
Amazon Relational Database Service

User Guide



Amazon Relational Database Service: User Guide

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is Amazon RDS?	1
Overview	1
DB instances	1
AWS Regions and Availability Zones	2
Security	2
Monitoring an Amazon RDS DB instance	3
How to work with Amazon RDS	3
AWS Management Console	3
Command line interface	3
Programming with Amazon RDS	3
How you are charged for Amazon RDS	3
What's next?	3
Getting started	4
Database engine-Specific topics	4
DB instances	5
DB instance classes	7
DB instance class types	7
Supported DB engines	8
Determining DB instance class support in AWS Regions	17
Changing your DB instance class	20
Configuring the processor	20
Hardware specifications	33
DB instance storage	40
Storage types	40
General Purpose SSD storage	40
Provisioned IOPS storage	42
Magnetic storage	44
Monitoring storage performance	44
Factors that affect storage performance	45
Regions, Availability Zones, and Local Zones	49
AWS Regions	49
Availability Zones	52
Local Zones	52
High availability (Multi-AZ)	53
Modifying a DB instance to be a Multi-AZ deployment	54
Failover process for Amazon RDS	54
DB instance billing for Amazon RDS	57
On-Demand DB instances	58
Reserved DB instances	59
Setting up	67
Sign up for AWS	67
Create an IAM user	67
Determine requirements	69
Provide access to your DB instance in your VPC by creating a security group	70
Getting started	73
Creating a MariaDB DB instance and connecting to a database	73
Creating a MariaDB DB instance	73
Connecting to a database on a DB instance running MariaDB	77
Deleting a DB instance	79
Creating a SQL Server DB instance and connecting to it	80
Creating a sample SQL Server DB instance	80
Connecting to your sample DB instance	83
Exploring your sample DB instance	84
Deleting your sample DB instance	85

Creating a MySQL DB instance and connecting to a database	86
Creating a MySQL DB instance	86
Connecting to a database on a DB instance running MySQL	90
Deleting a DB instance	92
Creating an Oracle DB instance and connecting to a database	93
Creating a sample Oracle DB instance	93
Connecting to your sample DB instance	97
Deleting your sample DB instance	99
Creating a PostgreSQL DB instance and connecting to a database	99
Creating a PostgreSQL DB instance	99
Connecting to a PostgreSQL DB instance	103
Deleting a DB instance	107
Tutorial: Create a web server and an Amazon RDS DB instance	108
Create a DB instance	109
Create a web server	114
Tutorials	126
Tutorials in this guide	126
Tutorials in other AWS guides	126
Best practices for Amazon RDS	128
Amazon RDS basic operational guidelines	128
DB instance RAM recommendations	129
Using Enhanced Monitoring to identify operating system issues	129
Using metrics to identify performance issues	129
Viewing performance metrics	129
Evaluating performance metrics	132
Tuning queries	133
Best practices for working with MySQL storage engines	134
Table size	134
Number of tables	134
Storage engine	135
Best practices for working with MariaDB storage engines	135
Table size	135
Number of tables	136
Storage engine	136
Best practices for working with Oracle	137
Best practices for working with PostgreSQL	137
Loading data into a PostgreSQL DB instance	137
Working with the PostgreSQL autovacuum feature	137
Best practices for working with SQL Server	138
Amazon RDS for SQL Server best practices video	139
Working with DB parameter groups	139
Amazon RDS new features and best practices presentation video	139
Configuring a DB instance	140
Creating a DB instance	141
Available settings	145
Original console example	157
Connecting to a DB instance	162
Finding the connection information	162
Database authentication options	165
Encrypted connections	166
Scenarios for accessing a DB instance	166
Connecting to a DB instance running a specific DB engine	166
Managing connections with RDS Proxy	167
Managing connections with RDS Proxy	167
Working with option groups	212
Option groups overview	212
Creating an option group	214

Copying an option group	215
Adding an option to an option group	216
Listing the options and option settings for an option group	220
Modifying an option setting	221
Removing an option from an option group	224
Deleting an option group	225
Working with parameter groups	228
Creating a DB parameter group	229
Associating a DB parameter group with a DB instance	231
Modifying parameters in a DB parameter group	232
Resetting parameters in a DB parameter group	234
Copying a DB parameter group	236
Listing DB parameter groups	238
Viewing parameter values for a DB parameter group	239
Comparing DB parameter groups	240
Specifying DB parameters	240
Managing a DB instance	245
Stopping a DB instance	246
Benefits	246
Limitations	247
Option and parameter group considerations	247
Public IP address	247
Stopping a DB instance	247
Starting a DB instance	249
Modifying a DB instance	250
Apply Immediately setting	251
Available settings	251
Maintaining a DB instance	264
Applying updates	266
Maintenance for Multi-AZ deployments	267
The maintenance window	268
Adjusting the maintenance window for a DB instance	269
Upgrading the engine version	271
Manually upgrading the engine version	271
Automatically upgrading the minor engine version	273
Renaming a DB instance	274
Renaming to replace an existing DB instance	274
Rebooting a DB instance	276
Working with read replicas	278
Overview	280
Creating a read replica	283
Promoting a read replica	285
Monitoring read replication	288
Creating a read replica in a different AWS Region	290
Tagging RDS resources	299
Overview	299
Using tags for access control with IAM	300
Using tags to produce detailed billing reports	300
Adding, listing, and removing tags	300
Using the AWS Tag Editor	303
Copying tags to DB instance snapshots	303
Tutorial: Use tags to specify which DB instances to stop	304
Enabling backups	306
Working with ARNs	309
Constructing an ARN	309
Getting an existing ARN	312
Working with storage	316

Increasing DB instance storage capacity	316
Managing capacity automatically with storage autoscaling	317
Modifying Provisioned IOPS	322
Deleting a DB instance	324
Deletion protection	324
Final snapshots and retained backups	324
Deleting a DB instance	325
Backing up and restoring a DB instance	327
Working with backups	328
Backup storage	328
Backup window	328
Backup retention period	330
Enabling automated backups	330
Retaining automated backups	331
Deleting retained automated backups	333
Disabling automated backups	334
Using AWS Backup	335
Unsupported MySQL storage engines	336
Unsupported MariaDB storage engines	336
Replicating automated backups to another Region	338
Enabling cross-Region automated backups	338
Finding information about replicated backups	339
Point-in-time recovery from a replicated backup	342
Stopping backup replication	343
Deleting replicated backups	344
.....	345
Creating a DB snapshot	346
Restoring from a DB snapshot	349
Parameter groups	349
Security groups	349
Option groups	349
Microsoft SQL Server	350
Oracle	350
Restoring from a snapshot	350
Copying a snapshot	352
Limitations	352
Snapshot retention	352
Copying shared snapshots	352
Handling encryption	353
Incremental snapshot copying	353
Cross-Region copying	353
Option groups	356
Parameter groups	357
Copying a DB snapshot	357
Sharing a snapshot	365
Sharing an encrypted snapshot	366
Sharing a snapshot	368
Exporting snapshot data to Amazon S3	373
Limitations	374
Overview of exporting snapshot data	374
Setting up access to an S3 bucket	374
Exporting a snapshot to an S3 bucket	377
Monitoring snapshot exports	379
Canceling a snapshot export	380
Troubleshooting PostgreSQL permissions errors	381
File naming convention	382
Data conversion	382

Point-in-time recovery	389
Deleting a snapshot	392
Deleting a DB snapshot	392
Tutorial: Restore a DB instance from a DB snapshot	394
Prerequisites for restoring a DB instance from a DB snapshot	394
Restoring a DB instance from a DB snapshot	395
Modifying a restored DB instance	396
Monitoring a DB instance	399
Overview of monitoring	400
Monitoring plan	400
Performance baseline	400
Performance guidelines	400
Monitoring tools	401
DB instance status	404
Using Amazon RDS recommendations	407
Responding to recommendations	409
Using Performance Insights	412
Overview	412
Enabling and Disabling Performance Insights	415
Accessing Performance Insights	419
Monitoring with the Performance Insights dashboard	421
Customizing the Performance Insights dashboard	444
Retrieving data with the Performance Insights API	454
Metrics published to CloudWatch	467
Logging Performance Insights calls by using AWS CloudTrail	468
Using Enhanced Monitoring	471
Enhanced Monitoring availability	471
Differences between CloudWatch and Enhanced Monitoring metrics	471
Setting up and enabling Enhanced Monitoring	471
Viewing Enhanced Monitoring	474
Viewing Enhanced Monitoring by using CloudWatch Logs	478
Using Amazon RDS event notification	487
Amazon RDS event categories and event messages	488
Subscribing to Amazon RDS event notification	494
Listing Amazon RDS event notification subscriptions	496
Modifying an Amazon RDS event notification subscription	497
Adding a source identifier to an Amazon RDS event notification subscription	499
Removing a source identifier from an Amazon RDS event notification subscription	500
Listing the Amazon RDS event notification categories	501
Deleting an Amazon RDS event notification subscription	502
Viewing Amazon RDS events	503
.....	503
Accessing database logs	504
Viewing and listing database log files	504
Downloading a database log file	504
Watching a database log file	506
Publishing to CloudWatch Logs	506
Reading log file contents using REST	506
MariaDB database log files	508
Microsoft SQL Server database log files	516
MySQL database log files	519
Oracle database log files	527
PostgreSQL database log files	534
Monitoring RDS with CloudWatch	540
Amazon RDS metrics	541
Amazon RDS dimensions	544
Viewing metrics and dimensions	544

Creating alarms	546
Publishing to CloudWatch Logs	547
Configuring CloudWatch log integration	547
Viewing DB instance metrics	548
Getting CloudWatch and EventBridge events for RDS	551
Overview of events for Amazon RDS	551
Creating rules to send Amazon RDS events to CloudWatch Events	553
Tutorial: Log Amazon RDS instance states	554
Working with AWS CloudTrail and Amazon RDS	557
CloudTrail integration with Amazon RDS	557
Amazon RDS log file entries	557
Working with RDS on AWS Outposts	561
Prerequisites	561
Support for Amazon RDS features	562
Supported DB instance classes	564
Customer-owned IP addresses	565
Creating DB instances	567
MariaDB on Amazon RDS	574
Common management tasks	574
MariaDB versions	576
Deprecation of MariaDB versions 10.0 and 10.1	577
MariaDB feature support	578
MariaDB 10.5 support	578
MariaDB 10.4 support	579
MariaDB 10.3 support	579
MariaDB 10.2 support	579
MariaDB 10.1 support	580
MariaDB 10.0 support	580
Features not supported	580
Supported storage engines	581
File size limits	581
MariaDB security	582
SSL support	584
Cache warming	585
Dumping and loading the buffer pool on demand	586
Database parameters	586
Common DBA tasks	586
Local time zone	587
Connecting to a DB instance running MariaDB	588
Finding the connection information	589
Connecting from the mysql utility	591
Connecting with SSL	592
Troubleshooting	592
Updating applications for new SSL/TLS certificates	594
Determining whether a client requires certificate verification in order to connect	594
Updating your application trust store	595
Example Java code for establishing SSL connections	597
Upgrading the MariaDB DB engine	598
Overview	598
Major version upgrades	599
Upgrading a MariaDB DB instance	600
Automatic minor version upgrades	600
Migrating data from a MySQL DB snapshot to a MariaDB DB instance	603
Incompatibilities between MariaDB and MySQL	603
Performing the migration	603
Working with MariaDB replication	605
Working with MariaDB read replicas	605

Configuring GTID-based replication	613
Importing data into a MariaDB DB instance	616
Options for MariaDB	616
MariaDB Audit Plugin support	617
Parameters for MariaDB	620
MariaDB on Amazon RDS SQL reference	625
mysql.rds_replica_status	625
mysql.rds_set_external_master_gtid	626
mysql.rds_kill_query_id	628
Microsoft SQL Server on Amazon RDS	630
Common management tasks	630
Limits	632
DB instance class support	634
Security	635
Compliance programs	636
HIPAA	636
SSL support	637
Version support	637
Version management	639
Database engine patches and versions	639
Deprecation schedule	639
Feature support	640
SQL Server 2019 features	640
SQL Server 2017 features	640
SQL Server 2016 features	641
SQL Server 2014 features	641
SQL Server 2012 features	641
SQL Server 2008 R2 deprecated on Amazon RDS	642
CDC support	642
Features not supported and features with limited support	643
Multi-AZ deployments	643
Using TDE	644
Functions and stored procedures	644
Local time zone	646
Supported time zones	647
Licensing SQL Server on Amazon RDS	655
Restoring license-terminated DB instances	655
SQL Server Developer Edition	655
Connecting to a DB instance running SQL Server	656
Connecting to your DB instance with SSMS	656
Connecting to your DB instance with SQL Workbench/J	658
Security group considerations	660
Troubleshooting	661
Updating applications for new SSL/TLS certificates	662
Determining whether any applications are connecting to your Microsoft SQL Server DB instance using SSL	662
Determining whether a client requires certificate verification in order to connect	663
Updating your application trust store	664
Upgrading the SQL Server DB engine	666
Overview	666
Major version upgrades	667
Multi-AZ and in-memory optimization considerations	668
Option and parameter group considerations	668
Testing an upgrade	669
Upgrading a SQL server DB instance	669
Upgrading deprecated DB instances before support ends	670
Importing and exporting SQL Server databases	671

Limitations and recommendations	671
Setting up	672
Using native backup and restore	675
Compressing backup files	685
Troubleshooting	685
.....	686
Importing and exporting SQL Server data using other methods	687
Working with SQL Server read replicas	696
Configuring read replicas for SQL Server	696
Read replica limitations with SQL Server	696
Troubleshooting a SQL Server read replica problem	697
Multi-AZ for SQL Server	698
Adding Multi-AZ to a SQL Server DB instance	699
Notes and recommendations	699
Determining the location of the secondary	701
Migrating to always on AGs	702
Additional features for SQL Server	703
Using SSL with a SQL Server DB instance	704
Configuring security protocols and ciphers	707
Using Windows Authentication with a SQL Server DB instance	711
Amazon S3 integration	721
Using Database Mail	734
Instance store support for tempdb	744
Using extended events	746
Options for SQL Server	749
Listing the available options for SQL Server versions and editions	750
Native backup and restore	751
Transparent Data Encryption	754
SQL Server Audit	757
SQL Server Analysis Services	762
SQL Server Integration Services	773
SQL Server Reporting Services	787
Microsoft Distributed Transaction Coordinator	797
Common DBA tasks for SQL Server	809
Accessing the tempdb database	810
Analyzing your database workload with SQL Server Tuning Advisor	812
Collations and character sets	814
Determining a recovery model	817
Determining the last failover time	817
Disabling fast inserts	818
Dropping a SQL Server database	818
Renaming a Multi-AZ database	818
Resetting the db_owner role password	819
Restoring license-terminated DB instances	819
Transitioning a database from OFFLINE to ONLINE	820
Using CDC	820
Using SQL Server Agent	822
Working with SQL Server logs	823
Working with trace and dump files	824
MySQL on Amazon RDS	826
Common management tasks	826
MySQL versions	828
Deprecation of MySQL version 5.6	830
Deprecation of MySQL version 5.5	831
MySQL features not supported by Amazon RDS	832
Supported storage engines	832
MySQL security	833

Password Validation Plugin	834
SSL support	835
Using memcached and other options with MySQL	836
InnoDB cache warming	837
Dumping and loading the buffer pool on demand	837
Local time zone	838
Known issues and limitations	839
Deprecated MySQL versions	839
Connecting to a DB instance running MySQL	840
Finding the connection information	840
Connecting from the MySQL client	843
Connecting with SSL	844
Connecting from MySQL Workbench	844
Troubleshooting	846
Updating applications for new SSL/TLS certificates	848
Determining whether any applications are connecting to your MySQL DB instance using SSL	849
Determining whether a client requires certificate verification to connect	849
Updating your application trust store	850
Example Java code for establishing SSL connections	851
Upgrading the MySQL DB engine	853
Overview	853
Major version upgrades	854
Testing an upgrade	858
Upgrading a MySQL DB instance	858
Automatic minor version upgrades	858
Upgrading with reduced downtime	860
Upgrading a MySQL DB snapshot	863
Importing data into a MySQL DB instance	865
Overview	865
Importing data considerations	866
Restoring a backup into an Amazon RDS MySQL DB instance	871
Importing data from a MySQL or MariaDB DB to a MySQL or MariaDB DB instance	879
Importing data to an Amazon RDS MySQL or MariaDB DB instance with reduced downtime	881
Importing data from any source to a MySQL or MariaDB DB instance	894
Working with MySQL replication	899
Working with MySQL read replicas	899
Using GTID-based replication	910
Replication with a MySQL or MariaDB instance running external to Amazon RDS	914
Exporting data from a MySQL DB instance	921
Prepare an external MySQL database	921
Prepare the source MySQL DB instance	922
Copy the database	923
Complete the export	924
Options for MySQL	925
MariaDB Audit Plugin	926
memcached	929
Common DBA tasks for MySQL	933
Ending a session or query	933
Skipping the current replication error	933
Working with InnoDB tablespaces to improve crash recovery times	934
Managing the global status history	936
Using Kerberos authentication for MySQL	938
Setting up Kerberos authentication for MySQL DB instances	939
Managing a DB instance in a domain	945
Connecting to MySQL with Kerberos authentication	946
Restoring a MySQL DB instance and adding it to a domain	947
Kerberos authentication MySQL limitations	947

Known issues and limitations	948
Inconsistent InnoDB buffer pool size	948
Index merge optimization returns wrong results	948
Log file size	949
MySQL parameter exceptions for Amazon RDS DB instances	949
MySQL file size limits in Amazon RDS	950
MySQL Keyring Plugin not supported	951
MySQL on Amazon RDS SQL reference	952
Overview	952
SQL reference conventions	953
mysql.rds_set_master_auto_position	953
mysql.rds_set_external_master	954
mysql.rds_set_external_master_with_delay	956
mysql.rds_set_external_master_with_auto_position	959
mysql.rds_reset_external_master	961
mysql.rds_import_binlog_ssl_material	962
mysql.rds_remove_binlog_ssl_material	963
mysql.rds_set_source_delay	964
mysql.rds_start_replication	964
mysql.rds_start_replication_until	965
mysql.rds_start_replication_until_gtid	966
mysql.rds_stop_replication	967
mysql.rds_skip_transaction_with_gtid	968
mysql.rds_skip_repl_error	968
mysql.rds_next_master_log	969
mysql.rds_innodb_buffer_pool_dump_now	971
mysql.rds_innodb_buffer_pool_load_now	971
mysql.rds_innodb_buffer_pool_load_abort	972
mysql.rds_set_configuration	972
mysql.rds_show_configuration	974
mysql.rds_kill	974
mysql.rds_kill_query	975
mysql.rds_rotate_general_log	975
mysql.rds_rotate_slow_log	976
mysql.rds_enable_gsh_collector	976
mysql.rds_set_gsh_collector	976
mysql.rds_disable_gsh_collector	976
mysql.rds_collect_global_status_history	977
mysql.rds_enable_gsh_rotation	977
mysql.rds_set_gsh_rotation	977
mysql.rds_disable_gsh_rotation	977
mysql.rds_rotate_global_status_history	978
Oracle on Amazon RDS	979
Oracle versions	979
Oracle Database 19c	979
Oracle Database 18c	980
Oracle Database 12c	981
Oracle licensing	990
License Included	990
Bring Your Own License (BYOL)	990
Licensing Oracle Multi-AZ deployments	992
Oracle instance classes	992
Deprecated DB instance classes	994
Oracle features	994
Supported features for RDS for Oracle	994
Unsupported features for RDS for Oracle	996
Oracle parameters	996

Oracle character sets	996
DB character set	997
National character set	999
Oracle limitations	999
File size limits	999
Public synonyms	1000
Schemas for unsupported features	1000
Limitations for Oracle DBA privileges	1000
Connecting to an Oracle instance	1001
Finding the endpoint	1001
SQL developer	1003
SQL*Plus	1005
Security group considerations	1006
Dedicated and shared server processes	1006
Troubleshooting	1006
Modifying Oracle sqlnet.ora parameters	1007
Securing Oracle connections	1010
Encrypting with SSL	1010
Using new SSL/TLS certificates	1011
Configuring Kerberos authentication	1014
Configuring outbound network access	1025
Administering your Oracle DB	1028
System tasks	1036
Database tasks	1049
Log tasks	1062
RMAN tasks	1069
Oracle Scheduler tasks	1085
Diagnostic tasks	1089
Other tasks	1095
Importing data into Oracle	1106
Importing using Oracle SQL Developer	1106
Importing using Oracle Data Pump	1106
Oracle Export/Import utilities	1115
Oracle SQL*Loader	1116
Oracle materialized views	1117
Working with Oracle replicas	1119
Overview of Oracle replicas	1119
Replica requirements for Oracle	1119
Preparing to create an Oracle replica	1121
Creating an Oracle replica in mounted mode	1122
Modifying the Oracle replica mode	1123
Troubleshooting Oracle replicas	1124
Options for Oracle	1126
Amazon S3 integration	1127
Application Express (APEX)	1140
Enterprise Manager	1149
Java virtual machine (JVM)	1164
Label security	1167
Locator	1170
Multimedia	1173
Native network encryption (NNE)	1176
OLAP	1180
Secure Sockets Layer (SSL)	1182
Spatial	1190
SQLT	1193
Statspack	1198
Time zone	1201

Transparent Data Encryption (TDE)	1204
UTL_MAIL	1206
XML DB	1208
Upgrading the Oracle DB engine	1209
Overview of Oracle upgrades	1209
Major version upgrades	1211
Minor version upgrades	1212
SE2 upgrade paths	1212
Upgrade considerations	1213
Automatic upgrade of Oracle Database 18c	1214
Testing an upgrade	1215
Upgrading an Oracle DB instance	1216
Upgrading an Oracle DB snapshot	1217
Console	1217
AWS CLI	1218
RDS API	1218
Tools and third-party software for Oracle	1219
Setting up	1219
Using Oracle GoldenGate	1225
Using the Oracle Repository Creation Utility	1237
Installing a Siebel database on Oracle on Amazon RDS	1242
Oracle database engine release notes	1245
Oracle Database 19c (19.0.0), Oracle Database 18c (18.0.0), and Oracle Database 12c Release 2 (12.2.0.1)	1245
Oracle versions 12.1.0.2 and 11.2.0.4	1246
Database engine: 19.0.0.0	1247
Database engine: 18.0.0.0	1293
Database engine: 12.2.0.1	1320
Database engine: 12.1.0.2	1359
Database engine: 11.2.0.4	1411
PostgreSQL on Amazon RDS	1453
Common management tasks	1454
The database preview environment	1457
Features not supported in the preview environment	1457
PostgreSQL extensions supported in the preview environment	1457
Creating a new DB instance in the preview environment	1459
PostgreSQL limitations	1460
PostgreSQL versions	1461
PostgreSQL 13 versions	1461
PostgreSQL 12 versions	1462
PostgreSQL 11 versions	1463
PostgreSQL 10 versions	1466
PostgreSQL 9.6 versions	1471
PostgreSQL 9.5 versions	1476
PostgreSQL extensions	1482
Restricting installation of PostgreSQL extensions	1482
PostgreSQL version 13 extensions supported on Amazon RDS	1483
PostgreSQL version 12 extensions supported on Amazon RDS	1486
PostgreSQL version 11.x extensions supported on Amazon RDS	1489
PostgreSQL version 10.x extensions supported on Amazon RDS	1491
PostgreSQL version 9.6.x extensions supported on Amazon RDS	1494
PostgreSQL version 9.5.x extensions supported on Amazon RDS	1496
PostgreSQL features	1499
Amazon RDS for PostgreSQL log_fdw extension	1499
Upgrading plv8	1500
Logical replication for PostgreSQL on Amazon RDS	1502
Event triggers for PostgreSQL on Amazon RDS	1504

Huge pages for Amazon RDS for PostgreSQL	1505
Tablespaces for PostgreSQL on Amazon RDS	1505
Autovacuum for PostgreSQL on Amazon RDS	1506
RAM disk for the stats_temp_directory	1506
ALTER ENUM for PostgreSQL	1506
Connecting to a PostgreSQL instance	1508
Using pgAdmin to connect to a PostgreSQL DB instance	1508
Using psql to connect to a PostgreSQL DB instance	1510
Troubleshooting connections to your PostgreSQL instance	1511
Security with RDS for PostgreSQL	1513
Using SSL with a PostgreSQL DB instance	1513
Using new SSL/TLS certificates in applications	1516
Using Kerberos authentication	1520
Upgrading the PostgreSQL DB engine	1533
Overview of upgrading	1533
PostgreSQL version numbers	1534
Choosing a major version upgrade	1534
How to perform a major version upgrade	1536
Automatic minor version upgrades	1540
Upgrading PostgreSQL extensions	1540
Upgrading a PostgreSQL DB snapshot engine version	1542
Working with PostgreSQL read replicas	1544
Read replica configuration with PostgreSQL	1544
Monitoring PostgreSQL read replicas	1545
Read replica limitations with PostgreSQL	1545
Replication interruptions with PostgreSQL read replicas	1545
Troubleshooting a PostgreSQL read replica problem	1546
Importing data into PostgreSQL	1548
Importing a PostgreSQL database from an Amazon EC2 instance	1549
Using the \copy command to import data to a table on a PostgreSQL DB instance	1551
Importing S3 data into RDS for PostgreSQL	1552
Transporting PostgreSQL databases between DB instances	1563
Exporting PostgreSQL data to Amazon S3	1568
Overview of exporting to S3	1568
Verify that your PostgreSQL version supports exports	1569
Specifying the Amazon S3 file path to export to	1569
Setting up access to an Amazon S3 bucket	1570
Exporting query data using the aws_s3.query_export_to_s3 function	1572
Function reference	1574
Common DBA tasks for PostgreSQL	1578
Creating roles	1578
Managing PostgreSQL database access	1579
Working with PostgreSQL parameters	1579
Audit logging for a PostgreSQL DB instance	1588
Working with the pgaudit extension	1588
Working with the pg_repack extension	1590
Using pgBadger for log analysis with PostgreSQL	1590
Viewing the contents of pg_config	1590
Working with the orafce extension	1591
Accessing external data with the postgres_fdw extension	1592
Restricting password management	1593
Working with PostgreSQL autovacuum	1593
Working with the PostGIS extension	1602
Using a custom DNS server for outbound network access	1605
Scheduling maintenance with the pg_cron extension	1607
Managing partitions with the pg_partman extension	1614
Invoking a Lambda function from RDS for PostgreSQL	1618

Security	1627
Database authentication	1628
Password authentication	1628
IAM database authentication	1629
Kerberos authentication	1629
Data protection	1629
Data encryption	1630
Internetwork traffic privacy	1643
Identity and access management	1644
Audience	1644
Authenticating with identities	1644
Managing access using policies	1646
How Amazon RDS works with IAM	1648
Identity-based policy examples	1650
IAM database authentication for MySQL and PostgreSQL	1660
Troubleshooting	1689
Logging and monitoring	1691
Compliance validation	1693
Resilience	1694
Backup and restore	1694
Replication	1694
Failover	1694
Infrastructure security	1695
Security groups	1695
Public accessibility	1695
VPC endpoints (AWS PrivateLink)	1696
Considerations	1696
Availability	1696
Creating an interface VPC endpoint	1697
Creating a VPC endpoint policy	1697
Security best practices	1698
Controlling access with security groups	1699
VPC security groups	1699
DB security groups	1699
DB security groups vs. VPC security groups	1700
Security group scenario	1700
Creating a VPC security group	1701
Associating with a DB instance	1701
Deleting DB VPC security groups	1701
DB security groups on EC2-Classic	1704
Master user account privileges	1712
Service-linked roles	1714
Service-linked role permissions for Amazon RDS	1714
Creating a service-linked role for Amazon RDS	1716
Editing a service-linked role for Amazon RDS	1716
Deleting a service-linked role for Amazon RDS	1716
Using Amazon RDS with Amazon VPC	1718
Determining whether you are using the EC2-VPC or EC2-Classic platform	1718
Scenarios for accessing a DB instance in a VPC	1720
Working with a DB instance in a VPC	1727
Updating the VPC for a DB instance	1734
Tutorial: Create an Amazon VPC for use with a DB instance	1737
Quotas and constraints	1742
Quotas in Amazon RDS	1742
Naming constraints in Amazon RDS	1743
Maximum number of database connections	1744
File size limits in Amazon RDS	1745

Troubleshooting	1746
Can't connect to DB instance	1746
Testing the DB instance connection	1747
Troubleshooting connection authentication	1748
Security issues	1748
Error message "failed to retrieve account attributes, certain console functions may be impaired."	1748
Resetting the DB instance owner password	1748
DB instance outage or reboot	1749
Parameter changes not taking effect	1749
DB instance out of storage	1750
Insufficient DB instance capacity	1751
MySQL and MariaDB issues	1751
Maximum MySQL and MariaDB connections	1752
Diagnosing and resolving incompatible parameters status for a memory limit	1752
Diagnosing and resolving lag between read replicas	1753
Diagnosing and resolving a MySQL or MariaDB read replication failure	1754
Creating triggers with binary logging enabled requires SUPER privilege	1755
Diagnosing and resolving point-in-time restore failures	1757
Replication stopped error	1757
Read replica create fails or replication breaks with fatal error 1236	1758
Can't set backup retention period to 0	1758
Amazon RDS API reference	1759
Using the Query API	1759
Query parameters	1759
Query request authentication	1759
Troubleshooting applications	1760
Retrieving errors	1760
Troubleshooting tips	1760
Document history	1761
Earlier updates	1798
AWS glossary	1818

What is Amazon Relational Database Service (Amazon RDS)?

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

Note

This guide covers Amazon RDS database engines other than Amazon Aurora. For information about using Amazon Aurora, see the [Amazon Aurora User Guide](#).

This guide covers using Amazon RDS in the AWS Cloud. For information about using Amazon RDS in on-premises VMware environments, see the [Amazon RDS on VMware User Guide](#).

Overview of Amazon RDS

Why do you want a managed relational database service? Because Amazon RDS takes over many of the difficult and tedious management tasks of a relational database:

- When you buy a server, you get CPU, memory, storage, and IOPS, all bundled together. With Amazon RDS, these are split apart so that you can scale them independently. If you need more CPU, less IOPS, or more storage, you can easily allocate them.
- Amazon RDS manages backups, software patching, automatic failure detection, and recovery.
- To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances. It also restricts access to certain system procedures and tables that require advanced privileges.
- You can have automated backups performed when you need them, or manually create your own backup snapshot. You can use these backups to restore a database. The Amazon RDS restore process works reliably and efficiently.
- You can use the database products you are already familiar with: MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server.
- You can get high availability with a primary instance and a synchronous secondary instance that you can fail over to when problems occur. You can also use MariaDB, Microsoft SQL Server, MySQL, Oracle, and PostgreSQL read replicas to increase read scaling.
- In addition to the security in your database package, you can help control who can access your RDS databases by using AWS Identity and Access Management (IAM) to define users and permissions. You can also help protect your databases by putting them in a virtual private cloud.

If you are new to AWS products and services, begin learning more with the following resources:

- For an overview of all AWS products, see [What is cloud computing?](#)
- Amazon Web Services provides a number of database services. For guidance on which service is best for your environment, see [Running databases on AWS](#).

DB instances

The basic building block of Amazon RDS is the DB instance. A *DB instance* is an isolated database environment in the AWS Cloud. Your DB instance can contain multiple user-created databases. You can access your DB instance by using the same tools and applications that you use with a standalone

database instance. You can create and modify a DB instance by using the AWS Command Line Interface, the Amazon RDS API, or the AWS Management Console.

Each DB instance runs a *DB engine*. Amazon RDS currently supports the MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server DB engines. Each DB engine has its own supported features, and each version of a DB engine may include specific features. Additionally, each DB engine has a set of parameters in a DB parameter group that control the behavior of the databases that it manages.

The computation and memory capacity of a DB instance is determined by its *DB instance class*. You can select the DB instance that best meets your needs. If your needs change over time, you can change DB instances. For information, see [DB instance classes \(p. 7\)](#).

Note

For pricing information on DB instance classes, see the Pricing section of the [Amazon RDS](#) product page.

DB instance storage comes in three types: Magnetic, General Purpose (SSD), and Provisioned IOPS (PIOPS). They differ in performance characteristics and price, allowing you to tailor your storage performance and cost to the needs of your database. Each DB instance has minimum and maximum storage requirements depending on the storage type and the database engine it supports. It's important to have sufficient storage so that your databases have room to grow. Also, sufficient storage makes sure that features for the DB engine have room to write content or log entries. For more information, see [Amazon RDS DB instance storage \(p. 40\)](#).

You can run a DB instance on a virtual private cloud (VPC) using the Amazon Virtual Private Cloud (Amazon VPC) service. When you use a VPC, you have control over your virtual networking environment. You can choose your own IP address range, create subnets, and configure routing and access control lists. The basic functionality of Amazon RDS is the same whether it's running in a VPC or not. Amazon RDS manages backups, software patching, automatic failure detection, and recovery. There's no additional cost to run your DB instance in a VPC. For more information on using Amazon VPC with RDS, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).

Amazon RDS uses Network Time Protocol (NTP) to synchronize the time on DB Instances.

AWS Regions and Availability Zones

Amazon cloud computing resources are housed in highly available data center facilities in different areas of the world (for example, North America, Europe, or Asia). Each data center location is called an AWS Region.

Each AWS Region contains multiple distinct locations called Availability Zones, or AZs. Each Availability Zone is engineered to be isolated from failures in other Availability Zones. Each is engineered to provide inexpensive, low-latency network connectivity to other Availability Zones in the same AWS Region. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location. For more information, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

You can run your DB instance in several Availability Zones, an option called a Multi-AZ deployment. When you choose this option, Amazon automatically provisions and maintains a secondary standby DB instance in a different Availability Zone. Your primary DB instance is synchronously replicated across Availability Zones to the secondary instance. This approach helps provide data redundancy and failover support, eliminate I/O freezes, and minimize latency spikes during system backups. For more information, see [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).

Security

A *security group* controls the access to a DB instance. It does so by allowing access to IP address ranges or Amazon EC2 instances that you specify.

For more information about security groups, see [Security in Amazon RDS \(p. 1627\)](#).

Monitoring an Amazon RDS DB instance

There are several ways that you can track the performance and health of a DB instance. You can use the Amazon CloudWatch service to monitor the performance and health of a DB instance. CloudWatch performance charts are shown in the Amazon RDS console. You can also subscribe to Amazon RDS events to be notified about changes to a DB instance, DB snapshot, DB parameter group, or DB security group. For more information, see [Monitoring an Amazon RDS DB instance \(p. 399\)](#).

How to work with Amazon RDS

There are several ways that you can interact with Amazon RDS.

AWS Management Console

The AWS Management Console is a simple web-based user interface. You can manage your DB instances from the console with no programming required. To access the Amazon RDS console, sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.

Command line interface

You can use the AWS Command Line Interface (AWS CLI) to access the Amazon RDS API interactively. To install the AWS CLI, see [Installing the AWS Command Line Interface](#). To begin using the AWS CLI for RDS, see [AWS Command Line Interface reference for Amazon RDS](#).

Programming with Amazon RDS

If you are a developer, you can access the Amazon RDS programmatically. For more information, see [Amazon RDS application programming interface \(API\) reference \(p. 1759\)](#).

For application development, we recommend that you use one of the AWS Software Development Kits (SDKs). The AWS SDKs handle low-level details such as authentication, retry logic, and error handling, so that you can focus on your application logic. AWS SDKs are available for a wide variety of languages. For more information, see [Tools for Amazon web services](#).

AWS also provides libraries, sample code, tutorials, and other resources to help you get started more easily. For more information, see [Sample code & libraries](#).

How you are charged for Amazon RDS

When you use Amazon RDS, you can choose to use on-demand DB instances or reserved DB instances. For more information, see [DB instance billing for Amazon RDS \(p. 57\)](#).

For Amazon RDS pricing information, see the [Amazon RDS product page](#).

What's next?

The preceding section introduced you to the basic infrastructure components that RDS offers. What should you do next?

Getting started

Create a DB instance using instructions in [Getting started with Amazon RDS \(p. 73\)](#).

Database engine–Specific topics

You can review information specific to a particular DB engine in the following sections:

- [MariaDB on Amazon RDS \(p. 574\)](#)
- [Microsoft SQL Server on Amazon RDS \(p. 630\)](#)
- [MySQL on Amazon RDS \(p. 826\)](#)
- [Oracle on Amazon RDS \(p. 979\)](#)
- [PostgreSQL on Amazon RDS \(p. 1453\)](#)

Amazon RDS DB instances

A *DB instance* is an isolated database environment running in the cloud. It is the basic building block of Amazon RDS. A DB instance can contain multiple user-created databases, and can be accessed using the same client tools and applications you might use to access a standalone database instance. DB instances are simple to create and modify with the Amazon AWS command line tools, Amazon RDS API operations, or the AWS Management Console.

Note

Amazon RDS supports access to databases using any standard SQL client application. Amazon RDS does not allow direct host access.

You can have up to 40 Amazon RDS DB instances, with the following limitations:

- 10 for each SQL Server edition (Enterprise, Standard, Web, and Express) under the "license-included" model
- 10 for Oracle under the "license-included" model
- 40 for MySQL, MariaDB, or PostgreSQL
- 40 for Oracle under the "bring-your-own-license" (BYOL) licensing model

Note

If your application requires more DB instances, you can request additional DB instances by using [this form](#).

Each DB instance has a DB instance identifier. This customer-supplied name uniquely identifies the DB instance when interacting with the Amazon RDS API and AWS CLI commands. The DB instance identifier must be unique for that customer in an AWS Region.

The identifier is used as part of the DNS hostname allocated to your instance by RDS. For example, if you specify db1 as the DB instance identifier, then RDS will automatically allocate a DNS endpoint for your instance, such as db1.123456789012.us-east-1.rds.amazonaws.com, where 123456789012 is the fixed identifier for a specific region for your account.

Each DB instance supports a database engine. Amazon RDS currently supports MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server, and Amazon Aurora database engines.

When creating a DB instance, some database engines require that a database name be specified. A DB instance can host multiple databases, or a single Oracle database with multiple schemas. The database name value depends on the database engine:

- For the MySQL and MariaDB database engines, the database name is the name of a database hosted in your DB instance. Databases hosted by the same DB instance must have a unique name within that instance.
- For the Oracle database engine, database name is used to set the value of ORACLE_SID, which must be supplied when connecting to the Oracle RDS instance.
- For the Microsoft SQL Server database engine, database name is not a supported parameter.
- For the PostgreSQL database engine, the database name is the name of a database hosted in your DB instance. A database name is not required when creating a DB instance. Databases hosted by the same DB instance must have a unique name within that instance.

Amazon RDS creates a master user account for your DB instance as part of the creation process. This master user has permissions to create databases and to perform create, delete, select, update, and insert operations on tables the master user creates. You must set the master user password when you create a DB instance, but you can change it at any time using the Amazon AWS command line tools, Amazon RDS

API operations, or the AWS Management Console. You can also change the master user password and manage users using standard SQL commands.

Note

This guide covers non-Aurora Amazon RDS database engines. For information about using Amazon Aurora, see the [Amazon Aurora User Guide](#).

DB instance classes

The DB instance class determines the computation and memory capacity of an Amazon RDS DB instance. The DB instance class you need depends on your processing power and memory requirements.

For more information about instance class pricing, see [Amazon RDS pricing](#).

Topics

- [DB instance class types \(p. 7\)](#)
- [Supported DB engines for DB instance classes \(p. 8\)](#)
- [Determining DB instance class support in AWS Regions \(p. 17\)](#)
- [Changing your DB instance class \(p. 20\)](#)
- [Configuring the processor for a DB instance class \(p. 20\)](#)
- [Hardware specifications for DB instance classes \(p. 33\)](#)

DB instance class types

Amazon RDS supports three types of instance classes: Standard, Memory Optimized, and Burstable Performance. For more information about Amazon EC2 instance types, see [Instance type](#) in the Amazon EC2 documentation.

The following are the Standard DB instance classes available:

- **db.m6g** – General-purpose instance classes powered by AWS Graviton2 processors. These deliver balanced compute, memory, and networking for a broad range of general purpose workloads.

You can modify a DB instance to use one of the DB instance classes powered by AWS Graviton2 processors by completing the same steps as any other DB instance modification.
- **db.m5d** – Newest generation instance classes that are optimized for low latency, very high random I/O performance, and high sequential read throughput.
- **db.m5** – Latest generation general-purpose instance classes that provide a balance of compute, memory, and network resources, and are a good choice for many applications. The db.m5 instance classes provide more computing capacity than the previous db.m4 instance classes. They are powered by the AWS Nitro System, a combination of dedicated hardware and lightweight hypervisor.
- **db.m4** – General-purpose instance classes that provide more computing capacity than the previous db.m3 instance classes.
- **db.m3** – General-purpose instance classes that provide more computing capacity than the previous db.m1 instance classes.
- **db.m1** – Earlier generation general-purpose instance classes.

The following are the Memory Optimized DB instance classes available:

- **db.z1d** – Instance classes optimized for memory-intensive applications. These offer both high compute capacity and a high memory footprint. High frequency z1d instances deliver a sustained all core frequency of up to 4.0 GHz.
- **db.x1e** – Instance classes optimized for memory-intensive applications. These offer one of the lowest price per gibibyte (GiB) of RAM among the DB instance classes and up to 3,904 GiB of DRAM-based instance memory.
- **db.x1** – Instance classes optimized for memory-intensive applications. These offer one of the lowest price per GiB of RAM among the DB instance classes and up to 1,952 GiB of DRAM-based instance memory.

- **db.r6g** – Instance classes powered by AWS Graviton2 processors. These are ideal for running memory-intensive workloads in open-source databases such as MySQL and PostgreSQL.

You can modify a DB instance to use one of the DB instance classes powered by AWS Graviton2 processors by completing the same steps as any other DB instance modification.

- **db.r5b** – Instance classes that are memory-optimized for throughput-intensive applications. Powered by the AWS Nitro System, db.r5b instances deliver up to 60 Gbps bandwidth and 260,000 IOPS of EBS performance, which is the fastest block storage performance on EC2.
- **db.r5d** – Instance classes that are optimized for low latency, very high random I/O performance, and high sequential read throughput.
- **db.r5** – Latest generation instance classes optimized for memory-intensive applications. These offer improved networking and Amazon Elastic Block Store (Amazon EBS) performance. They are powered by the AWS Nitro System, a combination of dedicated hardware and lightweight hypervisor.
- **db.r4** – Instance classes optimized for memory-intensive applications. These offer improved networking and Amazon EBS performance.
- **db.r3** – Instance classes that provide memory optimization.
- **db.m2** – Earlier generation memory-optimized instance classes.

The following are the Burstable Performance DB instance classes available:

- **db.t3** – Next generation instance classes that provide a baseline performance level, with the ability to burst to full CPU usage. These instance classes provide more computing capacity than the previous db.t2 instance classes. They are powered by the AWS Nitro System, a combination of dedicated hardware and lightweight hypervisor.
- **db.t2** – Instance classes that provide a baseline performance level, with the ability to burst to full CPU usage.

Note

The DB instance classes that use the AWS Nitro System (db.m5, db.r5, db.t3) are throttled on combined read plus write workload.

For DB instance class hardware specifications, see [Hardware specifications for DB instance classes \(p. 33\)](#).

Supported DB engines for DB instance classes

The following are DB engine considerations for DB instance classes:

MariaDB

The Graviton2 instance classes db.m6g and db.r6g are supported for all MariaDB 10.5 versions and MariaDB version 10.4.13 and higher 10.4 versions.

Microsoft SQL Server

Instance class support varies according to the version and edition of SQL Server. For instance class support by version and edition, see [DB instance class support for Microsoft SQL Server \(p. 634\)](#).

MySQL

The Graviton2 instance classes db.m6g and db.r6g are supported for RDS for MySQL versions 8.0.17 and higher.

Oracle

Instance class support varies according to the version and edition of Oracle. For instance class support by version and edition, see [RDS for Oracle instance classes \(p. 992\)](#).

PostgreSQL

PostgreSQL versions 13 and higher support the db.m6g, db.m5, db.r6g, db.r5, db.t3 instance classes. Previous generations of classes are supported only by PostgreSQL versions lower than 13 and include db.m4, db.m3, db.r4, db.r3, and db.t2.

In the following table, you can find details about supported Amazon RDS DB instance classes for each Amazon RDS DB engine.

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6g – Standard instance classes powered by AWS Graviton2 processors					
db.m6g.16xlarge	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 13,12.3 & higher
db.m6g.12xlarge	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 13,12.3 & higher
db.m6g.8xlarge	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 13,12.3 & higher
db.m6g.4xlarge	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 13,12.3 & higher
db.m6g.2xlarge	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 13,12.3 & higher
db.m6g.xlarge	All MariaDB 10.5 versions and MariaDB version	No	MySQL 8.0.17 & higher	No	PostgreSQL 13,12.3 & higher

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
	10.4.13 & higher 10.4 versions				
db.m6g.large	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 13, 12.3 & higher
db.m5d – Newest generation standard instance classes					
db.m5d.24xlarge	No	Yes	No	No	No
db.m5d.16xlarge	No	Yes	No	No	No
db.m5d.12xlarge	No	Yes	No	No	No
db.m5d.8xlarge	No	Yes	No	No	No
db.m5d.4xlarge	No	Yes	No	No	No
db.m5d.2xlarge	No	Yes	No	No	No
db.m5d.xlarge	No	Yes	No	No	No
db.m5d.large	No	Yes	No	No	No
db.m5 – Latest generation standard instance classes					
db.m5.24xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.m5.16xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.m5.12xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.m5.8xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.m5.4xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5.2xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.m5.xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.m5.large	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.m4 – Standard instance classes					
db.m4.16xlarge	Yes	Yes	MySQL 8.0, 5.7, 5.6	Yes	Lower than PostgreSQL 13
db.m4.10xlarge	Yes	Yes	Yes	Yes	Lower than PostgreSQL 13
db.m4.4xlarge	Yes	Yes	Yes	Yes	Lower than PostgreSQL 13
db.m4.2xlarge	Yes	Yes	Yes	Yes	Lower than PostgreSQL 13
db.m4.xlarge	Yes	Yes	Yes	Yes	Lower than PostgreSQL 13
db.m4.large	Yes	Yes	Yes	Yes	Lower than PostgreSQL 13
db.m3 – Standard instance classes					
db.m3.2xlarge	No	Yes	Yes	Deprecated	Lower than PostgreSQL 13
db.m3.xlarge	No	Yes	Yes	Deprecated	Lower than PostgreSQL 13
db.m3.large	No	Yes	Yes	Deprecated	Lower than PostgreSQL 13

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m3.medium	No	Yes	Yes	Deprecated	Lower than PostgreSQL 13
db.m1 – Standard instance classes					
db.m1.xlarge	No	Yes	Deprecated	Deprecated	Deprecated
db.m1.large	No	Yes	Deprecated	Deprecated	Deprecated
db.m1.medium	No	Yes	Deprecated	Deprecated	Deprecated
db.m1.small	No	Yes	Deprecated	Deprecated	Deprecated
db.z1d – Memory-optimized instance classes					
db.z1d.12xlarge	No	Yes	No	Yes	No
db.z1d.6xlarge	No	Yes	No	Yes	No
db.z1d.3xlarge	No	Yes	No	Yes	No
db.z1d.2xlarge	No	Yes	No	Yes	No
db.z1d.xlarge	No	Yes	No	Yes	No
db.z1d.large	No	Yes	No	Yes	No
db.x1e – Memory-optimized instance classes					
db.x1e.32xlarge	No	Yes	No	Yes	No
db.x1e.16xlarge	No	Yes	No	Yes	No
db.x1e.8xlarge	No	Yes	No	Yes	No
db.x1e.4xlarge	No	Yes	No	Yes	No
db.x1e.2xlarge	No	Yes	No	Yes	No
db.x1e.xlarge	No	Yes	No	Yes	No
db.x1 – Memory-optimized instance classes					
db.x1.32xlarge	No	Yes	No	Yes	No
db.x1.16xlarge	No	Yes	No	Yes	No
db.r6g – Memory-optimized instance classes powered by AWS Graviton2 processors					
db.r6g.16xlarge	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 13, 12.3 & higher

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6g.12xlarge	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 13, 12.3 & higher
db.r6g.8xlarge	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 12.3 & higher
db.r6g.4xlarge	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 13, 12.3 & higher
db.r6g.2xlarge	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 13, 12.3 & higher
db.r6g.xlarge	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 13, 12.3 & higher
db.r6g.large	All MariaDB 10.5 versions and MariaDB version 10.4.13 & higher 10.4 versions	No	MySQL 8.0.17 & higher	No	PostgreSQL 13, 12.3 & higher
db.r5d – Newest Generation Memory Optimized Instance Classes					
db.r5d.24xlarge	No	Yes	No	No	No
db.r5d.16xlarge	No	Yes	No	No	No
db.r5d.12xlarge	No	Yes	No	No	No

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5d.8xlarge	No	Yes	No	No	No
db.r5d.4xlarge	No	Yes	No	No	No
db.r5d.2xlarge	No	Yes	No	No	No
db.r5d.xlarge	No	Yes	No	No	No
db.r5d.large	No	Yes	No	No	No
db.r5b – Memory-optimized instance classes					
db.r5b.24xlarge	No	Yes	No	Yes	No
db.r5b.16xlarge	No	Yes	No	Yes	No
db.r5b.12xlarge	No	Yes	No	Yes	No
db.r5b.8xlarge	No	Yes	No	Yes	No
db.r5b.4xlarge	No	Yes	No	Yes	No
db.r5b.2xlarge	No	Yes	No	Yes	No
db.r5b.xlarge	No	Yes	No	Yes	No
db.r5b.large	No	Yes	No	Yes	No
db.r5 – Latest generation memory-optimized instance classes					
db.r5.24xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.r5.16xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.r5.12xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.r5.8xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.r5.4xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.2xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.r5.xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.r5.large	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10.4 & higher, 9.6.9 & higher
db.r4 – Memory-optimized instance classes					
db.r4.16xlarge	Yes	Yes	MySQL 8.0, 5.7, 5.6	Yes	Lower than PostgreSQL 13
db.r4.8xlarge	Yes	Yes	MySQL 8.0, 5.7, 5.6	Yes	Lower than PostgreSQL 13
db.r4.4xlarge	Yes	Yes	MySQL 8.0, 5.7, 5.6	Yes	Lower than PostgreSQL 13
db.r4.2xlarge	Yes	Yes	MySQL 8.0, 5.7, 5.6	Yes	Lower than PostgreSQL 13
db.r4.xlarge	Yes	Yes	MySQL 8.0, 5.7, 5.6	Yes	Lower than PostgreSQL 13
db.r4.large	Yes	Yes	MySQL 8.0, 5.7, 5.6	Yes	Lower than PostgreSQL 13
db.r3 – Memory-optimized instance classes					
db.r3.8xlarge**	Yes	Yes	Yes	Deprecated	Lower than PostgreSQL 13
db.r3.4xlarge	Yes	Yes	Yes	Deprecated	Lower than PostgreSQL 13
db.r3.2xlarge	Yes	Yes	Yes	Deprecated	Lower than PostgreSQL 13

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r3.xlarge	Yes	Yes	Yes	Deprecated	Lower than PostgreSQL 13
db.r3.large	Yes	Yes	Yes	Deprecated	Lower than PostgreSQL 13
db.m2 – Memory-optimized instance classes					
db.m2.4xlarge	No	Yes	Deprecated	Deprecated	Deprecated
db.m2.2xlarge	No	Yes	Deprecated	Deprecated	Deprecated
db.m2.xlarge	No	Yes	Deprecated	Deprecated	Deprecated
db.t3 – Next generation burstable performance instance classes					
db.t3.2xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10, 9.6.9 & higher
db.t3.xlarge	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10, 9.6.9 & higher
db.t3.large	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10, 9.6.9 & higher
db.t3.medium	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10, 9.6.9 & higher
db.t3.small	Yes	Yes	Yes	Yes	PostgreSQL 13, 12, 11, 10, 9.6.9 & higher
db.t3.micro	Yes	No	Yes	Yes	PostgreSQL 13, 12, 11, 10, 9.6.9 & higher
db.t2 – Burstable performance instance classes					
db.t2.2xlarge	Yes	No	MySQL 8.0, 5.7, 5.6	Deprecated	Lower than PostgreSQL 13
db.t2.xlarge	Yes	No	MySQL 8.0, 5.7, 5.6	Deprecated	Lower than PostgreSQL 13
db.t2.large	Yes	Yes	Yes	Deprecated	Lower than PostgreSQL 13

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.t2.medium	Yes	Yes	Yes	Deprecated	Lower than PostgreSQL 13
db.t2.small	Yes	Yes	Yes	Deprecated	Lower than PostgreSQL 13
db.t2.micro	Yes	Yes	Yes	Deprecated	Lower than PostgreSQL 13

Determining DB instance class support in AWS Regions

To determine the DB instance classes supported by each DB engine in a specific AWS Region, you can use the AWS Management Console, the [Amazon RDS Pricing](#) page, or the `describe-orderable-db-instance-options` command for the AWS Command Line Interface (AWS CLI).

Note

When you perform operations with the AWS CLI, such as creating or modifying a DB instance, it automatically shows the supported DB instance classes for a specific DB engine, DB engine version, and AWS Region.

Contents

- [Using the Amazon RDS pricing page to determine DB instance class support in AWS Regions \(p. 17\)](#)
- [Using the AWS CLI to determine DB instance class support in AWS Regions \(p. 18\)](#)
 - [Listing the DB instance classes that are supported by a specific DB engine version in an AWS Region \(p. 18\)](#)
 - [Listing the DB engine versions that support a specific DB instance class in an AWS Region \(p. 19\)](#)

Using the Amazon RDS pricing page to determine DB instance class support in AWS Regions

You can use the [Amazon RDS Pricing](#) page to determine the DB instance classes supported by each DB engine in a specific AWS Region.

To use the pricing page to determine the DB instance classes supported by each engine in a Region

1. Go to [Amazon RDS Pricing](#).
2. Choose a DB engine.
3. On the pricing page for the DB engine, choose **On-Demand DB Instances** or **Reserved DB Instances**.
4. To see the DB instance classes available in an AWS Region, choose the AWS Region in **Region**.

Other choices might be available for some DB engines, such as **Single-AZ Deployment** or **Multi-AZ Deployment**.

Using the AWS CLI to determine DB instance class support in AWS Regions

You can use the AWS CLI to determine which DB instance classes are supported for specific DB engines and DB engine versions in an AWS Region. The following table shows the valid DB engine values.

Engine names	Engine values in CLI commands	More information about versions
MariaDB	mariadb	MariaDB on Amazon RDS versions (p. 576)
Microsoft SQL Server	sqlserver-ee sqlserver-se sqlserver-ex sqlserver-web	Microsoft SQL Server versions on Amazon RDS (p. 637)
MySQL	mysql	MySQL on Amazon RDS versions (p. 828)
Oracle	oracle-ee oracle-se2 oracle-se	Oracle database engine release notes (p. 1245)
PostgreSQL	postgres	Supported PostgreSQL database versions (p. 1461)

For information about AWS Region names, see [AWS Regions \(p. 49\)](#).

The following examples demonstrate how to determine DB instance class support in an AWS Region using the `describe-orderable-db-instance-options` AWS CLI command.

Note

To limit the output, these examples show results only for the General Purpose SSD (gp2) storage type. If necessary, you can change the storage type to Provisioned IOPS (io1) or magnetic (standard) in the commands.

Topics

- [Listing the DB instance classes that are supported by a specific DB engine version in an AWS Region \(p. 18\)](#)
- [Listing the DB engine versions that support a specific DB instance class in an AWS Region \(p. 19\)](#)

Listing the DB instance classes that are supported by a specific DB engine version in an AWS Region

To list the DB instance classes that are supported by a specific DB engine version in an AWS Region, run the following command.

For Linux, macOS, or Unix:

```
aws rds describe-orderable-db-instance-options --engine engine --engine-version version \
--query "*[ ].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[? \
StorageType=='gp2']|[ ].{DBInstanceClass:DBInstanceClass}" \
--output text \
```

```
--region region
```

For Windows:

```
aws rds describe-orderable-db-instance-options --engine engine --engine-version version ^
--query "[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2'||[].{DBInstanceClass:DBInstanceClass}" ^
--output text ^
--region region
```

For example, the following command lists the supported DB instance classes for version 12.4 of the RDS for PostgreSQL DB engine in US East (N. Virginia).

For Linux, macOS, or Unix:

```
aws rds describe-orderable-db-instance-options --engine postgres --engine-version 12.4 \ 
--query "[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2'||[].{DBInstanceClass:DBInstanceClass}" \
--output text \
--region us-east-1
```

For Windows:

```
aws rds describe-orderable-db-instance-options --engine postgres --engine-version 12.4 ^
--query "[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2'||[].{DBInstanceClass:DBInstanceClass}" ^
--output text ^
--region us-east-1
```

Listing the DB engine versions that support a specific DB instance class in an AWS Region

To list the DB engine versions that support a specific DB instance class in an AWS Region, run the following command.

For Linux, macOS, or Unix:

```
aws rds describe-orderable-db-instance-options --engine engine --db-instance-
class DB_instance_class \
--query "[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2'||[].{EngineVersion:EngineVersion}" \
--output text \
--region region
```

For Windows:

```
aws rds describe-orderable-db-instance-options --engine engine --db-instance-
class DB_instance_class ^
--query "[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2'||[].{EngineVersion:EngineVersion}" ^
--output text ^
--region region
```

For example, the following command lists the DB engine versions of the RDS for PostgreSQL DB engine that support the db.r5.large DB instance class in US East (N. Virginia).

For Linux, macOS, or Unix:

```
aws rds describe-orderable-db-instance-options --engine postgres --db-instance-class db.r5.large \
    --query "[].{EngineVersion:EngineVersion,StorageType:StorageType}|[? StorageType=='gp2'||[].{EngineVersion:EngineVersion}]" \
    --output text \
    --region us-east-1
```

For Windows:

```
aws rds describe-orderable-db-instance-options --engine postgres --db-instance-class db.r5.large ^
    --query "[].{EngineVersion:EngineVersion,StorageType:StorageType}|[? StorageType=='gp2'||[].{EngineVersion:EngineVersion}]" ^
    --output text ^
    --region us-east-1
```

Changing your DB instance class

You can change the CPU and memory available to a DB instance by changing its DB instance class. To change the DB instance class, modify your DB instance by following the instructions in [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Some instance classes require that your DB instance is in a VPC. If your current DB instance isn't in a VPC, and you want to use an instance class that requires one, first move your DB instance into a VPC. For more information, see [Moving a DB instance not in a VPC into a VPC \(p. 1735\)](#).

Configuring the processor for a DB instance class

Amazon RDS DB instance classes support Intel Hyper-Threading Technology, which enables multiple threads to run concurrently on a single Intel Xeon CPU core. Each thread is represented as a virtual CPU (vCPU) on the DB instance. A DB instance has a default number of CPU cores, which varies according to DB instance type. For example, a db.m4.xlarge DB instance type has two CPU cores and two threads per core by default—four vCPUs in total.

Note

Each vCPU is a hyperthread of an Intel Xeon CPU core.

Topics

- [Overview of configuring the processor \(p. 20\)](#)
- [CPU cores and threads per CPU core per DB instance class \(p. 21\)](#)
- [Setting the CPU cores and threads per CPU core for a DB instance class \(p. 25\)](#)

Overview of configuring the processor

In most cases, you can find a DB instance class that has a combination of memory and number of vCPUs to suit your workloads. However, you can also specify the following processor features to optimize your DB instance for specific workloads or business needs:

- **Number of CPU cores** – You can customize the number of CPU cores for the DB instance. You might do this to potentially optimize the licensing costs of your software with a DB instance that has sufficient amounts of RAM for memory-intensive workloads but fewer CPU cores.
- **Threads per core** – You can disable Intel Hyper-Threading Technology by specifying a single thread per CPU core. You might do this for certain workloads, such as high-performance computing (HPC) workloads.

You can control the number of CPU cores and threads for each core separately. You can set one or both in a request. After a setting is associated with a DB instance, the setting persists until you change it.

The processor settings for a DB instance are associated with snapshots of the DB instance. When a snapshot is restored, its restored DB instance uses the processor feature settings used when the snapshot was taken.

If you modify the DB instance class for a DB instance with nondefault processor settings, either specify default processor settings or explicitly specify processor settings at modification. This requirement ensures that you are aware of the third-party licensing costs that might be incurred when you modify the DB instance.

There is no additional or reduced charge for specifying processor features on an Amazon RDS DB instance. You're charged the same as for DB instances that are launched with default CPU configurations.

CPU cores and threads per CPU core per DB instance class

In the following table, you can find the DB instance classes that support setting a number of CPU cores and CPU threads per core. You can also find the default value and the valid values for the number of CPU cores and CPU threads per core for each DB instance class.

DB instance class	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
db.m5.large	2	1	2	1	1, 2
db.m5.xlarge	4	2	2	2	1, 2
db.m5.2xlarge	8	4	2	2, 4	1, 2
db.m5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m5.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m5d.large	2	1	2	1	1, 2
db.m5d.xlarge	4	2	2	2	1, 2
db.m5d.2xlarge	8	4	2	2, 4	1, 2
db.m5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2

DB instance class	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
db.m5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m5d.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m4.10xlarge	40	20	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20	1, 2
db.m4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r3.large	2	1	2	1	1, 2
db.r3.xlarge	4	2	2	1, 2	1, 2
db.r3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.r3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.r3.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5.large	2	1	2	1	1, 2
db.r5.xlarge	4	2	2	2	1, 2
db.r5.2xlarge	8	4	2	2, 4	1, 2
db.r5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

DB instance class	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
db.r5.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r5b.large	2	1	2	1	1, 2
db.r5b.xlarge	4	2	2	2	1, 2
db.r5b.2xlarge	8	4	2	2, 4	1, 2
db.r5b.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r5b.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5b.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r5b.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5b.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r5d.large	2	1	2	1	1, 2
db.r5d.xlarge	4	2	2	2	1, 2
db.r5d.2xlarge	8	4	2	2, 4	1, 2
db.r5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

DB instance class	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
db.r5d.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r4.large	2	1	2	1	1, 2
db.r4.xlarge	4	2	2	1, 2	1, 2
db.r4.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.r4.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.r4.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
db.r4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x1.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
db.x1e.xlarge	4	2	2	1, 2	1, 2
db.x1e.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.x1e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.x1e.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2

DB instance class	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
db.x1e.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x1e.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
db.z1d.large	2	1	2	1	1, 2
db.z1d.xlarge	4	2	2	2	1, 2
db.z1d.2xlarge	8	4	2	2, 4	1, 2
db.z1d.3xlarge	12	6	2	2, 4, 6	1, 2
db.z1d.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
db.z1d.12xlarge	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

Currently, you can configure the number of CPU cores and threads per core only when the following conditions are met:

- You are configuring an Oracle DB instance. For information about the DB instance classes supported by different Oracle database editions, see [RDS for Oracle instance classes \(p. 992\)](#)
- Your instance is using the Bring Your Own License (BYOL) licensing option. For more information about Oracle licensing options, see [Oracle licensing options \(p. 990\)](#).

Note

You can use AWS CloudTrail to monitor and audit changes to the process configuration of Amazon RDS for Oracle DB instances. For more information about using CloudTrail, see [Working with AWS CloudTrail and Amazon RDS \(p. 557\)](#).

Setting the CPU cores and threads per CPU core for a DB instance class

You can configure the number of CPU cores and threads per core for the DB instance class when you perform the following operations:

- [Creating an Amazon RDS DB instance \(p. 141\)](#)
- [Modifying an Amazon RDS DB instance \(p. 250\)](#)
- [Restoring from a DB snapshot \(p. 349\)](#)
- [Restoring a DB instance to a specified time \(p. 389\)](#)

Note

When you modify a DB instance to configure the number of CPU cores or threads per core, there is a brief DB instance outage.

You can set the CPU cores and the threads per CPU core for a DB instance class using the AWS Management Console, the AWS CLI, or the RDS API.

Console

When you are creating, modifying, or restoring a DB instance, you set the DB instance class in the AWS Management Console. The **Instance specifications** section shows options for the processor. The following image shows the processor features options.

Instance specifications

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#)

DB engine
Oracle Database Enterprise Edition

License model [Info](#)
bring-your-own-license

DB engine version [Info](#)
Oracle 12.1.0.2.v12

DB instance class [Info](#)
db.r4.xlarge — 4 vCPU, 30.5 GiB RAM

Multi-AZ deployment [Info](#)
 Create replica in different zone
Creates a replica in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.
 No

Storage type [Info](#)
Provisioned IOPS (SSD)

Allocated storage
100 GiB
(Minimum: 100 GiB, Maximum: 16384 GiB)

Provisioned IOPS [Info](#)
1000

▼ Additional configuration

Processor features

Override default values
You can change the number of CPU cores and threads per core on the DB instance class.

Core count [Info](#)
2

Threads per core [Info](#)
2

Set the following options to the appropriate values for your DB instance class under **Processor features**:

- **Core count** – Set the number of CPU cores using this option. The value must be equal to or less than the maximum number of CPU cores for the DB instance class.
- **Threads per core** – Specify **2** to enable multiple threads per core, or specify **1** to disable multiple threads per core.

When you modify or restore a DB instance, you can also set the CPU cores and the threads per CPU core to the defaults for the instance class.

When you view the details for a DB instance in the console, you can view the processor information for its DB instance class on the **Configuration** tab. The following image shows a DB instance class with one CPU core and multiple threads per core enabled.

Instance and IOPS	
Instance Class	db.r4.large
Core count	1
Threads per core	2
vCPU enabled	2
Storage Type	Provisioned IOPS (SSD)
IOPS	1000
Storage	100 GiB

For Oracle DB instances, the processor information only appears for Bring Your Own License (BYOL) DB instances.

AWS CLI

You can set the processor features for a DB instance when you run one of the following AWS CLI commands:

- [create-db-instance](#)

- [modify-db-instance](#)
- [restore-db-instance-from-db-snapshot](#)
- [restore-db-instance-from-s3](#)
- [restore-db-instance-to-point-in-time](#)

To configure the processor of a DB instance class for a DB instance by using the AWS CLI, include the `--processor-features` option in the command. Specify the number of CPU cores with the `coreCount` feature name, and specify whether multiple threads per core are enabled with the `threadsPerCore` feature name.

The option has the following syntax.

```
--processor-features "Name=coreCount,Value=<value>" "Name=threadsPerCore,Value=<value>"
```

The following are examples that configure the processor:

Examples

- [Setting the number of CPU cores for a DB instance \(p. 29\)](#)
- [Setting the number of CPU cores and disabling multiple threads for a DB instance \(p. 29\)](#)
- [Viewing the valid processor values for a DB instance class \(p. 30\)](#)
- [Returning to default processor settings for a DB instance \(p. 31\)](#)
- [Returning to the default number of CPU cores for a DB instance \(p. 31\)](#)
- [Returning to the default number of threads per core for a DB instance \(p. 32\)](#)

Setting the number of CPU cores for a DB instance

Example

The following example modifies `mydbinstance` by setting the number of CPU cores to 4. The changes are applied immediately by using `--apply-immediately`. If you want to apply the changes during the next scheduled maintenance window, omit the `--apply-immediately` option.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--processor-features "Name=coreCount,Value=4" \
--apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--processor-features "Name=coreCount,Value=4" ^
--apply-immediately
```

Setting the number of CPU cores and disabling multiple threads for a DB instance

Example

The following example modifies `mydbinstance` by setting the number of CPU cores to 4 and disabling multiple threads per core. The changes are applied immediately by using `--apply-immediately`. If you want to apply the changes during the next scheduled maintenance window, omit the `--apply-immediately` option.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \  
  --processor-features "Name=coreCount,Value=4" "Name=threadsPerCore,Value=1" \  
  --apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^  
  --processor-features "Name=coreCount,Value=4" "Name=threadsPerCore,Value=1" ^  
  --apply-immediately
```

Viewing the valid processor values for a DB instance class

Example

You can view the valid processor values for a particular DB instance class by running the [describe-orderable-db-instance-options](#) command and specifying the instance class for the --db-instance-class option. For example, the output for the following command shows the processor options for the db.r3.large instance class.

```
aws rds describe-orderable-db-instance-options --engine oracle-ee --db-instance-class db.r3.large
```

Following is sample output for the command in JSON format.

```
{  
    "SupportsIops": true,  
    "MaxIopsPerGib": 50.0,  
    "LicenseModel": "bring-your-own-license",  
    "DBInstanceClass": "db.r3.large",  
    "SupportsIAMDatabaseAuthentication": false,  
    "MinStorageSize": 100,  
    "AvailabilityZones": [  
        {  
            "Name": "us-west-2a"  
        },  
        {  
            "Name": "us-west-2b"  
        },  
        {  
            "Name": "us-west-2c"  
        }  
    ],  
    "EngineVersion": "12.1.0.2.v2",  
    "MaxStorageSize": 32768,  
    "MinIopsPerGib": 1.0,  
    "MaxIopsPerDbInstance": 40000,  
    "ReadReplicaCapable": false,  
    "AvailableProcessorFeatures": [  
        {  
            "Name": "coreCount",  
            "DefaultValue": "1",  
            "AllowedValues": "1"  
        },  
        {  
            "Name": "threadsPerCore",  
            "DefaultValue": "2",  
            "AllowedValues": "1,2"  
        }  
    ]  
},
```

```
        "SupportsEnhancedMonitoring": true,  
        "SupportsPerformanceInsights": false,  
        "MinIopsPerDbInstance": 1000,  
        "StorageType": "io1",  
        "Vpc": false,  
        "SupportsStorageEncryption": true,  
        "Engine": "oracle-ee",  
        "MultiAZCapable": true  
    }
```

In addition, you can run the following commands for DB instance class processor information:

- [describe-db-instances](#) – Shows the processor information for the specified DB instance.
- [describe-db-snapshots](#) – Shows the processor information for the specified DB snapshot.
- [describe-valid-db-instance-modifications](#) – Shows the valid modifications to the processor for the specified DB instance.

In the output of the preceding commands, the values for the processor features are not null only if the following conditions are met:

- You are using an Oracle DB instance.
- Your Oracle DB instance supports changing processor values.
- The current CPU core and thread settings are set to nondefault values.

If the preceding conditions aren't met, you can get the instance type using [describe-db-instances](#). You can get the processor information for this instance type by running the EC2 operation [describe-instance-types](#).

Returning to default processor settings for a DB instance

Example

The following example modifies `mydbinstance` by returning its DB instance class to the default processor values for it. The changes are applied immediately by using `--apply-immediately`. If you want to apply the changes during the next scheduled maintenance window, omit the `--apply-immediately` option.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \  
  --use-default-processor-features \  
  --apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^  
  --use-default-processor-features ^  
  --apply-immediately
```

Returning to the default number of CPU cores for a DB instance

Example

The following example modifies `mydbinstance` by returning its DB instance class to the default number of CPU cores for it. The threads per core setting isn't changed. The changes are applied immediately by using `--apply-immediately`. If you want to apply the changes during the next scheduled maintenance window, omit the `--apply-immediately` option.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--processor-features "Name=coreCount,Value=DEFAULT" \
--apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--processor-features "Name=coreCount,Value=DEFAULT" ^
--apply-immediately
```

Returning to the default number of threads per core for a DB instance

Example

The following example modifies `mydbinstance` by returning its DB instance class to the default number of threads per core for it. The number of CPU cores setting isn't changed. The changes are applied immediately by using `--apply-immediately`. If you want to apply the changes during the next scheduled maintenance window, omit the `--apply-immediately` option.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--processor-features "Name=threadsPerCore,Value=DEFAULT" \
--apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--processor-features "Name=threadsPerCore,Value=DEFAULT" ^
--apply-immediately
```

RDS API

You can set the processor features for a DB instance when you call one of the following Amazon RDS API operations:

- [CreateDBInstance](#)
- [ModifyDBInstance](#)
- [RestoreDBInstanceFromDBSnapshot](#)
- [RestoreDBInstanceFromS3](#)
- [RestoreDBInstanceToPointInTime](#)

To configure the processor features of a DB instance class for a DB instance by using the Amazon RDS API, include the `ProcessFeatures` parameter in the call.

The parameter has the following syntax.

```
ProcessFeatures "Name=coreCount,Value=<value>" "Name=threadsPerCore,Value=<value>"
```

Specify the number of CPU cores with the `coreCount` feature name, and specify whether multiple threads per core are enabled with the `threadsPerCore` feature name.

You can view the valid processor values for a particular instance class by running the [DescribeOrderableDBInstanceOptions](#) operation and specifying the instance class for the `DBInstanceClass` parameter. You can also use the following operations:

- [DescribeDBInstances](#) – Shows the processor information for the specified DB instance.
- [DescribeDBSnapshots](#) – Shows the processor information for the specified DB snapshot.
- [DescribeValidDBInstanceModifications](#) – Shows the valid modifications to the processor for the specified DB instance.

In the output of the preceding operations, the values for the processor features are not null only if the following conditions are met:

- You are using an Oracle DB instance.
- Your Oracle DB instance supports changing processor values.
- The current CPU core and thread settings are set to nondefault values.

If the preceding conditions aren't met, you can get the instance type using [DescribeDBInstances](#). You can get the processor information for this instance type by running the EC2 operation [DescribeInstanceTypes](#).

Hardware specifications for DB instance classes

The following terminology is used to describe hardware specifications for DB instance classes:

vCPU

The number of virtual central processing units (CPUs). A *virtual CPU* is a unit of capacity that you can use to compare DB instance classes. Instead of purchasing or leasing a particular processor to use for several months or years, you are renting capacity by the hour. Our goal is to make a consistent and specific amount of CPU capacity available, within the limits of the actual underlying hardware.

ECU

The relative measure of the integer processing power of an Amazon EC2 instance. To make it easy for developers to compare CPU capacity between different instance classes, we have defined an Amazon EC2 Compute Unit. The amount of CPU that is allocated to a particular instance is expressed in terms of these EC2 Compute Units. One ECU currently provides CPU capacity equivalent to a 1.0–1.2 GHz 2007 Opteron or 2007 Xeon processor.

Memory (GiB)

The RAM, in gibibytes, allocated to the DB instance. There is often a consistent ratio between memory and vCPU. As an example, take the db.r4 instance class, which has a memory to vCPU ratio similar to the db.r5 instance class. However, for most use cases the db.r5 instance class provides better, more consistent performance than the db.r4 instance class.

VPC Only

The instance class is supported only for DB instances that are in a VPC based on the Amazon VPC service. In some cases, you might want to use an instance class that requires a VPC but your current DB instance isn't in a VPC. In these cases, start by moving your DB instance into a VPC. For more information, see [Moving a DB instance not in a VPC into a VPC \(p. 1735\)](#).

EBS-Optimized

The DB instance uses an optimized configuration stack and provides additional, dedicated capacity for I/O. This optimization provides the best performance by minimizing contention between I/O and other traffic from your instance. For more information about Amazon EBS-optimized instances, see [Amazon EBS-Optimized instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

Max. Bandwidth (Mbps)

The maximum bandwidth in megabits per second. Divide by 8 to get the expected throughput in megabytes per second.

Important

General Purpose SSD (gp2) volumes for Amazon RDS DB instances have a throughput limit of 250 MiB/s in most cases. However, the throughput limit can vary depending on volume size. For more information, see [Amazon EBS volume types](#) in the *Amazon EC2 User Guide for Linux Instances*. For information on estimating bandwidth for gp2 storage, see [General Purpose SSD storage \(p. 40\)](#).

Network Performance

The network speed relative to other DB instance classes.

In the following table, you can find hardware details about the Amazon RDS DB instance classes.

For information about Amazon RDS DB engine support for each DB instance class, see [Supported DB engines for DB instance classes \(p. 8\)](#).

Instance class	vCPU	ECU	Memory (GiB)	VPC only	EBS optimized	Max. bandwidth (mbps)	Network performance
db.m6g – Standard instance classes powered by AWS Graviton2 processors							
db.m6g.16xlarge	64	–	256	Yes	Yes	19,000	25 Gbps
db.m6g.12xlarge	48	–	192	Yes	Yes	13,500	20 Gbps
db.m6g.8xlarge	32	–	128	Yes	Yes	9,500	12 Gbps
db.m6g.4xlarge	16	–	64	Yes	Yes	6,800	Up to 10 Gbps
db.m6g.2xlarge*	8	–	32	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.m6g.xlarge*	4	–	16	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.m6g.large*	2	–	8	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.m5d – Latest generation standard instance classes							
db.m5d.24xlarge	96	345	384	Yes	Yes	19,000	25 Gbps
db.m5d.16xlarge	64	262	256	Yes	Yes	13,600	20 Gbps
db.m5d.12xlarge	48	173	192	Yes	Yes	9,500	10 Gbps
db.m5d.8xlarge	32	131	128	Yes	Yes	6,800	10 Gbps
db.m5d.4xlarge	16	61	64	Yes	Yes	4,750	Up to 10 Gbps
db.m5d.2xlarge*	8	31	32	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.m5d.xlarge*	4	15	16	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.m5d.large*	2	10	8	Yes	Yes	Up to 4,750	Up to 10 Gbps

Instance class	vCPU	ECU	Memory (GiB)	VPC only	EBS optimized	Max. bandwidth (mbps)	Network performance
db.m5 – Latest generation standard instance classes							
db.m5.24xlarge	96	345	384	Yes	Yes	19,000	25 Gbps
db.m5.16xlarge	64	262	256	Yes	Yes	13,600	20 Gbps
db.m5.12xlarge	48	173	192	Yes	Yes	9,500	10 Gbps
db.m5.8xlarge	32	131	128	Yes	Yes	6,800	10 Gbps
db.m5.4xlarge	16	61	64	Yes	Yes	4,750	Up to 10 Gbps
db.m5.2xlarge*	8	31	32	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.m5.xlarge*	4	15	16	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.m5.large*	2	10	8	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.m4 – Standard instance classes							
db.m4.16xlarge	64	188	256	Yes	Yes	10,000	25 Gbps
db.m4.10xlarge	40	124.5	160	Yes	Yes	4,000	10 Gbps
db.m4.4xlarge	16	53.5	64	Yes	Yes	2,000	High
db.m4.2xlarge	8	25.5	32	Yes	Yes	1,000	High
db.m4.xlarge	4	13	16	Yes	Yes	750	High
db.m4.large	2	6.5	8	Yes	Yes	450	Moderate
db.m3 – Standard instance classes							
db.m3.2xlarge	8	26	30	No	Yes	1,000	High
db.m3.xlarge	4	13	15	No	Yes	500	High
db.m3.large	2	6.5	7.5	No	No	—	Moderate
db.m3.medium	1	3	3.75	No	No	—	Moderate
db.m1 – Standard instance classes							
db.m1.xlarge	4	4	15	No	Yes	450	High
db.m1.large	2	2	7.5	No	Yes	450	Moderate
db.m1.medium	1	1	3.75	No	No	—	Moderate
db.m1.small	1	1	1.7	No	No	—	Very Low
db.z1d – Memory-optimized instance classes							
db.z1d.12xlarge	48	271	384	Yes	Yes	14,000	25 Gbps

Instance class	vCPU	ECU	Memory (GiB)	VPC only	EBS optimized	Max. bandwidth (mbps)	Network performance
db.z1d.6xlarge	24	134	192	Yes	Yes	7,000	10 Gbps
db.z1d.3xlarge	12	75	96	Yes	Yes	3,500	Up to 10 Gbps
db.z1d.2xlarge	8	53	64	Yes	Yes	2,333	Up to 10 Gbps
db.z1d.xlarge*	4	28	32	Yes	Yes	Up to 2,333	Up to 10 Gbps
db.z1d.large*	2	15	16	Yes	Yes	Up to 2,333	Up to 10 Gbps
db.x1e – Memory-optimized instance classes							
db.x1e.32xlarge	128	340	3,904	Yes	Yes	14,000	25 Gbps
db.x1e.16xlarge	64	179	1,952	Yes	Yes	7,000	10 Gbps
db.x1e.8xlarge	32	91	976	Yes	Yes	3,500	Up to 10 Gbps
db.x1e.4xlarge	16	47	488	Yes	Yes	1,750	Up to 10 Gbps
db.x1e.2xlarge	8	23	244	Yes	Yes	1,000	Up to 10 Gbps
db.x1e.xlarge	4	12	122	Yes	Yes	500	Up to 10 Gbps
db.x1 – Memory-optimized instance classes							
db.x1.32xlarge	128	349	1,952	Yes	Yes	14,000	25 Gbps
db.x1.16xlarge	64	174.5	976	Yes	Yes	7,000	10 Gbps
db.r6g – Memory-optimized instance classes powered by AWS Graviton2 processors							
db.r6g.16xlarge	64	–	512	Yes	Yes	19,000	25 Gbps
db.r6g.12xlarge	48	–	384	Yes	Yes	13,500	20 Gbps
db.r6g.8xlarge	32	–	256	Yes	Yes	9,000	12 Gbps
db.r6g.4xlarge	16	–	128	Yes	Yes	4,750	Up to 10 Gbps
db.r6g.2xlarge*	8	–	64	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.r6g.xlarge*	4	–	32	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.r6g.large*	2	–	16	Yes	Yes	Up to 4,750	Up to 10 Gbps

Instance class	vCPU	ECU	Memory (GiB)	VPC only	EBS optimized	Max. bandwidth (mbps)	Network performance
db.r5d – Latest generation memory optimized instance classes							
db.r5d.24xlarge	96	347	768	Yes	Yes	19,000	25 Gbps
db.r5d.16xlarge	64	264	512	Yes	Yes	13,600	20 Gbps
db.r5d.12xlarge	48	173	384	Yes	Yes	9,500	10 Gbps
db.r5d.8xlarge	32	132	256	Yes	Yes	6,800	10 Gbps
db.r5d.4xlarge	16	71	128	Yes	Yes	4,750	Up to 10 Gbps
db.r5d.2xlarge*	8	38	64	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.r5d.xlarge*	4	19	32	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.r5d.large*	2	10	16	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.r5b – Memory-optimized instance classes							
db.r5b.24xlarge	96	347	768	Yes	Yes	60,000	25 Gbps
db.r5b.16xlarge	64	264	512	Yes	Yes	40,000	20 Gbps
db.r5b.12xlarge	48	173	384	Yes	Yes	30,000	10 Gbps
db.r5b.8xlarge	32	132	256	Yes	Yes	20,000	10 Gbps
db.r5b.4xlarge	16	71	128	Yes	Yes	10,000	Up to 10 Gbps
db.r5b.2xlarge*	8	38	64	Yes	Yes	Up to 10,000	Up to 10 Gbps
db.r5b.xlarge*	4	19	32	Yes	Yes	Up to 10,000	Up to 10 Gbps
db.r5b.large*	2	10	16	Yes	Yes	Up to 10,000	Up to 10 Gbps
db.r5 – Latest generation memory-optimized instance classes							
db.r5.24xlarge	96	347	768	Yes	Yes	19,000	25 Gbps
db.r5.16xlarge	64	264	512	Yes	Yes	13,600	20 Gbps
db.r5.12xlarge	48	173	384	Yes	Yes	9,500	10 Gbps
db.r5.8xlarge	32	132	256	Yes	Yes	6,800	10 Gbps
db.r5.4xlarge	16	71	128	Yes	Yes	4,750	Up to 10 Gbps

Instance class	vCPU	ECU	Memory (GiB)	VPC only	EBS optimized	Max. bandwidth (mbps)	Network performance
db.r5.2xlarge*	8	38	64	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.r5.xlarge*	4	19	32	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.r5.large*	2	10	16	Yes	Yes	Up to 4,750	Up to 10 Gbps
db.r4 – Memory-optimized instance classes							
db.r4.16xlarge	64	195	488	Yes	Yes	14,000	25 Gbps
db.r4.8xlarge	32	99	244	Yes	Yes	7,000	10 Gbps
db.r4.4xlarge	16	53	122	Yes	Yes	3,500	Up to 10 Gbps
db.r4.2xlarge	8	27	61	Yes	Yes	1,700	Up to 10 Gbps
db.r4.xlarge	4	13.5	30.5	Yes	Yes	850	Up to 10 Gbps
db.r4.large	2	7	15.25	Yes	Yes	425	Up to 10 Gbps
db.r3 – Memory-optimized instance classes (deprecated)							
db.r3.8xlarge	32	104	244	No	No	—	10 Gbps
db.r3.4xlarge	16	52	122	No	Yes	2,000	High
db.r3.2xlarge	8	26	61	No	Yes	1,000	High
db.r3.xlarge	4	13	30.5	No	Yes	500	Moderate
db.r3.large	2	6.5	15.25	No	No	—	Moderate
db.m2 – Memory-optimized instance classes							
db.m2.4xlarge	8	26	68.4	No	Yes	1,000	High
db.m2.2xlarge	4	13	34.2	No	Yes	500	Moderate
db.m2.xlarge	2	6.5	17.1	No	No	—	Moderate
db.t3 – Next generation burstable performance instance classes							
db.t3.2xlarge*	8	Variable	32	Yes	Yes	Up to 2,048	Up to 5 Gbps
db.t3.xlarge*	4	Variable	16	Yes	Yes	Up to 2,048	Up to 5 Gbps
db.t3.large*	2	Variable	8	Yes	Yes	Up to 2,048	Up to 5 Gbps
db.t3.medium*	2	Variable	4	Yes	Yes	Up to 1,536	Up to 5 Gbps
db.t3.small*	2	Variable	2	Yes	Yes	Up to 1,536	Up to 5 Gbps

Instance class	vCPU	ECU	Memory (GiB)	VPC only	EBS optimized	Max. bandwidth (mbps)	Network performance
db.t3.micro*	2	Variable	1	Yes	Yes	Up to 1,536	Up to 5 Gbps
db.t2 – Burstable performance instance classes							
db.t2.2xlarge	8	Variable	32	Yes	No	—	Moderate
db.t2.xlarge	4	Variable	16	Yes	No	—	Moderate
db.t2.large	2	Variable	8	Yes	No	—	Moderate
db.t2.medium	2	Variable	4	Yes	No	—	Moderate
db.t2.small	1	Variable	2	Yes	No	—	Low
db.t2.micro	1	Variable	1	Yes	No	—	Low

* These DB instance classes can support maximum performance for 30 minutes at least once every 24 hours. For more information on baseline performance of these instance types, see [Amazon EBS-optimized instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

** These DB instance classes can support maximum performance for 30 minutes at least once every 24 hours. For more information on baseline performance of these instance types, see [Amazon EBS-optimized instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

*** The r3.8xlarge instance doesn't have dedicated EBS bandwidth and therefore doesn't offer EBS optimization. On this instance, network traffic and Amazon EBS traffic share the same 10-gigabit network interface.

Amazon RDS DB instance storage

DB instances for Amazon RDS for MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server use Amazon Elastic Block Store (Amazon EBS) volumes for database and log storage. Depending on the amount of storage requested, Amazon RDS automatically stripes across multiple Amazon EBS volumes to enhance performance.

Amazon RDS storage types

Amazon RDS provides three storage types: General Purpose SSD (also known as gp2), Provisioned IOPS SSD (also known as io1), and magnetic (also known as standard). They differ in performance characteristics and price, which means that you can tailor your storage performance and cost to the needs of your database workload. You can create MySQL, MariaDB, Oracle, and PostgreSQL RDS DB instances with up to 64 tebibytes (TiB) of storage. You can create SQL Server RDS DB instances with up to 16 TiB of storage. For this amount of storage, use the Provisioned IOPS SSD and General Purpose SSD storage types.

The following list briefly describes the three storage types:

- **General Purpose SSD** – General Purpose SSD volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Baseline performance for these volumes is determined by the volume's size.

For more information about General Purpose SSD storage, including the storage size ranges, see [General Purpose SSD storage \(p. 40\)](#).

- **Provisioned IOPS** – Provisioned IOPS storage is designed to meet the needs of I/O-intensive workloads, particularly database workloads, that require low I/O latency and consistent I/O throughput.

For more information about provisioned IOPS storage, including the storage size ranges, see [Provisioned IOPS SSD storage \(p. 42\)](#).

- **Magnetic** – Amazon RDS also supports magnetic storage for backward compatibility. We recommend that you use General Purpose SSD or Provisioned IOPS for any new storage needs. The maximum amount of storage allowed for DB instances on magnetic storage is less than that of the other storage types. For more information, see [Magnetic storage \(p. 44\)](#).

Several factors can affect the performance of Amazon EBS volumes, such as instance configuration, I/O characteristics, and workload demand. For more information about getting the most out of your Provisioned IOPS volumes, see [Amazon EBS volume performance](#).

General Purpose SSD storage

General Purpose SSD storage offers cost-effective storage that is acceptable for most database workloads. The following are the storage size ranges for General Purpose SSD DB instances:

- MariaDB, MySQL, Oracle, and PostgreSQL database instances: 20 GiB–64 TiB
- SQL Server for Enterprise, Standard, Web, and Express editions: 20 GiB–16 TiB

Baseline I/O performance for General Purpose SSD storage is 3 IOPS for each GiB, with a minimum of 100 IOPS. This relationship means that larger volumes have better performance. For example, baseline performance for a 100-GiB volume is 300 IOPS. Baseline performance for a 1-TiB volume is 3,000 IOPS. And baseline performance for a 5.34-TiB volume is 16,000 IOPS.

Volumes below 1 TiB in size also have ability to burst to 3,000 IOPS for extended periods of time. Burst is not relevant for volumes above 1 TiB. Instance I/O credit balance determines burst performance. For more information about instance I/O credits, see [I/O credits and burst performance \(p. 41\)](#).

Many workloads never deplete the burst balance, making General Purpose SSD an ideal storage choice for many workloads. However, some workloads can exhaust the 3,000 IOPS burst storage credit balance, so you should plan your storage capacity to meet the needs of your workloads.

Note

DB instances that use General Purpose SSD storage can experience much longer latency after read replica creation, Multi-AZ conversion, and DB snapshot restoration than instances that use Provisioned IOPS storage. If you need a DB instance with minimum latency after these operations, we recommend using Provisioned IOPS storage.

I/O credits and burst performance

General Purpose SSD storage performance is governed by volume size, which dictates the base performance level of the volume and how quickly it accumulates I/O credits. Larger volumes have higher base performance levels and accumulate I/O credits faster. *I/O credits* represent the available bandwidth that your General Purpose SSD storage can use to burst large amounts of I/O when more than the base level of performance is needed. The more I/O credits your storage has for I/O, the more time it can burst beyond its base performance level and the better it performs when your workload requires more performance.

When using General Purpose SSD storage, your DB instance receives an initial I/O credit balance of 5.4 million I/O credits. This initial credit balance is enough to sustain a burst performance of 3,000 IOPS for 30 minutes. This balance is designed to provide a fast initial boot cycle for boot volumes and to provide a good bootstrapping experience for other applications. Volumes earn I/O credits at the baseline performance rate of 3 IOPS for each GiB of volume size. For example, a 100-GiB SSD volume has a baseline performance of 300 IOPS.

When your storage requires more than the base performance I/O level, it uses I/O credits in the I/O credit balance to burst to the required performance level. Such a burst goes to a maximum of 3,000 IOPS. Storage larger than 1,000 GiB has a base performance that is equal or greater than the maximum burst performance. When your storage uses fewer I/O credits than it earns in a second, unused I/O credits are added to the I/O credit balance. The maximum I/O credit balance for a DB instance using General Purpose SSD storage is equal to the initial I/O credit balance (5.4 million I/O credits).

Suppose that your storage uses all of its I/O credit balance. If so, its maximum performance remains at the base performance level until I/O demand drops below the base level and unused I/O credits are added to the I/O credit balance. (The *base performance level* is the rate at which your storage earns I/O credits.) The more storage, the greater the base performance is and the faster it replenishes the I/O credit balance.

Note

Storage conversions between magnetic storage and General Purpose SSD storage can potentially deplete your I/O credit balance, resulting in longer conversion times. For more information about scaling storage, see [Working with storage for Amazon RDS DB instances \(p. 316\)](#).

The following table lists several storage sizes. For each storage size, it lists the associated base performance of the storage, which is also the rate at which it accumulates I/O credits. The table also lists the burst duration at the 3,000 IOPS maximum, when starting with a full I/O credit balance. In addition, the table lists the time in seconds that the storage takes to refill an empty I/O credit balance.

Note

The IOPS figure reaches its maximum value at a volume storage size of 5,334 GiB.

Storage size (GiB)	Base performance (IOPS)	Maximum burst duration at 3,000 IOPS (seconds)	Seconds to fill empty I/O credit balance
1	100	1,862	54,000
100	300	2,000	18,000
250	750	2,400	7,200
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	Infinite	N/A
5,334	16,000	N/A	N/A

The burst duration of your storage depends on the size of the storage, the burst IOPS required, and the I/O credit balance when the burst begins. This relationship is shown in the equation following.

$$\text{Burst duration} = \frac{(\text{Credit balance})}{(\text{Burst IOPS}) - 3 * (\text{Storage size in GiB})}$$

You might notice that your storage performance is frequently limited to the base level due to an empty I/O credit balance. If so, consider allocating more General Purpose SSD storage with a higher base performance level. Alternatively, you can switch to Provisioned IOPS storage for workloads that require sustained IOPS performance.

For workloads with steady state I/O requirements, provisioning less than 100 GiB of General Purpose SSD storage might result in higher latencies if you exhaust your I/O credit balance.

Note

In general, most workloads never exceed the I/O credit balance.

For a more detailed description of how baseline performance and I/O credit balance affect performance see [Understanding burst vs. baseline performance with Amazon RDS and GP2](#).

Provisioned IOPS SSD storage

For a production application that requires fast and consistent I/O performance, we recommend Provisioned IOPS (input/output operations per second) storage. Provisioned IOPS storage is a storage type that delivers predictable performance, and consistently low latency. Provisioned IOPS storage is optimized for online transaction processing (OLTP) workloads that have consistent performance requirements. Provisioned IOPS helps performance tuning of these workloads.

Note

Your database workload might not be able to achieve 100 percent of the IOPS that you have provisioned. For more information, see [Factors that affect storage performance \(p. 45\)](#).

When you create a DB instance, you specify the IOPS rate and the size of the volume. The ratio of IOPS to allocated storage (in GiB) must be at least 0.5. Amazon RDS provides that IOPS rate for the DB instance until you change it.

The following table shows the range of Provisioned IOPS and storage size range for each database engine.

Database engine	Range of Provisioned IOPS	Range of storage
MariaDB	1,000–80,000 IOPS	100 GiB–64 TiB
SQL Server Enterprise, Standard, and Web Editions	1,000–64,000 IOPS	20 GiB–16 TiB
SQL Server Express Edition	1,000–64,000 IOPS	100 GiB–16 TiB
MySQL	1,000–80,000 IOPS	100 GiB–64 TiB
Oracle	1,000–256,000 IOPS	100 GiB–64 TiB
PostgreSQL	1,000–80,000 IOPS	100 GiB–64 TiB

Note

For SQL Server, the maximum IOPS of 64,000 is guaranteed only on [Nitro-based instances](#) that are on the m5, m5d, r5, r5b, r5d, and z1d instance types. Other instance families guarantee performance up to 32,000 IOPS.

For Oracle, the maximum IOPS of 256,000 is guaranteed only on [Nitro-based instances](#) that are on the r5b instance type. Other instance families guarantee performance up to 80,000 IOPS.

For PostgreSQL, the maximum IOPS on the db.m5.8xlarge, db.m5.16xlarge, db.r5.8xlarge, and db.r5.16xlarge instance classes is 40,000.

Important

Depending on the instance class you're using, you might see lower IOPS performance than the maximum that RDS allows you to provision. For specific information on IOPS performance for DB instance classes, see [Amazon EBS-optimized instances](#). We recommend that you determine the maximum IOPS for the instance class before setting a Provisioned IOPS value for your DB instance.

Combining Provisioned IOPS storage with Multi-AZ deployments or read replicas

For production OLTP use cases, we recommend that you use Multi-AZ deployments for enhanced fault tolerance with Provisioned IOPS storage for fast and predictable performance.

You can also use Provisioned IOPS SSD storage with read replicas for MySQL, MariaDB or PostgreSQL. The type of storage for a read replica is independent of that on the primary DB instance. For example, you might use General Purpose SSD for read replicas with a primary DB instance that uses Provisioned IOPS SSD storage to reduce costs. However, your read replica's performance in this case might differ from that of a configuration where both the primary DB instance and the read replicas use Provisioned IOPS SSD storage.

Provisioned IOPS storage costs

With Provisioned IOPS storage, you are charged for the provisioned resources whether or not you use them in a given month.

For more information about pricing, see [Amazon RDS pricing](#).

Getting the best performance from Amazon RDS Provisioned IOPS SSD storage

If your workload is I/O constrained, using Provisioned IOPS SSD storage can increase the number of I/O requests that the system can process concurrently. Increased concurrency allows for decreased latency

because I/O requests spend less time in a queue. Decreased latency allows for faster database commits, which improves response time and allows for higher database throughput.

Provisioned IOPS SSD storage provides a way to reserve I/O capacity by specifying IOPS. However, as with any other system capacity attribute, its maximum throughput under load is constrained by the resource that is consumed first. That resource might be network bandwidth, CPU, memory, or database internal resources.

Magnetic storage

Amazon RDS also supports magnetic storage for backward compatibility. We recommend that you use General Purpose SSD or Provisioned IOPS SSD for any new storage needs. The following are some limitations for magnetic storage:

- Doesn't allow you to scale storage when using the SQL Server database engine.
- Doesn't support storage autoscaling.
- Doesn't support elastic volumes.
- Limited to a maximum size of 3 TiB.
- Limited to a maximum of 1,000 IOPS.

Monitoring storage performance

Amazon RDS provides several metrics that you can use to determine how your DB instance is performing. You can view the metrics on the summary page for your instance in Amazon RDS Management Console. You can also use Amazon CloudWatch to monitor these metrics. For more information, see [Viewing DB instance metrics \(p. 548\)](#). Enhanced Monitoring provides more detailed I/O metrics; for more information, see [Using Enhanced Monitoring \(p. 471\)](#).

The following metrics are useful for monitoring storage for your DB instance:

- **IOPS** – The number of I/O operations completed each second. This metric is reported as the average IOPS for a given time interval. Amazon RDS reports read and write IOPS separately on 1-minute intervals. Total IOPS is the sum of the read and write IOPS. Typical values for IOPS range from zero to tens of thousands per second.
- **Latency** – The elapsed time between the submission of an I/O request and its completion. This metric is reported as the average latency for a given time interval. Amazon RDS reports read and write latency separately on 1-minute intervals in units of seconds. Typical values for latency are in the millisecond (ms). For example, Amazon RDS reports 2 ms as 0.002 seconds.
- **Throughput** – The number of bytes each second that are transferred to or from disk. This metric is reported as the average throughput for a given time interval. Amazon RDS reports read and write throughput separately on 1-minute intervals using units of megabytes per second (MB/s). Typical values for throughput range from zero to the I/O channel's maximum bandwidth.
- **Queue Depth** – The number of I/O requests in the queue waiting to be serviced. These are I/O requests that have been submitted by the application but have not been sent to the device because the device is busy servicing other I/O requests. Time spent waiting in the queue is a component of latency and service time (not available as a metric). This metric is reported as the average queue depth for a given time interval. Amazon RDS reports queue depth in 1-minute intervals. Typical values for queue depth range from zero to several hundred.

Measured IOPS values are independent of the size of the individual I/O operation. This means that when you measure I/O performance, you should look at the throughput of the instance, not simply the number of I/O operations.

Factors that affect storage performance

Both system activities and database workload can affect storage performance.

System activities

The following system-related activities consume I/O capacity and might reduce database instance performance while in progress:

- Multi-AZ standby creation
- Read replica creation
- Changing storage types

Database workload

In some cases your database or application design results in concurrency issues, locking, or other forms of database contention. In these cases, you might not be able to use all the provisioned bandwidth directly. In addition, you may encounter the following workload-related situations:

- The throughput limit of the underlying instance type is reached.
- Queue depth is consistently less than 1 because your application is not driving enough I/O operations.
- You experience query contention in the database even though some I/O capacity is unused.

If there isn't at least one system resource that is at or near a limit, and adding threads doesn't increase the database transaction rate, the bottleneck is most likely contention in the database. The most common forms are row lock and index page lock contention, but there are many other possibilities. If this is your situation, you should seek the advice of a database performance tuning expert.

DB instance class

To get the most performance out of your Amazon RDS database instance, choose a current generation instance type with enough bandwidth to support your storage type. For example, you can choose EBS-optimized instances and instances with 10-gigabit network connectivity.

Important

Depending on the instance class you're using, you might see lower IOPS performance than the maximum that RDS allows you to provision. For specific information on IOPS performance for DB instance classes, see [Amazon EBS-optimized instances](#). We recommend that you determine the maximum IOPS for the instance class before setting a Provisioned IOPS value for your DB instance.

We encourage you to use the latest generation of instances to get the best performance. Previous generation DB instances have a lower instance storage limit. The following table shows the maximum storage that each DB instance class can scale to for each database engine. All values are in tebibytes (TiB).

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5 – Latest Generation Standard Instance Classes					
db.m5.24xlarge	64	16	64	64	64
db.m5.16xlarge	64	16	64	64	64
db.m5.12xlarge	64	16	64	64	64

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5.8xlarge	64	16	64	64	64
db.m5.4xlarge	64	16	64	64	64
db.m5.2xlarge	64	16	64	64	64
db.m5.xlarge	64	16	64	64	64
db.m5.large	64	16	64	64	64
db.m4 – Current Generation Standard Instance Classes					
db.m4.16xlarge	64	16	64	64	64
db.m4.10xlarge	64	16	64	64	64
db.m4.4xlarge	64	16	64	64	64
db.m4.2xlarge	64	16	64	64	64
db.m4.xlarge	64	16	64	64	64
db.m4.large	64	16	64	64	64
db.m3 – Previous Generation Standard Instance Classes					
db.m3.2xlarge	6	16	6	6	6
db.m3.xlarge	6	16	6	6	6
db.m3.large	6	16	6	6	6
db.m3.medium	32	16	32	32	32
db.r5 – Latest Generation Memory Optimized Instance Classes					
db.r5.24xlarge	64	16	64	64	64
db.r5.16xlarge	64	16	64	64	64
db.r5.12xlarge	64	16	64	64	64
db.r5.8xlarge	64	16	64	64	64
db.r5.4xlarge	64	16	64	64	64
db.r5.2xlarge	64	16	64	64	64
db.r5.xlarge	64	16	64	64	64
db.r5.large	64	16	64	64	64
db.r4 – Current Generation Memory Optimized Instance Classes					
db.r4.16xlarge	64	16	64	64	64
db.r4.8xlarge	64	16	64	64	64
db.r4.4xlarge	64	16	64	64	64

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r4.2xlarge	64	16	64	64	64
db.r4.xlarge	64	16	64	64	64
db.r4.large	64	16	64	64	64
db.r3 – Previous Generation Memory Optimized Instance Classes					
db.r3.8xlarge	64	16	64	64	64
db.r3.4xlarge	64	16	64	64	64
db.r3.2xlarge	64	16	64	64	64
db.r3.xlarge	64	16	64	64	64
db.r3.large	64	16	64	64	64
db.t3 – Latest Generation Burstable Performance Instance Classes					
db.t3.2xlarge	16	16	16	64	64
db.t3.xlarge	16	16	16	64	64
db.t3.large	16	16	16	64	64
db.t3.medium	16	16	16	32	32
db.t3.small	16	16	16	32	16
db.t3.micro	16	16	16	32	16
db.t2 – Current Generation Burstable Performance Instance Classes					
db.t2.2xlarge	64	16	64	64	64
db.t2.xlarge	64	16	64	64	64
db.t2.large	64	16	64	64	64
db.t2.medium	32	16	32	32	32
db.t2.small	16	16	16	16	16
db.t2.micro	16	16	16	16	16
db.x1e – Latest Generation Memory Optimized Instance Classes					
db.x1e.32xlarge		16		64	
db.x1e.16xlarge		16		64	
db.x1e.8xlarge		16		64	
db.x1e.4xlarge		16		64	
db.x1e.2xlarge		16		64	
db.x1e.xlarge		16		64	

Instance class	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x1 – Current Generation Memory Optimized Instance Classes					
db.x1.32xlarge		16		64	
db.x1.16xlarge		16		64	

For Oracle, scaling up to 80,000 IOPS is only supported on the following instance classes.

- db.m5.24xlarge
- db.r5.24xlarge
- db.x1.32xlarge
- db.x1e.32xlarge

For more details on all instance classes supported, see [Previous generation DB instances](#).

Regions, Availability Zones, and Local Zones

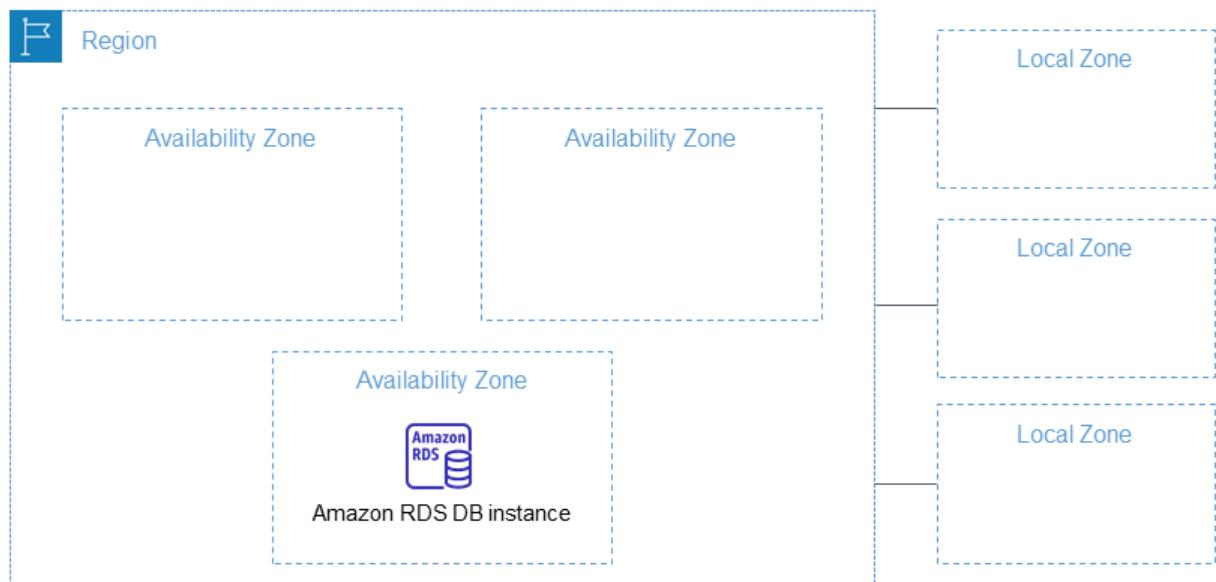
Amazon cloud computing resources are hosted in multiple locations world-wide. These locations are composed of AWS Regions, Availability Zones, and Local Zones. Each *AWS Region* is a separate geographic area. Each AWS Region has multiple, isolated locations known as *Availability Zones*.

Note

For information about finding the Availability Zones for an AWS Region, see [Describing your Regions, Availability Zones, and Local Zones](#) in the Amazon EC2 documentation.

By using Local Zones, you can place resources, such as compute and storage, in multiple locations closer to your users. Amazon RDS enables you to place resources, such as DB instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically.

Amazon operates state-of-the-art, highly-available data centers. Although rare, failures can occur that affect the availability of instances that are in the same location. If you host all your instances in a single location that is affected by such a failure, none of your instances is available.



It is important to remember that each AWS Region is completely independent. Any Amazon RDS activity you initiate (for example, creating database instances or listing available database instances) runs only in your current default AWS Region. The default AWS Region can be changed in the console, by setting the `AWS_DEFAULT_REGION` environment variable, or it can be overridden by using the `--region` parameter with the AWS Command Line Interface (AWS CLI). For more information, see [Configuring the AWS Command Line Interface](#), specifically the sections about environment variables and command line options.

Amazon RDS supports special AWS Regions called AWS GovCloud (US) that are designed to allow US government agencies and customers to move more sensitive workloads into the cloud. The AWS GovCloud (US) Regions address the US government's specific regulatory and compliance requirements. For more information, see [What is AWS GovCloud \(US\)?](#)

To create or work with an Amazon RDS DB instance in a specific AWS Region, use the corresponding regional service endpoint.

AWS Regions

Each AWS Region is designed to be isolated from the other AWS Regions. This design achieves the greatest possible fault tolerance and stability.

When you view your resources, you see only the resources that are tied to the AWS Region that you specified. This is because AWS Regions are isolated from each other, and we don't automatically replicate resources across AWS Regions.

Region availability

The following table shows the AWS Regions where Amazon RDS is currently available and the endpoint for each Region.

Region Name	Region	Endpoint	Protocol	
US East (Ohio)	us-east-2	rds.us-east-2.amazonaws.com rds-fips.us-east-2.amazonaws.com	HTTPS HTTPS	
US East (N. Virginia)	us-east-1	rds.us-east-1.amazonaws.com rds-fips.us-east-1.amazonaws.com	HTTPS HTTPS	
US West (N. California)	us-west-1	rds.us-west-1.amazonaws.com rds-fips.us-west-1.amazonaws.com	HTTPS HTTPS	
US West (Oregon)	us-west-2	rds.us-west-2.amazonaws.com rds-fips.us-west-2.amazonaws.com	HTTPS HTTPS	
Africa (Cape Town)	af-south-1	rds.af-south-1.amazonaws.com	HTTPS	
Asia Pacific (Hong Kong)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS	
Asia Pacific (Mumbai)	ap-south-1	rds.ap-south-1.amazonaws.com	HTTPS	
Asia Pacific (Osaka)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com	HTTPS	
Asia Pacific (Seoul)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com	HTTPS	
Asia Pacific (Singapore)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com	HTTPS	
Asia Pacific (Sydney)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com	HTTPS	

Region Name	Region	Endpoint	Protocol	
Asia Pacific (Tokyo)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS	
Canada (Central)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS	
		rds-fips.ca-central-1.amazonaws.com	HTTPS	
China (Beijing)	cn-north-1	rds.cn-north-1.amazonaws.com.cn	HTTPS	
China (Ningxia)	cn-northwest-1	rds.cn-northwest-1.amazonaws.com.cn	HTTPS	
Europe (Frankfurt)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS	
Europe (Ireland)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS	
Europe (London)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS	
Europe (Milan)	eu-south-1	rds.eu-south-1.amazonaws.com	HTTPS	
Europe (Paris)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS	
Europe (Stockholm)	eu-north-1	rds.eu-north-1.amazonaws.com	HTTPS	
Middle East (Bahrain)	me-south-1	rds.me-south-1.amazonaws.com	HTTPS	
South America (São Paulo)	sa-east-1	rds.sa-east-1.amazonaws.com	HTTPS	
AWS GovCloud (US-East)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com	HTTPS	
AWS GovCloud (US-West)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com	HTTPS	

If you do not explicitly specify an endpoint, the US West (Oregon) endpoint is the default.

When you work with a DB instance using the AWS CLI or API operations, make sure that you specify its regional endpoint.

Availability Zones

When you create a DB instance, you can choose an Availability Zone or have Amazon RDS choose one for you randomly. An Availability Zone is represented by an AWS Region code followed by a letter identifier (for example, us-east-1a).

You can't choose the Availability Zones for the primary and secondary DB instances in a Multi-AZ DB deployment. Amazon RDS chooses them for you randomly. For more information about Multi-AZ deployments, see [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).

Note

Random selection of Availability Zones by RDS doesn't guarantee an even distribution of DB instances among Availability Zones within a single account or DB subnet group. You can request a specific AZ when you create or modify a Single-AZ instance, and you can use more-specific DB subnet groups for Multi-AZ instances. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#) and [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Local Zones

A *Local Zone* is an extension of an AWS Region that is geographically close to your users. You can extend any VPC from the parent AWS Region into Local Zones by creating a new subnet and assigning it to the AWS Local Zone. When you create a subnet in a Local Zone, your VPC is extended to that Local Zone. The subnet in the Local Zone operates the same as other subnets in your VPC.

When you create a DB instance, you can choose a subnet in a Local Zone. Local Zones have their own connections to the internet and support AWS Direct Connect. Thus, resources created in a Local Zone can serve local users with very low-latency communications. For more information, see [AWS Local Zones](#).

A Local Zone is represented by an AWS Region code followed by an identifier that indicates the location, for example us-west-2-lax-1a.

Note

A Local Zone can't be included in a Multi-AZ deployment.

To use a Local Zone

1. Enable the Local Zone in the Amazon EC2 console.

For more information, see [Enabling Local Zones in the Amazon EC2 User Guide for Linux Instances](#).

2. Create a subnet in the Local Zone.

For more information, see [Creating a subnet in your VPC](#) in the *Amazon VPC User Guide*.

3. Create a DB subnet group in the Local Zone.

When you create a DB subnet group, choose the Availability Zone group for the Local Zone.

For more information, see [Creating a DB instance in a VPC \(p. 1730\)](#).

4. Create a DB instance that uses the DB subnet group in the Local Zone.

For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

Important

Currently, Local Zones are only available in the US West (Oregon) Region. In this AWS Region, the Los Angeles AWS Local Zone is available.

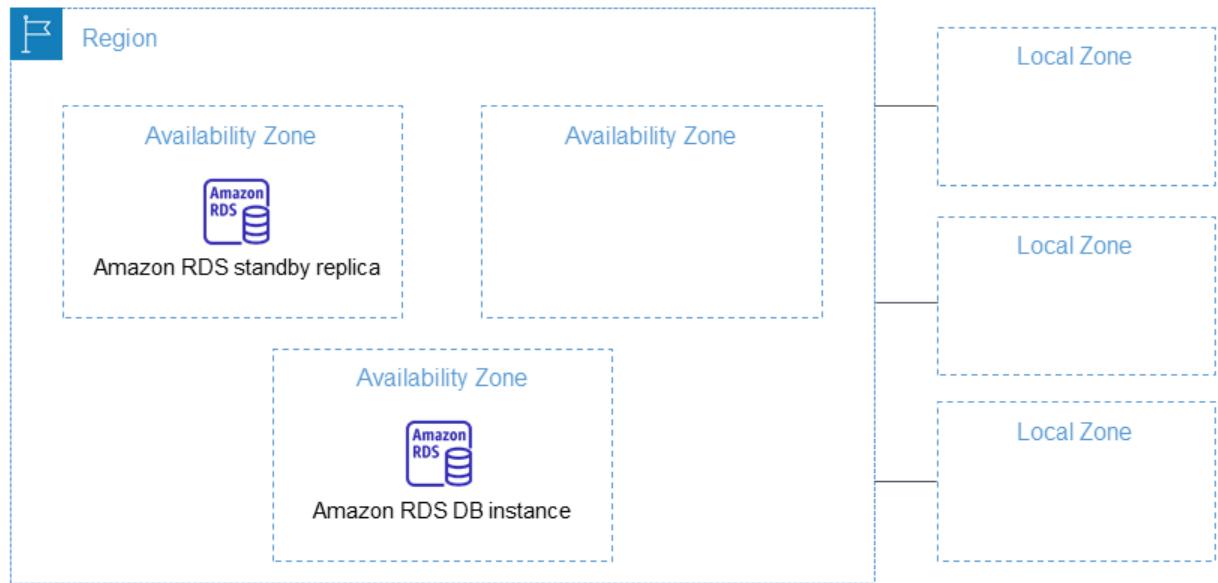
High availability (Multi-AZ) for Amazon RDS

Amazon RDS provides high availability and failover support for DB instances using Multi-AZ deployments. Amazon RDS uses several different technologies to provide failover support. Multi-AZ deployments for MariaDB, MySQL, Oracle, and PostgreSQL DB instances use Amazon's failover technology. SQL Server DB instances use SQL Server Database Mirroring (DBM) or Always On Availability Groups (AGs). For information on SQL Server version support for Multi-AZ, see [Multi-AZ deployments for Microsoft SQL Server \(p. 698\)](#).

In a Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone. The primary DB instance is synchronously replicated across Availability Zones to a standby replica to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups. Running a DB instance with high availability can enhance availability during planned system maintenance, and help protect your databases against DB instance failure and Availability Zone disruption. For more information on Availability Zones, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

Note

The high-availability feature isn't a scaling solution for read-only scenarios; you can't use a standby replica to serve read traffic. To serve read-only traffic, you use a read replica instead. For more information, see [Working with read replicas \(p. 278\)](#).



Using the RDS console, you can create a Multi-AZ deployment by simply specifying Multi-AZ when creating a DB instance. You can use the console to convert existing DB instances to Multi-AZ deployments by modifying the DB instance and specifying the Multi-AZ option. You can also specify a Multi-AZ deployment with the AWS CLI or Amazon RDS API. Use the [create-db-instance](#) or [modify-db-instance](#) CLI command, or the [CreateDBInstance](#) or [ModifyDBInstance](#) API operation.

The RDS console shows the Availability Zone of the standby replica (called the secondary AZ). You can also use the [describe-db-instances](#) CLI command or the [DescribeDBInstances](#) API operation to find the secondary AZ.

DB instances using Multi-AZ deployments can have increased write and commit latency compared to a Single-AZ deployment, due to the synchronous data replication that occurs. You might have a change in latency if your deployment fails over to the standby replica, although AWS is engineered with low-latency network connectivity between Availability Zones. For production workloads, we recommend

that you use Provisioned IOPS and DB instance classes that are optimized for Provisioned IOPS for fast, consistent performance. For more information about DB instance classes, see [DB instance classes \(p. 7\)](#).

Modifying a DB instance to be a Multi-AZ deployment

If you have a DB instance in a Single-AZ deployment and modify it to a Multi-AZ deployment (for engines other than Amazon Aurora), Amazon RDS takes several steps. First, Amazon RDS takes a snapshot of the primary DB instance from your deployment and then restores the snapshot into another Availability Zone. Amazon RDS then sets up synchronous replication between your primary DB instance and the new instance.

For information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Important

This action avoids downtime when you convert from Single-AZ to Multi-AZ, but you can experience a performance impact during and after converting to Multi-AZ. This impact can be significant for large write-intensive DB instances.

To enable Multi-AZ for a DB instance, RDS takes a snapshot of the primary DB instance's EBS volume and restores it on the newly created standby replica, and then synchronizes both volumes. New volumes created from existing EBS snapshots load lazily in the background. This capability permits large volumes to be restored from a snapshot quickly, but there is the possibility of added latency during and after the modification is complete. For more information, see [Restoring an Amazon EBS volume from a snapshot](#) in the Amazon EC2 documentation.

After the modification is complete, Amazon RDS triggers an event (RDS-EVENT-0025) that indicates the process is complete. You can monitor Amazon RDS events; for more information about events, see [Using Amazon RDS event notification \(p. 487\)](#).

Failover process for Amazon RDS

In the event of a planned or unplanned outage of your DB instance, Amazon RDS automatically switches to a standby replica in another Availability Zone if you have enabled Multi-AZ. The time it takes for the failover to complete depends on the database activity and other conditions at the time the primary DB instance became unavailable. Failover times are typically 60–120 seconds. However, large transactions or a lengthy recovery process can increase failover time. When the failover is complete, it can take additional time for the RDS console to reflect the new Availability Zone.

Note

You can force a failover manually when you reboot a DB instance. For more information, see [Rebooting a DB instance \(p. 276\)](#).

Amazon RDS handles failovers automatically so you can resume database operations as quickly as possible without administrative intervention. The primary DB instance switches over automatically to the standby replica if any of the conditions described in the following table occurs. You can view these failover reasons in the event log.

Failover reason	Description
The operating system underlying the RDS database instance is being patched in an offline operation.	A failover was triggered during the maintenance window for an OS patch or a security update. For more information, see Maintaining a DB instance (p. 264) .
The primary host of the RDS Multi-AZ instance is unhealthy.	The Multi-AZ deployment detected an impaired primary DB instance and failed over.

Failover reason	Description
The primary host of the RDS Multi-AZ instance is unreachable due to loss of network connectivity.	RDS monitoring detected a network reachability failure to the primary DB instance and triggered a failover.
The RDS instance was modified by customer.	<p>An RDS DB instance modification triggered a failover.</p> <p>For more information, see Modifying an Amazon RDS DB instance (p. 250).</p>
The RDS Multi-AZ primary instance is busy and unresponsive.	<p>The primary DB instance is unresponsive. We recommend that you do the following:</p> <ul style="list-style-type: none"> Examine the event and CloudWatch logs for excessive CPU, memory, or swap space usage. For more information, see Using Amazon RDS event notification (p. 487) and Getting CloudWatch Events and Amazon EventBridge events for Amazon RDS (p. 551). Evaluate your workload to determine whether you're using the appropriate DB instance class. For more information, see DB instance classes (p. 7). Use Enhanced Monitoring for real-time operating system metrics. For more information, see Using Enhanced Monitoring (p. 471). Use Performance Insights to help analyze any issues that affect your DB instance's performance. For more information, see Using Performance Insights on Amazon RDS (p. 412). <p>For more information on these recommendations, see Overview of monitoring Amazon RDS (p. 400) and Best practices for Amazon RDS (p. 128).</p>
The storage volume underlying the primary host of the RDS Multi-AZ instance experienced a failure.	The Multi-AZ deployment detected a storage issue on the primary DB instance and failed over.
The user requested a failover of the DB instance.	<p>You rebooted the DB instance and chose Reboot with failover.</p> <p>For more information, see Rebooting a DB instance (p. 276).</p>

There are several ways to determine if your Multi-AZ DB instance has failed over:

- DB event subscriptions can be set up to notify you by email or SMS that a failover has been initiated. For more information about events, see [Using Amazon RDS event notification \(p. 487\)](#).
- You can view your DB events by using the Amazon RDS console or API operations.

- You can view the current state of your Multi-AZ deployment by using the Amazon RDS console and API operations.

For information on how you can respond to failovers, reduce recovery time, and other best practices for Amazon RDS, see [Best practices for Amazon RDS \(p. 128\)](#).

Setting the JVM TTL for DNS name lookups

The failover mechanism automatically changes the Domain Name System (DNS) record of the DB instance to point to the standby DB instance. As a result, you need to re-establish any existing connections to your DB instance. In a Java virtual machine (JVM) environment, due to how the Java DNS caching mechanism works, you might need to reconfigure JVM settings.

The JVM caches DNS name lookups. When the JVM resolves a hostname to an IP address, it caches the IP address for a specified period of time, known as the *time-to-live* (TTL).

Because AWS resources use DNS name entries that occasionally change, we recommend that you configure your JVM with a TTL value of no more than 60 seconds. Doing this makes sure that when a resource's IP address changes, your application can receive and use the resource's new IP address by requerying the DNS.

On some Java configurations, the JVM default TTL is set so that it never refreshes DNS entries until the JVM is restarted. Thus, if the IP address for an AWS resource changes while your application is still running, it can't use that resource until you manually restart the JVM and the cached IP information is refreshed. In this case, it's crucial to set the JVM's TTL so that it periodically refreshes its cached IP information.

Note

The default TTL can vary according to the version of your JVM and whether a security manager is installed. Many JVMs provide a default TTL less than 60 seconds. If you're using such a JVM and not using a security manager, you can ignore the rest of this topic. For more information on security managers in Oracle, see [The security manager](#) in the Oracle documentation.

To modify the JVM's TTL, set the `networkaddress.cache.ttl` property value. Use one of the following methods, depending on your needs:

- To set the property value globally for all applications that use the JVM, set `networkaddress.cache.ttl` in the `$JAVA_HOME/jre/lib/security/java.security` file.

```
networkaddress.cache.ttl=60
```

- To set the property locally for your application only, set `networkaddress.cache.ttl` in your application's initialization code before any network connections are established.

```
java.security.Security.setProperty("networkaddress.cache.ttl" , "60");
```

DB instance billing for Amazon RDS

Amazon RDS instances are billed based on the following components:

- DB instance hours (per hour) – Based on the DB instance class of the DB instance (for example, db.t2.small or db.m4.large). Pricing is listed on a per-hour basis, but bills are calculated down to the second and show times in decimal form. RDS usage is billed in one second increments, with a minimum of 10 minutes. For more information, see [DB instance classes \(p. 7\)](#).
- Storage (per GiB per month) – Storage capacity that you have provisioned to your DB instance. If you scale your provisioned storage capacity within the month, your bill is pro-rated. For more information, see [Amazon RDS DB instance storage \(p. 40\)](#).
- I/O requests (per 1 million requests per month) – Total number of storage I/O requests that you have made in a billing cycle, for Amazon RDS magnetic storage only.
- Provisioned IOPS (per IOPS per month) – Provisioned IOPS rate, regardless of IOPS consumed, for Amazon RDS Provisioned IOPS (SSD) storage only. Provisioned storage for EBS volumes are billed in one second increments, with a minimum of 10 minutes.
- Backup storage (per GiB per month) – *Backup storage* is the storage that is associated with automated database backups and any active database snapshots that you have taken. Increasing your backup retention period or taking additional database snapshots increases the backup storage consumed by your database. Per second billing doesn't apply to backup storage (metered in GB-month).

For more information, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#).

- Data transfer (per GB) – Data transfer in and out of your DB instance from or to the internet and other AWS Regions.

Amazon RDS provides the following purchasing options to enable you to optimize your costs based on your needs:

- **On-Demand Instances** – Pay by the hour for the DB instance hours that you use. Pricing is listed on a per-hour basis, but bills are calculated down to the second and show times in decimal form. RDS usage is now billed in one second increments, with a minimum of 10 minutes.
- **Reserved Instances** – Reserve a DB instance for a one-year or three-year term and get a significant discount compared to the on-demand DB instance pricing. With Reserved Instance usage, you can launch, delete, start, or stop multiple instances within an hour and get the Reserved Instance benefit for all of the instances.

For Amazon RDS pricing information, see the [Amazon RDS product page](#).

Topics

- [On-Demand DB instances for Amazon RDS \(p. 58\)](#)
- [Reserved DB instances for Amazon RDS \(p. 59\)](#)

On-Demand DB instances for Amazon RDS

Amazon RDS on-demand DB instances are billed based on the class of the DB instance (for example, db.t2.small or db.m4.large). For Amazon RDS pricing information, see the [Amazon RDS product page](#).

Billing starts for a DB instance as soon as the DB instance is available. Pricing is listed on a per-hour basis, but bills are calculated down to the second and show times in decimal form. Amazon RDS usage is billed in one-second increments, with a minimum of 10 minutes. In the case of billable configuration change, such as scaling compute or storage capacity, you're charged a 10-minute minimum. Billing continues until the DB instance terminates, which occurs when you delete the DB instance or if the DB instance fails.

If you no longer want to be charged for your DB instance, you must stop or delete it to avoid being billed for additional DB instance hours. For more information about the DB instance states for which you are billed, see [DB instance status \(p. 404\)](#).

Stopped DB instances

While your DB instance is stopped, you're charged for provisioned storage, including Provisioned IOPS. You are also charged for backup storage, including storage for manual snapshots and automated backups within your specified retention window. You aren't charged for DB instance hours.

Multi-AZ DB instances

If you specify that your DB instance should be a Multi-AZ deployment, you're billed according to the Multi-AZ pricing posted on the Amazon RDS pricing page.

Reserved DB instances for Amazon RDS

Using reserved DB instances, you can reserve a DB instance for a one- or three-year term. Reserved DB instances provide you with a significant discount compared to on-demand DB instance pricing. Reserved DB instances are not physical instances, but rather a billing discount applied to the use of certain on-demand DB instances in your account. Discounts for reserved DB instances are tied to instance type and AWS Region.

The general process for working with reserved DB instances is: First get information about available reserved DB instance offerings, then purchase a reserved DB instance offering, and finally get information about your existing reserved DB instances.

Overview of reserved DB instances

When you purchase a reserved DB instance in Amazon RDS, you purchase a commitment to getting a discounted rate, on a specific DB instance type, for the duration of the reserved DB instance. To use an Amazon RDS reserved DB instance, you create a new DB instance just like you do for an on-demand instance. The new DB instance that you create must match the specifications of the reserved DB instance. If the specifications of the new DB instance match an existing reserved DB instance for your account, you are billed at the discounted rate offered for the reserved DB instance. Otherwise, the DB instance is billed at an on-demand rate.

For more information about reserved DB instances, including pricing, see [Amazon RDS reserved instances](#).

Offering types

Reserved DB instances are available in three varieties—No Upfront, Partial Upfront, and All Upfront—that let you optimize your Amazon RDS costs based on your expected usage.

No Upfront

This option provides access to a reserved DB instance without requiring an upfront payment. Your No Upfront reserved DB instance bills a discounted hourly rate for every hour within the term, regardless of usage, and no upfront payment is required. This option is only available as a one-year reservation.

Partial Upfront

This option requires a part of the reserved DB instance to be paid upfront. The remaining hours in the term are billed at a discounted hourly rate, regardless of usage. This option is the replacement for the previous Heavy Utilization option.

All Upfront

Full payment is made at the start of the term, with no other costs incurred for the remainder of the term regardless of the number of hours used.

If you are using consolidated billing, all the accounts in the organization are treated as one account. This means that all accounts in the organization can receive the hourly cost benefit of reserved DB instances that are purchased by any other account. For more information about consolidated billing, see [Amazon RDS reserved DB instances](#) in the *AWS Billing and Cost Management User Guide*.

Size-flexible reserved DB instances

When you purchase a reserved DB instance, one thing that you specify is the instance class, for example db.m4.large. For more information about instance classes, see [DB instance classes \(p. 7\)](#).

If you have a DB instance, and you need to scale it to larger capacity, your reserved DB instance is automatically applied to your scaled DB instance. That is, your reserved DB instances are automatically

applied across all DB instance class sizes. Size-flexible reserved DB instances are available for DB instances with the same AWS Region and database engine. Size-flexible reserved DB instances can only scale in their instance class type. For example, a reserved DB instance for a db.m4.large can apply to a db.m4.xlarge, but not to a db.m5.large, because db.m4 and db.m5 are different instance class types.

Reserved DB instance benefits also apply for both Multi-AZ and Single-AZ configurations. Flexibility means that you can move freely between configurations within the same DB instance class type. For example, you can move from a Single-AZ deployment running on one large DB instance (four normalized units) to a Multi-AZ deployment running on two small DB instances ($2 \times 2 = 4$ normalized units).

Size-flexible reserved DB instances are available for the following Amazon RDS database engines:

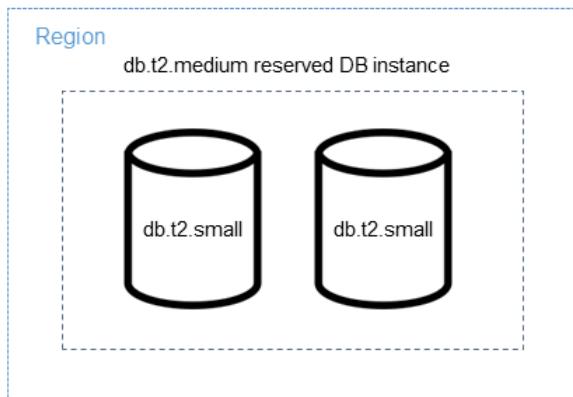
- MariaDB
- MySQL
- Oracle, Bring Your Own License
- PostgreSQL

For details about using size-flexible reserved instances with Aurora, see [Reserved DB instances for Aurora](#).

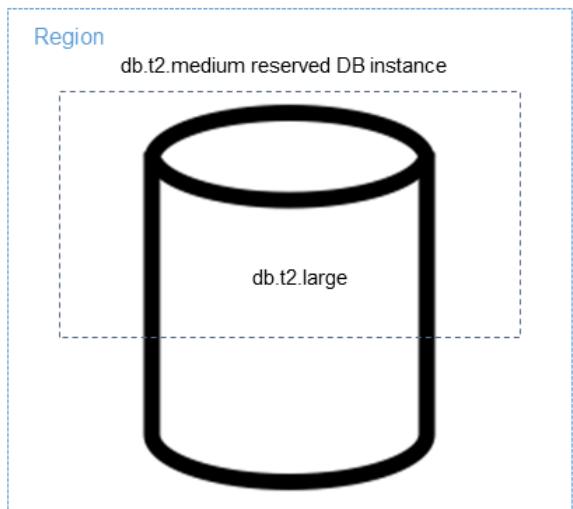
You can compare usage for different reserved DB instance sizes by using normalized units. For example, one unit of usage on two db.m3.large DB instances is equivalent to eight normalized units of usage on one db.m3.small. The following table shows the number of normalized units for each DB instance size.

Instance size	Single-AZ normalized units	Multi-AZ normalized units
micro	0.5	1
small	1	2
medium	2	4
large	4	8
xlarge	8	16
2xlarge	16	32
4xlarge	32	64
6xlarge	48	96
8xlarge	64	128
10xlarge	80	160
12xlarge	96	192
16xlarge	128	256
24xlarge	192	384
32xlarge	256	512

For example, suppose that you purchase a db.t2.medium reserved DB instance, and you have two running db.t2.small DB instances in your account in the same AWS Region. In this case, the billing benefit is applied in full to both instances.



Alternatively, if you have one db.t2.large instance running in your account in the same AWS Region, the billing benefit is applied to 50 percent of the usage of the DB instance.



Reserved DB instance billing example

The price for a reserved DB instance doesn't include regular costs associated with storage, backups, and I/O. The following example illustrates the total cost per month for a reserved DB instance:

- An RDS for MySQL reserved Single-AZ db.r4.large DB instance class in US East (N. Virginia) with the No Upfront option at a cost of \$0.12 for the instance, or \$90 per month
- 400 GiB of General Purpose SSD (gp2) storage at a cost of 0.115 per GiB per month, or \$45.60 per month
- 600 GiB of backup storage at \$0.095, or \$19 per month (400 GiB free)

Add all of these options (\$90 + \$45.60 + \$19) with the reserved DB instance, and the total cost per month is \$154.60.

If you chose to use an on-demand DB instance instead of a reserved DB instance, an RDS for MySQL Single-AZ db.r4.large DB instance class in US East (N. Virginia) costs \$0.1386 per hour, or \$101.18 per month. So, for an on-demand DB instance, add all of these options (\$101.18 + \$45.60 + \$19), and the total cost per month is \$165.78.

Note

The prices in this example are sample prices and might not match actual prices. For Amazon RDS pricing information, see the [Amazon RDS product page](#).

Deleting a reserved DB instance

The terms for a reserved DB instance involve a one-year or three-year commitment. You can't cancel a reserved DB instance. However, you can delete a DB instance that is covered by a reserved DB instance discount. The process for deleting a DB instance that is covered by a reserved DB instance discount is the same as for any other DB instance.

Your upfront payment for a reserved DB instance reserves the resources for your use. Because these resources are reserved for you, you are billed for the resources regardless of whether you use them.

If you delete a DB instance that is covered by a reserved DB instance discount, you can launch another DB instance with compatible specifications. In this case, you continue to get the discounted rate during the reservation term (one or three years).

Working with reserved DB instances

You can use the AWS Management Console, the AWS CLI, and the RDS API to work with reserved DB instances.

Console

You can use the AWS Management Console to work with reserved DB instances as shown in the following procedures.

To get pricing and information about available reserved DB instance offerings

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Reserved instances**.
3. Choose **Purchase Reserved DB Instance**.
4. For **Product description**, choose the DB engine and licensing type.
5. For **DB instance class**, choose the DB instance class.
6. For **Multi-AZ deployment**, choose whether you want a Multi-AZ deployment.
7. For **Term**, choose the length of time you want the DB instance reserved.
8. For **Offering type**, choose the offering type.

After you select the offering type, you can see the pricing information.

Important

Choose **Cancel** to avoid purchasing the reserved DB instance and incurring any charges.

After you have information about the available reserved DB instance offerings, you can use the information to purchase an offering as shown in the following procedure.

To purchase a reserved DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Reserved instances**.

3. Choose **Purchase Reserved DB Instance**.
4. For **Product description**, choose the DB engine and licensing type.
5. For **DB instance class**, choose the DB instance class.
6. For **Multi-AZ deployment**, choose whether you want a Multi-AZ deployment.
7. For **Term**, choose the length of time you want the DB instance reserved.
8. For **Offering type**, choose the offering type.

After you choose the offering type, you can see the pricing information.

RDS > Reserved instances > Purchase Reserved DB Instances

Purchase Reserved DB Instances

Choose from the options below, then enter the number of DB instances you wish to reserve with this order. When you are done, click the Continue button.

Options

Product description: sqlserver-ex(l1)

DB instance class: db.t3.small — 2 vCPU, 2 GiB RAM

Multi AZ deployment: Multi-AZ deployment model is not applicable for this database engine and edition

Term: 1 year

Offering type: Partial Upfront

Reserved Id (optional): Optional tag to track your reservation

Number of DB instances: Specify the number of DB instances you wish to reserve with this order

1

Pricing details

One-time payment (per instance)	Usage charges*
[Redacted]	(hourly) <small>*Additional taxes may apply</small>
Total one-time payment*	This hourly rate is charged for every hour for each instance in the Reserved Instance term you purchase, regardless of instance usage
*Additional taxes may apply	
Charges for your usage will appear on your monthly bill.	

Cancel **Continue**

9. (Optional) You can assign your own identifier to the reserved DB instances that you purchase to help you track them. For **Reserved Id**, type an identifier for your reserved DB instance.
10. Choose **Continue**.

The **Purchase Reserved DB Instances** dialog box appears, with a summary of the reserved DB instance attributes that you've selected and the payment due.

The screenshot shows the 'Purchase Reserved DB Instances' page. At the top, there's a breadcrumb trail: RDS > Reserved instances > Purchase Reserved DB Instances. The main title is 'Purchase Reserved DB Instances'. Below it is a 'Summary of Purchase' section with the following details:

Region	US West (Oregon)
Product Description	sqlserver-ex(l1)
DB Instance Class	db.t3.small
Offering Type	Partial Upfront
Multi AZ Deployment	No
Term	1 year
Reserved DB Instance	default
Quantity	1
Price Per Instance	[Redacted]
Total Payment Due Now	[Redacted]

At the bottom of the summary section, there's a warning message: **⚠️ Purchasing this Reserved DB Instance will charge [Redacted] to the payment method associated with this Amazon Web Services account. Are you sure you would like to proceed?**

At the very bottom right, there are three buttons: 'Cancel', 'Back', and a highlighted orange button labeled 'Order'.

11. On the confirmation page, review your reserved DB instance. If the information is correct, choose **Order** to purchase the reserved DB instance.

Alternatively, choose **Back** to edit your reserved DB instance.

After you have purchased reserved DB instances, you can get information about your reserved DB instances as shown in the following procedure.

To get information about reserved DB instances for your AWS account

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the Navigation pane, choose **Reserved instances**.

The reserved DB instances for your account appear. To see detailed information about a particular reserved DB instance, choose that instance in the list. You can then see detailed information about that instance in the detail pane at the bottom of the console.

AWS CLI

You can use the AWS CLI to work with reserved DB instances as shown in the following examples.

Example of getting available reserved DB instance offerings

To get information about available reserved DB instance offerings, call the AWS CLI command [describe-reserved-db-instances-offerings](#).

```
aws rds describe-reserved-db-instances-offerings
```

This call returns output similar to the following:

OFFERING	OfferingId	Class	Multi-AZ	Duration	Fixed
Price	Usage Price	Description	Offering Type		
OFFERING	438012d3-4052-4cc7-b2e3-8d3372e0e706	db.m1.large	y	1y	1820.00
USD	0.368 USD	mysql	Partial Upfront		
OFFERING	649fd0c8-cf6d-47a0-bfa6-060f8e75e95f	db.m1.small	n	1y	227.50
USD	0.046 USD	mysql	Partial Upfront		
OFFERING	123456cd-ab1c-47a0-bfa6-12345667232f	db.m1.small	n	1y	162.00
USD	0.00 USD	mysql	All Upfront		
Recurring Charges:	Amount	Currency	Frequency		
Recurring Charges:	0.123	USD	Hourly		
OFFERING	123456cd-ab1c-37a0-bfa6-12345667232d	db.m1.large	y	1y	700.00
USD	0.00 USD	mysql	All Upfront		
Recurring Charges:	Amount	Currency	Frequency		
Recurring Charges:	1.25	USD	Hourly		
OFFERING	123456cd-ab1c-17d0-bfa6-12345667234e	db.m1.xlarge	n	1y	4242.00
USD	2.42 USD	mysql	No Upfront		

After you have information about the available reserved DB instance offerings, you can use the information to purchase an offering.

To purchase a reserved DB instance, use the AWS CLI command [purchase-reserved-db-instances-offering](#) with the following parameters:

- **--reserved-db-instances-offering-id** – The ID of the offering that you want to purchase. See the preceding example to get the offering ID.
- **--reserved-db-instance-id** – You can assign your own identifier to the reserved DB instances that you purchase to help track them.

Example of purchasing a reserved DB instance

The following example purchases the reserved DB instance offering with ID [649fd0c8-cf6d-47a0-bfa6-060f8e75e95f](#), and assigns the identifier of [MyReservation](#).

For Linux, macOS, or Unix:

```
aws rds purchase-reserved-db-instances-offering \
--reserved-db-instances-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f \
--reserved-db-instance-id MyReservation
```

For Windows:

```
aws rds purchase-reserved-db-instances-offering ^
--reserved-db-instances-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^
--reserved-db-instance-id MyReservation
```

The command returns output similar to the following:

RESERVATION	ReservationId	Class	Multi-AZ	Start Time	Duration
Fixed Price	Usage Price	Count	State	Description	Offering Type
RESERVATION	MyReservation	db.m1.small	y	2011-12-19T00:30:23.247Z	1y
455.00 USD	0.092 USD	1	payment-pending	mysql	Partial Upfront

After you have purchased reserved DB instances, you can get information about your reserved DB instances.

To get information about reserved DB instances for your AWS account, call the AWS CLI command [describe-reserved-db-instances](#), as shown in the following example.

Example of getting your reserved DB instances

```
aws rds describe-reserved-db-instances
```

The command returns output similar to the following:

RESERVATION	ReservationId	Class	Multi-AZ	Start Time	Duration
Fixed Price	Usage Price	Count	State	Description	Offering Type
RESERVATION	MyReservation	db.m1.small	y	2011-12-09T23:37:44.720Z	1y
455.00 USD	0.092 USD	1	retired	mysql	Partial Upfront

RDS API

You can use the RDS API to work with reserved DB instances:

- To get information about available reserved DB instance offerings, call the Amazon RDS API operation [DescribeReservedDBInstancesOfferings](#).
- After you have information about the available reserved DB instance offerings, you can use the information to purchase an offering. Call the [PurchaseReservedDBInstancesOffering](#) RDS API operation with the following parameters:
 - **--reserved-db-instances-offering-id** – The ID of the offering that you want to purchase.
 - **--reserved-db-instance-id** – You can assign your own identifier to the reserved DB instances that you purchase to help track them.
- After you have purchased reserved DB instances, you can get information about your reserved DB instances. Call the [DescribeReservedDBInstances](#) RDS API operation.

Setting up for Amazon RDS

Complete the tasks in this section to set up Amazon Relational Database Service (Amazon RDS) for the first time. If you already have an AWS account, know your Amazon RDS requirements, and prefer to use the defaults for IAM and VPC security groups, skip ahead to [Getting started \(p. 4\)](#).

A couple things you should know about Amazon Web Services (AWS):

- When you sign up for AWS, your AWS account automatically has access to all services in AWS, including Amazon RDS. However, you are charged only for the services that you use.
- With Amazon RDS, you pay only for the RDS instances that are active. The Amazon RDS DB instance that you create is live (not running in a sandbox). You incur the standard Amazon RDS usage fees for the instance until you terminate it. For more information about Amazon RDS usage rates, see the [Amazon RDS product page](#).

Topics

- [Sign up for AWS \(p. 67\)](#)
- [Create an IAM user \(p. 67\)](#)
- [Determine requirements \(p. 69\)](#)
- [Provide access to your DB instance in your VPC by creating a security group \(p. 70\)](#)

Sign up for AWS

If you have an AWS account already, skip to the next section, [Create an IAM user \(p. 67\)](#).

If you don't have an AWS account, you can use the following procedure to create one. If you are a new AWS customer, you can get started with Amazon RDS for free; for more information, see [AWS free usage tier](#).

To create a new AWS account

1. Open <https://portal.aws.amazon.com/billing/signup>.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Create an IAM user

After you create an AWS account and successfully connect to the AWS Management Console, you can create an AWS Identity and Access Management (IAM) user. Instead of signing in with your AWS root account, we recommend that you use an IAM administrative user with Amazon RDS.

One way to do this is to create a new IAM user and grant it administrator permissions. Alternatively, you can add an existing IAM user to an IAM group with Amazon RDS administrative permissions. You can then access AWS from a special URL using the credentials for the IAM user.

If you signed up for AWS but haven't created an IAM user for yourself, you can create one using the IAM console.

To create an administrator user for yourself and add the user to an administrators group (console)

1. Sign in to the [IAM console](#) as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

Note

We strongly recommend that you adhere to the best practice of using the **Administrator** IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few [account and service management tasks](#).

2. In the navigation pane, choose **Users** and then choose **Add user**.
3. For **User name**, enter **Administrator**.
4. Select the check box next to **AWS Management Console access**. Then select **Custom password**, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to **User must create a new password at next sign-in** to allow the new user to reset their password after they sign in.
6. Choose **Next: Permissions**.
7. Under **Set permissions**, choose **Add user to group**.
8. Choose **Create group**.
9. In the **Create group** dialog box, for **Group name** enter **Administrators**.
10. Choose **Filter policies**, and then select **AWS managed -job function** to filter the table contents.
11. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.

Note

You must activate IAM user and role access to Billing before you can use the **AdministratorAccess** permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in [step 1 of the tutorial about delegating access to the billing console](#).

12. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
13. Choose **Next: Tags**.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see [Tagging IAM entities](#) in the *IAM User Guide*.
15. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see [Access management](#) and [Example policies](#).

To sign in as the new IAM user, first sign out of the AWS Management Console. Then use the following URL, where **your_aws_account_id** is your AWS account number without the hyphens. For example, if your AWS account number is 1234-5678-9012, your AWS account ID is 123456789012.

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Type the IAM user name and password that you just created. When you're signed in, the navigation bar displays "**your_user_name @ your_aws_account_id**".

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM dashboard, choose **Customize** and type an alias, such as your company name. To sign in after you create an account alias, use the following URL.

`https://your_account_alias.signin.aws.amazon.com/console/`

To verify the sign-in link for IAM users for your account, open the IAM console and check under **AWS Account Alias** on the dashboard.

You can also create access keys for your AWS account. These access keys can be used to access AWS through the AWS Command Line Interface (AWS CLI) or through the Amazon RDS API. For more information, see [Programmatic access](#), [Installing the AWS CLI](#), and the [Amazon RDS API reference](#).

Determine requirements

The basic building block of Amazon RDS is the DB instance. In a DB instance, you create your databases. A DB instance provides a network address called an *endpoint*. Your applications use this endpoint to connect to your DB instance. When you create a DB instance, you specify details like storage, memory, database engine and version, network configuration, security, and maintenance periods. You control network access to a DB instance through a security group.

Before you create a DB instance and a security group, you must know your DB instance and network needs. Here are some important things to consider:

- **Resource requirements** – What are the memory and processor requirements for your application or service? You use these settings to help you determine what DB instance class to use. For specifications about DB instance classes, see [DB instance classes \(p. 7\)](#).
- **VPC, subnet, and security group** – Your DB instance is most likely in a virtual private cloud (VPC). To connect to your DB instance, you need to set up security group rules. These rules are set up differently depending on what kind of VPC you use and how you use it: in a default VPC, in a user-defined VPC, or outside of a VPC.

Note

Some legacy accounts don't use a VPC. If you are accessing a new AWS Region or you are a new RDS user (after 2013), you are most likely creating a DB instance inside a VPC.

For information on how to determine if your account has a default VPC in a particular AWS Region, see [Determining whether you are using the EC2-VPC or EC2-Classic platform \(p. 1718\)](#).

The following list describes the rules for each VPC option:

- **Default VPC** – If your AWS account has a default VPC in the current AWS Region, that VPC is configured to support DB instances. If you specify the default VPC when you create the DB instance, do the following:
 - Create a *VPC security group* that authorizes connections from the application or service to the Amazon RDS DB instance with the database. Use the [Amazon EC2 API](#) or the **Security Group** option on the VPC console to create VPC security groups. For information, see [Step 4: Create a VPC security group \(p. 1734\)](#).
 - Specify the default DB subnet group. If this is the first DB instance you have created in this AWS Region, Amazon RDS creates the default DB subnet group when it creates the DB instance.
- **User-defined VPC** – If you want to specify a user-defined VPC when you create a DB instance, be aware of the following:
 - Make sure to create a *VPC security group* that authorizes connections from the application or service to the Amazon RDS DB instance with the database. Use the [Amazon EC2 API](#) or the **Security Group** option on the VPC console to create VPC security groups. For information, see [Step 4: Create a VPC security group \(p. 1734\)](#).
 - The VPC must meet certain requirements in order to host DB instances, such as having at least two subnets, each in a separate availability zone. For information, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).

- Make sure to specify a DB subnet group that defines which subnets in that VPC can be used by the DB instance. For information, see the DB subnet group section in [Working with a DB instance in a VPC \(p. 1728\)](#).
- **No VPC** – If your AWS account doesn't have a default VPC and you don't specify a user-defined VPC, create a DB security group. A *DB security group* authorizes connections from the devices and Amazon RDS instances running the applications or utilities to access the databases in the DB instance. For more information, see [Working with DB security groups \(EC2-Classic platform\) \(p. 1704\)](#).
- **High availability:** Do you need failover support? On Amazon RDS, a Multi-AZ deployment creates a primary DB instance and a secondary standby DB instance in another Availability Zone for failover support. We recommend Multi-AZ deployments for production workloads to maintain high availability. For development and test purposes, you can use a deployment that isn't Multi-AZ. For more information, see [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).
- **IAM policies:** Does your AWS account have policies that grant the permissions needed to perform Amazon RDS operations? If you are connecting to AWS using IAM credentials, your IAM account must have IAM policies that grant the permissions required to perform Amazon RDS operations. For more information, see [Identity and access management in Amazon RDS \(p. 1644\)](#).
- **Open ports:** What TCP/IP port does your database listen on? The firewall at some companies might block connections to the default port for your database engine. If your company firewall blocks the default port, choose another port for the new DB instance. When you create a DB instance that listens on a port you specify, you can change the port by modifying the DB instance.
- **AWS Region:** What AWS Region do you want your database in? Having your database in close proximity to your application or web service can reduce network latency.
- **DB disk subsystem:** What are your storage requirements? Amazon RDS provides three storage types:
 - Magnetic (Standard Storage)
 - General Purpose (SSD)
 - Provisioned IOPS (PIOPS)

Magnetic storage offers cost-effective storage that is ideal for applications with light or burst I/O requirements. General purpose, SSD-backed storage, also called *gp2*, can provide faster access than disk-based storage. Provisioned IOPS storage is designed to meet the needs of I/O-intensive workloads, particularly database workloads, which are sensitive to storage performance and consistency in random access I/O throughput. For more information on Amazon RDS storage, see [Amazon RDS DB instance storage \(p. 40\)](#).

When you have the information you need to create the security group and the DB instance, continue to the next step.

Provide access to your DB instance in your VPC by creating a security group

VPC security groups provide access to DB instances in a VPC. They act as a firewall for the associated DB instance, controlling both inbound and outbound traffic at the instance level. DB instances are created by default with a firewall and a default security group that protect the DB instance.

Before you can connect to your DB instance, you must add rules to security group that enable you to connect. Use your network and configuration information to create rules to allow access to your DB instance.

Note

If your legacy DB instance was created before March 2013 and isn't in a VPC, it might not have associated security groups. If your DB instance was created after this date, it might be inside a default VPC.

For example, suppose that you have an application that accesses a database on your DB instance in a VPC. In this case, you must add a custom TCP rule that specifies the port range and IP addresses that your application uses to access the database. If you have an application on an Amazon EC2 instance, you can use the security group that you set up for the Amazon EC2 instance.

To create a VPC security group

1. Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc>.
2. In the top right corner of the AWS Management Console, choose the AWS Region where you want to create your VPC security group and DB instance. In the list of Amazon VPC resources for that AWS Region, you should see at least one VPC and several subnets. If you don't, you don't have a default VPC in that AWS Region.
3. In the navigation pane, choose **Security Groups**.
4. Choose **Create Security Group**.
5. In the **Create Security Group** window, type **Name tag**, **Group name**, and **Description** values for your security group. For **VPC**, choose the VPC that you want to create your DB instance in. Choose **Yes, Create**.
6. The VPC security group that you created should still be selected. If not, locate it in the list, and choose it. The details pane at the bottom of the console window displays the details for the security group, and tabs for working with inbound and outbound rules. Choose the **Inbound Rules** tab.
7. On the **Inbound Rules** tab, choose **Edit**.
 - a. For **Type**, choose **Custom TCP Rule**.
 - b. For **Port Range**, type the port value to use for your DB instance.
 - c. For **Source**, choose a security group name or type the IP address range (CIDR value) from where you access the instance. If you choose **My IP**, this allows access to the DB instance from the IP address detected in your browser.
8. Choose **Add another rule** if you need to add more IP addresses or different port ranges.
9. (Optional) Use the **Outbound Rules** tab to add rules for outbound traffic. By default, all outbound traffic is allowed.

You can use the VPC security group that you just created as the security group for your DB instance when you create it. If your DB instance isn't going to be in a VPC, see [Working with DB security groups \(EC2-Classic platform\) \(p. 1704\)](#) to create a DB security group to use when you create your DB instance.

Note

If you use a default VPC, a default subnet group spanning all of the VPC's subnets is created for you. When you create a DB instance, you can select the default VPC and use **default for DB Subnet Group**.

Once you have completed the setup requirements, you can launch a DB instance using your requirements and security group. For information on creating a DB instance, see the relevant documentation in the following table.

Database engine	Documentation
MariaDB	Creating a MariaDB DB instance and connecting to a database on a MariaDB DB instance (p. 73)
Microsoft SQL Server	Creating a Microsoft SQL Server DB instance and connecting to it (p. 80)
MySQL	Creating a MySQL DB instance and connecting to a database on a MySQL DB instance (p. 86)

Database engine	Documentation
Oracle	Creating an Oracle DB instance and connecting to a database on an Oracle DB instance (p. 93)
PostgreSQL	Creating a PostgreSQL DB instance and connecting to a database on a PostgreSQL DB instance (p. 99)

Note

If you can't connect to a DB instance after you create it, see the troubleshooting information in [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).

Getting started with Amazon RDS

In the following examples, you can find how to create and connect to a DB instance using Amazon Relational Database Service (Amazon RDS). You can create a DB instance that uses MariaDB, MySQL, Microsoft SQL Server, Oracle, or PostgreSQL.

Important

Before you can create or connect to a DB instance, you must complete the tasks in [Setting up for Amazon RDS \(p. 67\)](#).

Creating a DB instance and connecting to a database on a DB instance is slightly different for each of the DB engines. Choose one of the following DB engines that you want to use for detailed information on creating and connecting to the DB instance. After you have created and connected to your DB instance, there are instructions to help you delete the DB instance.

Topics

- [Creating a MariaDB DB instance and connecting to a database on a MariaDB DB instance \(p. 73\)](#)
- [Creating a Microsoft SQL Server DB instance and connecting to it \(p. 80\)](#)
- [Creating a MySQL DB instance and connecting to a database on a MySQL DB instance \(p. 86\)](#)
- [Creating an Oracle DB instance and connecting to a database on an Oracle DB instance \(p. 93\)](#)
- [Creating a PostgreSQL DB instance and connecting to a database on a PostgreSQL DB instance \(p. 99\)](#)
- [Tutorial: Create a web server and an Amazon RDS DB instance \(p. 108\)](#)

Creating a MariaDB DB instance and connecting to a database on a MariaDB DB instance

The easiest way to create a MariaDB DB instance is to use the Amazon RDS console. After you create the DB instance, you can use command line tools such as mysql or standard graphical tools such as HeidiSQL to connect to a database on the DB instance.

Important

Before you can create or connect to a DB instance, you must complete the tasks in [Setting up for Amazon RDS \(p. 67\)](#).

Topics

- [Creating a MariaDB DB instance \(p. 73\)](#)
- [Connecting to a database on a DB instance running the MariaDB database engine \(p. 77\)](#)
- [Deleting a DB instance \(p. 79\)](#)

Creating a MariaDB DB instance

The basic building block of Amazon RDS is the DB instance. This environment is where you run your MariaDB databases.

Console

You can create a DB instance running MariaDB with the AWS Management Console with **Easy Create** enabled or not enabled. With **Easy Create** enabled, you specify only the DB engine type, DB instance size, and DB instance identifier. **Easy Create** uses the default setting for other configuration options. With **Easy Create** not enabled, you specify more configuration options when you create a database, including ones for availability, security, backups, and maintenance.

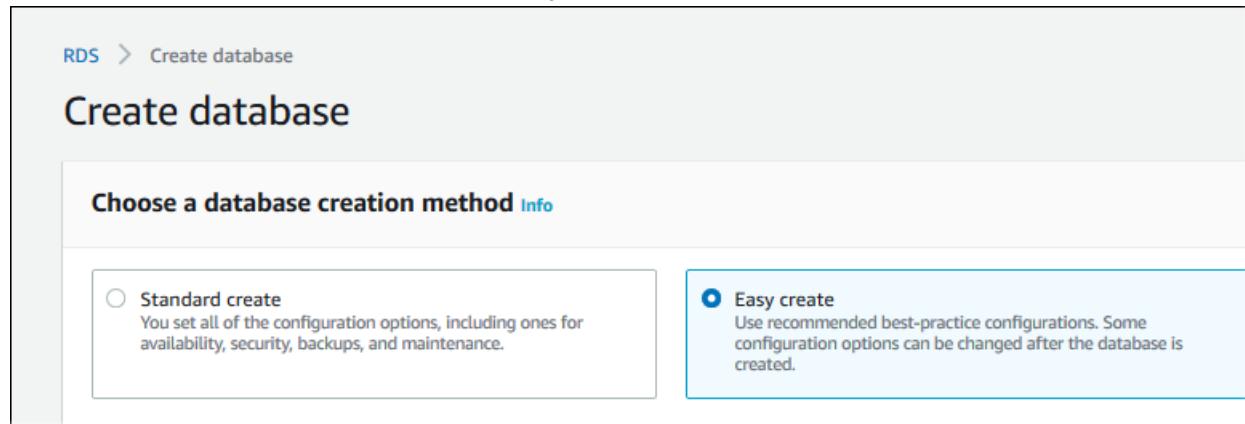
In this example, you use **Easy Create** to create a DB instance running the MariaDB database engine with a db.t2.micro DB instance class.

Note

For information about creating DB instances with **Easy Create** not enabled, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

To create a MariaDB DB instance with Easy Create enabled

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region in which you want to create the DB instance.
3. In the navigation pane, choose **Databases**.
4. Choose **Create database** and make sure that **Easy Create** is chosen.



5. In **Configuration**, choose **MariaDB**.
6. For **DB instance size**, choose **Free tier**.
7. For **DB instance identifier**, enter a name for the DB instance, or leave the default name.
8. For **Master username**, enter a name for the master user, or leave the default name.

The **Create database** page should look similar to the following image.

Create database

Choose a database creation method [Info](#)

Standard Create

You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy Create

Use recommended best-practice configuration options can be changed after the database is created.

Configuration

Engine type [Info](#)

Amazon Aurora



MySQL



MariaDB



PostgreSQL



Oracle

ORACLE®

Microsoft SQL Server



DB instance size

Production

db.r4.xlarge
4 vCPUs
30.5 GiB RAM
500 GiB

Dev/Test

db.r4.large
2 vCPUs
15.25 GiB RAM
100 GiB

Free tier

db.t2.micro
1 vCPUs
1 GiB RAM
20 GiB

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in this Region.

75

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1–63 characters, alphanumeric (A–Z, a–z, 0–9), and underscores (_).

9. To use an automatically generated master password for the DB instance, make sure that the **Auto generate a password** check box is chosen.

To enter your master password, clear the **Auto generate a password** check box, and then enter the same password in **Master password** and **Confirm password**.

10. (Optional) Open **View default settings for Easy create**.

▼ View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use [Standard Create](#).

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1234567a)	No
Option Group	default:oracle-se2-19	Yes
Subnet Group	default-vpc-1234567a	Yes
Automatic Backups	Enabled	Yes
VPC Security Group	sg-1a2bcd3e	Yes
Publicly Accessible	No	Yes
Database Port	1521	Yes

You can examine the default settings used when **Easy Create** is enabled. If you want to change one or more settings during database creation, choose **Standard Create** to set them. The **Editable after database creation** column shows which options you can change after database creation. To change a setting with **No** in that column, use **Standard Create**. For settings with **Yes** in that column, you can either use **Standard Create** or modify the DB instance after it's created to change the setting.

11. Choose **Create database**.

If you chose to use an automatically generated password, the **View credential details** button appears on the **Databases** page.

To view the master user name and password for the DB instance, choose **View credential details**.



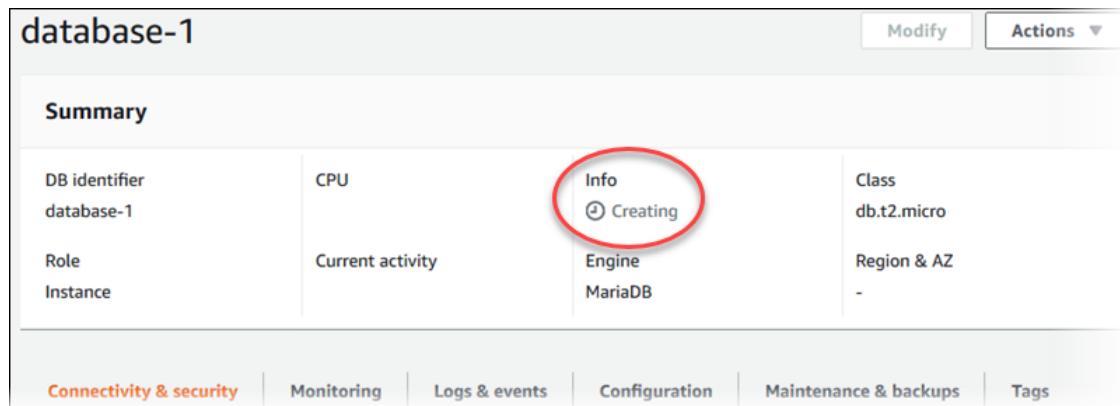
To connect to the DB instance as the master user, use the user name and password that appear.

Important

You can't view the master user password again. If you don't record it, you might have to change it. If you need to change the master user password after the DB instance is available, you can modify the DB instance to do so. For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

12. For **Databases**, choose the name of the new Maria DB instance.

On the RDS console, the details for new DB instance appear. The DB instance has a status of **creating** until the DB instance is ready to use. When the state changes to **available**, you can connect to the DB instance. Depending on the DB instance class and the amount of storage, it can take up to 20 minutes before the new instance is available.



Connecting to a database on a DB instance running the MariaDB database engine

After Amazon RDS provisions your DB instance, you can use any standard SQL client application to connect to a database on the DB instance. In this example, you connect to a database on a Maria DB instance using the mysql command-line tool. One GUI-based application you can use to connect is HeidiSQL. For more information, see the [Download HeidiSQL](#) page. For more information on using MariaDB, see the [MariaDB documentation](#).

To connect to a database on a DB instance using the mysql command-line tool

1. Find the endpoint (DNS name) and port number for your DB instance.
 - a. Open the RDS console and then choose **Databases** to display a list of your DB instances.
 - b. Choose the Maria DB instance name to display its details.
 - c. On the **Connectivity & security** tab, copy the endpoint. Also note the port number. You need both the endpoint and the port number to connect to the DB instance.

The screenshot shows the Amazon RDS console interface. At the top, the navigation path is RDS > Databases > mydb. Below this, the database name "mydb" is displayed in large blue text. A "Summary" section provides key metrics: DB identifier (mydb), Role (Instance), CPU usage (2.33%), and Current activity (0 Connections). Below the summary, there are tabs for Connectivity & security, Monitoring, Logs & events, and Configuration. The Connectivity & security tab is selected. Under this tab, the "Endpoint & port" section is highlighted with a red oval. It shows the Endpoint as "mydb.us-east-1.rds.amazonaws.com" and the Port as "3306".

2. Enter the following command at a command prompt on a client computer to connect to a database on a Maria DB instance. Substitute the DNS name (endpoint) for your DB instance for <endpoint>, the master user name you used for <mymasteruser>, and provide the master password you used when prompted for a password.

```
PROMPT> mysql -h <endpoint> -P 3306 -u <mymasteruser> -p
```

After you enter the password for the user, you should see output similar to the following.

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 272
Server version: 5.5.5-10.0.17-MariaDB-log MariaDB Server

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql >
```

For more information about connecting to a MariaDB DB instance, see [Connecting to a DB instance running the MariaDB database engine \(p. 588\)](#). For information on connection issues, see [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).

Deleting a DB instance

After you have connected to the sample DB instance that you created, you should delete the DB instance so you are no longer charged for it.

To delete a DB instance with no final DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the DB instance you want to delete.
4. For **Actions**, choose **Delete**.
5. For **Create final snapshot?**, choose **No**, and select the acknowledgment.
6. Choose **Delete**.

Creating a Microsoft SQL Server DB instance and connecting to it

The basic building block of Amazon RDS is the DB instance. Your Amazon RDS DB instance is similar to your on-premises Microsoft SQL Server. After you create your SQL Server DB instance, you can add one or more custom databases to it.

Important

You must have an AWS account before you can create a DB instance. If you don't have an AWS account, open <https://aws.amazon.com/>, and then choose **Create an AWS Account**.

In this topic, you create a sample SQL Server DB instance. You then connect to the DB instance and run a simple query. Finally, you delete the sample DB instance.

Creating a sample SQL Server DB instance

You can create a DB instance running Microsoft SQL Server with the AWS Management Console with **Easy create** enabled or not enabled. With **Easy create** enabled, you specify only the DB engine type, DB instance size, and DB instance identifier. **Easy create** uses the default settings for other configuration options. With **Easy create** not enabled (**Standard create**), you specify more configuration options when you create a database, including ones for availability, security, backups, and maintenance.

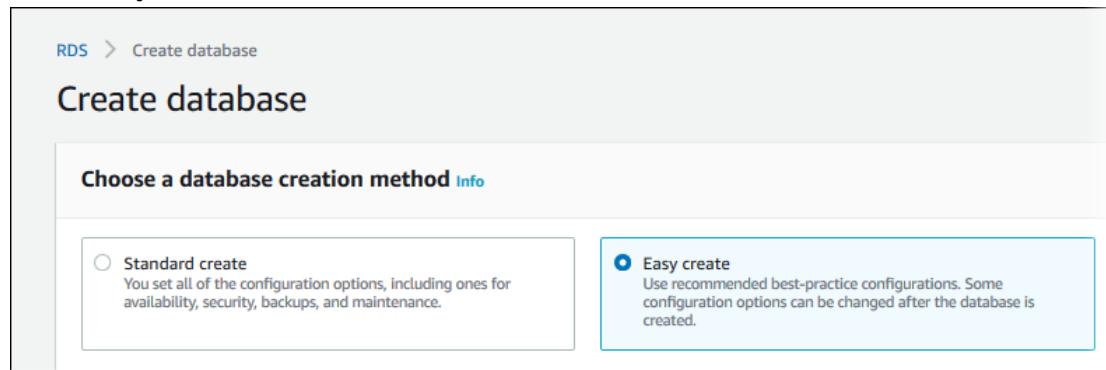
For this example, you use **Easy create** to create a DB instance running SQL Server Express Edition with a db.t2.micro DB instance class.

Note

For information about creating DB instances with **Standard create**, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

To create a Microsoft SQL Server DB instance with Easy create

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region in which you want to create the DB instance.
3. In the navigation pane, choose **Databases**.
4. Choose **Create database**.
5. Choose **Easy create**.



6. From **Engine type**, choose **Microsoft SQL Server**.
7. For **DB instance size**, choose **Free tier**.

8. For **DB instance identifier**, enter a name for the DB instance, or leave the default name.
9. For **Master username**, enter a name for the master user, or leave the default name.
10. To use an automatically generated master password for the DB instance, choose the **Auto generate a password** check box.

To enter your master password, clear the **Auto generate a password** check box, and then enter the same password in **Master password** and **Confirm password**.

The **Create database** page should look similar to the following image.

Create database

Choose a database creation method Info

Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Configuration

Engine type Info

Amazon Aurora

MySQL

MariaDB

PostgreSQL

Microsoft SQL Server

DB instance size

Production
db.r5.xlarge
4 vCPUs
32 GB RAM
300 GB
USD/hour

Dev/Test
db.m5.large
2 vCPUs
8 GB RAM
100 GB
USD/hour

Free tier
db.t2.micro
1 vCPU
1 GB RAM
20 GB
USD/hour

DB instance identifier
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username Info
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter

Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password

View default settings for Easy create
Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use [Standard Create](#).

ⓘ You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Cancel **Create database**

11. (Optional) Expand **View default settings for Easy create**.

▼ View default settings for Easy create		
Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use Standard Create .		
Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc- [REDACTED])	No
Option Group	defaultsqlserver-ex-14-00	Yes
Subnet Group	default	Yes
Automatic Backups	Enabled	Yes
VPC Security Group	sg- [REDACTED]	Yes
Publicly Accessible	No	Yes
Database Port	1433	Yes
DB Instance Identifier	database-2	Yes
DB Engine Version	14.00.3356.20.v1	Yes
DB Parameter Group	default.sqlserver-ex-14.0	Yes
Performance Insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto Minor Version Upgrade Enabled	Yes
Delete Protection	Not Enabled	Yes

You can examine the default settings used when **Easy create** is enabled. If you want to change one or more settings during database creation, choose **Standard create** to set them. The **Editable after database is created** column shows which options you can change after database creation. To change a setting with **No** in that column, use **Standard create**. For settings with **Yes** in that column, you can either use **Standard create** or modify the DB instance after it's created to change the setting.

12. Choose **Create database**.

If you chose to use an automatically generated password, the **View credential details** button appears on the **Databases** page.

To view the master user name and password for the DB instance, choose **View credential details**.



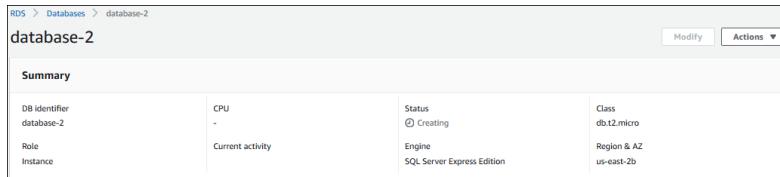
To connect to the DB instance as the master user, use the user name and password that appear.

Important

You can't view the master user password again. If you don't record it, you might have to change it. If you need to change the master user password after the DB instance is available, you can modify the DB instance to do so. For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

13. For **Databases**, choose the name of the new Microsoft SQL Server DB instance.

On the RDS console, the details for new DB instance appear. The DB instance has a status of **creating** until the DB instance is ready to use. When the state changes to **available**, you can connect to the DB instance. Depending on the DB instance class and the amount of storage, it can take up to 20 minutes before the new instance is available.



Summary			
DB identifier	CPU	Status	Class
database-2	-	Creating	db.t2.micro
Role	Current activity	Engine	Region & AZ
Instance		SQL Server Express Edition	us-east-2b

Connecting to your sample SQL Server DB instance

In this procedure, you connect to your sample DB instance by using Microsoft SQL Server Management Studio (SSMS).

Before you begin, your database should have a status of **Available**. If it has a status of **Creating** or **Backing-up**, wait until it's **Available**. The status updates without requiring you to refresh the page. This process can take up to 20 minutes.

Also, make sure you have SSMS installed. If you can also connect to SQL Server on RDS by using a different tools, such as an add-in for your development environment or some other database tool. However, this tutorial only covers using SSMS. To download a stand-alone version of this SSMS, see [Download SQL Server Management Studio \(SSMS\)](#) in the Microsoft documentation.

To connect to a DB instance using SSMS

1. Find the DNS name and port number for your DB instance.
 - a. Open the RDS console, and then choose **Databases** to display a list of your DB instances.
 - b. Hover your mouse cursor over the name **sample-instance**, which is blue. When you do this, the mouse cursor changes into a selection icon (for example, a pointing hand). Also, the DB instance name, becomes underlined.

Click on the DB instance name to choose it. The screen changes to display the information for the DB instance you choose.
- c. On the **Connectivity** tab, which opens by default, copy the endpoint. The **Endpoint** looks something like this: `sample-instance.abc2defghije.us-west-2.rds.amazonaws.com`. Also, take note of the port number. The default port for SQL Server is 1433. If yours is different, write it down.

2. Start SQL Server Management Studio.

The **Connect to Server** dialog box appears.

3. Provide the information for your sample DB instance.
 - a. For **Server type**, choose **Database Engine**.
 - b. For **Server name**, enter the DNS name, followed by a comma and the port number (the default port is 1433). For example, your server name should look like the following.

`sample-instance.abc2defghije.us-west-2.rds.amazonaws.com,1433`

- c. For **Authentication**, choose **SQL Server Authentication**.
- d. For **Login**, enter the user name that you chose to use for your sample DB instance. This is also known as the master user name.
- e. For **Password**, enter the password that you chose earlier for your sample DB instance. This is also known as the master user password.

4. Choose **Connect**.

After a few moments, SSMS connects to your DB instance.

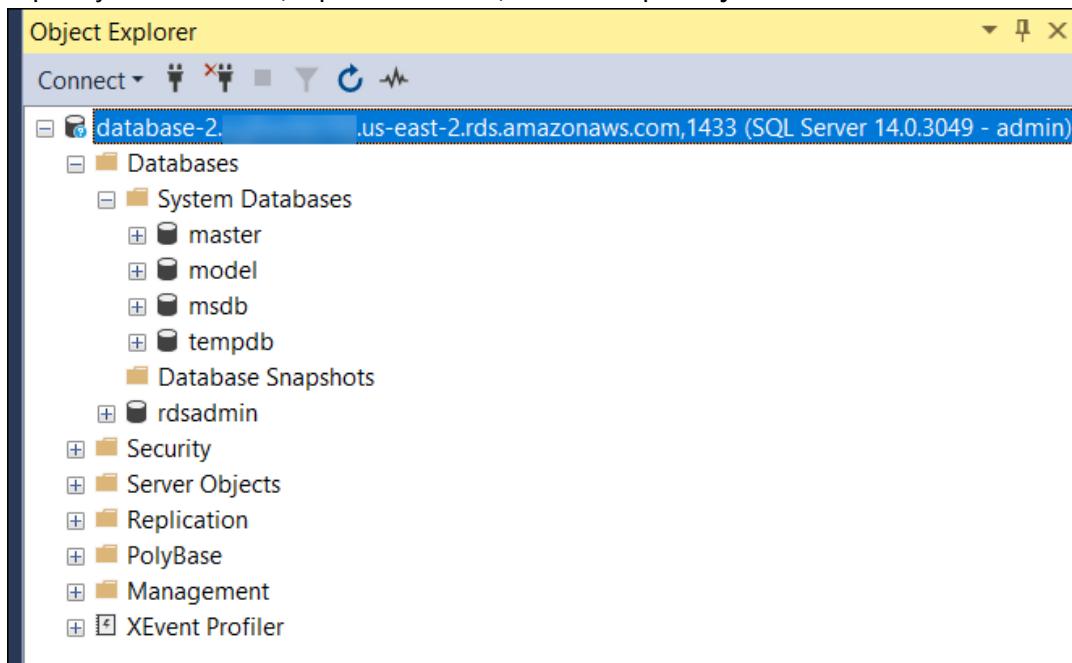
If you can't connect to your DB instance, see [Troubleshooting connections to your SQL Server DB instance \(p. 661\)](#).

Exploring your sample SQL Server DB instance

In this procedure, you continue the previous procedure and explore your sample DB instance by using Microsoft SQL Server Management Studio (SSMS).

To explore a DB instance using SSMS

1. Your SQL Server DB instance comes with SQL Server's standard built-in system databases (master, model, msdb, and tempdb). To explore the system databases, do the following:
 - a. In SSMS, on the **View** menu, choose **Object Explorer**.
 - b. Expand your DB instance, expand **Databases**, and then expand **System Databases** as shown.

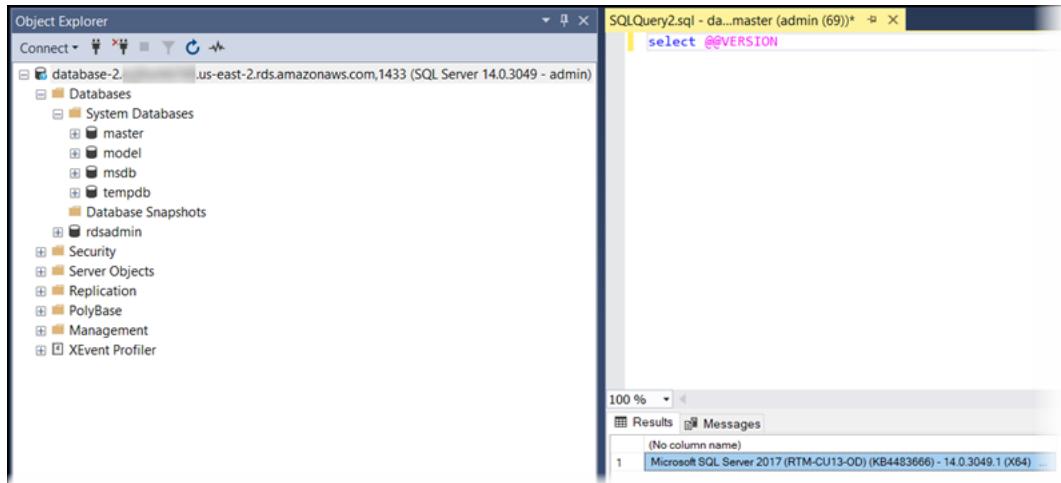


2. Your SQL Server DB instance also comes with a database named `rdsadmin`. Amazon RDS uses this database to store the objects that it uses to manage your database. The `rdsadmin` database also includes stored procedures that you can run to perform advanced tasks.
3. You can now start creating your own databases and running queries against your DB instance and databases as usual. To run a test query against your sample DB instance, do the following:

- a. In SSMS, on the **File** menu point to **New** and then choose **Query with Current Connection**.
- b. Enter the following SQL query.

```
select @@VERSION
```

- c. Run the query. SSMS returns the SQL Server version of your Amazon RDS DB instance.



Deleting your sample DB instance

After you are done exploring the sample DB instance that you created, you should delete the DB instance so that you are no longer charged for it.

To delete a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the button next to **sample-instance**, or whatever you named your sample DB instance.
4. From **Actions**, choose **Delete**.
5. If you see a message that says **This database has deletion protection option enabled**, follow these steps:
 - a. Choose **Modify**.
 - b. On the **Deletion protection** card (near the bottom of the page), clear the box next to **Enable deletion protection**. Then choose **Continue**.
 - c. On the **Scheduling of modifications** card, choose **Apply immediately**. Then choose **Modify DB instance**.
 - d. Try again to delete the instance by choosing **Delete** from the **Actions** menu.
6. Clear the box for **Create final snapshot**. Because this isn't a production database, you don't need to save a copy of it.
7. Verify that you selected the correct database to delete. The name "sample-instance" displays in the title of the screen: **Delete sample-instance instance?**

If you don't recognize the name of your sample instance in the title, choose **Cancel** and start over.

8. To confirm that you want to permanently delete the database that is displayed in the title of this screen, do the following:
 - Check the box to confirm: **I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available.**
 - Type "**delete me**" into the box **To confirm deletion, type *delete me* into the field**.
 - Choose **Delete**. This action can't be undone.

The database shows a status of **Deleting** until deletion is complete.

Creating a MySQL DB instance and connecting to a database on a MySQL DB instance

The easiest way to create a DB instance is to use the AWS Management Console. After you have created the DB instance, you can use standard MySQL utilities such as MySQL Workbench to connect to a database on the DB instance.

Important

Before you can create or connect to a DB instance, you must complete the tasks in [Setting up for Amazon RDS \(p. 67\)](#).

Topics

- [Creating a MySQL DB instance \(p. 86\)](#)
- [Connecting to a database on a DB instance running the MySQL database engine \(p. 90\)](#)
- [Deleting a DB instance \(p. 92\)](#)

Creating a MySQL DB instance

The basic building block of Amazon RDS is the DB instance. This environment is where you run your MySQL databases.

Console

You can create a DB instance running MySQL with the AWS Management Console with **Easy Create** enabled or disabled. With **Easy Create** enabled, you specify only the DB engine type, DB instance size, and DB instance identifier. **Easy Create** uses the default setting for other configuration options. With **Easy Create** not enabled, you specify more configuration options when you create a database, including ones for availability, security, backups, and maintenance.

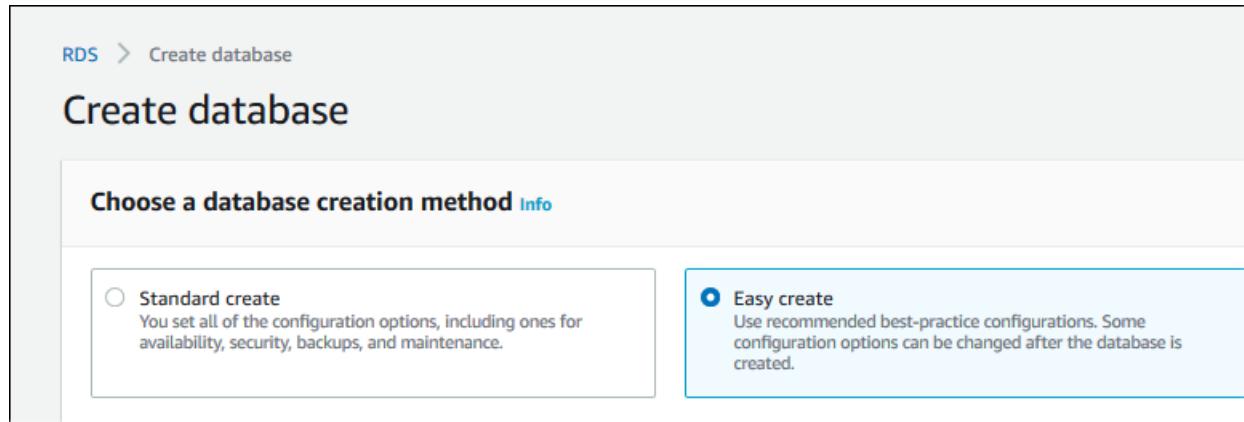
In this example, you use **Easy Create** to create a DB instance running the MySQL database engine with a db.t2.micro DB instance class.

Note

For information about creating DB instances with **Easy Create** not enabled, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

To create a MySQL DB instance with Easy Create enabled

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region in which you want to create the DB instance.
3. In the navigation pane, choose **Databases**.
4. Choose **Create database** and make sure that **Easy Create** is chosen.



5. In **Configuration**, choose **MySQL**.
6. For **DB instance size**, choose **Free tier**.
7. For **DB instance identifier**, enter a name for the DB instance, or leave the default name.
8. For **Master username**, enter a name for the master user, or leave the default name.

The **Create database** page should look similar to the following image.

Create database

Choose a database creation method Info

Standard Create

You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy Create

Use recommended best-practice configuration options can be changed after the database is created.

Configuration

Engine type Info

Amazon Aurora



MySQL



MariaDB

PostgreSQL



Oracle

ORACLE®

Microsoft SQL Server



DB instance size

Production

db.r5.xlarge
4 vCPUs
32 GiB RAM
500 GiB

Dev/Test

db.r5.large
2 vCPUs
16 GiB RAM
100 GiB

Free tier

db.t2.micro
1 vCPUs
1 GiB RAM
20 GiB

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account.

Region. 88

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1

9. To use an automatically generated master password for the DB instance, enable **Auto generate a password**.

To enter your master password, disable **Auto generate a password**, and then enter the same password in **Master password** and **Confirm password**.

10. (Optional) Open **View default settings for Easy create**.

▼ View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use [Standard Create](#).

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1234567a)	No
Option Group	default:oracle-se2-19	Yes
Subnet Group	default-vpc-1234567a	Yes
Automatic Backups	Enabled	Yes
VPC Security Group	sg-1a2bcd3e	Yes
Publicly Accessible	No	Yes
Database Port	1521	Yes

You can examine the default settings used when **Easy Create** is enabled. If you want to change one or more settings during database creation, choose **Standard Create** to set them. The **Editable after database creation** column shows which options you can change after database creation. To change a setting with **No** in that column, use **Standard Create**. For settings with **Yes** in that column, you can either use **Standard Create** or modify the DB instance after it is created to change the setting.

11. Choose **Create database**.

If you chose to use an automatically generated password, the **View credential details** button appears on the **Databases** page.

To view the master username and password for the DB instance, choose **View credential details**.



You can use the username and password that appears to connect to the DB instance as the master user.

Important

You won't be able to view master user password again. If you don't record it, you might have to change it. If you need to change the master user password after the DB instance is available, you can modify the DB instance to do so. For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

12. In the **Databases** list, choose the name of the new MySQL DB instance.

On the RDS console, the details for new DB instance appear. The DB instance has a status of **creating** until the DB instance is ready to use. When the state changes to **available**, you can connect to the

DB instance. Depending on the DB instance class and the amount of storage, it can take up to 20 minutes before the new instance is available.

database-1		Modify	Actions ▾
Summary			
DB identifier database-1	CPU	Info  Creating	Class db.t2.micro
Role Instance	Current activity	Engine MySQL	Region & AZ -

Connecting to a database on a DB instance running the MySQL database engine

After Amazon RDS provisions your DB instance, you can use any standard SQL client application to connect to a database on the DB instance. In this example, you connect to a database on a MySQL DB instance using MySQL monitor commands. One GUI-based application you can use to connect is MySQL Workbench; for more information, go to the [Download MySQL Workbench](#) page. For more information on using MySQL, go to the [MySQL documentation](#). For information about installing MySQL (including the MySQL client), see [Installing and upgrading MySQL](#).

To connect to a database on a DB instance using MySQL monitor

1. Find the endpoint (DNS name) and port number for your DB instance.
 - a. Open the RDS console and then choose **Databases** to display a list of your DB instances.
 - b. Choose the MySQL DB instance name to display its details.
 - c. On the **Connectivity & security** tab, copy the endpoint. Also, note the port number. You need both the endpoint and the port number to connect to the DB instance.

The screenshot shows the Amazon RDS console interface. At the top, the navigation path is RDS > Databases > mydb. Below this, the database identifier is listed as "mydb". On the right side, there are performance metrics: CPU usage at 2.33% and 0 connections. The "Role" is listed as "Instance". At the bottom, there are tabs for Connectivity & security, Monitoring, Logs & events, and Configuration. The Connectivity & security tab is selected. Under this tab, the endpoint and port information is displayed. The endpoint is "mydb.us-east-1.rds.amazonaws.com" and the port is "3306". Both the endpoint and port fields are circled in red.

2. Download a SQL client that you can use to connect to the DB instance.

You can connect to a MySQL DB instance by using tools like the MySQL command line utility. For more information on using the MySQL client, go to [mysql - the MySQL command-line client](#) in the MySQL documentation. One GUI-based application you can use to connect is MySQL Workbench. For more information, go to the [Download MySQL Workbench](#) page.

3. Connect to the a database on a MySQL DB instance. For example, enter the following command at a command prompt on a client computer to connect to a database on a MySQL DB instance using the MySQL client. Substitute the DNS name for your DB instance for `<endpoint>`, the master user name you used for `<mymasteruser>`, and provide the master password you used when prompted for a password.

```
PROMPT> mysql -h <endpoint> -P 3306 -u <mymasteruser> -p
```

After you enter the password for the user, you should see output similar to the following.

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 350
Server version: 5.6.40-log MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

If you can't connect to your MySQL DB instance, two common causes of connection failures to a new DB instance are:

- The DB instance was created using a security group that does not authorize connections from the device or Amazon EC2 instance where the MySQL application or utility is running. If the DB instance was created in a VPC, it must have a VPC security group that authorizes the connections. If the DB instance was created outside of a VPC, it must have a DB security group that authorizes the connections. For more information, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).
- The DB instance was created using the default port of 3306, and your company has firewall rules blocking connections to that port from devices in your company network. To fix this failure, recreate the instance with a different port.

For more information about connecting to a MySQL DB instance, see [Connecting to a DB instance running the MySQL database engine \(p. 840\)](#). For information on connection issues, see [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).

Deleting a DB instance

After you have connected to the sample DB instance that you created, you should delete the DB instance so you are no longer charged for it.

To delete a DB instance with no final DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the DB instance that you want to delete.
4. For **Actions**, choose **Delete**.
5. For **Create final snapshot?**, choose **No**, and select the acknowledgment.
6. Choose **Delete**.

Creating an Oracle DB instance and connecting to a database on an Oracle DB instance

The basic building block of Amazon RDS is the DB instance. Your Amazon RDS DB instance is similar to your on-premises Oracle database.

Important

You must have an AWS account before you can create a DB instance. If you don't have an AWS account, open <https://aws.amazon.com/>, and then choose **Create an AWS Account**.

In this topic, you create a sample Oracle DB instance. You then connect to the DB instance and run a simple query. Finally, you delete the sample DB instance.

Creating a sample Oracle DB instance

The DB instance is where you run your Oracle databases.

Console

You can create a DB instance running Oracle with the AWS Management Console with **Easy create** enabled or not enabled. With **Easy create** enabled, you specify only the DB engine type, DB instance size, and DB instance identifier. **Easy create** uses the default setting for other configuration options. With **Easy create** not enabled, you specify more configuration options when you create a database, including ones for availability, security, backups, and maintenance.

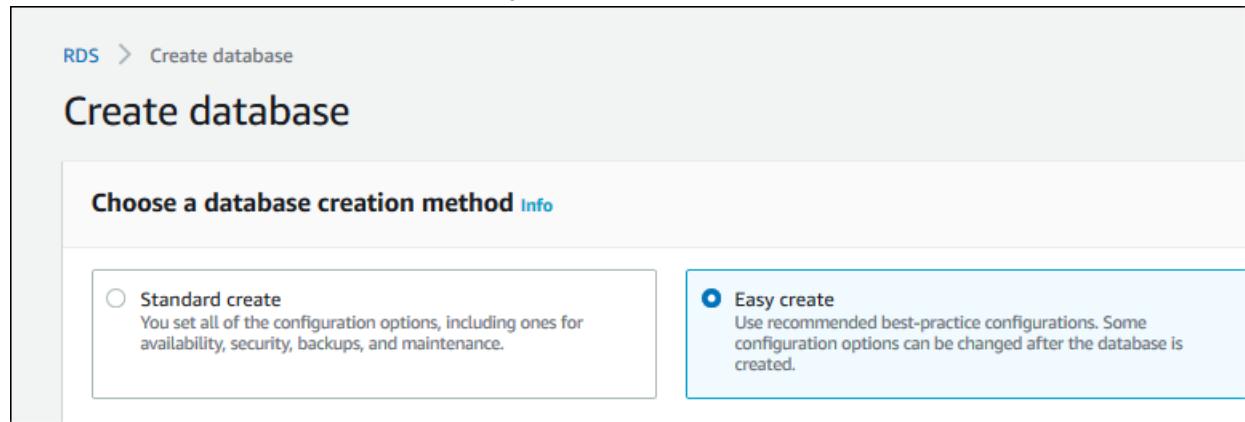
For this example, you use **Easy create** to create a DB instance running the Oracle database engine with a db.m4.large DB instance class.

Note

For information about creating DB instances with **Easy create** not enabled, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

To create an Oracle DB instance with Easy create enabled

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region in which you want to create the DB instance.
3. In the navigation pane, choose **Databases**.
4. Choose **Create database** and ensure that **Easy create** is chosen.



5. In **Configuration**, choose **Oracle**.
6. For **DB instance size**, choose **Free tier**. If **Free tier** isn't available, choose **Dev/Test**.
7. For **DB instance identifier**, enter a name for the DB instance, or leave the default name.
8. For **Master username**, enter a name for the master user, or leave the default name.

The **Create database** page should look similar to the following image.

Create database

Choose a database creation method Info

Standard Create

You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy Create

Use recommended best-practice configuration options can be changed after the database is created.

Configuration

Engine type Info

Amazon Aurora



MySQL



MariaDB



PostgreSQL



Oracle

ORACLE®

Microsoft SQL Server



DB instance size

Production

db.r4.large
2 vCPUs
15.25 GiB RAM
500 GiB

Dev/Test

db.m4.large
2 vCPUs
8 GiB RAM
100 GiB

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account.

database-1

9. To use an automatically generated master password for the DB instance, make sure that the **Auto generate a password** check box is chosen.

To enter your master password, clear the **Auto generate a password** check box, and then enter the same password in **Master password** and **Confirm password**.

10. (Optional) Open **View default settings for Easy create**.

▼ View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use [Standard Create](#).

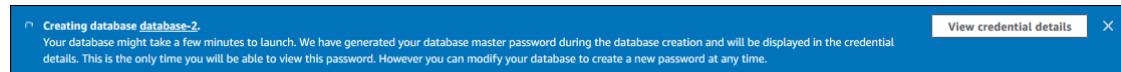
Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1234567a)	No
Option Group	default:oracle-se2-19	Yes
Subnet Group	default-vpc-1234567a	Yes
Automatic Backups	Enabled	Yes
VPC Security Group	sg-1a2bcd3e	Yes
Publicly Accessible	No	Yes
Database Port	1521	Yes

You can examine the default settings that are used when **Easy create** is enabled. If you want to change one or more settings during database creation, choose **Standard create** to set them. The **Editable after database creation** column shows which options you can change after database creation. To change a setting with **No** in that column, use **Standard create**. For settings with **Yes** in that column, you can either use **Standard create** or modify the DB instance after it's created to change the setting.

11. Choose **Create database**.

If you used an automatically generated password, the **View credential details** button appears on the **Databases** page.

To view the master user name and password for the DB instance, choose **View credential details**.



To connect to the DB instance as the master user, use the user name and password that appear.

Important

You can't view the master user password again. If you don't record it, you might have to change it. If you need to change the master user password after the DB instance is available, you can modify the DB instance to do so. For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

12. For **Databases**, choose the name of the new Oracle DB instance.

On the RDS console, the details for new DB instance appear. The DB instance has a status of **creating** until the DB instance is ready to use. When the state changes to **available**, you can connect to the DB instance. Depending on the DB instance class and the amount of storage, it can take up to 20 minutes before the new instance is available.

The screenshot shows the 'Summary' tab of the RDS instance details for 'database-1'. The 'Info' column displays the status as 'Creating', which is highlighted with a red circle. Other visible details include the DB identifier 'database-1', CPU usage, role 'Instance', current activity with 0 sessions, engine 'Oracle Enterprise Edition', and class 'db.m5.xlarge'. Below the summary, there are tabs for Connectivity & security, Monitoring, Logs & events, Configuration, Maintenance & backups, and Tags.

Connecting to your sample Oracle DB instance

After Amazon RDS provisions your DB instance, you can use any standard SQL client application to connect to the DB instance. In this procedure, you connect to your sample DB instance by using the Oracle sqlplus command line utility. To download a stand-alone version of this utility, see [SQL*Plus User's Guide and Reference](#).

To connect to a DB instance using SQL*Plus

1. Find the endpoint (DNS name) and port number for your DB Instance.
 - a. Open the RDS console and then choose **Databases** to display a list of your DB instances.
 - b. Choose the Oracle DB instance name to display its details.
 - c. On the **Connectivity & security** tab, copy the following pieces of information:
 - Endpoint
 - Port

You need both the endpoint and the port number to connect to the DB instance.

DB identifier	CPU	Status	Class
database-1	<div style="width: 2.3%;">2.30%</div>	Available	db.m4.large
Role	Current activity	Engine	Region & AZ
Instance	<div style="width: 0.01%;">0.01 Sessions</div>	Oracle Standard Edition Two	us-east-1f

Connectivity & security

Endpoint & port	Networking	Security
Endpoint database-1.abcdefghijkl.us-east-1.rds.amazonaws.com	Availability zone us-east-1f	VPC security groups default (sg-0a5cba2b) (active)
Port 1521	VPC vpc-1234567f	Public accessibility No
	Subnet group	

- d. On the **Configuration** tab, copy the following pieces of information:

- DB name (not the DB instance ID)
- Master username

You need both the DB name and the master username to connect to the DB instance.

2. Enter the following command on one line at a command prompt to connect to your DB instance by using the sqlplus utility. Use the following values:
- For **dbuser**, enter the name of the master user that you copied in the preceding steps.
 - For **HOST=endpoint**, enter the endpoint that you copied in the preceding steps.
 - For **PORT=portnum**, enter the port number that you copied in the preceding steps.
 - For **SID=DB_NAME**, enter the Oracle database name (not the instance name) that you copied in the preceding steps.

```
sqlplus 'dbuser@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=endpoint)(PORT=portnum))(CONNECT_DATA=(SID=DB_NAME)))'
```

You should see output similar to the following.

```
SQL*Plus: Release 11.1.0.7.0 - Production on Wed May 25 15:13:59 2011
SQL>
```

For more information about connecting to an Oracle DB instance, see [Connecting to your Oracle DB instance \(p. 1001\)](#). For information on connection issues, see [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).

Deleting your sample DB instance

After you are done exploring the sample DB instance that you created, you should delete the DB instance so that you are no longer charged for it.

To delete a DB instance with no final DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the DB instance that you want to delete.
4. For **Actions**, choose **Delete**.
5. For **Create final snapshot?**, choose **No**, and choose the acknowledgment.
6. Choose **Delete**.

Creating a PostgreSQL DB instance and connecting to a database on a PostgreSQL DB instance

The easiest way to create a DB instance is to use the RDS console. After you have created the DB instance, you can use standard SQL client utilities to connect to the DB instance, such as the pgAdmin utility. In this example, you create a DB instance running the PostgreSQL database engine called database-1, with a db.t2.micro DB instance class and 20 gibibytes (GiB) of storage.

Important

Before you can create or connect to a DB instance, you must complete the tasks in [Setting up for Amazon RDS \(p. 67\)](#).

Contents

- [Creating a PostgreSQL DB instance \(p. 99\)](#)
- [Connecting to a PostgreSQL DB instance \(p. 103\)](#)
 - [Using pgAdmin to connect to a PostgreSQL DB instance \(p. 103\)](#)
 - [Using psql to connect to a PostgreSQL DB instance \(p. 107\)](#)
- [Deleting a DB instance \(p. 107\)](#)

Creating a PostgreSQL DB instance

The basic building block of Amazon RDS is the DB instance. This environment is where you run your PostgreSQL databases.

You can create a DB instance running PostgreSQL with the AWS Management Console with **Easy Create** enabled or disabled. With **Easy Create** enabled, you specify only the DB engine type, DB instance size, and DB instance identifier. **Easy Create** uses the default setting for other configuration options. With **Easy Create** not enabled, you specify more configuration options when you create a database, including ones for availability, security, backups, and maintenance.

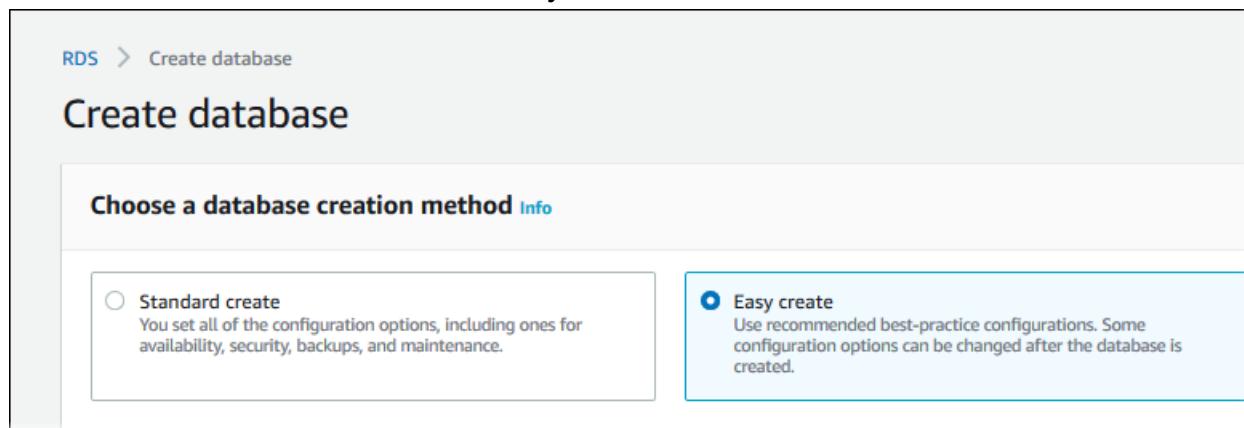
In this example, you use **Easy Create** to create a DB instance running the PostgreSQL database engine with a db.t2.micro DB instance class.

Note

For information about creating DB instances with **Easy Create** not enabled, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

To create a PostgreSQL DB instance with Easy Create enabled

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region in which you want to create the DB instance.
3. In the navigation pane, choose **Databases**.
4. Choose **Create database** and make sure that **Easy Create** is chosen.



5. In **Configuration**, choose **PostgreSQL**.
6. For **DB instance size**, choose **Free tier**.
7. For **DB instance identifier**, enter a name for the DB instance, or leave the default name.
8. For **Master username**, enter a name for the master user, or leave the default name.

The **Create database** page should look similar to the following image.

Create database

Choose a database creation method [Info](#)

Standard Create

You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy Create

Use recommended best-practice configuration options can be changed after the database is created.

Configuration

Engine type [Info](#)

Amazon Aurora



MySQL



MariaDB



PostgreSQL



Oracle

ORACLE®

Microsoft SQL Server



DB instance size

Production

db.r4.large
2 vCPUs
15.25 GiB RAM
500 GiB

Dev/Test

db.m4.large
2 vCPUs
8 GiB RAM
100 GiB

Free tier

db.t2.micro
1 vCPUs
1 GiB RAM
20 GiB

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in this Region.

101

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1–63 characters or backticks (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens.

- To use an automatically generated master password for the DB instance, make sure that the **Auto generate a password** check box is chosen.

To enter your master password, clear the **Auto generate a password** check box, and then enter the same password in **Master password** and **Confirm password**.

- (Optional) Open **View default settings for Easy create**.

View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use [Standard Create](#).

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1234567a)	No
Option Group	default:oracle-se2-19	Yes
Subnet Group	default-vpc-1234567a	Yes
Automatic Backups	Enabled	Yes
VPC Security Group	sg-1a2bcd3e	Yes
Publicly Accessible	No	Yes
Database Port	1521	Yes

You can examine the default settings used when **Easy Create** is enabled. If you want to change one or more settings during database creation, choose **Standard Create** to set them. The **Editable after database creation** column shows which options you can change after database creation. To change a setting with **No** in that column, use **Standard Create**. For settings with **Yes** in that column, you can either use **Standard Create** or modify the DB instance after it's created to change the setting.

- Choose **Create database**.

If you chose to use an automatically generated password, the **View credential details** button appears on the **Databases** page.

To view the master user name and password for the DB instance, choose **View credential details**.



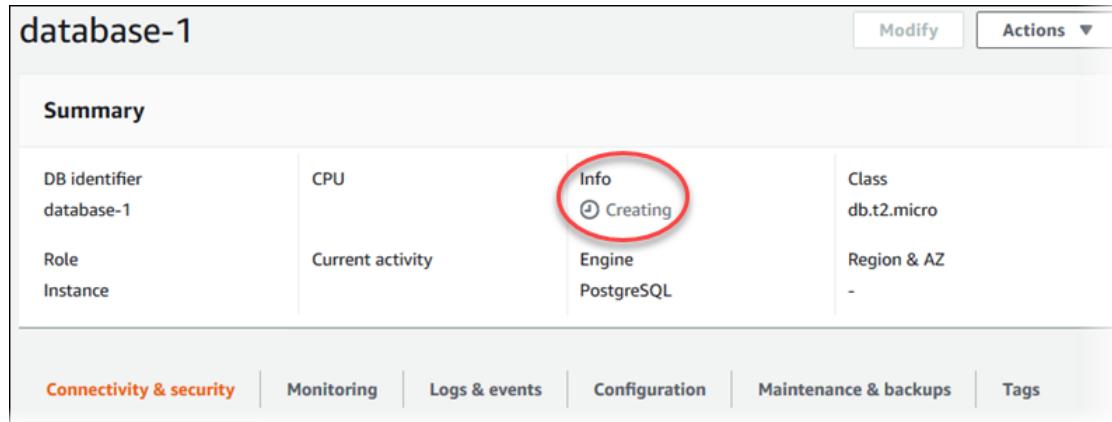
To connect to the DB instance as the master user, use the user name and password that appear.

Important

You can't view the master user password again. If you don't record it, you might have to change it. If you need to change the master user password after the DB instance is available, you can modify the DB instance to do so. For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

- For **Databases**, choose the name of the new PostgreSQL DB instance.

On the RDS console, the details for new DB instance appear. The DB instance has a status of **creating** until the DB instance is ready to use. When the state changes to **available**, you can connect to the DB instance. Depending on the DB instance class and the amount of storage, it can take up to 20 minutes before the new instance is available.



Connecting to a PostgreSQL DB instance

After Amazon RDS provisions your DB instance, you can use any standard SQL client application to connect to the instance. The security group that you assigned to the DB instance when you created it must allow access to the DB instance. If you have difficulty connecting to the DB instance, the problem is most often with the access rules you set up in the security group you assigned to the DB instance.

This section shows two ways to connect to a PostgreSQL DB instance. The first example uses pgAdmin, a popular open-source administration and development tool for PostgreSQL. You can download and use pgAdmin without having a local instance of PostgreSQL on your client computer. The second example uses psql, a command line utility that is part of a PostgreSQL installation. To use psql, you must have a PostgreSQL installed on your client computer or have installed the psql client on your machine.

For more information about connecting to a PostgreSQL DB instance, see [Connecting to a DB instance running the PostgreSQL database engine \(p. 1508\)](#). If you can't connect to your DB instance, see [Troubleshooting connections to your PostgreSQL instance \(p. 1511\)](#).

Topics

- [Using pgAdmin to connect to a PostgreSQL DB instance \(p. 103\)](#)
- [Using psql to connect to a PostgreSQL DB instance \(p. 107\)](#)

Using pgAdmin to connect to a PostgreSQL DB instance

To connect to a PostgreSQL DB instance using pgAdmin

1. Find the endpoint (DNS name) and port number for your DB instance.
 - a. Open the RDS console and then choose **Databases** to display a list of your DB instances.
 - b. Choose the PostgreSQL DB instance name to display its details.
 - c. On the **Connectivity & security** tab, copy the endpoint. Also, note the port number. You need both the endpoint and the port number to connect to the DB instance.

database-1

Summary

DB identifier
database-1

Role
Instance

Connectivity & security Monitoring Logs & events

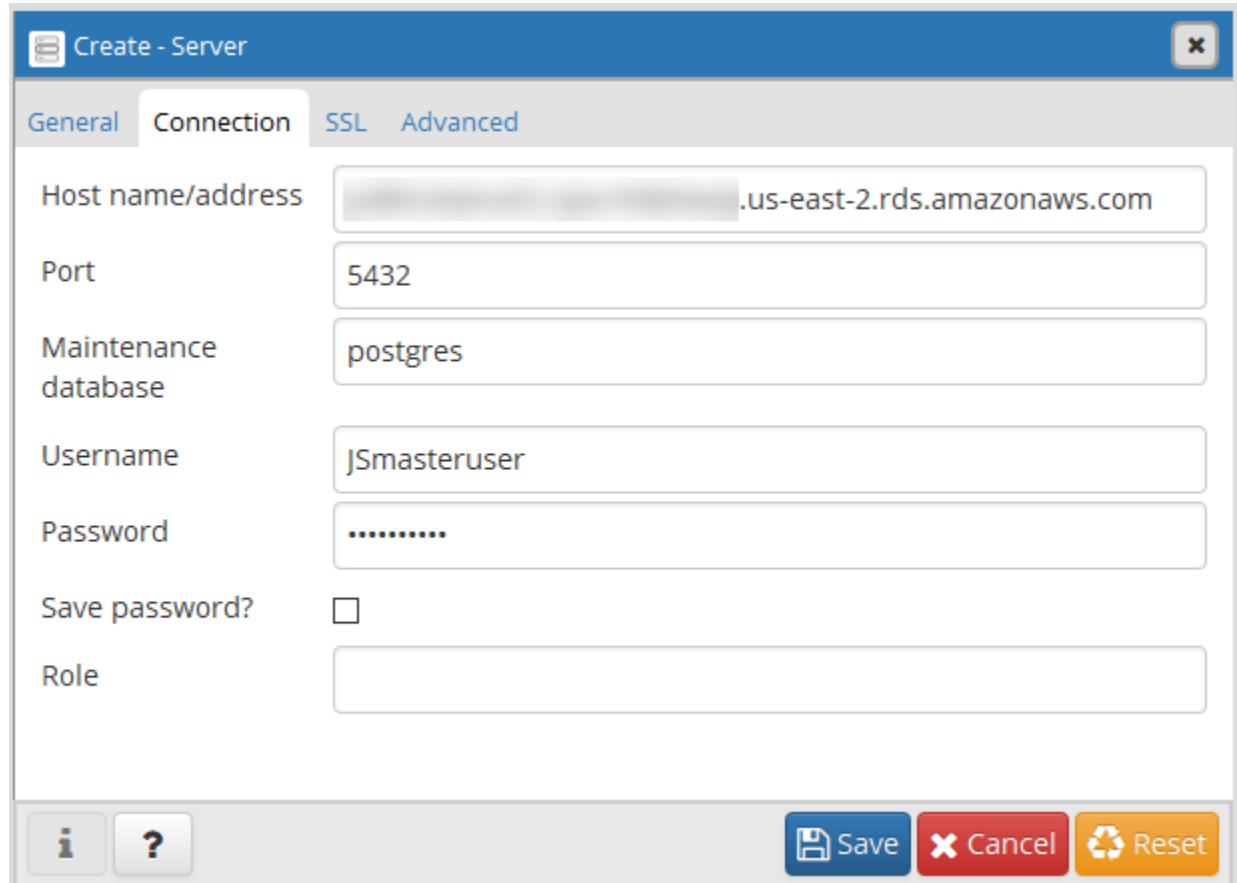
Connectivity & security

Endpoint
database-1.c6c8dntfzzhgv0.us-west-1.rds.amazonaws.com

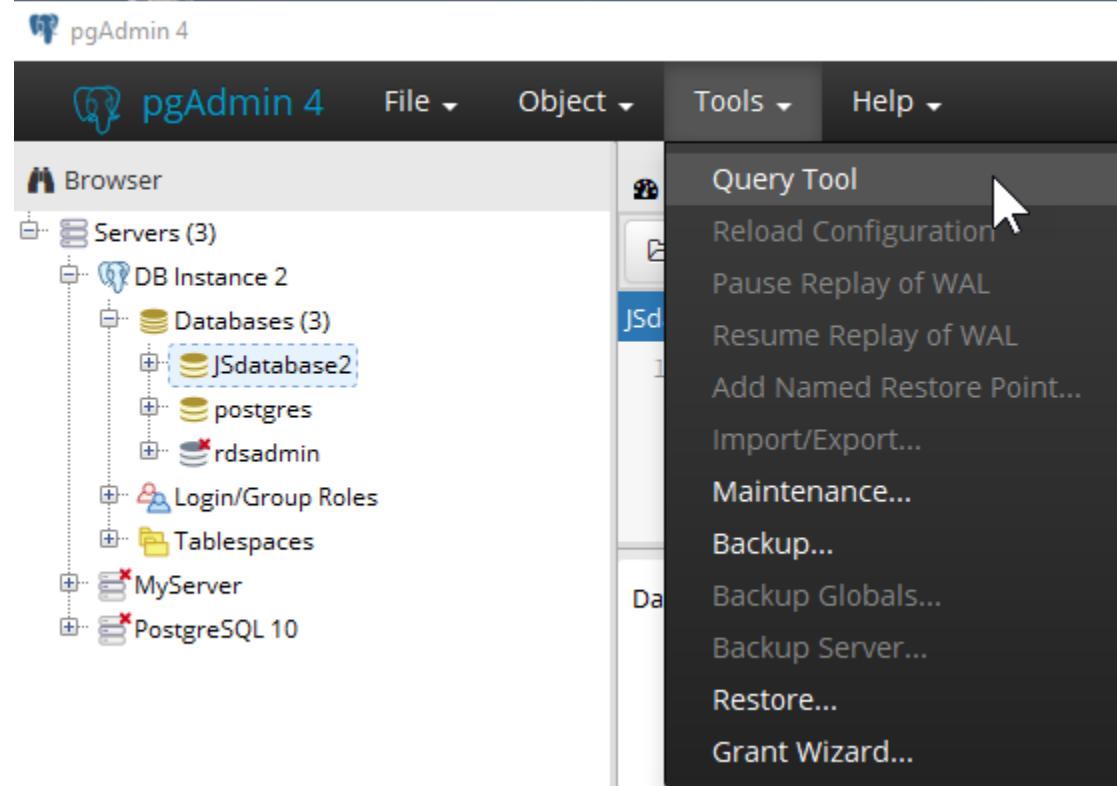
Port
5432

2. Install pgAdmin from <https://www.pgadmin.org/>. You can download and use pgAdmin without having a local instance of PostgreSQL on your client computer.
3. Launch the pgAdmin application on your client computer.
4. Choose **Add Server** from the **File** menu.
5. In the **New Server Registration** dialog box, enter the DB instance endpoint (for example, `database-1.c6c8dntfzzhgv0.us-west-1.rds.amazonaws.com`) in the **Host** box. Don't include the colon or port number as shown on the Amazon RDS console (`database-1.c6c8dntfzzhgv0.us-west-1.rds.amazonaws.com:5432`).

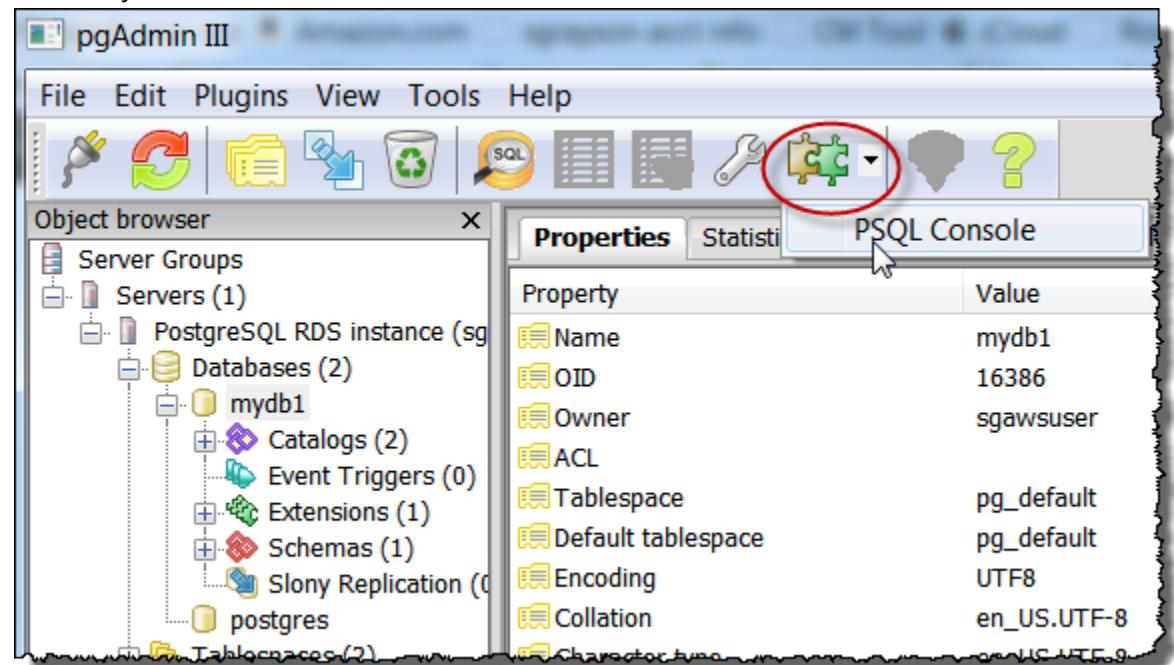
Enter the port you assigned to the DB instance for **Port**. Enter the user name and user password that you entered when you created the DB instance for **Username** and **Password**.



6. Choose **OK**.
7. In the Object browser, expand **Server Groups**. Choose the server (the DB instance) you created, and then choose the database name.



8. Choose the plugin icon and choose **PSQL Console**. The psql command window opens for the default database you created.



9. Use the command window to enter SQL or psql commands. Enter \q to close the window.

Using psql to connect to a PostgreSQL DB instance

If your client computer has PostgreSQL installed, you can use a local instance of psql to connect to a PostgreSQL DB instance. To connect to your PostgreSQL DB instance using psql, provide host information and access credentials.

The following format is used to connect to a PostgreSQL DB instance on Amazon RDS.

```
psql --host=DB_instance_endpoint --port=port --username=master_user_name --password --  
dbname=database_name
```

For example, the following command connects to a database called mypgdb on a PostgreSQL DB instance called mypostgresql using fictitious credentials.

```
psql --host=database-1.c6c8dntfzzhgv0.us-west-1.rds.amazonaws.com --port=5432 --  
username=awsuser --password --dbname=postgres
```

Deleting a DB instance

After you have connected to the sample DB instance that you created, you should delete the DB instance so you are no longer charged for it.

To delete a DB instance with no final DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the DB instance that you want to delete.
4. For **Actions**, choose **Delete**.
5. For **Create final snapshot?**, choose **No**, and select the acknowledgment.
6. Choose **Delete**.

Tutorial: Create a web server and an Amazon RDS DB instance

This tutorial helps you install an Apache web server with PHP and create a MySQL database. The web server runs on an Amazon EC2 instance using Amazon Linux, and the MySQL database is an MySQL DB instance. Both the Amazon EC2 instance and the DB instance run in a virtual private cloud (VPC) based on the Amazon VPC service.

Important

There's no charge for creating an AWS account. However, by completing this tutorial, you might incur costs for the AWS resources you use. You can delete these resources after you complete the tutorial if they are no longer needed.

Note

This tutorial works with Amazon Linux and might not work for other versions of Linux such as Ubuntu.

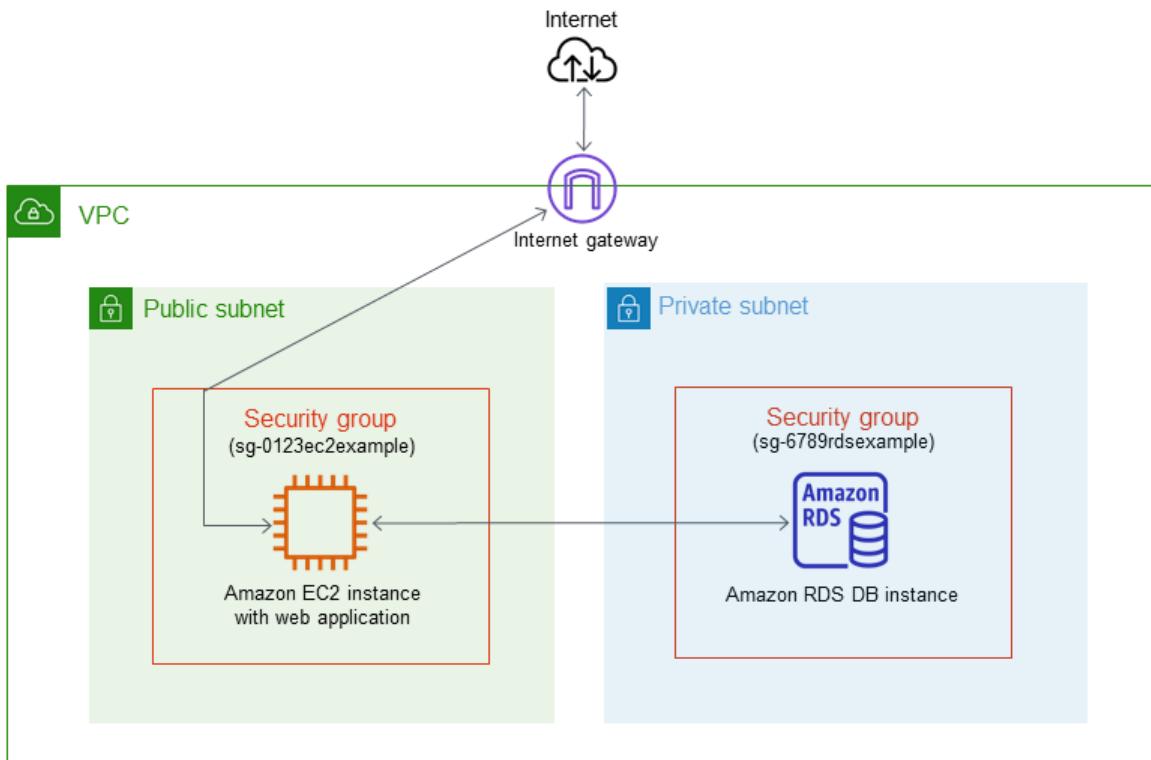
In the tutorial that follows, you specify the VPC, subnets, and security groups when you create the DB instance. You also specify them when you create the EC2 instance to host your web server. The VPC, subnets, and security groups are required for the DB instance and the web server to communicate. After the VPC is set up, this tutorial shows you how to create the DB instance and install the web server. You connect your web server to your DB instance in the VPC using the DB instance endpoint endpoint.

1. Complete the tasks in [Tutorial: Create an Amazon VPC for use with a DB instance \(p. 1737\)](#).

Before you begin this tutorial, make sure that you have a VPC with both public and private subnets, and corresponding security groups. If you don't have these, complete the following tasks in the tutorial:

- a. [Create a VPC with private and public subnets \(p. 1737\)](#)
 - b. [Create additional subnets \(p. 1738\)](#)
 - c. [Create a VPC security group for a public web server \(p. 1739\)](#)
 - d. [Create a VPC security group for a private DB instance \(p. 1740\)](#)
 - e. [Create a DB subnet group \(p. 1740\)](#)
2. [Create a DB instance \(p. 109\)](#)
 3. [Create an EC2 instance and install a web server \(p. 114\)](#)

The following diagram shows the configuration when the tutorial is complete.



Create a DB instance

In this step, you create an Amazon RDS for MySQL DB instance that maintains the data used by a web application.

Important

Before you begin this step, make sure that you have a VPC with both public and private subnets, and corresponding security groups. If you don't have these, see [Tutorial: Create an Amazon VPC for use with a DB instance \(p. 1737\)](#). Complete the steps in [Create a VPC with private and public subnets \(p. 1737\)](#), [Create additional subnets \(p. 1738\)](#), [Create a VPC security group for a public web server \(p. 1739\)](#), and [Create a VPC security group for a private DB instance \(p. 1740\)](#).

To create a MySQL DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the AWS Management Console, choose the AWS Region where you want to create the DB instance. This example uses the US West (Oregon) Region.
3. In the navigation pane, choose **Databases**.
4. Choose **Create database**.
5. On the **Create database** page, shown following, make sure that the **Standard create** option is chosen, and then choose **MySQL**.

RDS > Create database

Create database

Choose a database creation method [Info](#)

Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)

Amazon Aurora

MySQL

MariaDB

PostgreSQL

Oracle

Microsoft SQL Server

Edition

MySQL Community

Known issues/limitations
Review the [Known issues/limitations](#) to learn about potential compatibility issues with specific database versions.

Version

6. In the **Templates** section, choose **Free tier**.
7. In the **Settings** section, set these values:
 - **DB instance identifier** – `tutorial-db-instance`
 - **Master username** – `tutorial_user`
 - **Auto generate a password** – Disable the option.
 - **Master password** – Choose a password.
 - **Confirm password** – Retype the password.

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the current AWS Region.

tutorial-db-instance

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

tutorial_user

1 to 16 alphanumeric characters. First character must be a letter

Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), " (double quote) and @ (at sign).

Confirm password [Info](#)

8. In the **DB instance class** section, enable **Include previous generation classes**, and set these values:
 - **Burstable classes (includes t classes)**
 - **db.t2.micro**

DB instance class

DB instance class [Info](#)
Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

Standard classes (includes m classes)
 Memory optimized classes (includes r and x classes)
 Burstable classes (includes t classes)

db.t2.small
1 vCPUs 2 GiB RAM Not EBS Optimized

New instance classes are available for specific engine versions. [Info](#)

Include previous generation classes

9. In the **Storage and Availability & durability** sections, use the default values.
 10. In the **Connectivity** section, set these values:
 - **Virtual private cloud (VPC)** – Choose an existing VPC with both public and private subnets, such as the tutorial-vpc (vpc-*identifier*) created in [Create a VPC with private and public subnets \(p. 1737\)](#)
- Note**
The VPC must have subnets in different Availability Zones.
- **Subnet group** – The DB subnet group for the VPC, such as the tutorial-db-subnet-group created in [Create a DB subnet group \(p. 1740\)](#)
 - **Public access** – No
 - **VPC security group** – Choose existing
 - **Existing VPC security groups** – Choose an existing VPC security group that is configured for private access, such as the tutorial-db-securitygroup created in [Create a VPC security group for a private DB instance \(p. 1740\)](#).

Remove other security groups, such as the default security group, by choosing the **X** associated with each.

- **Availability Zone** – No preference
- Open **Additional configuration**, and make sure **Database port** uses the default value **3306**.

Connectivity

Virtual private cloud (VPC) [Info](#)
VPC that defines the virtual networking environment for this DB instance.

tutorial-vpc (vpc-08bf0876fa2e229cf)

Only VPCs with a corresponding DB subnet group are listed.

Subnet group [Info](#)
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

tutorial-db-subnet-group

Public access [Info](#)

Yes
Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

No
RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group
Choose a VPC security group to allow access to your database. Ensure that the security group rules allow the appropriate incoming traffic.

Choose existing
Choose existing VPC security groups

Create new
Create new VPC security group

Existing VPC security groups

Choose VPC security groups

tutorial-db-securitygroup X

Availability Zone [Info](#)

No preference

▼ Additional configuration

Database port [Info](#)
TCP/IP port that the database will use for application connections.

3306

11. In the **Database authentication** section, make sure **Password authentication** is selected.
 12. Open the **Additional configuration** section, and enter **sample** for **Initial database name**. Keep the default settings for the other options.
 13. To create your MySQL DB instance, choose **Create database**.
- Your new DB instance appears in the **Databases** list with the status **Creating**.
14. Wait for the **Status** of your new DB instance to show as **Available**. Then choose the DB instance name to show its details.
 15. In the **Connectivity & security** section, view the **Endpoint** and **Port** of the DB instance.

tutorial-db-instance

Summary

DB identifier	CPU
tutorial-db-instance	
Role	Current activity
Instance	0 Connections

Connectivity & security Monitoring Logs & events Configuration

Connectivity & security

Endpoint & port	Network
Endpoint	Availability
tutorial-db-instance. [REDACTED].us-west-2.rds.amazonaws.com	us-west-
Port	VPC
3306	tutorial-
	Subnet [REDACTED]

Note the endpoint and port for your DB instance. You use this information to connect your web server to your DB instance.

16. Complete [Create an EC2 instance and install a web server \(p. 114\)](#).

Create an EC2 instance and install a web server

In this step, you create a web server to connect to the Amazon RDS DB instance that you created in [Create a DB instance \(p. 109\)](#).

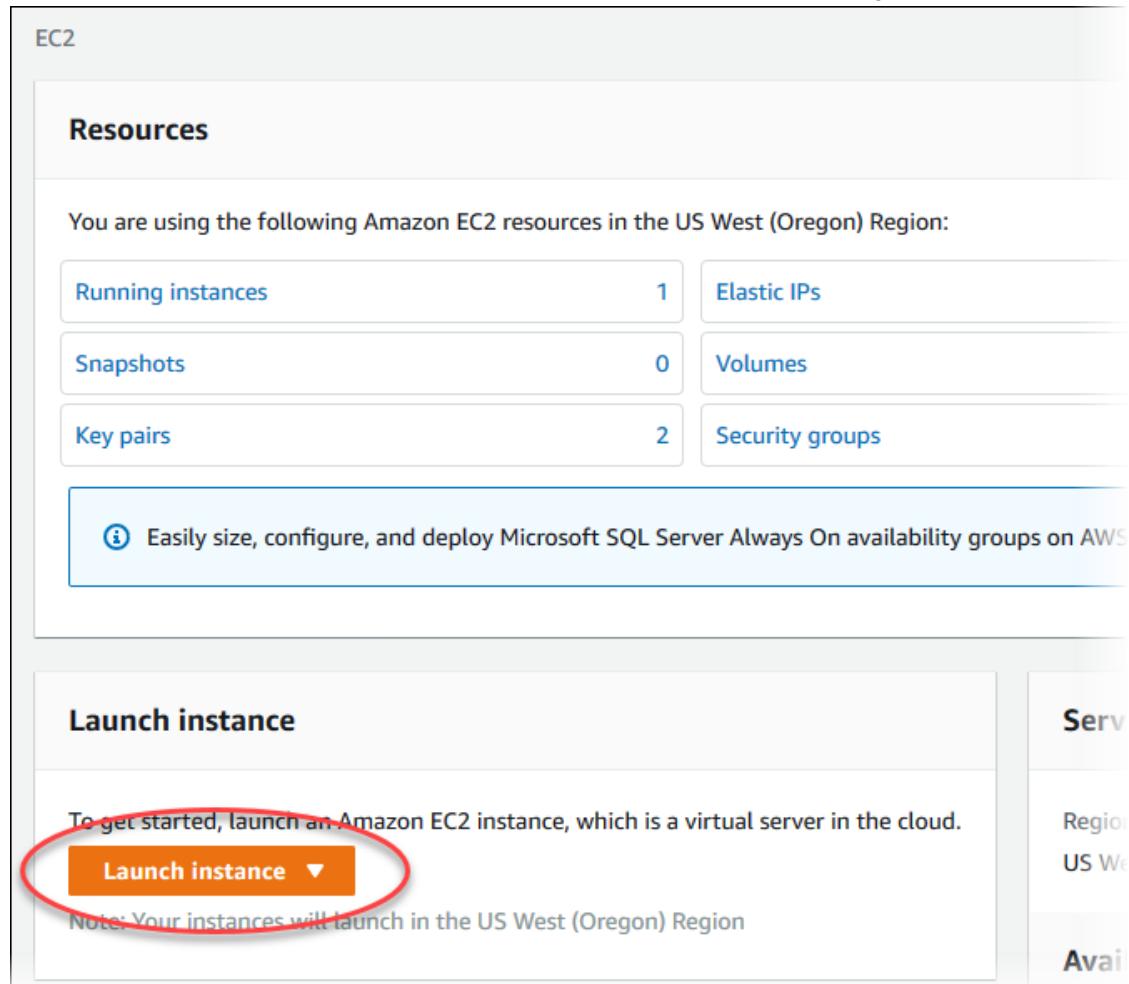
Launch an EC2 instance

First, you create an Amazon EC2 instance in the public subnet of your VPC.

114

To launch an EC2 instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **EC2 Dashboard**, and then choose **Launch instance**, as shown following.



3. Choose the **Amazon Linux 2 AMI**.

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Cancel and Exit

Quick Start

My AMIs

AWS Marketplace

Community AMIs

Free tier only ⓘ

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-07a0da1997b55b23e (64-bit x86) / ami-0787fda5708b00aa3 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is approaching end of life on December 31, 2020 and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-01e78c5619c5e68b4 (64-bit x86) / ami-0a1158e1a81fe09a (64-bit Arm)

Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-063c2d222d223d0e9 (64-bit x86) / ami-0bfc92b18fd79372c (64-bit Arm)

Select

4. Choose the **t2.micro** instance type, as shown following, and then choose **Next: Configure Instance Details**.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families Current generation Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

Family	Type	vCPUs ⓘ	Memory (GiB)	Instance Storage (GB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ	IPv6 Support ⓘ
t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

5. On the **Configure Instance Details** page, shown following, set these values and keep the other values as their defaults:

- Network:** Choose the VPC with both public and private subnets that you chose for the DB instance, such as the `vpc-identifier | tutorial-vpc` created in [Create a VPC with private and public subnets \(p. 1737\)](#).
- Subnet:** Choose an existing public subnet, such as `subnet-identifier | Tutorial public | us-west-2a` created in [Create a VPC security group for a public web server \(p. 1739\)](#).
- Auto-assign Public IP:** Choose **Enable**.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of Instances <input type="text" value="1"/>	<input type="checkbox"/> Launch into Auto Scaling Group
Purchasing option <input type="checkbox"/> Request Spot instances	
Network <input type="text" value="vpc"/> tutorial-vpc <input type="button" value="Create new VPC"/> Subnet <input type="text" value="subnet-000000000000000000"/> Tutorial public us-east-1 <input type="button" value="Create new subnet"/> 249 IP Addresses available Auto-assign Public IP <input checked="" type="checkbox"/> Enable	
Placement group <input type="checkbox"/> Add instance to placement group Capacity Reservation <input type="button" value="Open"/> Domain join directory <input type="text" value="No directory"/> <input type="button" value="Create new directory"/> IAM role <input type="text" value="None"/> <input type="button" value="Create new IAM role"/>	
CPU options <input type="checkbox"/> Specify CPU options	
Shutdown behavior <input type="button" value="Stop"/> Stop - Hibernate behavior <input type="checkbox"/> Enable hibernation as an additional stop behavior Enable termination protection <input type="checkbox"/> Protect against accidental termination Monitoring <input type="checkbox"/> Enable CloudWatch detailed monitoring <small>Additional charges apply.</small> Tenancy <input type="button" value="Shared - Run a shared hardware instance"/> <small>Additional charges will apply for dedicated tenancy.</small> Elastic Inference <input type="checkbox"/> Add an Elastic Inference accelerator	
<input type="button" value="Cancel"/> <input type="button" value="Previous"/> <input style="background-color: #0070C0; color: white; font-weight: bold; border-radius: 5px; border: none; padding: 2px 10px;" type="button" value="Review and Launch"/> <input type="button" value="Next: Add Storage"/>	

6. Choose **Next: Add Storage**.
7. On the **Add Storage** page, keep the default values and choose **Next: Add Tags**.
8. On the **Add Tags** page, shown following, choose **Add Tag**, then enter **Name** for **Key** and enter **tutorial-web-server** for **Value**.

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances <input type="checkbox"/>	Volumes <input type="checkbox"/>
<input type="text" value="Name"/>	<input type="text" value="tutorial-web-server"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="button" value="Add another tag"/> (Up to 50 tags maximum)			
<input type="button" value="Cancel"/> <input type="button" value="Previous"/> <input style="background-color: #0070C0; color: white; font-weight: bold; border-radius: 5px; border: none; padding: 2px 10px;" type="button" value="Review and Launch"/> <input type="button" value="Next: Configure Security Group"/>			

9. Choose **Next: Configure Security Group**.
10. On the **Configure Security Group** page, shown following, choose **Select an existing security group**. Then choose an existing security group, such as the **tutorial-securitygroup** created in [Create a VPC security group for a public web server \(p. 1739\)](#). Make sure that the security group that you choose includes inbound rules for Secure Shell (SSH) and HTTP access.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group
 Select an existing security group

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-[REDACTED]	default	default VPC security group	Copy to new
<input type="checkbox"/> sg-[REDACTED]	tutorial-db-securitygroup	Tutorial DB Instance Security Group	Copy to new
<input checked="" type="checkbox"/> sg-[REDACTED]	tutorial-securitygroup	Tutorial Security Group	Copy to new

Inbound rules for sg-0ef508f81f84a5764 (Selected security groups: sg-0ef508f81f84a5764)

Type <small>(i)</small>	Protocol <small>(i)</small>	Port Range <small>(i)</small>	Source <small>(i)</small>	Description <small>(i)</small>
HTTP	TCP	80	0.0.0.0/0	
SSH	TCP	22	0.0.0.0/0	

[Cancel](#) [Previous](#) [Review and Launch](#)

11. Choose **Review and Launch**.

12. On the **Review Instance Launch** page, shown following, verify your settings and then choose **Launch**.

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Your instance configuration is not eligible for the free usage tier

To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about [free usage tier](#) eligibility and usage restrictions.

[Don't show me this again](#)

AMI Details [Edit AMI](#)

	Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-061392db613a6357b
<small>Free tier eligible</small>	Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root Device Type: ebs Virtualization type: hvm	

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.small	Variable	1	2	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security Group ID	Name	Description
<input type="checkbox"/> sg-[REDACTED]	tutorial-securitygroup	Tutorial Security Group

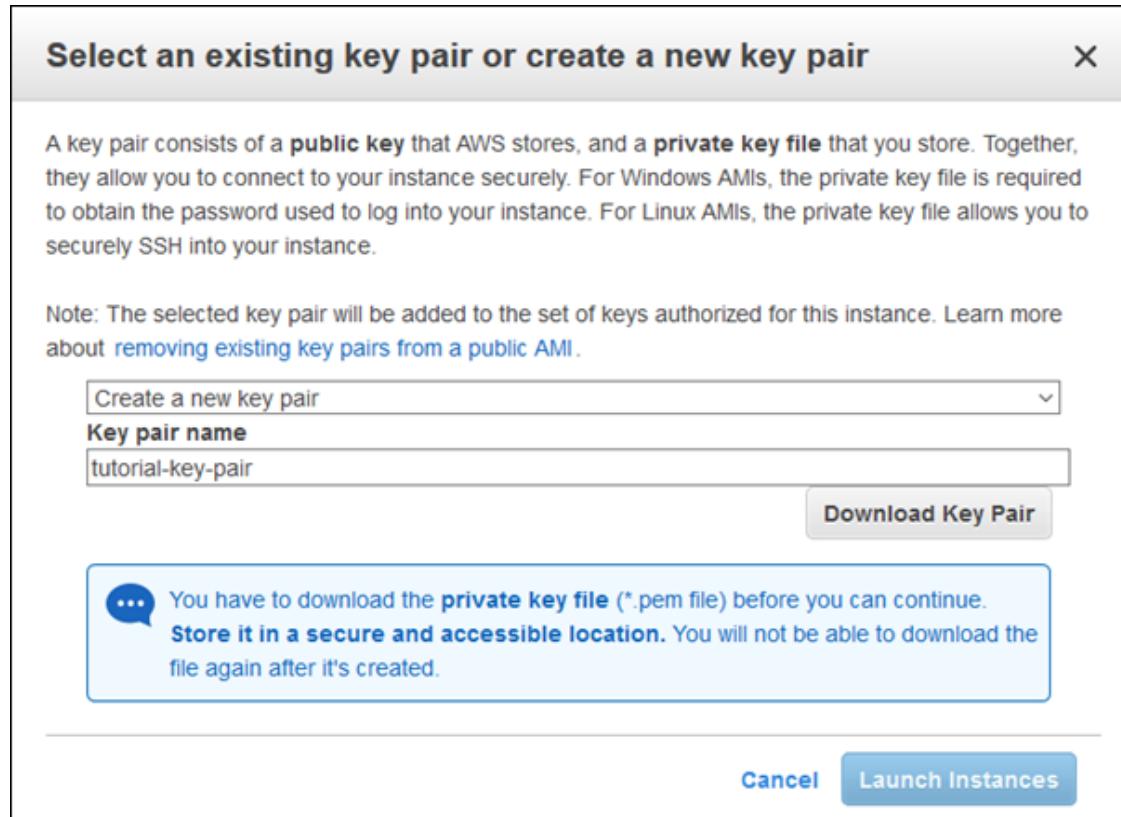
All selected security groups inbound rules

Type <small>(i)</small>	Protocol <small>(i)</small>	Port Range <small>(i)</small>	Source <small>(i)</small>	Description <small>(i)</small>
HTTP	TCP	80	0.0.0.0/0	
SSH	TCP	22	0.0.0.0/0	

Instance Details [Edit instance details](#)

[Cancel](#) [Previous](#) [Launch](#)

13. On the **Select an existing key pair or create a new key pair** page, shown following, choose **Create a new key pair** and set **Key pair name** to **tutorial-key-pair**. Choose **Download Key Pair**, and then save the key pair file on your local machine. You use this key pair file to connect to your EC2 instance.



14. To launch your EC2 instance, choose **Launch Instances**. On the **Launch Status** page, shown following, note the identifier for your new EC2 instance, for example: i-0288d65fd4470b6a9.

Launch Status

Your instances are now launching
The following instance launches have been initiated: [i-0288d65fd4470b6a9](#) [View launch log](#)

Get notified of estimated charges
Create [billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click [View Instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can [connect](#) to them from the Instances screen. [Find out](#) how to connect to your instances.

▼ Here are some helpful resources to get you started

- [How to connect to your Linux instance](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: User Guide](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

- [Create status check alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)
- [Create and attach additional EBS volumes](#) (Additional charges may apply)
- [Manage security groups](#)

[View Instances](#)

15. Choose [View Instances](#) to find your instance.
16. Wait until **Instance Status** for your instance reads as **Running** before continuing.

Install an Apache web server with PHP

Next, you connect to your EC2 instance and install the web server.

To connect to your EC2 instance and install the Apache web server with PHP

1. Connect to the EC2 instance that you created earlier by following the steps in [Connect to your Linux instance](#).
2. Get the latest bug fixes and security updates by updating the software on your EC2 instance. To do this, use the following command.

Note

The `-y` option installs the updates without asking for confirmation. To examine updates before installing, omit this option.

```
sudo yum update -y
```

3. After the updates complete, install the PHP software using the `amazon-linux-extras install` command. This command installs multiple software packages and related dependencies at the same time.

```
sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
```

If you receive an error stating `sudo: amazon-linux-extras: command not found`, then your instance was not launched with an Amazon Linux 2 AMI (perhaps you are using the Amazon Linux AMI instead). You can view your version of Amazon Linux using the following command.

```
cat /etc/system-release
```

For more information, see [Updating instance software](#).

4. Install the Apache web server.

```
sudo yum install -y httpd
```

5. Start the web server with the command shown following.

```
sudo systemctl start httpd
```

You can test that your web server is properly installed and started. To do this, enter the public Domain Name System (DNS) name of your EC2 instance in the address bar of a web browser, for example: `http://ec2-42-8-168-21.us-west-1.compute.amazonaws.com`. If your web server is running, then you see the Apache test page.

If you don't see the Apache test page, check your inbound rules for the VPC security group that you created in [Tutorial: Create an Amazon VPC for use with a DB instance \(p. 1737\)](#). Make sure that your inbound rules include a rule allowing HTTP (port 80) access for the IP address you use to connect to the web server.

Note

The Apache test page appears only when there is no content in the document root directory, `/var/www/html`. After you add content to the document root directory, your content appears at the public DNS address of your EC2 instance instead of the Apache test page.

6. Configure the web server to start with each system boot using the `systemctl` command.

```
sudo systemctl enable httpd
```

To allow `ec2-user` to manage files in the default root directory for your Apache web server, modify the ownership and permissions of the `/var/www` directory. There are many ways to accomplish this task. In this tutorial, you add `ec2-user` to the `apache` group, to give the `apache` group ownership of the `/var/www` directory and assign write permissions to the group.

To set file permissions for the Apache web server

1. Add the `ec2-user` user to the `apache` group.

```
sudo usermod -a -G apache ec2-user
```

2. Log out to refresh your permissions and include the new `apache` group.

```
exit
```

3. Log back in again and verify that the `apache` group exists with the `groups` command.

```
groups
```

Your output looks similar to the following:

```
ec2-user adm wheel apache systemd-journal
```

4. Change the group ownership of the /var/www directory and its contents to the apache group.

```
sudo chown -R ec2-user:apache /var/www
```

5. Change the directory permissions of /var/www and its subdirectories to add group write permissions and set the group ID on subdirectories created in the future.

```
sudo chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
```

6. Recursively change the permissions for files in the /var/www directory and its subdirectories to add group write permissions.

```
find /var/www -type f -exec sudo chmod 0664 {} \;
```

Now, `ec2-user` (and any future members of the `apache` group) can add, delete, and edit files in the Apache document root, enabling you to add content, such as a static website or a PHP application.

Note

A web server running the HTTP protocol provides no transport security for the data that it sends or receives. When you connect to an HTTP server using a web browser, the URLs that you visit, the content of web pages that you receive, and the contents (including passwords) of any HTML forms that you submit are all visible to eavesdroppers anywhere along the network pathway. The best practice for securing your web server is to install support for HTTPS (HTTP Secure), which protects your data with SSL/TLS encryption. For more information, see [Tutorial: Configure SSL/TLS with the Amazon Linux AMI](#) in the *Amazon EC2 User Guide*.

Connect your Apache web server to your DB instance

Next, you add content to your Apache web server that connects to your Amazon RDS DB instance.

To add content to the Apache web server that connects to your DB instance

1. While still connected to your EC2 instance, change the directory to /var/www and create a new subdirectory named `inc`.

```
cd /var/www
mkdir inc
cd inc
```

2. Create a new file in the `inc` directory named `dbinfo.inc`, and then edit the file by calling nano (or the editor of your choice).

```
>dbinfo.inc
nano dbinfo.inc
```

3. Add the following contents to the `dbinfo.inc` file. Here, `db_instance_endpoint` is your DB instance endpoint, without the port, and `master_password` is the master password for your DB instance.

Note

We recommend placing the user name and password information in a folder that isn't part of the document root for your web server. Doing this reduces the possibility of your security information being exposed.

```
<?php

define('DB_SERVER', 'db_instance_endpoint');
define('DB_USERNAME', 'tutorial_user');
define('DB_PASSWORD', 'master password');
define('DB_DATABASE', 'sample');

?>
```

4. Save and close the dbinfo.inc file.
5. Change the directory to /var/www/html.

```
cd /var/www/html
```

6. Create a new file in the html directory named SamplePage.php, and then edit the file by calling nano (or the editor of your choice).

```
>SamplePage.php
nano SamplePage.php
```

7. Add the following contents to the SamplePage.php file:

Note

We recommend placing the user name and password information in a folder that isn't part of the document root for your web server. Doing this reduces the possibility of your security information being exposed.

```
<?php include "../inc/dbinfo.inc"; ?>
<html>
<body>
<h1>Sample page</h1>
<?php

/* Connect to MySQL and select the database. */
$connection = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD);

if (mysqli_connect_errno()) echo "Failed to connect to MySQL: " .
mysqli_connect_error();

$database = mysqli_select_db($connection, DB_DATABASE);

/* Ensure that the EMPLOYEES table exists. */
VerifyEmployeesTable($connection, DB_DATABASE);

/* If input fields are populated, add a row to the EMPLOYEES table. */
$employee_name = htmlentities($_POST['NAME']);
$employee_address = htmlentities($_POST['ADDRESS']);

if (strlen($employee_name) || strlen($employee_address)) {
    AddEmployee($connection, $employee_name, $employee_address);
}
?>
```

```

<!-- Input form -->
<form action="<?PHP echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
  <table border="0">
    <tr>
      <td>NAME</td>
      <td>ADDRESS</td>
    </tr>
    <tr>
      <td>
        <input type="text" name="NAME" maxlength="45" size="30" />
      </td>
      <td>
        <input type="text" name="ADDRESS" maxlength="90" size="60" />
      </td>
      <td>
        <input type="submit" value="Add Data" />
      </td>
    </tr>
  </table>
</form>

<!-- Display table data. -->
<table border="1" cellpadding="2" cellspacing="2">
  <tr>
    <td>ID</td>
    <td>NAME</td>
    <td>ADDRESS</td>
  </tr>
<?php

$result = mysqli_query($connection, "SELECT * FROM EMPLOYEES");

while($query_data = mysqli_fetch_row($result)) {
  echo "<tr>";
  echo "<td>",$query_data[0], "</td>",
        "<td>",$query_data[1], "</td>",
        "<td>",$query_data[2], "</td>";
  echo "</tr>";
}
?>

</table>

<!-- Clean up. -->
<?php

mysqli_free_result($result);
mysqli_close($connection);

?>

</body>
</html>

<?php

/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
  $n = mysqli_real_escape_string($connection, $name);
  $a = mysqli_real_escape_string($connection, $address);

  $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";

  if(!mysqli_query($connection, $query)) echo("<p>Error adding employee data.</p>");
```

```
}

/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
    if(!TableExists("EMPLOYEES", $connection, $dbName))
    {
        $query = "CREATE TABLE EMPLOYEES (
            ID int(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
            NAME VARCHAR(45),
            ADDRESS VARCHAR(90)
        )";

        if(!mysqli_query($connection, $query)) echo("<p>Error creating table.</p>");
    }
}

/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
    $t = mysqli_real_escape_string($connection, $tableName);
    $d = mysqli_real_escape_string($connection, $dbName);

    $checktable = mysqli_query($connection,
        "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME = '$t' AND
        TABLE_SCHEMA = '$d'");

    if(mysqli_num_rows($checktable) > 0) return true;

    return false;
}
?>
```

8. Save and close the SamplePage.php file.
9. Verify that your web server successfully connects to your DB instance by opening a web browser and browsing to <http://EC2 instance endpoint/SamplePage.php>, for example: <http://ec2-55-122-41-31.us-west-2.compute.amazonaws.com/SamplePage.php>.

You can use SamplePage.php to add data to your DB instance. The data that you add is then displayed on the page. To verify that the data was inserted into the table, you can install MySQL on the Amazon EC2 instance, connect to the DB instance, and query the table.

To make sure that your DB instance is as secure as possible, verify that sources outside of the VPC can't connect to your DB instance.

After you have finished testing your web server and your database, you should delete your DB instance and your Amazon EC2 instance.

- To delete a DB instance, follow the instructions in [Deleting a DB instance \(p. 324\)](#). You don't need to create a final snapshot.
- To terminate an Amazon EC2 instance, follow the instruction in [Terminate your instance](#) in the *Amazon EC2 User Guide*.

Amazon RDS Tutorials

The AWS documentation includes several tutorials that guide you through common Amazon RDS use cases. Many of these tutorials show you how to use Amazon RDS with other AWS services.

Note

You can find more tutorials at the [AWS Database Blog](#). For information about training, see [AWS Training and Certification](#).

Tutorials in this guide

The following tutorials in this guide show you how to perform common tasks with Amazon RDS:

- [Tutorial: Create an Amazon VPC for use with a DB instance \(p. 1737\)](#)

Learn how to include a DB instance in an Amazon virtual private cloud (VPC) that shares data with a web server that is running on an Amazon EC2 instance in the same VPC.

- [Tutorial: Create a web server and an Amazon RDS DB instance \(p. 108\)](#)

Learn how to install an Apache web server with PHP and create a MySQL database. The web server runs on an Amazon EC2 instance using Amazon Linux, and the MySQL database is a MySQL DB instance. Both the Amazon EC2 instance and the DB instance run in an Amazon VPC.

- [Tutorial: Restore a DB instance from a DB snapshot \(p. 394\)](#)

Learn how to restore a DB instance from a DB snapshot.

- [Tutorial: Use tags to specify which DB instances to stop \(p. 304\)](#)

Learn how to use tags to specify which DB instances to stop.

- [Tutorial: Log the state of an Amazon RDS instance using EventBridge \(p. 554\)](#)

Learn how to log a DB instance stage change using Amazon EventBridge and AWS Lambda.

Tutorials in other AWS guides

The following tutorials in other AWS guides show you how to perform common tasks with Amazon RDS:

- [Tutorial: Rotating a Secret for an AWS Database](#) in the *AWS Secrets Manager User Guide*

Learn how to create a secret for an AWS database and configure the secret to rotate on a schedule. You trigger one rotation manually, and then confirm that the new version of the secret continues to provide access.

- [Tutorial: Configuring a Lambda function to access Amazon RDS in an Amazon VPC](#) in the *AWS Lambda Developer Guide*

Learn how to create a Lambda function to access a database, create a table, add a few records, and retrieve the records from the table. You also learn how to invoke the Lambda function and verify the query results.

- [Tutorials and samples](#) in the *AWS Elastic Beanstalk Developer Guide*

Learn how to deploy applications that use Amazon RDS databases with AWS Elastic Beanstalk.

- [Using Data from an Amazon RDS Database to Create an Amazon ML Datasource in the Amazon Machine Learning Developer Guide](#)

Learn how to create an Amazon Machine Learning (Amazon ML) datasource object from data stored in a MySQL DB instance.

- [Manually Enabling Access to an Amazon RDS Instance in a VPC in the Amazon QuickSight User Guide](#)

Learn how to enable Amazon QuickSight access to an Amazon RDS DB instance in a VPC.

Best practices for Amazon RDS

Learn best practices for working with Amazon RDS. As new best practices are identified, we will keep this section up to date.

Topics

- [Amazon RDS basic operational guidelines \(p. 128\)](#)
- [DB instance RAM recommendations \(p. 129\)](#)
- [Using Enhanced Monitoring to identify operating system issues \(p. 129\)](#)
- [Using metrics to identify performance issues \(p. 129\)](#)
- [Best practices for working with MySQL storage engines \(p. 134\)](#)
- [Best practices for working with MariaDB storage engines \(p. 135\)](#)
- [Best practices for working with Oracle \(p. 137\)](#)
- [Best practices for working with PostgreSQL \(p. 137\)](#)
- [Best practices for working with SQL Server \(p. 138\)](#)
- [Working with DB parameter groups \(p. 139\)](#)
- [Amazon RDS new features and best practices presentation video \(p. 139\)](#)

Note

For common recommendations for Amazon RDS, see [Using Amazon RDS recommendations \(p. 407\)](#).

Amazon RDS basic operational guidelines

The following are basic operational guidelines that everyone should follow when working with Amazon RDS. Note that the Amazon RDS Service Level Agreement requires that you follow these guidelines:

- Monitor your memory, CPU, and storage usage. Amazon CloudWatch can be set up to notify you when usage patterns change or when you approach the capacity of your deployment, so that you can maintain system performance and availability.
- Scale up your DB instance when you are approaching storage capacity limits. You should have some buffer in storage and memory to accommodate unforeseen increases in demand from your applications.
- Enable automatic backups and set the backup window to occur during the daily low in write IOPS. That's when a backup is least disruptive to your database usage.
- If your database workload requires more I/O than you have provisioned, recovery after a failover or database failure will be slow. To increase the I/O capacity of a DB instance, do any or all of the following:
 - Migrate to a different DB instance class with high I/O capacity.
 - Convert from magnetic storage to either General Purpose or Provisioned IOPS storage, depending on how much of an increase you need. For information on available storage types, see [Amazon RDS storage types \(p. 40\)](#).

If you convert to Provisioned IOPS storage, make sure you also use a DB instance class that is optimized for Provisioned IOPS. For information on Provisioned IOPS, see [Provisioned IOPS SSD storage \(p. 42\)](#).

- If you are already using Provisioned IOPS storage, provision additional throughput capacity.
- If your client application is caching the Domain Name Service (DNS) data of your DB instances, set a time-to-live (TTL) value of less than 30 seconds. Because the underlying IP address of a DB instance can change after a failover, caching the DNS data for an extended time can lead to connection failures if your application tries to connect to an IP address that no longer is in service.
- Test failover for your DB instance to understand how long the process takes for your particular use case and to ensure that the application that accesses your DB instance can automatically connect to the new DB instance after failover occurs.

DB instance RAM recommendations

An Amazon RDS performance best practice is to allocate enough RAM so that your *working set* resides almost completely in memory. The working set is the data and indexes that are frequently in use on your instance. The more you use the DB instance, the more the working set will grow.

To tell if your working set is almost all in memory, check the ReadIOPS metric (using Amazon CloudWatch) while the DB instance is under load. The value of ReadIOPS should be small and stable. If scaling up the DB instance class—to a class with more RAM—results in a dramatic drop in ReadIOPS, your working set was not almost completely in memory. Continue to scale up until ReadIOPS no longer drops dramatically after a scaling operation, or ReadIOPS is reduced to a very small amount. For information on monitoring a DB instance's metrics, see [Viewing DB instance metrics \(p. 548\)](#).

Using Enhanced Monitoring to identify operating system issues

When Enhanced Monitoring is enabled, Amazon RDS provides metrics in real time for the operating system (OS) that your DB instance runs on. You can view the metrics for your DB instance using the console, or consume the Enhanced Monitoring JSON output from Amazon CloudWatch Logs in a monitoring system of your choice. For more information about Enhanced Monitoring, see [Using Enhanced Monitoring \(p. 471\)](#).

Using metrics to identify performance issues

To identify performance issues caused by insufficient resources and other common bottlenecks, you can monitor the metrics available for your Amazon RDS DB instance.

Viewing performance metrics

You should monitor performance metrics on a regular basis to see the average, maximum, and minimum values for a variety of time ranges. If you do so, you can identify when performance is degraded. You can also set Amazon CloudWatch alarms for particular metric thresholds so you are alerted if they are reached.

To troubleshoot performance issues, it's important to understand the baseline performance of the system. When you set up a new DB instance and get it running with a typical workload, you should

capture the average, maximum, and minimum values of all of the performance metrics at a number of different intervals (for example, one hour, 24 hours, one week, two weeks) to get an idea of what is normal. It helps to get comparisons for both peak and off-peak hours of operation. You can then use this information to identify when performance is dropping below standard levels.

To view performance metrics

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose a DB instance.
3. Choose **Monitoring**. The first eight performance metrics display. The metrics default to showing information for the current day.
4. Use the numbered buttons at top right to page through the additional metrics, or choose adjust the settings to see more metrics.
5. Choose a performance metric to adjust the time range in order to see data for other than the current day. You can change the **Statistic**, **Time Range**, and **Period** values to adjust the information displayed. For example, to see the peak values for a metric for each day of the last two weeks, set **Statistic** to **Maximum**, **Time Range** to **Last 2 Weeks**, and **Period** to **Day**.

Note

Changing the **Statistic**, **Time Range**, and **Period** values changes them for all metrics. The updated values persist for the remainder of your session or until you change them again.

You can also view performance metrics using the CLI or API. For more information, see [Viewing DB instance metrics \(p. 548\)](#).

To set a CloudWatch alarm

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose a DB instance.
3. Choose **Logs & events**.
4. In the **CloudWatch alarms** section, choose **Create alarm**.

Create alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

Settings

[Refresh](#)

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

Send notifications

- Yes
 No

Send notifications to

- ARN
 New email or SMS topic

Topic name

Name of the topic.

Manually enter a topic name...

With these recipients

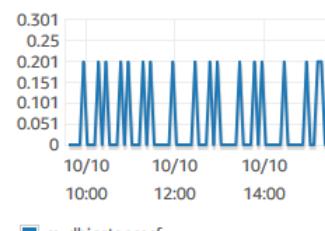
Email addresses or phone numbers of SMS enabled devices to send the notifications to

awsAccount@domain.com

Metric

Average ▼ of CPU Utilization ▼

CPU Utilization Percent



Threshold

>= ▼ Percent

Evaluation period

1 consecutive period(s) of 5 Minutes ▼

Name of alarm

awsrds-mydbinstancecf-High-CPU-Utilization

[Cancel](#)

[Create alarm](#)

5. For **Send notifications**, choose **Yes**, and for **Send notifications to**, choose **New email or SMS topic**.
6. For **Topic name**, enter a name for the notification, and for **With these recipients**, enter a comma-separated list of email addresses and phone numbers.
7. For **Metric**, choose the alarm statistic and metric to set.
8. For **Threshold**, specify whether the metric must be greater than, less than, or equal to the threshold, and specify the threshold value.
9. For **Evaluation period**, choose the evaluation period for the alarm, and for **consecutive period(s) of**, choose the period during which the threshold must have been reached in order to trigger the alarm.
10. For **Name of alarm**, enter a name for the alarm.
11. Choose **Create Alarm**.

The alarm appears in the **CloudWatch alarms** section.

Evaluating performance metrics

A DB instance has a number of different categories of metrics, and how to determine acceptable values depends on the metric.

CPU

- CPU Utilization – Percentage of computer processing capacity used.

Memory

- Freeable Memory – How much RAM is available on the DB instance, in megabytes. The red line in the Monitoring tab metrics is marked at 75% for CPU, Memory and Storage Metrics. If instance memory consumption frequently crosses that line, then this indicates that you should check your workload or upgrade your instance.
- Swap Usage – How much swap space is used by the DB instance, in megabytes.

Disk space

- Free Storage Space – How much disk space is not currently being used by the DB instance, in megabytes.

Input/output operations

- Read IOPS, Write IOPS – The average number of disk read or write operations per second.
- Read Latency, Write Latency – The average time for a read or write operation in milliseconds.
- Read Throughput, Write Throughput – The average number of megabytes read from or written to disk per second.
- Queue Depth – The number of I/O operations that are waiting to be written to or read from disk.

Network traffic

- Network Receive Throughput, Network Transmit Throughput – The rate of network traffic to and from the DB instance in bytes per second.

Database connections

- DB Connections – The number of client sessions that are connected to the DB instance.

For more detailed individual descriptions of the performance metrics available, see [Monitoring Amazon RDS metrics with Amazon CloudWatch \(p. 540\)](#).

Generally speaking, acceptable values for performance metrics depend on what your baseline looks like and what your application is doing. Investigate consistent or trending variances from your baseline. Advice about specific types of metrics follows:

- **High CPU or RAM consumption** – High values for CPU or RAM consumption might be appropriate, provided that they are in keeping with your goals for your application (like throughput or concurrency) and are expected.

- **Disk space consumption** – Investigate disk space consumption if space used is consistently at or above 85 percent of the total disk space. See if it is possible to delete data from the instance or archive data to a different system to free up space.
- **Network traffic** – For network traffic, talk with your system administrator to understand what expected throughput is for your domain network and Internet connection. Investigate network traffic if throughput is consistently lower than expected.
- **Database connections** – Consider constraining database connections if you see high numbers of user connections in conjunction with decreases in instance performance and response time. The best number of user connections for your DB instance will vary based on your instance class and the complexity of the operations being performed. You can determine the number of database connections by associating your DB instance with a parameter group where the *User Connections* parameter is set to other than 0 (unlimited). You can either use an existing parameter group or create a new one. For more information, see [Working with DB parameter groups \(p. 228\)](#).
- **IOPS metrics** – The expected values for IOPS metrics depend on disk specification and server configuration, so use your baseline to know what is typical. Investigate if values are consistently different than your baseline. For best IOPS performance, make sure your typical working set will fit into memory to minimize read and write operations.

For issues with any performance metrics, one of the first things you can do to improve performance is tune the most used and most expensive queries to see if that lowers the pressure on system resources. For more information, see [Tuning queries \(p. 133\)](#)

If your queries are tuned and an issue persists, consider upgrading your Amazon RDS [DB instance classes \(p. 7\)](#) to one with more of the resource (CPU, RAM, disk space, network bandwidth, I/O capacity) that is related to the issue you are experiencing.

Tuning queries

One of the best ways to improve DB instance performance is to tune your most commonly used and most resource-intensive queries to make them less expensive to run.

MySQL Query Tuning

Go to [Optimizing SELECT statements](#) in the MySQL documentation for more information on writing queries for better performance. You can also go to [MySQL performance tuning and optimization resources](#) for additional query tuning resources.

Oracle Query Tuning

Go to the [Database SQL Tuning Guide](#) in the Oracle documentation for more information on writing and analyzing queries for better performance.

SQL Server Query Tuning

Go to [Analyzing a query](#) in the SQL Server documentation to improve queries for SQL Server DB instances. You can also use the execution-, index- and I/O-related data management views (DMVs) described in the [Dynamic management views and functions](#) documentation to troubleshoot SQL Server query issues.

A common aspect of query tuning is creating effective indexes. You can use the [Database engine Tuning Advisor](#) to get potential index improvements for your DB instance. For more information, see [Analyzing your database workload on an Amazon RDS DB instance with SQL Server Tuning Advisor \(p. 812\)](#).

PostgreSQL Query Tuning

Go to [Using EXPLAIN](#) in the PostgreSQL documentation to learn how to analyze a query plan. You can use this information to modify a query or underlying tables in order to improve query performance. You

can also go to [Controlling the planner with explicit JOIN clauses](#) to get tips about how to specify joins in your query for the best performance.

MariaDB Query Tuning

Go to [Query optimizations](#) in the MariaDB documentation for more information on writing queries for better performance.

Best practices for working with MySQL storage engines

Both table sizes and number of tables in a MySQL database can affect performance.

Table size

Typically, operating system constraints on file sizes determine the effective maximum table size for MySQL databases. So, the limits usually aren't determined by internal MySQL constraints.

On a MySQL DB instance, avoid tables in your database growing too large. Although the general storage limit is 64 TiB, provisioned storage limits restrict the maximum size of a MySQL table file to 16 TiB. Partition your large tables so that file sizes are well under the 16 TiB limit. This approach can also improve performance and recovery time. For more information, see [MySQL file size limits in Amazon RDS \(p. 950\)](#).

Very large tables (greater than 100 GB in size) can negatively affect performance for both reads and writes (including DML statements and especially DDL statements). Indexes on large tables can significantly improve select performance, but they can also degrade the performance of DML statements. DDL statements, such as `ALTER TABLE`, can be significantly slower for the large tables because those operations might completely rebuild a table in some cases. These DDL statements might lock the tables for the duration of the operation.

The amount of memory required by MySQL for reads and writes depends on the tables involved in the operations. It is a best practice to have at least enough RAM to hold the indexes of *actively used* tables. To find the ten largest tables and indexes in a database, use the following query:

```
SELECT CONCAT(table_schema, '.', table_name),
       CONCAT(ROUND(table_rows / 1000000, 2), 'M')                                rows,
       CONCAT(ROUND(data_length / ( 1024 * 1024 * 1024 ), 2), 'G')                  DATA,
       CONCAT(ROUND(index_length / ( 1024 * 1024 * 1024 ), 2), 'G')                  idx,
       CONCAT(ROUND(( data_length + index_length ) / ( 1024 * 1024 * 1024 ), 2), 'G') total_size,
       ROUND(index_length / data_length, 2)                                         idxfrac
  FROM   information_schema.TABLES
 ORDER  BY data_length + index_length DESC
 LIMIT  10;
```

Number of tables

While the underlying file system might have a limit on the number of files that represent tables, MySQL has no limit on the number of tables. However, the total number of tables in the MySQL InnoDB storage engine can contribute to the performance degradation, regardless of the size of those tables. To limit the operating system impact, you can split the tables across multiple databases in the same MySQL DB instance. Doing so might limit the number of files in a directory but won't solve the overall problem.

When there is performance degradation because of a large number of tables (more than 10 thousand), it is caused by MySQL working with storage files, including opening and closing them. To address this issue, you can increase the size of the `table_open_cache` and `table_definition_cache` parameters. However, increasing the values of those parameters might significantly increase the amount of memory MySQL uses, and might even use all of the available memory. For more information, see [How MySQL Opens and Closes Tables](#) in the MySQL documentation.

In addition, too many tables can significantly affect MySQL startup time. Both a clean shutdown and restart and a crash recovery can be affected, especially in versions prior to MySQL 8.0.

We recommend having fewer than ten thousand tables total across all of the databases in a DB instance. For a use case with a large number of tables in a MySQL database, see [One Million Tables in MySQL 8.0](#).

Storage engine

The point-in-time restore and snapshot restore features of Amazon RDS for MySQL require a crash-recoverable storage engine and are supported for the InnoDB storage engine only. Although MySQL supports multiple storage engines with varying capabilities, not all of them are optimized for crash recovery and data durability. For example, the MyISAM storage engine does not support reliable crash recovery and might prevent a Point-In-Time Restore or snapshot restore from working as intended. This might result in lost or corrupt data when MySQL is restarted after a crash.

InnoDB is the recommended and supported storage engine for MySQL DB instances on Amazon RDS. InnoDB instances can also be migrated to Aurora, while MyISAM instances can't be migrated. However, MyISAM performs better than InnoDB if you require intense, full-text search capability. If you still choose to use MyISAM with Amazon RDS, following the steps outlined in [Automated backups with unsupported MySQL storage engines \(p. 336\)](#) can be helpful in certain scenarios for snapshot restore functionality.

If you want to convert existing MyISAM tables to InnoDB tables, you can use the process outlined in the [MySQL documentation](#). MyISAM and InnoDB have different strengths and weaknesses, so you should fully evaluate the impact of making this switch on your applications before doing so.

In addition, Federated Storage Engine is currently not supported by Amazon RDS for MySQL.

Best practices for working with MariaDB storage engines

Both table sizes and number of tables in a MariaDB database can affect performance.

Table size

Typically, operating system constraints on file sizes determine the effective maximum table size for MariaDB databases. So, the limits usually aren't determined by internal MariaDB constraints.

On a MariaDB DB instance, avoid tables in your database growing too large. Although the general storage limit is 64 TiB, provisioned storage limits restrict the maximum size of a MariaDB table file to 16 TiB. Partition your large tables so that file sizes are well under the 16 TiB limit. This approach can also improve performance and recovery time.

Very large tables (greater than 100 GB in size) can negatively affect performance for both reads and writes (including DML statements and especially DDL statements). Indexes on large tables can significantly improve select performance, but they can also degrade the performance of DML

statements. DDL statements, such as `ALTER TABLE`, can be significantly slower for the large tables because those operations might completely rebuild a table in some cases. These DDL statements might lock the tables for the duration of the operation.

The amount of memory required by MariaDB for reads and writes depends on the tables involved in the operations. It is a best practice to have at least enough RAM to hold the indexes of *actively* used tables. To find the ten largest tables and indexes in a database, use the following query:

```
SELECT CONCAT(table_schema, '.', table_name),
       CONCAT(ROUND(table_rows / 1000000, 2), 'M')                                     rows,
       CONCAT(ROUND(data_length / ( 1024 * 1024 * 1024 ), 2), 'G')                      DATA,
       CONCAT(ROUND(index_length / ( 1024 * 1024 * 1024 ), 2), 'G')                     idx,
       CONCAT(ROUND(( data_length + index_length ) / ( 1024 * 1024 * 1024 ), 2), 'G')    total_size,
       ROUND(index_length / data_length, 2)                                              idxfrac
  FROM  information_schema.TABLES
 ORDER  BY data_length + index_length DESC
 LIMIT  10;
```

Number of tables

While the underlying file system might have a limit on the number of files that represent tables, MariaDB has no limit on the number of tables. However, the total number of tables in the MariaDB InnoDB storage engine can contribute to the performance degradation, regardless of the size of those tables. To limit the operating system impact, you can split the tables across multiple databases in the same MariaDB DB instance. Doing so might limit the number of files in a directory but won't solve the overall problem.

When there is performance degradation because of a large number of tables (more than 10 thousand), it is caused by MariaDB working with storage files, including opening and closing them. To address this issue, you can increase the size of the `table_open_cache` and `table_definition_cache` parameters. However, increasing the values of those parameters might significantly increase the amount of memory MariaDB uses, and might even use all of the available memory. For more information, see [Optimizing table_open_cache](#) in the MariaDB documentation.

In addition, too many tables can significantly affect MariaDB startup time. Both a clean shutdown and restart and a crash recovery can be affected. We recommend having fewer than ten thousand tables total across all of the databases in a DB instance.

Storage engine

The point-in-time restore and snapshot restore features of Amazon RDS for MariaDB require a crash-recoverable storage engine. Although MariaDB supports multiple storage engines with varying capabilities, not all of them are optimized for crash recovery and data durability. For example, although Aria is a crash-safe replacement for MyISAM, it might still prevent a point-in-time restore or snapshot restore from working as intended. This might result in lost or corrupt data when MariaDB is restarted after a crash. InnoDB (for version 10.2 and higher) and XtraDB (for version 10.0 and 10.1) are the recommended and supported storage engines for MariaDB DB instances on Amazon RDS. If you still choose to use Aria with Amazon RDS, following the steps outlined in [Automated backups with unsupported MariaDB storage engines \(p. 336\)](#) can be helpful in certain scenarios for snapshot restore functionality.

If you want to convert existing MyISAM tables to InnoDB tables, you can use the process outlined in the [MariaDB documentation](#). MyISAM and InnoDB have different strengths and weaknesses, so you should fully evaluate the impact of making this switch on your applications before doing so.

Best practices for working with Oracle

For information about best practices for working with Amazon RDS for Oracle, see [Best practices for running Oracle database on Amazon Web Services](#).

A 2020 AWS virtual workshop included a presentation on running production Oracle databases on Amazon RDS. A video of the presentation is available [here](#).

Best practices for working with PostgreSQL

Two important areas where you can improve performance with PostgreSQL on Amazon RDS are when loading data into a DB instance and when using the PostgreSQL autovacuum feature. The following sections cover some of the practices we recommend for these areas.

Loading data into a PostgreSQL DB instance

When loading data into an Amazon RDS PostgreSQL DB instance, you should modify your DB instance settings and your DB parameter group values to allow for the most efficient importing of data into your DB instance.

Modify your DB instance settings to the following:

- Disable DB instance backups (set `backup_retention` to 0)
- Disable Multi-AZ

Modify your DB parameter group to include the following settings. You should test the parameter settings to find the most efficient settings for your DB instance:

- Increase the value of the `maintenance_work_mem` parameter. For more information about PostgreSQL resource consumption parameters, see the [PostgreSQL documentation](#).
- Increase the value of the `checkpoint_segments` and `checkpoint_timeout` parameters to reduce the number of writes to the wal log.
- Disable the `synchronous_commit` parameter (do not turn off FSYNC).
- Disable the PostgreSQL autovacuum parameter.
- Make sure none of the tables you are importing are unlogged. Data stored in unlogged tables can be lost during a failover. For more information, see [CREATE TABLE UNLOGGED](#).

Use the `pg_dump -Fc` (compressed) or `pg_restore -j` (parallel) commands with these settings.

After the load operation completes, return your DB instance and DB parameters to their normal settings.

Working with the PostgreSQL autovacuum feature

The autovacuum feature for PostgreSQL databases is a feature that we strongly recommend you use to maintain the health of your PostgreSQL DB instance. Autovacuum automates the execution of the `VACUUM` and `ANALYZE` command; using autovacuum is required by PostgreSQL, not imposed by Amazon RDS, and its use is critical to good performance. The feature is enabled by default for all new Amazon RDS PostgreSQL DB instances, and the related configuration parameters are appropriately set by default.

Your database administrator needs to know and understand this maintenance operation. For the PostgreSQL documentation on autovacuum, see [Routine vacuuming](#).

Autovacuum is not a "resource free" operation, but it works in the background and yields to user operations as much as possible. When enabled, autovacuum checks for tables that have had a large number of updated or deleted tuples. It also protects against loss of very old data due to transaction ID wraparound. For more information, see [Preventing transaction ID wraparound failures](#).

Autovacuum should not be thought of as a high-overhead operation that can be reduced to gain better performance. On the contrary, tables that have a high velocity of updates and deletes will quickly deteriorate over time if autovacuum is not run.

Important

Not running autovacuum can result in an eventual required outage to perform a much more intrusive vacuum operation. When an Amazon RDS PostgreSQL DB instance becomes unavailable because of an over conservative use of autovacuum, the PostgreSQL database will shut down to protect itself. At that point, Amazon RDS must perform a single-user-mode full vacuum directly on the DB instance , which can result in a multi-hour outage. Thus, we strongly recommend that you do not turn off autovacuum, which is enabled by default.

The autovacuum parameters determine when and how hard autovacuum works. The `autovacuum_vacuum_threshold` and `autovacuum_vacuum_scale_factor` parameters determine when autovacuum is run. The `autovacuum_max_workers`, `autovacuum_nap_time`, `autovacuum_cost_limit`, and `autovacuum_cost_delay` parameters determine how hard autovacuum works. For more information about autovacuum, when it runs, and what parameters are required, see the [PostgreSQL documentation](#).

The following query shows the number of "dead" tuples in a table named `table1` :

```
PROMPT> select relname, n_dead_tup, last_vacuum, last_autovacuum from pg_catalog.pg_stat_all_tables where n_dead_tup > 0 and relname = 'table1';
```

The results of the query will resemble the following:

relname	n_dead_tup	last_vacuum	last_autovacuum
tasks	81430522		

(1 row)

Best practices for working with SQL Server

Best practices for a Multi-AZ deployment with a SQL Server DB instance include the following:

- Use Amazon RDS DB events to monitor failovers. For example, you can be notified by text message or email when a DB instance fails over. For more information about Amazon RDS events, see [Using Amazon RDS event notification \(p. 487\)](#).
- If your application caches DNS values, set time to live (TTL) to less than 30 seconds. Setting TTL as so is a good practice in case there is a failover, where the IP address might change and the cached value might no longer be in service.
- We recommend that you *do not* enable the following modes because they turn off transaction logging, which is required for Multi-AZ:
 - Simple recover mode
 - Offline mode
 - Read-only mode
- Test to determine how long it takes for your DB instance to failover. Failover time can vary due to the type of database, the instance class, and the storage type you use. You should also test your application's ability to continue working if a failover occurs.

- To shorten failover time, you should do the following:
 - Ensure that you have sufficient Provisioned IOPS allocated for your workload. Inadequate I/O can lengthen failover times. Database recovery requires I/O.
 - Use smaller transactions. Database recovery relies on transactions, so if you can break up large transactions into multiple smaller transactions, your failover time should be shorter.
- Take into consideration that during a failover, there will be elevated latencies. As part of the failover process, Amazon RDS automatically replicates your data to a new standby instance. This replication means that new data is being committed to two different DB instances, so there might be some latency until the standby DB instance has caught up to the new primary DB instance.
- Deploy your applications in all Availability Zones. If an Availability Zone does go down, your applications in the other Availability Zones will still be available.

When working with a Multi-AZ deployment of SQL Server, remember that Amazon RDS creates replicas for all SQL Server databases on your instance. If you don't want specific databases to have secondary replicas, set up a separate DB instance that doesn't use Multi-AZ for those databases.

Amazon RDS for SQL Server best practices video

The 2019 AWS re:Invent conference included a presentation on new features and best practices for working with SQL Server on Amazon RDS. A video of the presentation is available [here](#).

Working with DB parameter groups

We recommend that you try out DB parameter group changes on a test DB instance before applying parameter group changes to your production DB instances. Improperly setting DB engine parameters in a DB parameter group can have unintended adverse effects, including degraded performance and system instability. Always exercise caution when modifying DB engine parameters and back up your DB instance before modifying a DB parameter group.

For information about backing up your DB instance, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#).

Amazon RDS new features and best practices presentation video

The 2019 AWS re:Invent conference included a presentation on new Amazon RDS features and best practices for monitoring, analyzing, and tuning database performance using RDS. A video of the presentation is available [here](#).

Configuring an Amazon RDS DB instance

This section shows how to set up your Amazon RDS DB instance. Before creating a DB instance, decide on the DB instance class that will run the DB instance. Also, decide where the DB instance will run by choosing an AWS Region. Next, create the DB instance.

You can configure a DB instance with an option group and a DB parameter group.

- An *option group* specifies features, called options, that are available for a particular Amazon RDS DB instance.
- A *DB parameter group* acts as a container for engine configuration values that are applied to one or more DB instances.

The options and parameters that are available depend on the DB engine and DB engine version. You can specify an option group and a DB parameter group when you create a DB instance, or you can modify a DB instance to specify them.

Topics

- [Creating an Amazon RDS DB instance \(p. 141\)](#)
- [Connecting to an Amazon RDS DB instance \(p. 162\)](#)
- [Working with option groups \(p. 212\)](#)
- [Working with DB parameter groups \(p. 228\)](#)

Creating an Amazon RDS DB instance

The basic building block of Amazon RDS is the DB instance, where you create your databases. You choose the engine-specific characteristics of the DB instance when you create it. You also choose the storage capacity, CPU, memory, and so on, of the AWS instance on which the database server runs.

Important

Before you can create or connect to a DB instance, you must complete the tasks in [Setting up for Amazon RDS \(p. 67\)](#).

Console

You can create a DB instance by using the AWS Management Console with **Easy Create** enabled or not enabled. With **Easy Create** enabled, you specify only the DB engine type, DB instance size, and DB instance identifier. **Easy Create** uses the default setting for other configuration options. With **Easy Create** not enabled, you specify more configuration options when you create a database, including ones for availability, security, backups, and maintenance.

Note

In the following procedure, **Standard Create** is enabled, and **Easy Create** isn't enabled. This procedure uses Microsoft SQL Server as an example.

For examples that use **Easy Create** to walk you through creating and connecting to sample DB instances for each engine, see [Getting started with Amazon RDS \(p. 73\)](#). For an example that uses the original console to create a DB instance, see [Original console example \(p. 157\)](#).

To create a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region in which you want to create the DB instance.
3. In the navigation pane, choose **Databases**.
4. Choose **Create database**.
5. In **Choose a database creation method**, select **Standard Create**.
6. In **Engine options**, choose the engine type: MariaDB, Microsoft SQL Server, MySQL, Oracle, or PostgreSQL. **Microsoft SQL Server** is shown here.

Create database

Choose a database creation method Info

Standard Create

You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy Create

Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

Amazon Aurora



MySQL



MariaDB



PostgreSQL



Oracle



Microsoft SQL Server



Edition

SQL Server Express Edition

Affordable database management system that supports database sizes up to 10 GB.

SQL Server Web Edition

In accordance with Microsoft's licensing policies, it can only be used to support public and Internet-accessible webpages, websites, web applications, and web services.

SQL Server Standard Edition

Core data management and business intelligence capabilities for mission-critical applications and mixed workloads.

SQL Server Enterprise Edition

Comprehensive high-end capabilities for mission-critical applications with demanding database workloads and business intelligence requirements.

Version Info

SQL Server 2017 14.00.3010.1-1

7. For **Edition**, if you're using Oracle or SQL Server choose the DB engine edition that you want to use. MySQL has only one option for the edition, and MariaDB and PostgreSQL have none.
8. For **Version**, choose the engine version.
9. In **Templates**, choose the template that matches your use case. If you choose **Production**, the following are preselected in a later step:
 - **Multi-AZ failover option**
 - **Provisioned IOPS storage option**
 - **Enable deletion protection** option

We recommend these features for any production environment.

Note

Template choices vary by edition.

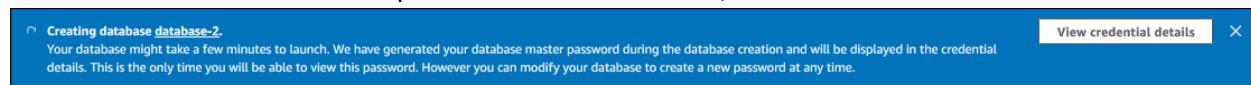
10. To enter your master password, do the following:
 - a. In the **Settings** section, open **Credential Settings**.
 - b. Clear the **Auto generate a password** check box.
 - c. (Optional) Change the **Master username** value and enter the same password in **Master password** and **Confirm password**.

By default, the new DB instance uses an automatically generated password for the master user.

11. For the remaining sections, specify your DB instance settings. For information about each setting, see [Settings for DB instances \(p. 145\)](#).
12. Choose **Create database**.

If you chose to use an automatically generated password, the **View credential details** button appears on the **Databases** page.

To view the master user name and password for the DB instance, choose **View credential details**.



To connect to the DB instance as the master user, use the user name and password that appear.

Important

You can't view the master user password again. If you don't record it, you might have to change it. If you need to change the master user password after the DB instance is available, modify the DB instance to do so. For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

13. For **Databases**, choose the name of the new DB instance.

On the RDS console, the details for the new DB instance appear. The DB instance has a status of **creating** until the DB instance is created and ready for use. When the state changes to **available**, you can connect to the DB instance. Depending on the DB instance class and storage allocated, it can take several minutes for the new instance to be available.

database-1				Modify	Actions ▾
Summary					
DB identifier	CPU	Info	Creating	Class	db.t2.micro
database-1		(?)		Region & AZ	-
Role	Current activity	Engine	SQL Server Express Edition	Maintenance & backups	Tags
Instance					
Connectivity & security		Monitoring	Logs & events	Configuration	Maintenance & backups

AWS CLI

To create a DB instance by using the AWS CLI, call the [create-db-instance](#) command with the following parameters. This example uses Microsoft SQL Server.

For information about each setting, see [Settings for DB instances \(p. 145\)](#).

- `--db-instance-identifier`
- `--db-instance-class`
- `--vpc-security-group-ids`
- `--db-subnet-group`
- `--engine`
- `--master-username`
- `--master-user-password`
- `--allocated-storage`
- `--backup-retention-period`

Example

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
  --engine sqlserver-se \
  --db-instance-identifier mymssqlserver \
  --allocated-storage 250 \
  --db-instance-class db.t3.large \
  --vpc-security-group-ids mysecuritygroup \
  --db-subnet-group mydbsubnetgroup \
  --master-username masterawsuser \
  --master-user-password masteruserpassword \
  --backup-retention-period 3
```

For Windows:

```
aws rds create-db-instance ^
  --engine sqlserver-se ^
  --db-instance-identifier mydbinstance ^
  --allocated-storage 250 ^
  --db-instance-class db.t3.large ^
  --vpc-security-group-ids mysecuritygroup ^
  --db-subnet-group mydbsubnetgroup ^
  --master-username masterawsuser ^
  --master-user-password masteruserpassword ^
  --backup-retention-period 3
```

This command produces output similar to the following.

```
DBINSTANCE mydbinstance db.t3.large sqlserver-se 250 sa creating 3 **** n
10.50.2789
SECGROUP default active
PARAMGRP default.sqlserver-se-14 in-sync
```

RDS API

To create a DB instance by using the Amazon RDS API, call the [CreateDBInstance](#) operation with the following parameters.

For information about each setting, see [Settings for DB instances \(p. 145\)](#).

- `AllocatedStorage`
- `BackupRetentionPeriod`
- `DBInstanceClass`
- `DBInstanceIdentifier`
- `VpcSecurityGroupIds`
- `DBSubnetGroup`
- `Engine`
- `MasterUsername`
- `MasterUserPassword`

Settings for DB instances

In the following table, you can find details about settings that you choose when you create a DB instance. The table also shows the DB engines for which each setting is supported.

You can create a DB instance using the console, the `create-db-instance` CLI command, or the `CreateDBInstance` RDS API operation.

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
Allocated storage	<p>The amount of storage to allocate for your DB instance (in gigabytes). In some cases, allocating a higher amount of storage for your DB instance than the size of your database can improve I/O performance.</p> <p>For more information, see Amazon RDS DB instance storage (p. 40).</p>	CLI option: <code>--allocated-storage</code> API parameter: <code>AllocatedStorage</code>	All
Auto minor version upgrade	<p>Enable auto minor version upgrade to enable your DB instance to receive preferred minor DB engine version upgrades automatically when they become available. Amazon RDS performs automatic minor version upgrades in the maintenance window.</p>	CLI option: <code>--auto-minor-version-upgrade</code> <code>--no-auto-minor-version-upgrade</code> API parameter: <code>AutoMinorVersionUpgrade</code>	All except SQL Server
Availability zone	<p>The Availability Zone for your DB instance. Use the default value of No Preference unless you want to specify an Availability Zone.</p> <p>For more information, see Regions, Availability Zones, and Local Zones (p. 49).</p>	CLI option: <code>--availability-zone</code> API parameter: <code>AvailabilityZone</code>	All

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
Backup replication	<p>Choose Enable replication to another AWS Region to create backups in an additional Region for disaster recovery.</p> <p>Then choose the Destination Region for the additional backups.</p>	<p>Not available when creating a DB instance. For information on enabling cross-Region backups using the AWS CLI or RDS API, see Enabling cross-Region automated backups (p. 338).</p>	Oracle
Backup retention period	<p>The number of days that you want automatic backups of your DB instance to be retained. For any nontrivial DB instance, set this value to 1 or greater.</p> <p>For more information, see Working with backups (p. 328).</p>	CLI option: <code>--backup-retention-period</code> API parameter: <code>BackupRetentionPeriod</code>	All
Backup window	<p>The time period during which Amazon RDS automatically takes a backup of your DB instance. Unless you have a specific time that you want to have your database backed up, use the default of No Preference.</p> <p>For more information, see Working with backups (p. 328).</p>	CLI option: <code>--preferred-backup-window</code> API parameter: <code>PreferredBackupWindow</code>	All
Character set	<p>The character set for your DB instance. The default value of AL32UTF8 for the DB character set is for the Unicode 5.0 UTF-8 Universal character set. You can't change the DB character set after you create the DB instance.</p> <p>The DB character set is different from the national character set, which is called the NCHAR character set. Unlike the DB character set, the NCHAR character set specifies the encoding for NCHAR data types (NCHAR, NVARCHAR2, and NCLOB) columns without affecting database metadata.</p> <p>For more information, see RDS for Oracle character sets (p. 996).</p>	CLI option: <code>--character-set-name</code> API parameter: <code>CharacterSetName</code>	Oracle
Collation	<p>A server-level collation for your DB instance.</p> <p>For more information, see Server-level collation for Microsoft SQL Server (p. 814).</p>	CLI option: <code>--character-set-name</code> API parameter: <code>CharacterSetName</code>	SQL Server

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
Copy tags to snapshots	<p>This option copies any DB instance tags to a DB snapshot when you create a snapshot.</p> <p>For more information, see Tagging Amazon RDS resources (p. 299).</p>	CLI option: <code>--copy-tags-to-snapshot</code> <code>--no-copy-tags-to-snapshot</code> RDS API parameter: <code>CopyTagsToSnapshot</code>	All
Database port	<p>The port that you want to access the DB instance through. The default port is shown. If you use a DB security group with your DB instance, this port value must be the same one that you provided when creating the DB security group.</p> <p>Note The firewalls at some companies block connections to the default MariaDB, MySQL, and PostgreSQL ports. If your company firewall blocks the default port, enter another port for your DB instance.</p>	CLI option: <code>--port</code> RDS API parameter: <code>Port</code>	All

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
Database authentication	<p>The database authentication option that you want to use.</p> <p>Choose Password authentication to authenticate database users with database passwords only.</p> <p>Choose Password and IAM DB authentication to authenticate database users with database passwords and user credentials through IAM users and roles. For more information, see IAM database authentication for MySQL and PostgreSQL (p. 1660). This option is only supported for MySQL and PostgreSQL.</p> <p>Choose Password and Kerberos authentication to authenticate database users with database passwords and Kerberos authentication through an AWS Managed Microsoft AD created with AWS Directory Service. Next, choose the directory or choose Create a new Directory.</p> <p>For more information, see one of the following:</p> <ul style="list-style-type: none"> • Using Kerberos authentication for MySQL (p. 938) • Configuring Kerberos authentication for Amazon RDS for Oracle (p. 1014) • Using Kerberos authentication with Amazon RDS for PostgreSQL (p. 1520) 	<p>IAM:</p> <p>CLI option:</p> <pre>--enable-iam-database-authentication</pre> <p>RDS API parameter:</p> <pre>EnableIAMDatabaseAuthentication</pre> <p>Kerberos:</p> <p>CLI option:</p> <pre>--domain</pre> <pre>--domain-iam-role-name</pre> <p>RDS API parameter:</p> <pre>Domain</pre> <pre>DomainIAMRoleName</pre>	MySQL Oracle PostgreSQL
DB engine version	The version of database engine that you want to use.	<p>CLI option:</p> <pre>--engine-version</pre> <p>RDS API parameter:</p> <pre>EngineVersion</pre>	All

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
DB instance class	<p>The configuration for your DB instance. For example, a <code>db.t3.small</code> instance class has 2 GiB memory, 2 vCPUs, 1 virtual core, a variable ECU, and a moderate I/O capacity.</p> <p>If possible, choose an instance class large enough that a typical query working set can be held in memory. When working sets are held in memory the system can avoid writing to disk, which improves performance.</p> <p>For more information, see DB instance classes (p. 7).</p>	CLI option: <code>--db-instance-class</code> RDS API parameter: <code>DBInstanceClass</code>	All
DB instance identifier	The name for your DB instance. Name your DB instances in the same way that you name your on-premises servers. Your DB instance identifier can contain up to 63 alphanumeric characters, and must be unique for your account in the AWS Region you chose. You can add some intelligence to the name, such as including the AWS Region and DB engine you chose, for example <code>sqlsrvr-instance1</code> .	CLI option: <code>--db-instance-identifier</code> RDS API parameter: <code>DBInstanceIdentifier</code>	All
DB parameter group	<p>A parameter group for your DB instance. You can choose the default parameter group or you can create a custom parameter group.</p> <p>For more information, see Working with DB parameter groups (p. 228).</p>	CLI option: <code>--db-parameter-group-name</code> RDS API parameter: <code>DBParameterGroupName</code>	All
Deletion protection	<p>Enable deletion protection to prevent your DB instance from being deleted. If you create a production DB instance with the AWS Management Console, deletion protection is enabled by default.</p> <p>For more information, see Deleting a DB instance (p. 324).</p>	CLI option: <code>--deletion-protection</code> <code>--no-deletion-protection</code> RDS API parameter: <code>DeletionProtection</code>	All

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
Encryption	<p>Enable Encryption to enable encryption at rest for this DB instance.</p> <p>For more information, see Encrypting Amazon RDS resources (p. 1630).</p>	<p>CLI option: <code>--storage-encrypted</code> <code>--no-storage-encrypted</code></p> <p>RDS API parameter: <code>StorageEncrypted</code></p>	All
Enhanced Monitoring	<p>Enable enhanced monitoring to enable gathering metrics in real time for the operating system that your DB instance runs on.</p> <p>For more information, see Using Enhanced Monitoring (p. 471).</p>	<p>CLI options: <code>--monitoring-interval</code> <code>--monitoring-role-arn</code></p> <p>RDS API parameters: <code>MonitoringInterval</code> <code>MonitoringRoleArn</code></p>	All

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
Initial database name	<p>The name for the database on your DB instance. If you don't provide a name, Amazon RDS doesn't create a database on the DB instance (except for Oracle and PostgreSQL). The name can't be a word reserved by the database engine, and has other constraints depending on the DB engine.</p> <p>MariaDB and MySQL:</p> <ul style="list-style-type: none"> • It must contain 1–64 alphanumeric characters. <p>Oracle:</p> <ul style="list-style-type: none"> • It must contain 1–8 alphanumeric characters. • It can't be <code>NULL</code>. The default value is <code>ORCL</code>. • It must begin with a letter. <p>PostgreSQL:</p> <ul style="list-style-type: none"> • It must contain 1–63 alphanumeric characters. • It must begin with a letter or an underscore. Subsequent characters can be letters, underscores, or digits (0–9). • The initial database name is <code>postgres</code>. 	<p>CLI option: <code>--db-name</code></p> <p>RDS API parameter: <code>DBName</code></p>	All except SQL Server
License	<p>The license model:</p> <ul style="list-style-type: none"> • Choose license-included for Microsoft SQL Server. • Choose license-included or bring-your-own-license for Oracle. 	<p>CLI option: <code>--license-model</code></p> <p>RDS API parameter: <code>LicenseModel</code></p>	SQL Server Oracle
Maintenance window	<p>The 30-minute window in which pending modifications to your DB instance are applied. If the time period doesn't matter, choose No Preference.</p> <p>For more information, see The Amazon RDS maintenance window (p. 268).</p>	<p>CLI option: <code>--preferred-maintenance-window</code></p> <p>RDS API parameter: <code>PreferredMaintenanceWindow</code></p>	All

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
Master password	<p>The password for your master user account. The password has the following number of printable ASCII characters (excluding /, ", a space, and @) depending on the DB engine:</p> <ul style="list-style-type: none"> • Oracle: 8–30 • MariaDB and MySQL: 8–41 • SQL Server and PostgreSQL: 8–128 	<p>CLI option: <code>--master-user-password</code></p> <p>RDS API parameter: <code>MasterUserPassword</code></p>	All
Master username	<p>The name that you use as the master user name to log on to your DB instance with all database privileges.</p> <ul style="list-style-type: none"> • It can contain 1–16 alphanumeric characters and underscores. • Its first character must be a letter. • It can't be a word reserved by the database engine. <p>For more information on privileges granted to the master user, see the following topics:</p> <ul style="list-style-type: none"> • MariaDB security on Amazon RDS (p. 582) • Microsoft SQL Server security (p. 635) • MySQL security on Amazon RDS (p. 833) • Securing Oracle DB instance connections (p. 1010) • Using SSL with a PostgreSQL DB instance (p. 1513) 	<p>CLI option: <code>--master-username</code></p> <p>RDS API parameter: <code>MasterUsername</code></p>	All
Microsoft SQL Server Windows Authentication	<p>Enable Microsoft SQL Server Windows authentication, then Browse Directory to choose the directory where you want to allow authorized domain users to authenticate with this SQL Server instance using Windows Authentication.</p>	<p>CLI options: <code>--domain</code> <code>--domain-iam-role-name</code></p> <p>RDS API parameters: <code>Domain</code> <code>DomainIAMRoleName</code></p>	SQL Server

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
Multi-AZ deployment	<p>Create a standby instance to create a passive secondary replica of your DB instance in another Availability Zone for failover support. We recommend Multi-AZ for production workloads to maintain high availability.</p> <p>For development and testing, you can choose Do not create a standby instance.</p> <p>For more information, see High availability (Multi-AZ) for Amazon RDS (p. 53).</p>	CLI option: --multi-az --no-multi-az RDS API parameter: MultiAZ	All
National character set (NCHAR)	<p>The national character set for your DB instance, commonly called the NCHAR character set. You can set the national character set to either AL16UTF16 (default) or UTF-8. You can't change the national character set after you create the DB instance.</p> <p>The national character set is different from the DB character set. Unlike the DB character set, the national character set specifies the encoding only for NCHAR data types (NCHAR, NVARCHAR2, and NCLOB) columns without affecting database metadata.</p> <p>For more information, see RDS for Oracle character sets (p. 996).</p>	CLI option: --nchar-character-set-name API parameter: NcharCharacterSetName	Oracle
Option group	<p>An option group for your DB instance. You can choose the default option group or you can create a custom option group.</p> <p>For more information, see Working with option groups (p. 212).</p>	CLI option: --option-group-name RDS API parameter: OptionGroupName	All

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
Performance Insights	<p>Enable Performance Insights to monitor your DB instance load so that you can analyze and troubleshoot your database performance.</p> <p>Choose a retention period to determine how much rolling data history to keep. The default of seven days is in the free tier. Long-term retention (two years) is priced per vCPU per month.</p> <p>Choose a master key to use to protect the key used to encrypt this database volume. Choose from the master keys in your account, or enter the key from a different account.</p> <p>For more information, see Using Performance Insights on Amazon RDS (p. 412).</p>	<p>CLI options:</p> <pre>--enable-performance-insights --no-enable-performance-insights --performance-insights-retention-period --performance-insights-kms-key-id</pre> <p>RDS API parameters:</p> <pre>EnablePerformanceInsights PerformanceInsightsRetentionPeriod PerformanceInsightsKMSKeyId</pre>	All

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
Public access	<p>Publicly accessible to give the DB instance a public IP address, meaning that it's accessible outside the VPC. To be publicly accessible, the DB instance also has to be in a public subnet in the VPC.</p> <p>Not publicly accessible to make the DB instance accessible only from inside the VPC.</p> <p>For more information, see Hiding a DB instance in a VPC from the internet (p. 1729).</p> <p>To connect to a DB instance from outside of its Amazon VPC, the DB instance must be publicly accessible, access must be granted using the inbound rules of the DB instance's security group, and other requirements must be met. For more information, see Can't connect to Amazon RDS DB instance (p. 1746).</p> <p>If your DB instance is isn't publicly accessible, you can also use an AWS Site-to-Site VPN connection or an AWS Direct Connect connection to access it from a private network. For more information, see Internetwork traffic privacy (p. 1643).</p>	<p>CLI option:</p> <pre>--publicly-accessible --no-publicly-accessible</pre> <p>RDS API parameter:</p> <pre>PubliclyAccessible</pre>	All
Storage autoscaling	<p>Enable storage autoscaling to enable Amazon RDS to automatically increase storage when needed to avoid having your DB instance run out of storage space.</p> <p>Use Maximum storage threshold to set the upper limit for Amazon RDS to automatically increase storage for your DB instance. The default is 1,000 GiB.</p> <p>For more information, see Managing capacity automatically with Amazon RDS storage autoscaling (p. 317).</p>	<p>CLI option:</p> <pre>--max-allocated-storage</pre> <p>RDS API parameter:</p> <pre>MaxAllocatedStorage</pre>	All

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
Storage type	The storage type for your DB instance. For more information, see Amazon RDS storage types (p. 40) .	CLI option: <code>--storage-type</code> RDS API parameter: <code>StorageType</code>	All
Subnet group	This setting depends on the platform that you are on. If you are a new customer to AWS, choose default , which is the default DB subnet group that was created for your account. If you are creating a DB instance on the earlier E2-Classic platform, you might want your DB instance in a specific VPC. In this case, choose the DB subnet group that you created for that VPC.	CLI option: <code>--db-subnet-group-name</code> RDS API parameter: <code>DBSubnetGroupName</code>	All
Time zone	The time zone for your DB instance. If you don't choose a time zone, your DB instance uses the default time zone. You can't change the time zone after the DB instance is created. For more information, see Local time zone for Microsoft SQL Server DB instances (p. 646) .	CLI option: <code>--timezone</code> RDS API parameter: <code>Timezone</code>	SQL Server
Virtual Private Cloud (VPC)	This setting depends on the platform that you are on. If you are a new customer to AWS, choose the default VPC shown. If you are creating a DB instance on the earlier E2-Classic platform that doesn't use a VPC, choose Not in VPC . For more information, see Amazon Virtual Private Cloud VPCs and Amazon RDS (p. 1718) .	For the CLI and API, you specify the VPC security group IDs.	All

Console setting	Setting description	CLI option and RDS API parameter	Supported DB engines
VPC security group	<p>If you are a new customer to AWS, Create new to create a new VPC security group. Otherwise, Choose existing, then choose from security groups that you previously created.</p> <p>When you choose Create new in the RDS console, a new security group is created. This new security group has an inbound rule that allows access to the DB instance from the IP address detected in your browser.</p> <p>For more information, see Working with DB security groups (EC2-Classic platform) (p. 1704).</p>	<p>CLI option: <code>--vpc-security-group-ids</code></p> <p>RDS API parameter: <code>VpcSecurityGroupIds</code></p>	All

Original console example

You can create a DB instance with the original AWS Management Console. This example uses Microsoft SQL Server.

To launch a SQL Server DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region in which you want to create the DB instance.
3. In the navigation pane, choose **Databases**.
If the navigation pane is closed, choose the menu icon at the top left to open it.
4. Choose **Create database** to open the **Select engine** page.
5. Choose the **Microsoft SQL Server** icon.

Select engine

Engine options

Amazon Aurora
Amazon Aurora

MySQL


MariaDB


PostgreSQL


Oracle
ORACLE

Microsoft SQL Server


Microsoft SQL Server

Edition

SQL Server Express Edition
Affordable database management system that supports database sizes up to 10 GiB.

SQL Server Web Edition
In accordance with Microsoft's licensing policies, it can only be used to support public and Internet-accessible webpages, websites, web applications, and web services.

SQL Server Standard Edition
Core data management and business intelligence capabilities for mission-critical applications and mixed workloads.

SQL Server Enterprise Edition
Comprehensive high-end capabilities for mission-critical applications with demanding database workloads and business intelligence requirements.

Aurora global database feature is now available.
This feature is now available in our new database creation flow.

[Try it now](#)

Only enable options eligible for RDS Free Usage Tier [Info](#)

[Cancel](#) **Next**

6. Choose the SQL Server DB engine edition that you want to use. The SQL Server editions that are available vary by AWS Region.
7. For some editions, the **Use Case** step asks if you are planning to use the DB instance you are creating for production. If you are, choose **Production**. If you choose **Production**, the following are all preselected in a later step:
 - Multi-AZ failover option

- **Provisioned IOPS** storage option
- **Enable deletion protection** option

We recommend these features for any production environment.

8. Choose **Next** to continue. The **Specify DB Details** page appears.

On the **Specify DB Details** page, specify your DB instance information. For information about each setting, see [Settings for DB instances \(p. 145\)](#).

Specify DB details

Instance specifications

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#) 

DB engine

Microsoft SQL Server Express Edition

License model [Info](#)

license-included

DB engine version [Info](#)

SQL Server 2017 14.00.3035.2.v1



Free tier

The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GiB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions [here](#).

Only enable options eligible for RDS Free Usage Tier [Info](#)

DB instance class [Info](#)

db.t2.small — 1 vCPU, 2 GiB RAM

Time zone (optional)

No preference

Storage type [Info](#)

Standard storage (General Purpose)

9. Choose **Next** to continue. The **Configure Advanced Settings** page appears.

On the **Configure Advanced Settings** page, provide additional information that Amazon RDS needs to launch the DB instance. For information about each setting, see [Settings for DB instances \(p. 145\)](#).

Configure advanced settings

Network & Security

Virtual Private Cloud (VPC) [Info](#)

VPC defines the virtual networking environment for this DB instance.

[C](#)

Only VPCs with a corresponding DB subnet group are listed.

Subnet group [Info](#)

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

[C](#)

Public accessibility [Info](#)

Yes

EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

No

DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

Availability zone [Info](#)

[C](#)

VPC security groups

Security groups have rules authorizing connections from all the EC2 instances and devices that need to access the DB instance.

Create new VPC security group

Choose existing VPC security groups

10. Choose **Launch DB Instance**.

11. On the final page of the wizard, choose **Close**.

On the RDS console, the new DB instance appears in the list of DB instances. The DB instance has a status of **creating** until the DB instance is ready to use. When the state changes to **available**, you can connect to the DB instance. Depending on the DB instance class and the amount of storage, it can take up to 20 minutes before the new instance is available.

The screenshot shows the 'Databases' section of the Amazon RDS console. At the top left, there's a breadcrumb navigation: 'RDS > Databases'. Below it is a title 'Databases' and a search bar labeled 'Filter databases'. A table lists five databases:

	DB Name	Role	Engine	Region
○	mymariadb	Instance	MariaDB	us-east-1
○	myoracledb	Instance	Oracle Enterprise Edition	us-east-1
○	mypostgresql	Instance	PostgreSQL	us-east-1
○	mysqldb	Instance	SQL Server Express Edition	us-east-1
○	testauroramysql-cl	Regional	Aurora MySQL	us-east-1

Connecting to an Amazon RDS DB instance

Before you can connect to a DB instance, you must create the DB instance. For information, see [Creating an Amazon RDS DB instance \(p. 141\)](#). After Amazon RDS provisions your DB instance, you can use any standard client application or utility for your DB engine to connect to the DB instance. In the connection string, you specify the DNS address from the DB instance endpoint as the host parameter, and specify the port number from the DB instance endpoint as the port parameter.

Topics

- [Finding the connection information for an Amazon RDS DB instance \(p. 162\)](#)
- [Database authentication options \(p. 165\)](#)
- [Encrypted connections \(p. 166\)](#)
- [Scenarios for accessing a DB instance in a VPC \(p. 166\)](#)
- [Connecting to a DB instance that is running a specific DB engine \(p. 166\)](#)
- [Managing connections with RDS Proxy \(p. 167\)](#)
- [Managing connections with Amazon RDS Proxy \(p. 167\)](#)

Finding the connection information for an Amazon RDS DB instance

The connection information for a DB instance includes its endpoint, port, and a valid database user, such as the master user. For example, for a MySQL DB instance, suppose that the endpoint value is `mydb.123456789012.us-east-1.rds.amazonaws.com`. In this case, the port value is 3306, and the database user is `admin`. Given this information, you specify the following values in a connection string:

- For host or host name or DNS name, specify `mydb.123456789012.us-east-1.rds.amazonaws.com`.
- For port, specify 3306.
- For user, specify `admin`.

The endpoint is unique for each DB instance, and the values of the port and user can vary. The following list shows the most common port for each DB engine:

- MariaDB – 3306
- Microsoft SQL Server – 1433
- MySQL – 3306
- Oracle – 1521
- PostgreSQL – 5432

To connect to a DB instance, use any client for a DB engine. For example, you might use the `mysql` utility to connect to a MariaDB or MySQL DB instance. You might use Microsoft SQL Server Management Studio to connect to a SQL Server DB instance. You might use Oracle SQL Developer to connect to an Oracle DB instance, or the `psql` command line utility to connect to a PostgreSQL DB instance.

To find the connection information for a DB instance, you can use the AWS Management Console, the AWS Command Line Interface (AWS CLI) [describe-db-instances](#) command, or the Amazon RDS API [DescribeDBInstances](#) operation to list its details.

Console

To find the connection information for a DB instance in the AWS Management Console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases** to display a list of your DB instances.
3. Choose the name of the DB instance to display its details.
4. On the **Connectivity & security** tab, copy the endpoint. Also, note the port number. You need both the endpoint and the port number to connect to the DB instance.

RDS > Databases > mydb

mydb

Summary

DB identifier	mydb	CPU	2.33%
Role	Instance	Current activity	0 Connections

Connectivity & security | Monitoring | Logs & events | Configuration

Connectivity & security

Endpoint & port	Network
Endpoint	Available
mydb. [REDACTED].us-east-1.rds.amazonaws.com	us-eas
Port	VPC
3306	vpc-65
Subnet	default
Security groups	[REDACTED]

5. If you need to find the master user name, choose the **Configuration** tab and view the **Master username** value.

AWS CLI

To find the connection information for a DB instance by using the AWS CLI, call the [describe-db-instances](#) command. In the call, query for the DB instance ID, endpoint, port, and master user name.

For Linux, macOS, or Unix:

```
aws rds describe-db-instances \
--query "[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

For Windows:

```
aws rds describe-db-instances ^
--query "[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Your output should be similar to the following.

```
[  
  [  
    "mydb",  
    "mydb.123456789012.us-east-1.rds.amazonaws.com",  
    3306,  
    "admin"  
  ],  
  [  
    "myoracledb",  
    "myoracledb.123456789012.us-east-1.rds.amazonaws.com",  
    1521,  
    "dbadmin"  
  ],  
  [  
    "mypostgresql",  
    "mypostgresql.123456789012.us-east-1.rds.amazonaws.com",  
    5432,  
    "postgresadmin"  
  ]  
]
```

RDS API

To find the connection information for a DB instance by using the Amazon RDS API, call the [DescribeDBInstances](#) operation. In the output, find the values for the endpoint address, endpoint port, and master user name.

Database authentication options

Amazon RDS supports the following ways to authenticate database users:

- **Password authentication** – Your DB instance performs all administration of user accounts. You create users and specify passwords with SQL statements. The SQL statements you can use depend on your DB engine.
- **AWS Identity and Access Management (IAM) database authentication** – You don't need to use a password when you connect to a DB instance. Instead, you use an authentication token.
- **Kerberos authentication** – You use external authentication of database users using Kerberos and Microsoft Active Directory. Kerberos is a network authentication protocol that uses tickets and symmetric-key cryptography to eliminate the need to transmit passwords over the network. Kerberos has been built into Active Directory and is designed to authenticate users to network resources, such as databases.

IAM database authentication and Kerberos authentication are available only for specific DB engines and versions.

For more information, see [Database authentication with Amazon RDS \(p. 1628\)](#).

Encrypted connections

You can use Secure Socket Layer (SSL) or Transport Layer Security (TLS) from your application to encrypt a connection to a DB instance. Each DB engine has its own process for implementing SSL/TLS. For more information, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

Scenarios for accessing a DB instance in a VPC

Using Amazon Virtual Private Cloud (Amazon VPC), you can launch AWS resources, such as Amazon RDS DB instances, into a virtual private cloud (VPC). When you use Amazon VPC, you have control over your virtual networking environment. You can choose your own IP address range, create subnets, and configure routing and access control lists.

A VPC security group controls access to DB instances inside a VPC. Each VPC security group rule enables a specific source to access a DB instance in a VPC that is associated with that VPC security group. The source can be a range of addresses (for example, 203.0.113.0/24), or another VPC security group. By specifying a VPC security group as the source, you allow incoming traffic from all instances (typically application servers) that use the source VPC security group.

Before attempting to connect to your DB instance, configure your VPC for your use case. The following are common scenarios for accessing a DB instance in a VPC:

- **A DB instance in a VPC accessed by an Amazon EC2 instance in the same VPC** – A common use of a DB instance in a VPC is to share data with an application server that is running in an EC2 instance in the same VPC. The EC2 instance might run a web server with an application that interacts with the DB instance.
- **A DB instance in a VPC accessed by an EC2 instance in a different VPC** – When your DB instance is in a different VPC from the EC2 instance that you're using to access it, you can use VPC peering to access the DB instance.
- **A DB instance in a VPC accessed by a client application through the internet** – To access a DB instance in a VPC from a client application through the internet, you configure a VPC with a single public subnet, and an internet gateway to enable communication over the internet.

To connect to a DB instance from outside of its VPC, the DB instance must be publicly accessible. Also, access must be granted using the inbound rules of the DB instance's security group, and other requirements must be met. For more information, see [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).

- **A DB instance in a VPC accessed by a private network** – If your DB instance isn't publicly accessible, you can use an AWS Site-to-Site VPN connection or an AWS Direct Connect connection to access it from a private network.
- **A DB instance in a VPC accessed by an EC2 instance not in a VPC** – You can communicate between a DB instance that is in a VPC and an EC2 instance that is not in a VPC by using ClassicLink.

For more information, see [Scenarios for accessing a DB instance in a VPC \(p. 1720\)](#).

Connecting to a DB instance that is running a specific DB engine

For information about connecting to a DB instance that is running a specific DB engine, follow the instructions for your DB engine:

- [Connecting to a DB instance running the MariaDB database engine \(p. 588\)](#)
- [Connecting to a DB instance running the Microsoft SQL Server database engine \(p. 656\)](#)
- [Connecting to a DB instance running the MySQL database engine \(p. 840\)](#)

- [Connecting to your Oracle DB instance \(p. 1001\)](#)
- [Connecting to a DB instance running the PostgreSQL database engine \(p. 1508\)](#)

Managing connections with RDS Proxy

You can also use Amazon RDS Proxy to manage connections to MySQL and PostgreSQL DB instances. RDS Proxy allows applications to pool and share database connections to improve scalability.

- [Managing connections with Amazon RDS Proxy \(p. 167\)](#)

Managing connections with Amazon RDS Proxy

By using Amazon RDS Proxy, you can allow your applications to pool and share database connections to improve their ability to scale. RDS Proxy makes applications more resilient to database failures by automatically connecting to a standby DB instance while preserving application connections. RDS Proxy also enables you to enforce AWS Identity and Access Management (IAM) authentication for databases, and securely store credentials in AWS Secrets Manager.

Note

RDS Proxy is fully compatible with MySQL and PostgreSQL. You can enable RDS Proxy for most applications with no code changes.

Using RDS Proxy, you can handle unpredictable surges in database traffic that otherwise might cause issues due to oversubscribing connections or creating new connections at a fast rate. RDS Proxy establishes a database connection pool and reuses connections in this pool without the memory and CPU overhead of opening a new database connection each time. To protect the database against oversubscription, you can control the number of database connections that are created.

RDS Proxy queues or throttles application connections that can't be served immediately from the pool of connections. Although latencies might increase, your application can continue to scale without abruptly failing or overwhelming the database. If connection requests exceed the limits you specify, RDS Proxy rejects application connections (that is, it sheds load). At the same time, it maintains predictable performance for the load that can be served with the available capacity.

You can reduce the overhead to process credentials and establish a secure connection for each new connection. RDS Proxy can handle some of that work on behalf of the database.

Topics

- [RDS Proxy concepts and terminology \(p. 167\)](#)
- [Planning for and setting up RDS Proxy \(p. 171\)](#)
- [Connecting to a database through RDS Proxy \(p. 182\)](#)
- [Managing an RDS Proxy \(p. 184\)](#)
- [Monitoring RDS Proxy using Amazon CloudWatch \(p. 192\)](#)
- [Endpoints for Amazon RDS Proxy \(p. 197\)](#)
- [Command-line examples for RDS Proxy \(p. 203\)](#)
- [Troubleshooting for RDS Proxy \(p. 205\)](#)
- [Using RDS Proxy with AWS CloudFormation \(p. 211\)](#)

RDS Proxy concepts and terminology

You can simplify connection management for your Amazon RDS DB instances and Amazon Aurora DB clusters by using RDS Proxy.

RDS Proxy handles the network traffic between the client application and the database. It does so in an active way first by understanding the database protocol. It then adjusts its behavior based on the SQL operations from your application and the result sets from the database.

RDS Proxy reduces the memory and CPU overhead for connection management on your database. The database needs less memory and CPU resources when applications open many simultaneous connections. It also doesn't require logic in your applications to close and reopen connections that stay idle for a long time. Similarly, it requires less application logic to reestablish connections in case of a database problem.

The infrastructure for RDS Proxy is highly available and deployed over multiple Availability Zones (AZs). The computation, memory, and storage for RDS Proxy are independent of your RDS DB instances and Aurora DB clusters. This separation helps lower overhead on your database servers, so that they can devote their resources to serving database workloads. The RDS Proxy compute resources are serverless, automatically scaling based on your database workload.

Topics

- [Overview of RDS Proxy concepts \(p. 168\)](#)
- [Connection pooling \(p. 169\)](#)
- [RDS Proxy security \(p. 169\)](#)
- [Failover \(p. 170\)](#)
- [Transactions \(p. 171\)](#)

Overview of RDS Proxy concepts

RDS Proxy handles the infrastructure to perform connection pooling and the other features described following. You see the proxies represented in the RDS console on the [Proxies](#) page.

Each proxy handles connections to a single RDS DB instance or Aurora DB cluster. The proxy automatically determines the current writer instance for RDS Multi-AZ DB instances and Aurora provisioned clusters. For Aurora multi-master clusters, the proxy connects to one of the writer instances and uses the other writer instances as hot standby targets.

The connections that a proxy keeps open and available for your database application to use make up the *connection pool*.

By default, RDS Proxy can reuse a connection after each transaction in your session. This transaction-level reuse is called *multiplexing*. When RDS Proxy temporarily removes a connection from the connection pool to reuse it, that operation is called *borrowing* the connection. When it's safe to do so, RDS Proxy returns that connection to the connection pool.

In some cases, RDS Proxy can't be sure that it's safe to reuse a database connection outside of the current session. In these cases, it keeps the session on the same connection until the session ends. This fallback behavior is called *pinning*.

A proxy has a default endpoint. You connect to this endpoint when you work with an RDS DB instance or Aurora DB cluster, instead of connecting to the read/write endpoint that connects directly to the instance or cluster. The special-purpose endpoints for an Aurora cluster remain available for you to use. For Aurora DB clusters, you can also create additional read/write and read-only endpoints. For more information, see [Overview of proxy endpoints \(p. 197\)](#).

For example, you can still connect to the cluster endpoint for read/write connections without connection pooling. You can still connect to the reader endpoint for load-balanced read-only connections. You can still connect to the instance endpoints for diagnosis and troubleshooting of specific DB instances within an Aurora cluster. If you are using other AWS services such as AWS Lambda to connect to RDS databases, you change their connection settings to use the proxy endpoint. For example, you specify the proxy endpoint to allow Lambda functions to access your database while taking advantage of RDS Proxy functionality.

Each proxy contains a target group. This *target group* embodies the RDS DB instance or Aurora DB cluster that the proxy can connect to. For an Aurora cluster, by default the target group is associated with all the DB instances in that cluster. That way, the proxy can connect to whichever Aurora DB instance is promoted to be the writer instance in the cluster. The RDS DB instance associated with a proxy, or the Aurora DB cluster and its instances, are called the *targets* of that proxy. For convenience, when you create a proxy through the console, RDS Proxy also creates the corresponding target group and registers the associated targets automatically.

An *engine family* is a related set of database engines that use the same DB protocol. You choose the engine family for each proxy that you create.

Connection pooling

Each proxy performs connection pooling for the writer instance of its associated RDS or Aurora database. *Connection pooling* is an optimization that reduces the overhead associated with opening and closing connections and with keeping many connections open simultaneously. This overhead includes memory needed to handle each new connection. It also involves CPU overhead to close each connection and open a new one, such as Transport Layer Security/Secure Sockets Layer (TLS/SSL) handshaking, authentication, negotiating capabilities, and so on. Connection pooling simplifies your application logic. You don't need to write application code to minimize the number of simultaneous open connections.

Each proxy also performs connection multiplexing, also known as connection reuse. With *multiplexing*, RDS Proxy performs all the operations for a transaction using one underlying database connection, then can use a different connection for the next transaction. You can open many simultaneous connections to the proxy, and the proxy keeps a smaller number of connections open to the DB instance or cluster. Doing so further minimizes the memory overhead for connections on the database server. This technique also reduces the chance of "too many connections" errors.

RDS Proxy security

RDS Proxy uses the existing RDS security mechanisms such as TLS/SSL and AWS Identity and Access Management (IAM). For general information about those security features, see [Security in Amazon RDS \(p. 1627\)](#). If you aren't familiar with how RDS and Aurora work with authentication, authorization, and other areas of security, make sure to familiarize yourself with how RDS and Aurora work with those areas first.

RDS Proxy can act as an additional layer of security between client applications and the underlying database. For example, you can connect to the proxy using TLS 1.2, even if the underlying DB instance supports only TLS 1.0 or 1.1. You can connect to the proxy using an IAM role, even if the proxy connects to the database using the native user and password authentication method. By using this technique, you can enforce strong authentication requirements for database applications without a costly migration effort for the DB instances themselves.

You store the database credentials used by RDS Proxy in AWS Secrets Manager. Each database user for the RDS DB instance or Aurora DB cluster accessed by a proxy must have a corresponding secret in Secrets Manager. You can also set up IAM authentication for users of RDS Proxy. By doing so, you can enforce IAM authentication for database access even if the databases use native password authentication. We recommend using these security features instead of embedding database credentials in your application code.

Using TLS/SSL with RDS Proxy

You can connect to RDS Proxy using the TLS/SSL protocol.

Note

RDS Proxy uses certificates from the AWS Certificate Manager (ACM). If you use RDS Proxy, when you rotate your TLS/SSL certificate you don't need to update applications that use RDS Proxy connections.

To enforce TLS for all connections between the proxy and your database, you can specify a setting **Require Transport Layer Security** when you create or modify a proxy.

RDS Proxy can also ensure that your session uses TLS/SSL between your client and the RDS Proxy endpoint. To have RDS Proxy do so, specify the requirement on the client side. SSL session variables are not set for SSL connections to a database using RDS Proxy.

- For RDS for MySQL and Aurora MySQL, specify the requirement on the client side with the `--ssl-mode` parameter when you run the `mysql` command.
- For Amazon RDS PostgreSQL and Aurora PostgreSQL, specify `sslmode=require` as part of the `conninfo` string when you run the `psql` command.

RDS Proxy supports TLS protocol version 1.0, 1.1, and 1.2. You can connect to the proxy using a higher version of TLS than you use in the underlying database.

By default, client programs establish an encrypted connection with RDS Proxy, with further control available through the `--ssl-mode` option. From the client side, RDS Proxy supports all SSL modes.

For the client, the SSL modes are the following:

PREFERRED

SSL is the first choice, but it isn't required.

DISABLED

No SSL is allowed.

REQUIRED

Enforce SSL.

VERIFY_CA

Enforce SSL and verify the certificate authority (CA).

VERIFY_IDENTITY

Enforce SSL and verify the CA and CA hostname.

Note

You can use the SSL mode `VERIFY_IDENTITY` when connecting to the default proxy endpoint. You can't use that SSL mode when you connect to proxy endpoints that you create.

When using a client with `--ssl-mode VERIFY_CA` or `VERIFY_IDENTITY`, specify the `--ssl-ca` option pointing to a CA in `.pem` format. For a `.pem` file that you can use, download the [Amazon root CA 1 trust store](#) from Amazon Trust Services.

RDS Proxy uses wildcard certificates, which apply to both a domain and its subdomains. If you use the `mysql` client to connect with SSL mode `VERIFY_IDENTITY`, currently you must use the MySQL 8.0-compatible `mysql` command.

Failover

Failover is a high-availability feature that replaces a database instance with another one when the original instance becomes unavailable. A failover might happen because of a problem with a database instance. It might also be part of normal maintenance procedures, such as during a database upgrade. Failover applies to RDS DB instances in a Multi-AZ configuration, and Aurora DB clusters with one or more reader instances in addition to the writer instance.

Connecting through a proxy makes your application more resilient to database failovers. When the original DB instance becomes unavailable, RDS Proxy connects to the standby database without dropping idle application connections. Doing so helps to speed up and simplify the failover process. The result is faster failover that's less disruptive to your application than a typical reboot or database problem.

Without RDS Proxy, a failover involves a brief outage. During the outage, you can't perform write operations on that database. Any existing database connections are disrupted and your application must reopen them. The database becomes available for new connections and write operations when a read-only DB instance is promoted to take the place of the one that's unavailable.

During DB failovers, RDS Proxy continues to accept connections at the same IP address and automatically directs connections to the new primary DB instance. Clients connecting through RDS Proxy are not susceptible to the following:

- Domain Name System (DNS) propagation delays on failover.
- Local DNS caching.
- Connection timeouts.
- Uncertainty about which DB instance is the current writer.
- Waiting for a query response from a former writer that became unavailable without closing connections.

For applications that maintain their own connection pool, going through RDS Proxy means that most connections stay alive during failovers or other disruptions. Only connections that are in the middle of a transaction or SQL statement are canceled. RDS Proxy immediately accepts new connections. When the database writer is unavailable, RDS Proxy queues up incoming requests.

For applications that don't maintain their own connection pools, RDS Proxy offers faster connection rates and more open connections. It offloads the expensive overhead of frequent reconnects from the database. It does so by reusing database connections maintained in the RDS Proxy connection pool. This approach is particularly important for TLS connections, where setup costs are significant.

Transactions

All the statements within a single transaction always use the same underlying database connection. The connection becomes available for use by a different session when the transaction ends. Using the transaction as the unit of granularity has the following consequences:

- Connection reuse can happen after each individual statement when the RDS for MySQL or Aurora MySQL `autocommit` setting is enabled.
- Conversely, when the `autocommit` setting is disabled, the first statement you issue in a session begins a new transaction. Thus, if you enter a sequence of `SELECT`, `INSERT`, `UPDATE`, and other data manipulation language (DML) statements, connection reuse doesn't happen until you issue a `COMMIT`, `ROLLBACK`, or otherwise end the transaction.
- Entering a data definition language (DDL) statement causes the transaction to end after that statement completes.

RDS Proxy detects when a transaction ends through the network protocol used by the database client application. Transaction detection doesn't rely on keywords such as `COMMIT` or `ROLLBACK` appearing in the text of the SQL statement.

In some cases, RDS Proxy might detect a database request that makes it impractical to move your session to a different connection. In these cases, it turns off multiplexing for that connection the remainder of your session. The same rule applies if RDS Proxy can't be certain that multiplexing is practical for the session. This operation is called *pinning*. For ways to detect and minimize pinning, see [Avoiding pinning \(p. 190\)](#).

Planning for and setting up RDS Proxy

In the following sections, you can find how to set up RDS Proxy. You can also find how to set the related security options that control who can access each proxy and how each proxy connects to DB instances.

Topics

- [Limits for RDS Proxy \(p. 172\)](#)
- [Identifying DB instances, clusters, and applications to use with RDS Proxy \(p. 173\)](#)
- [Setting up network prerequisites \(p. 174\)](#)
- [Setting up database credentials in AWS Secrets Manager \(p. 175\)](#)
- [Setting up AWS Identity and Access Management \(IAM\) policies \(p. 176\)](#)
- [Creating an RDS Proxy \(p. 178\)](#)
- [Viewing an RDS Proxy \(p. 181\)](#)

Limits for RDS Proxy

The following limitations apply to RDS Proxy:

- RDS Proxy is available only in certain AWS Regions only. For more information, see [Amazon RDS Proxy](#).
You can have up to 20 proxies for each AWS account ID. If your application requires more proxies, you can request additional proxies by opening a ticket with the AWS Support organization.
- Each proxy can have up to 200 associated Secrets Manager secrets. Thus, each proxy can connect to up to 200 different user accounts at any given time.
- You can create, view, modify, and delete up to 20 endpoints for each proxy. These endpoints are in addition to the default endpoint that's automatically created for each proxy.
- In an Aurora cluster, all of the connections using the default proxy endpoint are handled by the Aurora writer instance. To perform load balancing for read-intensive workloads, you can create a read-only endpoint for a proxy. That endpoint passes connections to the reader endpoint of the cluster. That way, your proxy connections can take advantage of Aurora read scalability. For more information, see [Overview of proxy endpoints \(p. 197\)](#).

For RDS DB instances in replication configurations, you can associate a proxy only with the writer DB instance, not a read replica.

- You can't use RDS Proxy with Aurora Serverless clusters.
- You can't use RDS Proxy with Aurora clusters that are part of an Aurora global database.
- Your RDS Proxy must be in the same VPC as the database. The proxy can't be publicly accessible, although the database can be.

Note

For Aurora DB clusters, you can enable cross-VPC access by creating an additional endpoint for a proxy and specifying a different VPC, subnets, and security groups with that endpoint. For more information, see [Accessing Aurora and RDS databases across VPCs \(p. 198\)](#).

- You can't use RDS Proxy with a VPC that has dedicated tenancy.
- If you use RDS Proxy with an RDS DB instance or Aurora DB cluster that has IAM authentication enabled, make sure that all users who connect through a proxy authenticate through user names and passwords. See [Setting up AWS Identity and Access Management \(IAM\) policies \(p. 176\)](#) for details about IAM support in RDS Proxy.
- You can't use RDS Proxy with custom DNS.
- RDS Proxy is available for the MySQL and PostgreSQL engine families.
- Each proxy can be associated with a single target DB instance or cluster. However, you can associate multiple proxies with the same DB instance or cluster.

The following RDS Proxy prerequisites and limitations apply to MySQL:

- For RDS for MySQL, RDS Proxy supports MySQL 5.6 and 5.7. For Aurora MySQL, RDS Proxy supports version 1 (compatible with MySQL 5.6) and version 2 (compatible with MySQL 5.7).

- Currently, all proxies listen on port 3306 for MySQL. The proxies still connect to your database using the port that you specified in the database settings.
- You can't use RDS Proxy with RDS for MySQL 8.0.
- You can't use RDS Proxy with self-managed MySQL databases in EC2 instances.
- Proxies don't support MySQL compressed mode. For example, they don't support the compression used by the `--compress` or `-C` options of the `mysql` command.
- Some SQL statements and functions can change the connection state without causing pinning. For the most current pinning behavior, see [Avoiding pinning \(p. 190\)](#).

The following RDS Proxy prerequisites and limitations apply to PostgreSQL:

- For RDS PostgreSQL, RDS Proxy supports version 10.10 and higher minor versions, and version 11.5 and higher minor versions. For Aurora PostgreSQL, RDS Proxy supports version 10.11 and higher minor versions, and 11.6 and higher minor versions.
- Currently, all proxies listen on port 5432 for PostgreSQL.
- Query cancellation isn't supported for PostgreSQL.
- The results of the PostgreSQL function `lastval` aren't always accurate. As a work-around, use the `INSERT` statement with the `RETURNING` clause.

Identifying DB instances, clusters, and applications to use with RDS Proxy

You can determine which of your DB instances, clusters, and applications might benefit the most from using RDS Proxy. To do so, consider these factors:

- RDS Proxy is highly available and deployed over multiple Availability Zones (AZs). To ensure overall high availability for your database, deploy your Amazon RDS DB instance or Aurora cluster in a Multi-AZ configuration.
- Any DB instance or cluster that encounters "too many connections" errors is a good candidate for associating with a proxy. The proxy enables applications to open many client connections, while the proxy manages a smaller number of long-lived connections to the DB instance or cluster.
- For DB instances or clusters that use smaller AWS instance classes, such as T2 or T3, using a proxy can help avoid out-of-memory conditions. It can also help reduce the CPU overhead for establishing connections. These conditions can occur when dealing with large numbers of connections.
- You can monitor certain Amazon CloudWatch metrics to determine whether a DB instance or cluster is approaching certain types of limit. These limits are for the number of connections and the memory associated with connection management. You can also monitor certain CloudWatch metrics to determine whether a DB instance or cluster is handling many short-lived connections. Opening and closing such connections can impose performance overhead on your database. For information about the metrics to monitor, see [Monitoring RDS Proxy using Amazon CloudWatch \(p. 192\)](#).
- AWS Lambda functions can also be good candidates for using a proxy. These functions make frequent short database connections that benefit from connection pooling offered by RDS Proxy. You can take advantage of any IAM authentication you already have for Lambda functions, instead of managing database credentials in your Lambda application code.
- Applications that use languages and frameworks such as PHP and Ruby on Rails are typically good candidates for using a proxy. Such applications typically open and close large numbers of database connections, and don't have built-in connection pooling mechanisms.
- Applications that keep a large number of connections open for long periods are typically good candidates for using a proxy. Applications in industries such as software as a service (SaaS) or ecommerce often minimize the latency for database requests by leaving connections open. With RDS Proxy, an application can keep more connections open than it can when connecting directly to the DB instance or cluster.

- You might not have adopted IAM authentication and Secrets Manager due to the complexity of setting up such authentication for all DB instances and clusters. If so, you can leave the existing authentication methods in place and delegate the authentication to a proxy. The proxy can enforce the authentication policies for client connections for particular applications. You can take advantage of any IAM authentication you already have for Lambda functions, instead of managing database credentials in your Lambda application code.

Setting up network prerequisites

Using RDS Proxy requires you to have a set of networking resources in place. These include a virtual private cloud (VPC), two or more subnets, an Amazon EC2 instance within the same VPC, and an internet gateway. If you've successfully connected to any RDS DB instances or Aurora DB clusters, you already have the required network resources.

The following Linux example shows AWS CLI commands that examine the VPCs and subnets owned by your AWS account. In particular, you pass subnet IDs as parameters when you create a proxy using the CLI.

```
aws ec2 describe-vpcs
aws ec2 describe-internet-gateways
aws ec2 describe-subnets --query '*[].[VpcId,SubnetId]' --output text | sort
```

The following Linux example shows AWS CLI commands to determine the subnet IDs corresponding to a specific Aurora DB cluster or RDS DB instance. For an Aurora cluster, first you find the ID for one of the associated DB instances. You can extract the subnet IDs used by that DB instance by examining the nested fields within the `DBSubnetGroup` and `Subnets` attributes in the describe output for the DB instance. You specify some or all of those subnet IDs when setting up a proxy for that database server.

```
$ # Optional first step, only needed if you're starting from an Aurora cluster. Find the ID of any DB instance in the cluster.
$ aws rds describe-db-clusters --db-cluster-id my_cluster_id --query '*[].[DBClusterMembers|[0]|[0][*].DBInstanceIdentifier' --output text
my_instance_id
instance_id_2
instance_id_3
...

$ # From the DB instance, trace through the DBSubnetGroup and Subnets to find the subnet IDs.
$ aws rds describe-db-instances --db-instance-id my_instance_id --query '*[].[DBSubnetGroup|[0]|[0][[Subnets]|[0][*].SubnetIdentifier' --output text
subnet_id_1
subnet_id_2
subnet_id_3
...
```

As an alternative, you can first find the VPC ID for the DB instance. Then you can examine the VPC to find its subnets. The following Linux example shows how.

```
$ # From the DB instance, find the VPC.
$ aws rds describe-db-instances --db-instance-id my_instance_id --query '*[].[VpcId]' --output text
my_vpc_id

$ aws ec2 describe-subnets --filters Name=vpc-id,Values=my_vpc_id --query '*[].[SubnetId]' --output text
subnet_id_1
subnet_id_2
subnet_id_3
```

```
subnet_id_4
subnet_id_5
subnet_id_6
```

Setting up database credentials in AWS Secrets Manager

For each proxy that you create, you first use the Secrets Manager service to store sets of user name and password credentials. You create a separate Secrets Manager secret for each database user account that the proxy connects to on the RDS DB instance or Aurora DB cluster.

In Secrets Manager, you create these secrets with values for the `username` and `password` fields. Doing so allows the proxy to connect to the corresponding database users on whichever RDS DB instances or Aurora DB clusters that you associate with the proxy. To do this, you can use the setting **Credentials for other database**, **Credentials for RDS database**, or **Other type of secrets**. Fill in the appropriate values for the **User name** and **Password** fields, and placeholder values for any other required fields. The proxy ignores other fields such as **Host** and **Port** if they're present in the secret. Those details are automatically supplied by the proxy.

You can also choose **Other type of secrets**. In this case, you create the secret with keys named `username` and `password`.

Because the secrets used by your proxy aren't tied to a specific database server, you can reuse a secret across multiple proxies if you use the same credentials across multiple database servers. For example, you might use the same credentials across a group of development and test servers.

To connect through the proxy as a specific user, make sure that the password associated with a secret matches the database password for that user. If there's a mismatch, you can update the associated secret in Secrets Manager. In this case, you can still connect to other accounts where the secret credentials and the database passwords do match.

When you create a proxy through the AWS CLI or RDS API, you specify the Amazon Resource Names (ARNs) of the corresponding secrets for all the DB user accounts that the proxy can access. In the AWS Management Console, you choose the secrets by their descriptive names.

For instructions about creating secrets in Secrets Manager, see the [Creating a secret](#) page in the Secrets Manager documentation. Use one of the following techniques:

- Use [Secrets Manager](#) in the console.
- To use the CLI to create a Secrets Manager secret for use with RDS Proxy, use a command such as the following.

```
aws secretsmanager create-secret \
--name "secret_name"
--description "secret_description"
--region region_name
--secret-string '{"username":"db_user","password":"db_user_password"}'
```

For example, the following commands create Secrets Manager secrets for two database users, one named `admin` and the other named `app-user`.

```
aws secretsmanager create-secret \
--name admin_secret_name --description "db admin user" \
--secret-string '{"username":"admin","password":"choose_your_own_password"}'

aws secretsmanager create-secret \
--name proxy_secret_name --description "application user" \
--secret-string '{"username":"app-user","password":"choose_your_own_password"}'
```

To see the secrets owned by your AWS account, use a command such as the following.

```
aws secretsmanager list-secrets
```

When you create a proxy using the CLI, you pass the Amazon resource names (ARNs) of one or more secrets to the `--auth` parameter. The following Linux example shows how to prepare a report with only the name and ARN of each secret owned by your AWS account. This example uses the `--output table` parameter that is available in AWS CLI version 2. If you are using AWS CLI version 1, use `--output text` instead.

```
aws secretsmanager list-secrets --query '*[].[Name,ARN]' --output table
```

To verify that you stored the correct credentials and in the right format in a secret, use a command such as the following. Substitute the short name or the ARN of the secret for `your_secret_name`.

```
aws secretsmanager get-secret-value --secret-id your_secret_name
```

The output should include a line displaying a JSON-encoded value like the following.

```
"SecretString": "{\"username\":\"your_username\", \"password\":\"your_password\"}"}
```

Setting up AWS Identity and Access Management (IAM) policies

After you create the secrets in Secrets Manager, you create an IAM policy that can access those secrets. For general information about using IAM with RDS and Aurora, see [Identity and access management in Amazon RDS \(p. 1644\)](#).

Tip

The following procedure applies if you use the IAM console. If you use the AWS Management Console for RDS, RDS can create the IAM policy for you automatically. In that case, you can skip the following procedure.

To create an IAM policy that accesses your Secrets Manager secrets for use with your proxy

1. Sign in to the IAM console. Follow the **Create role** process, as described in [Creating IAM roles](#). Include the **Add Role to Database** step.
 2. For the new role, perform the **Add inline policy** step. Use the same general procedures as in [Editing IAM policies](#). Paste the following JSON into the JSON text box. Substitute your own account ID. Substitute your AWS Region for us-east-2. Substitute the Amazon Resource Names (ARNs) for the secrets that you created. For the kms : Decrypt action, substitute the ARN of the default AWS KMS customer master key (CMK) or your own AWS KMS CMK, depending on which one you used to encrypt the Secrets Manager secrets.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "VisualEditor0",  
            "Effect": "Allow",  
            "Action": "secretsmanager:GetSecretValue",  
            "Resource": [  
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_1",  
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_2"  
            ]  
        },  
        {  
            "Sid": "VisualEditor1",  
            "Effect": "Allow",  
            "Action": "kms:Decrypt",  
            "Resource": "arn:aws:kms:us-east-2:account_id:key/key_id".  
    ]  
}
```

```
        "Condition": {
            "StringEquals": {
                "kms:ViaService": "secretsmanager.us-east-2.amazonaws.com"
            }
        }
    ]
}
```

3. Edit the trust policy for this IAM role. Paste the following JSON into the JSON text box.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

The following commands perform the same operation through the AWS CLI.

```
PREFIX=choose_an_identifier

aws iam create-role --role-name choose_role_name \
--assume-role-policy-document '{"Version":"2012-10-17","Statement":\
[{"Effect":"Allow","Principal": {"Service":\
["rds.amazonaws.com"]}, "Action": "sts:AssumeRole"}]}'"

aws iam put-role-policy --role-name same_role_name_as_previous \
--policy-name $PREFIX-secret-reader-policy --policy-document """
same_json_as_in_previous_example
"""

aws kms create-key --description "$PREFIX-test-key" --policy """
{
    "Id": "$PREFIX-kms-policy",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {"AWS": "arn:aws:iam::account_id:root"},
            "Action": "kms:*", "Resource": "*"
        },
        {
            "Sid": "Allow access for Key Administrators",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "$USER_ARN", "arn:aws:iam::account_id:role/Admin"
                ],
                "Action": [
                    "kms>Create*",
                    "kms:Describe*",

```

```
"kms:Enable*",
"kms>List*",
"kms:Put*",
"kms:Update*",
"kms:Revoke*",
"kms:Disable*",
"kms:Get*",
"kms>Delete*",
"kms:TagResource",
"kms:UntagResource",
"kms:ScheduleKeyDeletion",
"kms:CancelKeyDeletion"
],
"Resource": "*"
},
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": { "AWS": "$ROLE_ARN" },
  "Action": [ "kms:Decrypt", "kms:DescribeKey" ],
  "Resource": "*"
}
]
"""
}
```

Creating an RDS Proxy

To manage connections for a specified set of DB instances, you can create a proxy. You can associate a proxy with an RDS for MySQL DB instance, PostgreSQL DB instance, or an Aurora DB cluster.

AWS Management Console

To create a proxy

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Proxies**.
3. Choose **Create proxy**.
4. Choose all the settings for your proxy.

For **Proxy configuration**, provide information for the following:

- **Proxy identifier.** Specify a name of your choosing, unique within your AWS account ID and current AWS Region.
- **Engine compatibility.** Choose either **MySQL** or **POSTGRESQL**.
- **Require Transport Layer Security.** Choose this setting if you want the proxy to enforce TLS/SSL for all client connections. When you use an encrypted or unencrypted connection to a proxy, the proxy uses the same encryption setting when it makes a connection to the underlying database.
- **Idle client connection timeout.** Choose a time period that a client connection can be idle before the proxy can close it. The default is 1,800 seconds (30 minutes). A client connection is considered idle when the application doesn't submit a new request within the specified time after the previous request completed. The underlying database connection stays open and is returned to the connection pool. Thus, it's available to be reused for new client connections.

Consider lowering the idle client connection timeout if you want the proxy to proactively remove stale connections. If your workload is spiking, consider raising the idle client connection timeout to save the cost of establishing connections.

For **Target group configuration**, provide information for the following:

- **Database.** Choose one RDS DB instance or Aurora DB cluster to access through this proxy. The list only includes DB instances and clusters with compatible database engines, engine versions, and other settings. If the list is empty, create a new DB instance or cluster that's compatible with RDS Proxy. To do so, follow the procedure in [Creating an Amazon RDS DB instance \(p. 141\)](#). Then try creating the proxy again.
- **Connection pool maximum connections.** Specify a value from 1 through 100. This setting represents the percentage of the `max_connections` value that RDS Proxy can use for its connections. If you only intend to use one proxy with this DB instance or cluster, you can set this value to 100. For details about how RDS Proxy uses this setting, see [Controlling connection limits and timeouts \(p. 189\)](#).
- **Session pinning filters.** (Optional) This is an advanced setting, for troubleshooting performance issues with particular applications. Currently, the only choice is `EXCLUDE_VARIABLE_SETS`. Choose a filter only if both of following are true: Your application isn't reusing connections due to certain kinds of SQL statements, and you can verify that reusing connections with those SQL statements doesn't affect application correctness. For more information, see [Avoiding pinning \(p. 190\)](#).
- **Connection borrow timeout.** In some cases, you might expect the proxy to sometimes use all available database connections. In such cases, you can specify how long the proxy waits for a database connection to become available before returning a timeout error. You can specify a period up to a maximum of five minutes. This setting only applies when the proxy has the maximum number of connections open and all connections are already in use.

For **Connectivity**, provide information for the following:

- **Secrets Manager ARNs.** Choose at least one Secrets Manager secret that contains DB user credentials for the RDS DB instance or Aurora DB cluster that you intend to access with this proxy.
- **IAM role.** Choose an IAM role that has permission to access the Secrets Manager secrets that you chose earlier. You can also choose for the AWS Management Console to create a new IAM role for you and use that.
- **IAM Authentication.** Choose whether to require or disallow IAM authentication for connections to your proxy. The choice of IAM authentication or native database authentication applies to all DB users that access this proxy.
- **Subnets.** This field is prepopulated with all the subnets associated with your VPC. You can remove any subnets that you don't need for this proxy. You must leave at least two subnets.

Provide additional connectivity configuration:

- **VPC security group.** Choose an existing VPC security group. You can also choose for the AWS Management Console to create a new security group for you and use that.

Note

This security group must allow access to the database the proxy connects to. The same security group is used for ingress from your applications to the proxy, and for egress from the proxy to the database. For example, suppose that you use the same security group for your database and your proxy. In this case, make sure that you specify that resources in that security group can communicate with other resources in the same security group.

(Optional) Provide advanced configuration:

- **Enable enhanced logging.** You can enable this setting to troubleshoot proxy compatibility or performance issues.

When this setting is enabled, RDS Proxy includes detailed information about SQL statements in its logs. This information helps you to debug issues involving SQL behavior or the performance and scalability of the proxy connections. The debug information includes the text of SQL statements that you submit through the proxy. Thus, only enable this setting when needed for debugging, and only when you have security measures in place to safeguard any sensitive information that appears in the logs.

To minimize overhead associated with your proxy, RDS Proxy automatically turns this setting off 24 hours after you enable it. Enable it temporarily to troubleshoot a specific issue.

5. Choose **Create Proxy**.

AWS CLI

To create a proxy, use the AWS CLI command [create-db-proxy](#). The --engine-family value is case-sensitive.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-proxy \
--db-proxy-name proxy_name \
--engine-family { MYSQL | POSTGRESOL } \
--auth ProxyAuthenticationConfig_JSON_string \
--role-arn iam_role \
--vpc-subnet-ids space_separated_list \
[--vpc-security-group-ids space_separated_list] \
[--require-tls | --no-require-tls] \
[--idle-client-timeout value] \
[--debug-logging | --no-debug-logging] \
[--tags comma_separated_list]
```

For Windows:

```
aws rds create-db-proxy ^
--db-proxy-name proxy_name ^
--engine-family { MYSQL | POSTGRESOL } ^
--auth ProxyAuthenticationConfig_JSON_string ^
--role-arn iam_role ^
--vpc-subnet-ids space_separated_list ^
[--vpc-security-group-ids space_separated_list] ^
[--require-tls | --no-require-tls] ^
[--idle-client-timeout value] ^
[--debug-logging | --no-debug-logging] ^
[--tags comma_separated_list]
```

Tip

If you don't already know the subnet IDs to use for the --vpc-subnet-ids parameter, see [Setting up network prerequisites \(p. 174\)](#) for examples of how to find the subnet IDs that you can use.

To create the required information and associations for the proxy, you also use the [register-db-proxy-targets](#) command. Specify the target group name default. RDS Proxy automatically creates a target group with this name when you create each proxy.

```
aws rds register-db-proxy-targets
--db-proxy-name value
```

```
[--target-group-name target_group_name]
[--db-instance-identifiers space_separated_list] # rds db instances, or
[--db-cluster-identifiers cluster_id]           # rds db cluster (all instances), or
[--db-cluster-endpoint endpoint_name]          # rds db cluster endpoint (all
instances)
```

RDS API

To create an RDS proxy, call the Amazon RDS API operation [CreateDBProxy](#). You pass a parameter with the [AuthConfig](#) data structure.

RDS Proxy automatically creates a target group named `default` when you create each proxy. You associate an RDS DB instance or Aurora DB cluster with the target group by calling the function [RegisterDBProxyTargets](#).

Viewing an RDS Proxy

After you create one or more RDS proxies, you can view them all to examine their configuration details and choose which ones to modify, delete, and so on.

Any database applications that use the proxy require the proxy endpoint to use in the connection string.

AWS Management Console

To view your proxy

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the AWS Management Console, choose the AWS Region in which you created the RDS Proxy.
3. In the navigation pane, choose **Proxies**.
4. Choose the name of an RDS proxy to display its details.
5. On the details page, the **Target groups** section shows how the proxy is associated with a specific RDS DB instance or Aurora DB cluster. You can follow the link to the **default** target group page to see more details about the association between the proxy and the database. This page is where you see settings that you specified when creating the proxy, such as maximum connection percentage, connection borrow timeout, engine compatibility, and session pinning filters.

CLI

To view your proxy using the CLI, use the [describe-db-proxies](#) command. By default, it displays all proxies owned by your AWS account. To see details for a single proxy, specify its name with the `--db-proxy-name` parameter.

```
aws rds describe-db-proxies [--db-proxy-name proxy_name]
```

To view the other information associated with the proxy, use the following commands.

```
aws rds describe-db-proxy-target-groups --db-proxy-name proxy_name
aws rds describe-db-proxy-targets --db-proxy-name proxy_name
```

Use the following sequence of commands to see more detail about the things that are associated with the proxy:

1. To get a list of proxies, run [describe-db-proxies](#).

2. To show connection parameters such as the maximum percentage of connections that the proxy can use, run `describe-db-proxy-target-groups` --db-proxy-name and use the name of the proxy as the parameter value.
3. To see the details of the RDS DB instance or Aurora DB cluster associated with the returned target group, run `describe-db-proxy-targets`.

RDS API

To view your proxies using the RDS API, use the `DescribeDBProxies` operation. It returns values of the `DBProxy` data type.

To see details of the connection settings for the proxy, use the proxy identifiers from this return value with the `DescribeDBProxyTargetGroups` operation. It returns values of the `DBProxyTargetGroup` data type.

To see the RDS instance or Aurora DB cluster associated with the proxy, use the `DescribeDBProxyTargets` operation. It returns values of the `DBProxyTarget` data type.

Connecting to a database through RDS Proxy

You connect to an RDS DB instance or Aurora DB cluster through a proxy in generally the same way as you connect directly to the database. The main difference is that you specify the proxy endpoint instead of the instance or cluster endpoint. For an Aurora DB cluster, by default all proxy connections have read/write capability and use the writer instance. If you normally use the reader endpoint for read-only connections, you can create an additional read-only endpoint for the proxy and use that endpoint the same way. For more information, see [Overview of proxy endpoints \(p. 197\)](#).

Topics

- [Connecting to a proxy using native authentication \(p. 182\)](#)
- [Connecting to a proxy using IAM authentication \(p. 183\)](#)
- [Considerations for connecting to a proxy with PostgreSQL \(p. 183\)](#)

Connecting to a proxy using native authentication

Use the following basic steps to connect to a proxy using native authentication:

1. Find the proxy endpoint. In the AWS Management Console, you can find the endpoint on the details page for the corresponding proxy. With the AWS CLI, you can use the `describe-db-proxies` command. The following example shows how.

```
# Add --output text to get output as a simple tab-separated list.
$ aws rds describe-db-proxies --query '*[*].{DBProxyName:DBProxyName,Endpoint:Endpoint}'
[
    [
        {
            "Endpoint": "the-proxy.proxy-demo.us-east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy"
        },
        {
            "Endpoint": "the-proxy-other-secret.proxy-demo.us-east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy-other-secret"
        },
        {
            "Endpoint": "the-proxy-rds-secret.proxy-demo.us-east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy-rds-secret"
        }
    ]
]
```

```
        "Endpoint": "the-proxy-t3.proxy-demo.us-east-1.rds.amazonaws.com",
        "DBProxyName": "the-proxy-t3"
    }
]
```

2. Specify that endpoint as the host parameter in the connection string for your client application. For example, specify the proxy endpoint as the value for the `mysql -h` option or `psql -h` option.
3. Supply the same database user name and password as you usually do.

Connecting to a proxy using IAM authentication

When you use IAM authentication with RDS Proxy, set up your database users to authenticate with regular user names and passwords. The IAM authentication applies to RDS Proxy retrieving the user name and password credentials from Secrets Manager. The connection from RDS Proxy to the underlying database doesn't go through IAM.

To connect to RDS Proxy using IAM authentication, follow the same general procedure as for connecting to an RDS DB instance or Aurora cluster using IAM authentication. For general information about using IAM with RDS and Aurora, see [Security in Amazon RDS \(p. 1627\)](#).

The major differences in IAM usage for RDS Proxy include the following:

- You don't configure each individual database user with an authorization plugin. The database users still have regular user names and passwords within the database. You set up Secrets Manager secrets containing these user names and passwords, and authorize RDS Proxy to retrieve the credentials from Secrets Manager.

Important

The IAM authentication applies to the connection between your client program and the proxy. The proxy then authenticates to the database using the user name and password credentials retrieved from Secrets Manager. When you use IAM for the connection to a proxy, make sure that the underlying RDS DB instance or Aurora DB cluster doesn't have IAM enabled.

- Instead of the instance, cluster, or reader endpoint, you specify the proxy endpoint. For details about the proxy endpoint, see [Connecting to your DB instance using IAM authentication \(p. 1668\)](#).
- In the direct DB IAM auth case, you selectively pick database users and configure them to be identified with a special auth plugin. You can then connect to those users using IAM auth.

In the proxy use case, you need to provide the proxy with Secrets that contain some user's username and password (native auth). You then connect to the proxy using IAM auth (by generating an auth token with the proxy endpoint, not the database endpoint) and using a username which matches one of the usernames for the secrets you previously provided.

- Make sure that you use Transport Layer Security (TLS) / Secure Sockets Layer (SSL) when connecting to a proxy using IAM authentication.

You can grant a specific user access to the proxy by modifying the IAM policy. An example follows.

```
"Resource": "arn:aws:rds-db:us-east-2:1234567890:dbuser:prx-ABCDEFGHIJKLM01234/db_user"
```

Considerations for connecting to a proxy with PostgreSQL

For PostgreSQL, when a client starts a connection to a PostgreSQL database, it sends a startup message that includes pairs of parameter name and value strings. For details, see the `StartupMessage` in [PostgreSQL message formats](#) in the PostgreSQL documentation.

When connecting through an RDS proxy, the startup message can include the following currently recognized parameters:

- `user`
- `database`
- `replication`

The startup message can also include the following additional runtime parameters:

- `application_name`
- `client_encoding`
- `DateStyle`
- `TimeZone`
- `extra_float_digits`

For more information about PostgreSQL messaging, see the [Frontend/Backend protocol](#) in the PostgreSQL documentation.

For PostgreSQL, if you use JDBC we recommend the following to avoid pinning:

- Set the JDBC connection parameter `assumeMinServerVersion` to at least `9.0` to avoid pinning. Doing this prevents the JDBC driver from performing an extra round trip during connection startup when it runs `SET extra_float_digits = 3`.
- Set the JDBC connection parameter `ApplicationName` to `any/your-application-name` to avoid pinning. Doing this prevents the JDBC driver from performing an extra round trip during connection startup when it runs `SET application_name = "PostgreSQL JDBC Driver"`. Note the JDBC parameter is `ApplicationName` but the PostgreSQL `StartupMessage` parameter is `application_name`.
- Set the JDBC connection parameter `preferQueryMode` to `extendedForPrepared` to avoid pinning. The `extendedForPrepared` ensures that the extended mode is used only for prepared statements.

The default for the `preferQueryMode` parameter is `extended`, which uses the extended mode for all queries. The extended mode uses a series of `Prepare`, `Bind`, `Execute`, and `Sync` requests and corresponding responses. This type of series causes connection pinning in an RDS proxy.

For more information, see [Avoiding pinning \(p. 190\)](#). For more information about connecting using JDBC, see [Connecting to the database](#) in the PostgreSQL documentation.

Managing an RDS Proxy

Following, you can find an explanation of how to manage RDS proxy operation and configuration. These procedures help your application make the most efficient use of database connections and achieve maximum connection reuse. The more that you can take advantage of connection reuse, the more CPU and memory overhead that you can save. This in turn reduces latency for your application and enables the database to devote more of its resources to processing application requests.

Topics

- [Modifying an RDS Proxy \(p. 185\)](#)
- [Adding a new database user \(p. 189\)](#)
- [Changing the password for a database user \(p. 189\)](#)
- [Controlling connection limits and timeouts \(p. 189\)](#)
- [Managing and monitoring connection pooling \(p. 189\)](#)
- [Avoiding pinning \(p. 190\)](#)
- [Deleting an RDS Proxy \(p. 192\)](#)

Modifying an RDS Proxy

You can change certain settings associated with a proxy after you create the proxy. You do so by modifying the proxy itself, its associated target group, or both. Each proxy has an associated target group.

AWS Management Console

To modify the settings for a proxy

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Proxies**.
3. In the list of proxies, choose the proxy whose settings you want to modify or go to its details page.
4. For **Actions**, choose **Modify**.
5. Enter or choose the properties to modify. You can do the following:
 - Rename the proxy by entering a new identifier.
 - Turn the requirement for Transport layer Security (TLS) on or off.
 - Enter a time period for the idle connection timeout.
 - Add or remove Secrets Manager secrets. These secrets correspond to database user names and passwords.
 - Change the IAM role used to retrieve the secrets from Secrets Manager.
 - Require or disallow IAM authentication for connections to the proxy.
 - Add or remove VPC subnets for the proxy to use.
 - Add or remove VPC security groups for the proxy to use.
 - Enable or disable enhanced logging.
6. Choose **Modify**.

If you didn't find the settings listed that you want to change, use the following procedure to update the target group for the proxy. The *target group* associated with a proxy controls the settings related to the physical database connections. Each proxy has one associated target group named **default**, which is created automatically along with the proxy.

You can only modify the target group from the proxy details page, not from the list on the **Proxies** page.

To modify the settings for a proxy target group

1. On the **Proxies** page, go to the details page for a proxy.
2. For **Target groups**, choose the **default** link. Currently, all proxies have a single target group named **default**.
3. On the details page for the **default** target group, choose **Modify**.
4. Choose new settings for the properties that you can modify:
 - Choose a different RDS DB instance or Aurora cluster.
 - Adjust what percentage of the maximum available connections the proxy can use.
 - Choose a session pinning filter. Doing this can help reduce performance issues due to insufficient transaction-level reuse for connections. Using this setting requires understanding of application behavior and the circumstances under which RDS Proxy pins a session to a database connection.
 - Adjust the connection borrow timeout interval. This setting applies when the maximum number of connections is already being used for the proxy. The setting determines how long the proxy waits for a connection to become available before returning a timeout error.

You can't change certain properties, such as the target group identifier and the database engine.

5. Choose **Modify target group**.

AWS CLI

To modify a proxy using the AWS CLI, use the commands [modify-db-proxy](#), [modify-db-proxy-target-group](#), [deregister-db-proxy-targets](#), and [register-db-proxy-targets](#).

With the `modify-db-proxy` command, you can change properties such as the following:

- The set of Secrets Manager secrets used by the proxy.
- Whether TLS is required.
- The idle client timeout.
- Whether to log additional information from SQL statements for debugging.
- The IAM role used to retrieve Secrets Manager secrets.
- The security groups used by the proxy.

The following example shows how to rename an existing proxy.

```
aws rds modify-db-proxy --db-proxy-name the-proxy --new-db-proxy-name the_new_name
```

To modify connection-related settings or rename the target group, use the `modify-db-proxy-target-group` command. Currently, all proxies have a single target group named `default`. When working with this target group, you specify the name of the proxy and `default` for the name of the target group.

The following example shows how to first check the `MaxIdleConnectionsPercent` setting for a proxy and then change it, using the target group.

```
aws rds describe-db-proxy-target-groups --db-proxy-name the-proxy
{
    "TargetGroups": [
        {
            "Status": "available",
            "UpdatedDate": "2019-11-30T16:49:30.342Z",
            "ConnectionPoolConfig": {
                "MaxIdleConnectionsPercent": 50,
                "ConnectionBorrowTimeout": 120,
                "MaxConnectionsPercent": 100,
                "SessionPinningFilters": []
            },
            "TargetGroupName": "default",
            "CreatedDate": "2019-11-30T16:49:27.940Z",
            "DBProxyName": "the-proxy",
            "IsDefault": true
        }
    ]
}

aws rds modify-db-proxy-target-group --db-proxy-name the-proxy --target-group-name default
--connection-pool-config '
{ "MaxIdleConnectionsPercent": 75 }'

{
```

```

    "DBProxyTargetGroup": {
        "Status": "available",
        "UpdatedDate": "2019-12-02T04:09:50.420Z",
        "ConnectionPoolConfig": {
            "MaxIdleConnectionsPercent": 75,
            "ConnectionBorrowTimeout": 120,
            "MaxConnectionsPercent": 100,
            "SessionPinningFilters": []
        },
        "TargetGroupName": "default",
        "CreatedDate": "2019-11-30T16:49:27.940Z",
        "DBProxyName": "the-proxy",
        "IsDefault": true
    }
}

```

With the `deregister-db-proxy-targets` and `register-db-proxy-targets` commands, you change which RDS DB instance or Aurora DB cluster the proxy is associated with through its target group. Currently, each proxy can connect to one RDS DB instance or Aurora DB cluster. The target group tracks the connection details for all the RDS DB instances in a Multi-AZ configuration, or all the DB instances in an Aurora cluster.

The following example starts with a proxy that is associated with an Aurora MySQL cluster named `cluster-56-2020-02-25-1399`. The example shows how to change the proxy so that it can connect to a different cluster named `provisioned-cluster`.

When you work with an RDS DB instance, you specify the `--db-instance-identifier` option. When you work with an Aurora DB cluster, you specify the `--db-cluster-identifier` option instead.

The following example modifies an Aurora MySQL proxy. An Aurora PostgreSQL proxy has port 5432.

```

aws rds describe-db-proxy-targets --db-proxy-name the-proxy

{
    "Targets": [
        {
            "Endpoint": "instance-9814.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-9814"
        },
        {
            "Endpoint": "instance-8898.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-8898"
        },
        {
            "Endpoint": "instance-1018.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-1018"
        },
        {
            "Type": "TRACKED_CLUSTER",
            "Port": 0,
            "RdsResourceId": "cluster-56-2020-02-25-1399"
        },
        {
            "Endpoint": "instance-4330.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-4330"
        }
    ]
}

```

```

        }
    }

aws rds deregister-db-proxy-targets --db-proxy-name the-proxy --db-cluster-identifier
cluster-56-2020-02-25-1399

aws rds describe-db-proxy-targets --db-proxy-name the-proxy

{
    "Targets": []
}

aws rds register-db-proxy-targets --db-proxy-name the-proxy --db-cluster-identifier
provisioned-cluster

{
    "DBProxyTargets": [
        {
            "Type": "TRACKED_CLUSTER",
            "Port": 0,
            "RdsResourceId": "provisioned-cluster"
        },
        {
            "Endpoint": "gkldje.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "gkldje"
        },
        {
            "Endpoint": "provisioned-1.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "provisioned-1"
        }
    ]
}

```

RDS API

To modify a proxy using the RDS API, you use the operations [ModifyDBProxy](#), [ModifyDBProxyTargetGroup](#), [DeregisterDBProxyTargets](#), and [RegisterDBProxyTargets](#) operations.

With [ModifyDBProxy](#), you can change properties such as the following:

- The set of Secrets Manager secrets used by the proxy.
- Whether TLS is required.
- The idle client timeout.
- Whether to log additional information from SQL statements for debugging.
- The IAM role used to retrieve Secrets Manager secrets.
- The security groups used by the proxy.

With [ModifyDBProxyTargetGroup](#), you can modify connection-related settings or rename the target group. Currently, all proxies have a single target group named `default`. When working with this target group, you specify the name of the proxy and `default` for the name of the target group.

With [DeregisterDBProxyTargets](#) and [RegisterDBProxyTargets](#), you change which RDS DB instance or Aurora DB cluster the proxy is associated with through its target group. Currently, each proxy can connect to one RDS DB instance or Aurora DB cluster. The target group tracks the connection details for all the RDS DB instances in a Multi-AZ configuration, or all the DB instances in an Aurora cluster.

Adding a new database user

In some cases, you might add a new database user to an RDS DB instance or Aurora cluster that's associated with a proxy. If so, add or repurpose a Secrets Manager secret to store the credentials for that user. To do this, choose one of the following options:

- Create a new Secrets Manager secret, using the procedure described in [Setting up database credentials in AWS Secrets Manager \(p. 175\)](#).
- Update the IAM role to give RDS Proxy access to the new Secrets Manager secret. To do so, update the resources section of the IAM role policy.
- If the new user takes the place of an existing one, update the credentials stored in the proxy's Secrets Manager secret for the existing user.

Changing the password for a database user

In some cases, you might change the password for a database user in an RDS DB instance or Aurora cluster that's associated with a proxy. If so, update the corresponding Secrets Manager secret with the new password.

Controlling connection limits and timeouts

RDS Proxy uses the `max_connections` setting for your RDS DB instance or Aurora DB cluster. This setting represents the overall upper limit on the connections that the proxy can open at any one time. In Aurora clusters and RDS Multi-AZ configurations, the `max_connections` value that the proxy uses is the one for the Aurora primary instance or the RDS writer instance.

To set this value for your RDS DB instance or Aurora DB cluster, follow the procedures in [Working with DB parameter groups \(p. 228\)](#). These procedures demonstrate how to associate a parameter group with your database and edit the `max_connections` value in the parameter group.

The proxy setting for maximum connections represents a percentage of the `max_connections` value for the database that's associated with the proxy. If you have multiple applications all using the same database, you can effectively divide their connection quotas by using a proxy for each application with a specific percentage of `max_connections`. If you do so, ensure that the percentages add up to 100 or less for all proxies associated with the same database.

RDS Proxy periodically disconnects idle connections and returns them to the connection pool. You can adjust this timeout interval. Doing so helps your applications to deal with stale resources, especially if the application mistakenly leaves a connection open while holding important database resources.

Managing and monitoring connection pooling

As described in [Connection pooling \(p. 169\)](#), connection pooling is a crucial RDS Proxy feature. Following, you can learn how to make the most efficient use of connection pooling and transaction-level connection reuse (multiplexing).

Because the connection pool is managed by RDS Proxy, you can monitor it and adjust connection limits and timeout intervals without changing your application code.

For each proxy, you can specify an upper limit on the number of connections used by the connection pool. You specify the limit as a percentage. This percentage applies to the maximum connections configured in the database. The exact number varies depending on the DB instance size and configuration settings.

For example, suppose that you configured RDS Proxy to use 75 percent of the maximum connections for the database. For MySQL, the maximum value is defined by the `max_connections` configuration parameter. In this case, the other 25 percent of maximum connections remain available to assign to other proxies or for connections that don't go through a proxy. In some cases, the proxy might keep

less than 75 percent of the maximum connections open at a particular time. Those cases might include situations where the database doesn't have many simultaneous connections, or some connections stay idle for long periods.

The overall number of connections available for the connection pool changes as you update the `max_connections` configuration setting that applies to an RDS DB instance or an Aurora cluster.

The proxy doesn't reserve all of these connections in advance. Thus, you can specify a relatively large percentage, and those connections are only opened when the proxy becomes busy enough to need them.

You can choose how long to wait for a connection to become available for use by your application. This setting is represented by the **Connection borrow timeout** option when you create a proxy. This setting specifies how long to wait for a connection to become available in the connection pool before returning a timeout error. It applies when the number of connections is at the maximum, and so no connections are available in the connection pool. It also applies if no writer instance is available because a failover operation is in process. Using this setting, you can set the best wait period for your application without having to change the query timeout in your application code.

Avoiding pinning

Multiplexing is more efficient when database requests don't rely on state information from previous requests. In that case, RDS Proxy can reuse a connection at the conclusion of each transaction. Examples of such state information include most variables and configuration parameters that you can change through `SET` or `SELECT` statements. SQL transactions on a client connection can multiplex between underlying database connections by default.

Your connections to the proxy can enter a state known as *pinning*. When a connection is pinned, each later transaction uses the same underlying database connection until the session ends. Other client connections also can't reuse that database connection until the session ends. The session ends when the client connection is dropped.

RDS Proxy automatically pins a client connection to a specific DB connection when it detects a session state change that isn't appropriate for other sessions. Pinning reduces the effectiveness of connection reuse. If all or almost all of your connections experience pinning, consider modifying your application code or workload to reduce the conditions that cause the pinning.

For example, if your application changes a session variable or configuration parameter, later statements can rely on the new variable or parameter to be in effect. Thus, when RDS Proxy processes requests to change session variables or configuration settings, it pins that session to the DB connection. That way, the session state remains in effect for all later transactions in the same session.

This rule doesn't apply to all parameters you can set. RDS Proxy tracks changes to the character set, collation, time zone, autocommit, current database, SQL mode, and `session_track_schema` settings. Thus RDS Proxy doesn't pin the session when you modify these. In this case, RDS Proxy only reuses the connection for other sessions that have the same values for those settings.

Performance tuning for RDS Proxy involves trying to maximize transaction-level connection reuse (multiplexing) by minimizing pinning. You can do so by doing the following:

- Avoid unnecessary database requests that might cause pinning.
- Set variables and configuration settings consistently across all connections. That way, later sessions are more likely to reuse connections that have those particular settings.

However, for PostgreSQL setting a variable leads to session pinning.

- Apply a session pinning filter to the proxy. You can exempt certain kinds of operations from pinning the session if you know that doing so doesn't affect the correct operation of your application.
- See how frequently pinning occurs by monitoring the CloudWatch metric `DatabaseConnectionsCurrentlySessionPinned`. For information about this and other CloudWatch metrics, see [Monitoring RDS Proxy using Amazon CloudWatch \(p. 192\)](#).

- If you use `SET` statements to perform identical initialization for each client connection, you can do so while preserving transaction-level multiplexing. In this case, you move the statements that set up the initial session state into the initialization query used by a proxy. This property is a string containing one or more SQL statements, separated by semicolons.

For example, you can define an initialization query for a proxy that sets certain configuration parameters. Then, RDS Proxy applies those settings whenever it sets up a new connection for that proxy. You can remove the corresponding `SET` statements from your application code, so that they don't interfere with transaction-level multiplexing.

Important

For proxies associated with MySQL databases, don't set the configuration parameter `sql_auto_is_null` to `true` or a nonzero value in the initialization query. Doing so might cause incorrect application behavior.

The proxy pins the session to the current connection in the following situations where multiplexing might cause unexpected behavior:

- Any statement with a text size greater than 16 KB causes the proxy to pin the session.
- Prepared statements cause the proxy to pin the session. This rule applies whether the prepared statement uses SQL text or the binary protocol.
- Explicit MySQL statements `LOCK TABLE`, `LOCK TABLES`, or `FLUSH TABLES WITH READ LOCK` cause the proxy to pin the session.
- Setting a user variable or a system variable (with some exceptions) causes the proxy to pin the session. If this situation reduces your connection reuse too much, you can choose for `SET` operations not to cause pinning. For information about how to do so by setting the `SessionPinningFilters` property, see [Creating an RDS Proxy \(p. 178\)](#).
- Creating a temporary table causes the proxy to pin the session. That way, the contents of the temporary table are preserved throughout the session regardless of transaction boundaries.
- Calling the MySQL functions `ROW_COUNT`, `FOUND_ROWS`, and `LAST_INSERT_ID` sometimes causes pinning.

The exact circumstances where these functions cause pinning might differ between Aurora MySQL versions that are compatible with MySQL 5.6 and MySQL 5.7.

Calling MySQL stored procedures and stored functions doesn't cause pinning. RDS Proxy doesn't detect any session state changes resulting from such calls. Therefore, make sure that your application doesn't change session state inside stored routines and rely on that session state to persist across transactions. For example, if a stored procedure creates a temporary table that is intended to persist across transactions, that application currently isn't compatible with RDS Proxy.

For PostgreSQL, the following interactions cause pinning:

- Using `SET` commands
- Using the extended query protocol such as by using JDBC default settings
- Creating temporary sequences, tables, or views
- Declaring cursors
- Discarding the session state
- Listening on a notification channel
- Loading a library module such as `auto_explain`
- Manipulating sequences using functions such as `nextval` and `setval`
- Interacting with locks using functions such as `pg_advisory_lock` and `pg_try_advisory_lock`
- Using prepared statements, setting parameters, or resetting a parameter to its default

If you have expert knowledge about your application behavior, you can skip the pinning behavior for certain application statements. To do so, choose the **Session pinning filters** option when creating the proxy. Currently, you can opt out of session pinning for setting session variables and configuration settings.

For metrics about how often pinning occurs for a proxy, see [Monitoring RDS Proxy using Amazon CloudWatch \(p. 192\)](#).

Deleting an RDS Proxy

You can delete a proxy if you no longer need it. You might delete a proxy because the application that was using it is no longer relevant. Or you might delete a proxy if you take the DB instance or cluster associated with it out of service.

AWS Management Console

To delete a proxy

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Proxies**.
3. Choose the proxy to delete from the list.
4. Choose **Delete Proxy**.

AWS CLI

To delete a DB proxy, use the AWS CLI command `delete-db-proxy`. To remove related associations, also use the `deregister-db-proxy-targets` command.

```
aws rds delete-db-proxy --name proxy_name
```

```
aws rds deregister-db-proxy-targets
  --db-proxy-name proxy_name
  [--target-group-name target_group_name]
  [--target-ids comma_separated_list]      # or
  [--db-instance-identifiers instance_id]      # or
  [--db-cluster-identifiers cluster_id]
```

RDS API

To delete a DB proxy, call the Amazon RDS API function `DeleteDBProxy`. To delete related items and associations, you also call the functions `DeleteDBProxyTargetGroup` and `DeregisterDBProxyTargets`.

Monitoring RDS Proxy using Amazon CloudWatch

You can monitor RDS Proxy by using Amazon CloudWatch. CloudWatch collects and processes raw data from the proxies into readable, near-real-time metrics. To find these metrics in the CloudWatch console, choose **Metrics**, then choose **RDS**, and choose **Per-Proxy Metrics**. For more information, see [Using Amazon CloudWatch metrics](#) in the Amazon CloudWatch User Guide.

Note

RDS publishes these metrics for each underlying Amazon EC2 instance associated with a proxy. A single proxy might be served by more than one EC2 instance. Use CloudWatch statistics to aggregate the values for a proxy across all the associated instances.

Some of these metrics might not be visible until after the first successful connection by a proxy.

In the RDS Proxy logs, each entry is prefixed with the name of the associated proxy endpoint. This name can be the name you specified for a user-defined endpoint, or the special name `default` for read/write requests using the default endpoint of a proxy.

All RDS Proxy metrics are in the group `proxy`.

Each proxy endpoint has its own CloudWatch metrics. You can monitor the usage of each proxy endpoint independently. For more information about proxy endpoints, see [Endpoints for Amazon RDS Proxy \(p. 197\)](#).

You can aggregate the values for each metric using one of the following dimension sets. For example, by using the `ProxyName` dimension set, you can analyze all the traffic for a particular proxy. By using the other dimension sets, you can split the metrics in different ways. You can split the metrics based on the different endpoints or target databases of each proxy, or the read/write and read-only traffic to each database.

- Dimension set 1: `ProxyName`
- Dimension set 2: `ProxyName, EndpointName`
- Dimension set 3: `ProxyName, TargetGroup, Target`
- Dimension set 4: `ProxyName, TargetGroup, TargetRole`

Metric	Description	Valid period	CloudWatch dimension set
<code>AvailabilityPercentage</code>	The percentage of time for which the target group was available in the role indicated by the dimension. This metric is reported every minute. The most useful statistic for this metric is <code>Average</code> .	1 minute	Dimension set 4 (p. 193)
<code>ClientConnections</code>	The current number of client connections. This metric is reported every minute. The most useful statistic for this metric is <code>Sum</code> .	1 minute	Dimension set 1 (p. 193), Dimension set 2 (p. 193)
<code>ClientConnectionsClosed</code>	The number of client connections closed. The most useful statistic for this metric is <code>Sum</code> .	1 minute and above	Dimension set 1 (p. 193), Dimension set 2 (p. 193)
<code>ClientConnectionsNotEncrypted</code>	The current number of client connections without Transport Layer Security (TLS). This metric is reported every minute. The most useful statistic for this metric is <code>Sum</code> .	1 minute and above	Dimension set 1 (p. 193), Dimension set 2 (p. 193)

Metric	Description	Valid period	CloudWatch dimension set
ClientConnectionsReceived	The number of client connection requests received. The most useful statistic for this metric is Sum.	1 minute and above	Dimension set 1 (p. 193) , Dimension set 2 (p. 193)
ClientConnectionsSetFailed	The number of client connection attempts that failed due to misconfigured authentication or TLS. The most useful statistic for this metric is Sum.	1 minute and above	Dimension set 1 (p. 193) , Dimension set 2 (p. 193)
ClientConnectionsSetEstablished	The number of client connections successfully established with any authentication mechanism with or without TLS. The most useful statistic for this metric is Sum.	1 minute and above	Dimension set 1 (p. 193) , Dimension set 2 (p. 193)
ClientConnectionsTLS	The current number of client connections with TLS. This metric is reported every minute. The most useful statistic for this metric is Sum.	1 minute and above	Dimension set 1 (p. 193) , Dimension set 2 (p. 193)
DatabaseConnectionRequests	The number of requests to create a database connection. The most useful statistic for this metric is Sum.	1 minute and above	Dimension set 1 (p. 193) , Dimension set 3 (p. 193) , Dimension set 4 (p. 193)
DatabaseConnectionRequestsTLS	The number of requests to create a database connection with TLS. The most useful statistic for this metric is Sum.	1 minute and above	Dimension set 1 (p. 193) , Dimension set 3 (p. 193) , Dimension set 4 (p. 193)
DatabaseConnections	The current number of database connections. This metric is reported every minute. The most useful statistic for this metric is Sum.	1 minute	Dimension set 1 (p. 193) , Dimension set 3 (p. 193) , Dimension set 4 (p. 193)

Metric	Description	Valid period	CloudWatch dimension set
DatabaseConnectionsTimeToBorrow	The time in microseconds that it takes for the proxy being monitored to get a database connection. The most useful statistic for this metric is Average.	1 minute and above	Dimension set 1 (p. 193) , Dimension set 2 (p. 193)
DatabaseConnectionsCurrentBorrow	The current number of database connections in the borrow state. This metric is reported every minute. The most useful statistic for this metric is Sum.	1 minute	Dimension set 1 (p. 193) , Dimension set 3 (p. 193) , Dimension set 4 (p. 193)
DatabaseConnectionsCurrentInTransaction	The current number of database connections in a transaction. This metric is reported every minute. The most useful statistic for this metric is Sum.	1 minute	Dimension set 1 (p. 193) , Dimension set 3 (p. 193) , Dimension set 4 (p. 193)
DatabaseConnectionsCurrentPinned	The current session count of database connections currently pinned because of operations in client requests that change session state. This metric is reported every minute. The most useful statistic for this metric is Sum.	1 minute	Dimension set 1 (p. 193) , Dimension set 3 (p. 193) , Dimension set 4 (p. 193)
DatabaseConnectionsFailed	The number of database connection requests that failed. The most useful statistic for this metric is Sum.	1 minute and above	Dimension set 1 (p. 193) , Dimension set 3 (p. 193) , Dimension set 4 (p. 193)
DatabaseConnectionsSuccessful	The number of database connections successfully established with or without TLS. The most useful statistic for this metric is Sum.	1 minute and above	Dimension set 1 (p. 193) , Dimension set 3 (p. 193) , Dimension set 4 (p. 193)

Metric	Description	Valid period	CloudWatch dimension set
DatabaseConnectionsTLS	The current number of database connections with TLS. This metric is reported every minute. The most useful statistic for this metric is Sum.	1 minute	Dimension set 1 (p. 193), Dimension set 3 (p. 193), Dimension set 4 (p. 193)
MaxDatabaseConnections	The maximum number of database connections allowed. This metric is reported every minute. The most useful statistic for this metric is Sum.	1 minute	Dimension set 1 (p. 193), Dimension set 3 (p. 193), Dimension set 4 (p. 193)
QueryDatabaseResponseTime	The time in microseconds that the database took to respond to the query. The most useful statistic for this metric is Average.	1 minute and above	Dimension set 1 (p. 193), Dimension set 2 (p. 193), Dimension set 3 (p. 193), Dimension set 4 (p. 193)
QueryRequests	The number of queries received. A query including multiple statements is counted as one query. The most useful statistic for this metric is Sum.	1 minute and above	Dimension set 1 (p. 193), Dimension set 2 (p. 193)
QueryRequestsNoTLS	The number of queries received from non-TLS connections. A query including multiple statements is counted as one query. The most useful statistic for this metric is Sum.	1 minute and above	Dimension set 1 (p. 193), Dimension set 2 (p. 193)
QueryRequestsTLS	The number of queries received from TLS connections. A query including multiple statements is counted as one query. The most useful statistic for this metric is Sum.	1 minute and above	Dimension set 1 (p. 193), Dimension set 2 (p. 193)

Metric	Description	Valid period	CloudWatch dimension set
QueryResponseLatency	The time in microseconds between getting a query request and the proxy responding to it. The most useful statistic for this metric is Average.	1 minute and above	Dimension set 1 (p. 193) , Dimension set 2 (p. 193)

Endpoints for Amazon RDS Proxy

Following, you can learn about endpoints for RDS Proxy and how to use them. By using endpoints, you can take advantage of the following capabilities:

- You can use multiple endpoints with a proxy to monitor and troubleshoot connections from different applications independently.
- You can use reader endpoints with Aurora DB clusters to improve read scalability and high availability for your query-intensive applications.
- You can use a cross-VPC endpoint to allow access to databases in one VPC from resources such as Amazon EC2 instances in a different VPC.

Topics

- [Overview of proxy endpoints \(p. 197\)](#)
- [Reader endpoints \(p. 198\)](#)
- [Accessing Aurora and RDS databases across VPCs \(p. 198\)](#)
- [Creating a proxy endpoint \(p. 199\)](#)
- [Viewing proxy endpoints \(p. 201\)](#)
- [Modifying a proxy endpoint \(p. 201\)](#)
- [Deleting a proxy endpoint \(p. 202\)](#)
- [Limits for proxy endpoints \(p. 203\)](#)

Overview of proxy endpoints

Working with RDS Proxy endpoints involves the same kinds of procedures as with Aurora DB cluster and reader endpoints and RDS instance endpoints. If you aren't familiar with RDS endpoints, find more information in [Connecting to a DB instance running the MySQL database engine](#) and [Connecting to a DB instance running the PostgreSQL database engine](#).

By default, the endpoint that you connect to when you use RDS Proxy with an Aurora cluster has read/write capability. As a consequence, this endpoint sends all requests to the writer instance of the cluster, and all of those connections count against the `max_connections` value for the writer instance. If your proxy is associated with an Aurora DB cluster, you can create additional read/write or read-only endpoints for that proxy.

You can use a read-only endpoint with your proxy for read-only queries, the same way that you use the reader endpoint for an Aurora provisioned cluster. Doing so helps you to take advantage of the read scalability of an Aurora cluster with one or more reader DB instances. You can run more simultaneous queries and make more simultaneous connections by using a read-only endpoint and adding more reader DB instances to your Aurora cluster as needed.

For a proxy endpoint that you create, you can also associate the endpoint with a different virtual private cloud (VPC) than the proxy itself uses. By doing so, you can connect to the proxy from a different VPC, for example a VPC used by a different application within your organization. Both VPCs must be owned by the same AWS account.

For information about limits associated with proxy endpoints, see [Limits for proxy endpoints \(p. 203\)](#).

In the RDS Proxy logs, each entry is prefixed with the name of the associated proxy endpoint. This name can be the name you specified for a user-defined endpoint, or the special name `default` for read/write requests using the default endpoint of a proxy.

Each proxy endpoint has its own set of CloudWatch metrics. You can monitor the metrics for all endpoints of a proxy. You can also monitor metrics for a specific endpoint, or for all the read/write or read-only endpoints of a proxy. For more information, see [Monitoring RDS Proxy using Amazon CloudWatch \(p. 192\)](#).

A proxy endpoint uses the same authentication mechanism as its associated proxy. RDS Proxy automatically sets up permissions and authorizations for the user-defined endpoint, consistent with the properties of the associated proxy.

Reader endpoints

With RDS Proxy, you can create and use reader endpoints. However, these endpoints only work for proxies associated with Aurora DB clusters. You might see references to reader endpoints in the AWS Management Console. If you use the RDS CLI or API, you might see the `TargetRole` attribute with a value of `READ_ONLY`. You can take advantage of these features by changing the target of a proxy from an RDS DB instance to an Aurora DB cluster. To learn about reader endpoints, see [Managing connections with Amazon RDS Proxy in the Aurora User Guide](#).

Accessing Aurora and RDS databases across VPCs

By default, the components of your RDS and Aurora technology stack are all in the same Amazon VPC. For example, suppose that an application running on an Amazon EC2 instance connects to an Amazon RDS DB instance or an Aurora DB cluster. In this case, the application server and database must both be within the same VPC.

With RDS Proxy, you can set up access to an Aurora cluster or RDS instance in one VPC from resources such as EC2 instances in another VPC. For example, your organization might have multiple applications that access the same database resources. Each application might be in its own VPC. To use cross-VPC capability with RDS Proxy, all the VPCs must be owned by the same AWS account.

To enable cross-VPC access, you create a new endpoint for the proxy. If you aren't familiar with creating proxy endpoints, see [Endpoints for Amazon RDS Proxy \(p. 197\)](#) for details. The proxy itself resides in the same VPC as the Aurora DB cluster or RDS instance. However, the cross-VPC endpoint resides in the other VPC, along with the other resources such as the EC2 instances. The cross-VPC endpoint is associated with subnets and security groups from the same VPC as the EC2 and other resources. These associations let you connect to the endpoint from the applications that otherwise can't access the database due to the VPC restrictions.

The following steps explain how to create and access a cross-VPC endpoint through RDS Proxy:

1. Create two VPCs, or choose two VPCs that you already use for Aurora and RDS work. Each VPC should have its own associated network resources such as an Internet gateway, route tables, subnets, and security groups. If you only have one VPC, you can consult [Getting started with Amazon RDS \(p. 197\)](#) for the steps to set up another VPC to use RDS successfully. You can also examine your existing VPC in the Amazon EC2 console to see what kinds of resources to connect together.
2. Create a DB proxy associated with the Aurora DB cluster or RDS instance that you want to connect to. Follow the procedure in [Creating an RDS Proxy \(p. 178\)](#).

3. On the **Details** page for your proxy in the RDS console, under the **Proxy endpoints** section, choose **Create endpoint**. Follow the procedure in [Creating a proxy endpoint \(p. 199\)](#).
 4. Choose whether to make the cross-VPC endpoint read/write or read-only.
 5. Instead of accepting the default of the same VPC as the Aurora DB cluster or RDS instance, choose a different VPC. This VPC must be in the same AWS Region as the VPC where the proxy resides.
 6. Now instead of accepting the defaults for subnets and security groups from the same VPC as the Aurora DB cluster or RDS instance, make new selections. Make these based on the subnets and security groups from the VPC that you chose.
 7. You don't need to change any of the settings for the Secrets Manager secrets. The same credentials work for all endpoints for your proxy, regardless of which VPC each endpoint is in.
 8. Wait for the new endpoint to reach the **Available** state.
 9. Make a note of the full endpoint name. This is the value ending in **Region_name.rds.amazonaws.com** that you supply as part of the connection string for your database application.
- 10 Access the new endpoint from a resource in the same VPC as the endpoint. A simple way to test this process is to create a new EC2 instance in this VPC. Then you can log into the EC2 instance and run the `mysql` or `psql` commands to connect by using the endpoint value in your connection string.

Creating a proxy endpoint

Console

To create a proxy endpoint

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Proxies**.
3. Click the name of the proxy that you want to create a new endpoint for.

The details page for that proxy appears.

4. In the **Proxy endpoints** section, choose **Create proxy endpoint**.

The **Create proxy endpoint** window appears.

5. For **Proxy endpoint name**, enter a descriptive name of your choice.
6. For **Target role**, choose whether to make the endpoint read/write or read-only.

Connections that use a read/write endpoint can perform any kind of operation: data definition language (DDL) statements, data manipulation language (DML) statements, and queries. These endpoints always connect to the primary instance of the Aurora cluster. You can use read/write endpoints for general database operations when you only use a single endpoint in your application. You can also use read/write endpoints for administrative operations, online transaction processing (OLTP) applications, and extract-transform-load (ETL) jobs.

Connections that use a read-only endpoint can only perform queries. When there are multiple reader instances in the Aurora cluster, RDS Proxy can use a different reader instance for each connection to the endpoint. That way, a query-intensive application can take advantage of Aurora's clustering capability. You can add more query capacity to the cluster by adding more reader DB instances. These read-only connections don't impose any overhead on the primary instance of the cluster. That way, your reporting and analysis queries don't slow down the write operations of your OLTP applications.

7. For **Virtual Private Cloud (VPC)**, choose the default if you plan to access the endpoint from the same EC2 instances or other resources where you normally access the proxy or its associated database. If you want to set up cross-VPC access for this proxy, choose a VPC other than the default.

For more information about cross-VPC access, see [Accessing Aurora and RDS databases across VPCs \(p. 198\)](#).

8. For **Subnets**, RDS Proxy fills in the same subnets as the associated proxy by default. If you want to restrict access to the endpoint so that only a portion of the address range of the VPC can connect to it, remove one or more subnets from the set of choices.
9. For **VPC security group**, you can choose an existing security group or create a new one. RDS Proxy fills in the same security group or groups as the associated proxy by default. If the inbound and outbound rules for the proxy are appropriate for this endpoint, you can leave the default choice.

If you choose to create a new security group, specify a name for the security group on this page. Then edit the security group settings from the EC2 console afterward.

10. Choose **Create proxy endpoint**.

AWS CLI

To create a proxy endpoint, use the AWS CLI `create-db-proxy-endpoint` command.

Include the following required parameters:

- `--db-proxy-name value`
- `--db-proxy-endpoint-name value`
- `--vpc-subnet-ids list_of_ids`. Separate the subnet IDs with spaces. You don't specify the ID of the VPC itself.

You can also include the following optional parameters:

- `--target-role { READ_WRITE | READ_ONLY }`. This parameter defaults to `READ_WRITE`. The `READ_ONLY` value only has an effect on Aurora provisioned clusters that contain one or more reader DB instances. When the proxy is associated with an RDS instance or with an Aurora cluster that only contains a writer DB instance, you can't specify `READ_ONLY`.
- `--vpc-security-group-ids value`. Separate the security group IDs with spaces. If you omit this parameter, RDS Proxy uses the default security group for the VPC. RDS Proxy determines the VPC based on the subnet IDs that you specify for the `--vpc-subnet-ids` parameter.

Example

The following example creates a proxy endpoint named `my-endpoint`.

For Linux, macOS, or Unix:

```
aws rds create-db-proxy-endpoint \
--db-proxy-name my-proxy \
--db-proxy-endpoint-name my-endpoint \
--vpc-subnet-ids subnet_id subnet_id subnet_id ... \
--target-role READ_ONLY \
--vpc-security-group-ids security_group_id ]
```

For Windows:

```
aws rds create-db-proxy-endpoint ^
--db-proxy-name my-proxy ^
--db-proxy-endpoint-name my-endpoint ^
--vpc-subnet-ids subnet_id_1 subnet_id_2 subnet_id_3 ... ^
--target-role READ_ONLY ^
--vpc-security-group-ids security_group_id
```

RDS API

To create a proxy endpoint, use the RDS API [CreateProxyEndpoint](#) action.

Viewing proxy endpoints

Console

To view the details for a proxy endpoint

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Proxies**.
3. In the list, choose the proxy whose endpoint you want to view. Click the proxy name to view its details page.
4. In the **Proxy endpoints** section, choose the endpoint that you want to view. Click its name to view the details page.
5. Examine the parameters whose values you're interested in. You can check properties such as the following:
 - Whether the endpoint is read/write or read-only.
 - The endpoint address that you use in a database connection string.
 - The VPC, subnets, and security groups associated with the endpoint.

AWS CLI

To view one or more DB proxy endpoints, use the AWS CLI [describe-db-proxy-endpoints](#) command.

You can include the following optional parameters:

- `--db-proxy-endpoint-name`
- `--db-proxy-name`

The following example describes the `my-endpoint` proxy endpoint.

Example

For Linux, macOS, or Unix:

```
aws rds describe-db-proxy-endpoints \
--db-proxy-endpoint-name my-endpoint
```

For Windows:

```
aws rds describe-db-proxy-endpoints ^
--db-proxy-endpoint-name my-endpoint
```

RDS API

To describe one or more proxy endpoints, use the RDS API [DescribeDBProxyEndpoints](#) operation.

Modifying a proxy endpoint

Console

To modify one or more proxy endpoints

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Proxies**.
3. In the list, choose the proxy whose endpoint you want to modify. Click the proxy name to view its details page.
4. In the **Proxy endpoints** section, choose the endpoint that you want to modify. You can select it in the list, or click its name to view the details page.
5. On the proxy details page, under the **Proxy endpoints** section, choose **Edit**. Or on the proxy endpoint details page, for **Actions**, choose **Edit**.
6. Change the values of the parameters that you want to modify.
7. Choose **Save changes**.

AWS CLI

To modify a DB proxy endpoint, use the AWS CLI `modify-db-proxy-endpoint` command with the following required parameters:

- `--db-proxy-endpoint-name`

Specify changes to the endpoint properties by using one or more of the following parameters:

- `--new-db-proxy-endpoint-name`
- `--vpc-security-group-ids`. Separate the security group IDs with spaces.

The following example renames the `my-endpoint` proxy endpoint to `new-endpoint-name`.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-proxy-endpoint \
--db-proxy-endpoint-name my-endpoint \
--new-db-proxy-endpoint-name new-endpoint-name
```

For Windows:

```
aws rds modify-db-proxy-endpoint ^
--db-proxy-endpoint-name my-endpoint ^
--new-db-proxy-endpoint-name new-endpoint-name
```

RDS API

To modify a proxy endpoint, use the RDS API [ModifyDBProxyEndpoint](#) operation.

Deleting a proxy endpoint

You can delete an endpoint for your proxy using the console as described following.

Note

You can't delete the default endpoint that RDS Proxy automatically creates for each proxy. When you delete a proxy, RDS Proxy automatically deletes all the associated endpoints.

Console

To delete a proxy endpoint using the AWS Management Console

1. In the navigation pane, choose **Proxies**.
2. In the list, choose the proxy whose endpoint you want to endpoint. Click the proxy name to view its details page.
3. In the **Proxy endpoints** section, choose the endpoint that you want to delete. You can select one or more endpoints in the list, or click the name of a single endpoint to view the details page.
4. On the proxy details page, under the **Proxy endpoints** section, choose **Delete**. Or on the proxy endpoint details page, for **Actions**, choose **Delete**.

AWS CLI

To delete a proxy endpoint, run the `delete-db-proxy-endpoint` command with the following required parameters:

- `--db-proxy-endpoint-name`

The following command deletes the proxy endpoint named `my-endpoint`.

For Linux, macOS, or Unix:

```
aws rds delete-db-proxy-endpoint \
--db-proxy-endpoint-name my-endpoint
```

For Windows:

```
aws rds delete-db-proxy-endpoint ^
--db-proxy-endpoint-name my-endpoint
```

RDS API

To delete a proxy endpoint with the RDS API, run the `DeleteDBProxyEndpoint` operation. Specify the name of the proxy endpoint for the `DBProxyEndpointName` parameter.

Limits for proxy endpoints

Each proxy has a default endpoint that you can modify but not create or delete.

The maximum number of user-defined endpoints for a proxy is 20. Thus, a proxy can have up to 21 endpoints: the default endpoint, plus 20 that you create.

When you associate additional endpoints with a proxy, RDS Proxy automatically determines which DB instances in your cluster to use for each endpoint. You can't choose specific instances the way that you can with Aurora custom endpoints.

To use cross-VPC capability with RDS Proxy, all the VPCs must be owned by the same AWS account.

Reader endpoints aren't available for Aurora multi-writer clusters.

You can connect to proxy endpoints that you create using the SSL modes `REQUIRED` and `VERIFY_CA`. You can't connect to an endpoint that you create using the SSL mode `VERIFY_IDENTITY`.

Command-line examples for RDS Proxy

To see how combinations of connection commands and SQL statements interact with RDS Proxy, look at the following examples.

Examples

- [Preserving Connections to a MySQL Database Across a Failover](#)
- [Adjusting the max_connections Setting for an Aurora DB Cluster](#)

Example Preserving connections to a MySQL database across a failover

This MySQL example demonstrates how open connections continue working during a failover, for example when you reboot a database or it becomes unavailable due to a problem. This example uses a proxy named `the-proxy` and an Aurora DB cluster with DB instances `instance-8898` and `instance-9814`. When you run the `failover-db-cluster` command from the Linux command line, the writer instance that the proxy is connected to changes to a different DB instance. You can see that the DB instance associated with the proxy changes while the connection remains open.

```
$ mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com -u admin_user -p
Enter password:
...
mysql> select @@aurora_server_id;
+-----+
| @@aurora_server_id |
+-----+
| instance-9814      |
+-----+
1 row in set (0.01 sec)

mysql>
[1]+ Stopped                  mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com -
u admin_user -p
$ # Initially, instance-9814 is the writer.
$ aws rds failover-db-cluster --db-cluster-id cluster-56-2019-11-14-1399
JSON output
$ # After a short time, the console shows that the failover operation is complete.
$ # Now instance-8898 is the writer.
$ fg
mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com -u admin_user -p

mysql> select @@aurora_server_id;
+-----+
| @@aurora_server_id |
+-----+
| instance-8898      |
+-----+
1 row in set (0.01 sec)

mysql>
[1]+ Stopped                  mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com -
u admin_user -p
$ aws rds failover-db-cluster --db-cluster-id cluster-56-2019-11-14-1399
JSON output
$ # After a short time, the console shows that the failover operation is complete.
$ # Now instance-9814 is the writer again.
$ fg
mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com -u admin_user -p

mysql> select @@aurora_server_id;
+-----+
| @@aurora_server_id |
+-----+
| instance-9814      |
+-----+
1 row in set (0.01 sec)
+-----+
```

```
| Variable_name | Value      |
+-----+-----+
| hostname    | ip-10-1-3-178 |
+-----+-----+
1 row in set (0.02 sec)
```

Example Adjusting the max_connections setting for an Aurora DB cluster

This example demonstrates how you can adjust the max_connections setting for an Aurora MySQL DB cluster. To do so, you create your own DB cluster parameter group based on the default parameter settings for clusters that are compatible with MySQL 5.6 or 5.7. You specify a value for the max_connections setting, overriding the formula that sets the default value. You associate the DB cluster parameter group with your DB cluster.

```
export REGION=us-east-1
export CLUSTER_PARAM_GROUP=rds-proxy-mysql-56-max-connections-demo
export CLUSTER_NAME=rds-proxy-mysql-56

aws rds create-db-parameter-group --region $REGION \
--db-parameter-group-family aurora5.6 \
--db-parameter-group-name $CLUSTER_PARAM_GROUP \
--description "Aurora MySQL 5.6 cluster parameter group for RDS Proxy demo."

aws rds modify-db-cluster --region $REGION \
--db-cluster-identifier $CLUSTER_NAME \
--db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP

echo "New cluster param group is assigned to cluster:"
aws rds describe-db-clusters --region $REGION \
--db-cluster-identifier $CLUSTER_NAME \
--query '*[*].{DBClusterParameterGroup:DBClusterParameterGroup}'

echo "Current value for max_connections:"
aws rds describe-db-cluster-parameters --region $REGION \
--db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP \
--query '*[*].{ParameterName:ParameterName,ParameterValue:ParameterValue}' \
--output text | grep "max_connections"

echo -n "Enter number for max_connections setting: "
read answer

aws rds modify-db-cluster-parameter-group --region $REGION --db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP \
--parameters "ParameterName=max_connections,ParameterValue=$answer,ApplyMethod=immediate"

echo "Updated value for max_connections:"
aws rds describe-db-cluster-parameters --region $REGION \
--db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP \
--query '*[*].{ParameterName:ParameterName,ParameterValue:ParameterValue}' \
--output text | grep "max_connections"
```

Troubleshooting for RDS Proxy

Following, you can find troubleshooting ideas for some common RDS Proxy issues and information on CloudWatch logs for RDS Proxy.

In the RDS Proxy logs, each entry is prefixed with the name of the associated proxy endpoint. This name can be the name you specified for a user-defined endpoint, or the special name default for read/write requests using the default endpoint of a proxy. For more information about proxy endpoints, see [Endpoints for Amazon RDS Proxy \(p. 197\)](#).

Topics

- [Common issues and solutions \(p. 206\)](#)
- [Working with CloudWatch logs for RDS Proxy \(p. 210\)](#)
- [Verifying connectivity for a proxy \(p. 210\)](#)

Common issues and solutions

For possible causes and solutions to some common problems that you might encounter using RDS Proxy, see the following.

You might encounter the following issues while creating a new proxy or connecting to a proxy.

Error	Causes or workarounds
403: The security token included in the request is invalid	Select an existing IAM role instead of choosing to create a new one.

You might encounter the following issues while connecting to a MySQL proxy.

Error	Causes or workarounds
ERROR 1040 (HY000): Connections rate limit exceeded (<i>limit_value</i>)	The rate of connection requests from the client to the proxy has exceeded the limit.
ERROR 1040 (HY000): IAM authentication rate limit exceeded	The number of simultaneous requests with IAM authentication from the client to the proxy has exceeded the limit.
ERROR 1040 (HY000): Number simultaneous connections exceeded (<i>limit_value</i>)	The number of simultaneous connection requests from the client to the proxy exceeded the limit.
ERROR 1045 (28000): Access denied for user ' <i>DB_USER</i> '@'%' (using password: YES)	Some possible reasons include the following: <ul style="list-style-type: none">• The Secrets Manager secret used by the proxy doesn't match the user name and password of an existing database user. Either update the credentials in the Secrets Manager secret, or make sure the database user exists and has the same password as in the secret.
ERROR 1105 (HY000): Unknown error	An unknown error occurred.
ERROR 1231 (42000): Variable	The value set for the character_set_client parameter is not valid. For example, the value ucs2 is not valid because it can crash the MySQL server.

Error	Causes or workarounds
'character_set_client' can't be set to the value of <code>value</code>	
ERROR 3159 (HY000): This RDS Proxy requires TLS connections.	You enabled the setting Require Transport Layer Security in the proxy but your connection included the parameter <code>ssl-mode=DISABLED</code> in the MySQL client. Do either of the following: <ul style="list-style-type: none"> Disable the setting Require Transport Layer Security for the proxy. Connect to the database using the minimum setting of <code>ssl-mode=REQUIRED</code> in the MySQL client.
ERROR 2026 (HY000): SSL connection error: Internal Server <code>Error</code>	The TLS handshake to the proxy failed. Some possible reasons include the following: <ul style="list-style-type: none"> SSL is required but the server doesn't support it. An internal server error occurred. A bad handshake occurred.
ERROR 9501 (HY000): Timed- out waiting to acquire database connection	The proxy timed-out waiting to acquire a database connection. Some possible reasons include the following: <ul style="list-style-type: none"> The proxy is unable to establish a database connection because the maximum connections have been reached The proxy is unable to establish a database connection because the database is unavailable.

You might encounter the following issues while connecting to a PostgreSQL proxy.

Error	Cause	Solution
IAM authentication is allowed only with SSL connections.	The user tried to connect to the database using IAM authentication with the setting <code>sslmode=disable</code> in the PostgreSQL client.	The user needs to connect to the database using the minimum setting of <code>sslmode=require</code> in the PostgreSQL client. For more information, see the PostgreSQL SSL support documentation.
This RDS Proxy requires TLS connections.	The user enabled the option Require Transport Layer Security but tried to connect with <code>sslmode=disable</code> in the PostgreSQL client.	To fix this error, do one of the following: <ul style="list-style-type: none"> Disable the proxy's Require Transport Layer Security option. Connect to the database using the minimum setting of <code>sslmode=allow</code> in the PostgreSQL client.
IAM authentication failed for user <code>user_name</code> . Check the IAM token for this user and try again.	This error might be due to the following reasons: <ul style="list-style-type: none"> The client supplied the incorrect IAM user name. 	To fix this error, do the following: <ol style="list-style-type: none"> Confirm that the provided IAM user exists.

Error	Cause	Solution
	<ul style="list-style-type: none"> The client supplied an incorrect IAM authorization token for the user. The client is using an IAM policy that does not have the necessary permissions. The client supplied an expired IAM authorization token for the user. 	2. Confirm that the IAM authorization token belongs to the provided IAM user. 3. Confirm that the IAM policy has adequate permissions for RDS. 4. Check the validity of the IAM authorization token used.
This RDS proxy has no credentials for the role <i>role_name</i> . Check the credentials for this role and try again.	There is no Secrets Manager secret for this role.	Add a Secrets Manager secret for this role.
RDS supports only IAM or MD5 authentication.	The database client being used to connect to the proxy is using an authentication mechanism not currently supported by the proxy, such as SCRAM-SHA-256.	If you're not using IAM authentication, use the MD5 password authentication only.
A user name is missing from the connection startup packet. Provide a user name for this connection.	The database client being used to connect to the proxy isn't sending a user name when trying to establish a connection.	Make sure to define a user name when setting up a connection to the proxy using the PostgreSQL client of your choice.
Feature not supported: RDS Proxy supports only version 3.0 of the PostgreSQL messaging protocol.	The PostgreSQL client used to connect to the proxy uses a protocol older than 3.0.	Use a newer PostgreSQL client that supports the 3.0 messaging protocol. If you're using the PostgreSQL <code>psql</code> CLI, use a version greater than or equal to 7.4.
Feature not supported: RDS Proxy currently doesn't support streaming replication mode.	The PostgreSQL client used to connect to the proxy is trying to use the streaming replication mode, which isn't currently supported by RDS Proxy.	Turn off the streaming replication mode in the PostgreSQL client being used to connect.
Feature not supported: RDS Proxy currently doesn't support the option <i>option_name</i> .	Through the startup message, the PostgreSQL client used to connect to the proxy is requesting an option that isn't currently supported by RDS Proxy.	Turn off the option being shown as not supported from the message above in the PostgreSQL client being used to connect.
The IAM authentication failed because of too many competing requests.	The number of simultaneous requests with IAM authentication from the client to the proxy has exceeded the limit.	Reduce the rate in which connections using IAM authentication from a PostgreSQL client are established.

Error	Cause	Solution
The maximum number of client connections to the proxy exceeded <code>number_value</code> .	The number of simultaneous connection requests from the client to the proxy exceeded the limit.	Reduce the number of active connections from PostgreSQL clients to this RDS proxy.
Rate of connection to proxy exceeded <code>number_value</code> .	The rate of connection requests from the client to the proxy has exceeded the limit.	Reduce the rate at which connections from a PostgreSQL client are established.
The password that was provided for the role <code>role_name</code> is wrong.	The password for this role doesn't match the Secrets Manager secret.	Check the secret for this role in Secrets Manager to see if the password is the same as what's being used in your PostgreSQL client.
The IAM authentication failed for the role <code>role_name</code> . Check the IAM token for this role and try again.	There is a problem with the IAM token used for IAM authentication.	Generate a new authentication token and use it in a new connection.
IAM is allowed only with SSL connections.	A client tried to connect using IAM authentication, but SSL wasn't enabled.	Enable SSL in the PostgreSQL client.
Unknown error.	An unknown error occurred.	Reach out to AWS Support for us to investigate the issue.
Timed-out waiting to acquire database connection.	<p>The proxy timed-out waiting to acquire a database connection. Some possible reasons include the following:</p> <ul style="list-style-type: none"> The proxy can't establish a database connection because the maximum connections have been reached. The proxy can't establish a database connection because the database is unavailable. 	<p>Possible solutions are:</p> <ul style="list-style-type: none"> Check the target of the RDS DB instance or Aurora DB cluster status to see if it's unavailable. Check if there are long-running transactions and/or queries being executed. They can use database connections from the connection pool for a long time.
Request returned an error: <code>database_error</code> .	The database connection established from the proxy returned an error.	The solution depends on the specific database error. One example is: Request returned an error: database "your-database-name" does not exist. This means the specified database name, or the user name used as a database name (in case a database name hasn't been specified), doesn't exist in the database server.

Working with CloudWatch logs for RDS Proxy

You can find logs of RDS Proxy activity under CloudWatch in the AWS Management Console. Each proxy has an entry in the **Log groups** page.

Important

These logs are intended for human consumption for troubleshooting purposes and not for programmatic access. The format and content of the logs is subject to change.

In particular, older logs don't contain any prefixes indicating the endpoint for each request. In newer logs, each entry is prefixed with the name of the associated proxy endpoint. This name can be the name that you specified for a user-defined endpoint, or the special name `default` for requests using the default endpoint of a proxy.

Verifying connectivity for a proxy

You can use the following commands to verify that all components of the connection mechanism can communicate with the other components.

Examine the proxy itself using the [describe-db-proxies](#) command. Also examine the associated target group using the [describe-db-proxy-target-groups](#). Check that the details of the targets match the RDS DB instance or Aurora DB cluster that you intend to associate with the proxy. Use commands such as the following.

```
aws rds describe-db-proxies --db-proxy-name $DB_PROXY_NAME
aws rds describe-db-proxy-target-groups --db-proxy-name $DB_PROXY_NAME
```

To confirm that the proxy can connect to the underlying database, examine the targets specified in the target groups using the [describe-db-proxy-targets](#) command. Use a command such as the following.

```
aws rds describe-db-proxy-targets --db-proxy-name $DB_PROXY_NAME
```

The output of the [describe-db-proxy-targets](#) command includes a `TargetHealth` field. You can examine the fields `State`, `Reason`, and `Description` inside `TargetHealth` to check if the proxy can communicate with the underlying DB instance.

- A `State` value of `AVAILABLE` indicates that the proxy can connect to the DB instance.
- A `State` value of `UNAVAILABLE` indicates a temporary or permanent connection problem. In this case, examine the `Reason` and `Description` fields. For example, if `Reason` has a value of `PENDING_PROXY_CAPACITY`, try connecting again after the proxy finishes its scaling operation. If `Reason` has a value of `UNREACHABLE`, `CONNECTION_FAILED`, or `AUTH_FAILURE`, use the explanation from the `Description` field to help you diagnose the issue.
- The `State` field might have a value of `REGISTERING` for a brief time before changing to `AVAILABLE` or `UNAVAILABLE`.

If the following Netcat command (`nc`) is successful, you can access the proxy endpoint from the EC2 instance or other system where you're logged in. This command reports failure if you're not in the same VPC as the proxy and the associated database. You might be able to log directly in to the database without being in the same VPC. However, you can't log into the proxy unless you're in the same VPC.

```
nc -zx MySQL_proxy_endpoint 3306
nc -zx PostgreSQL_proxy_endpoint 5432
```

You can use the following commands to make sure that your EC2 instance has the required properties. In particular, the VPC for the EC2 instance must be the same as the VPC for the RDS DB instance or Aurora DB cluster that the proxy connects to.

```
aws ec2 describe-instances --instance-ids your_ec2_instance_id
```

Examine the Secrets Manager secrets used for the proxy.

```
aws secretsmanager list-secrets
aws secretsmanager get-secret-value --secret-id your_secret_id
```

Make sure that the `SecretString` field displayed by `get-secret-value` is encoded as a JSON string that includes `username` and `password` fields. The following example shows the format of the `SecretString` field.

```
{
  "ARN": "some_arn",
  "Name": "some_name",
  "VersionId": "some_version_id",
  "SecretString": '{"username":"some_username","password":"some_password"}',
  "VersionStages": [ "some_stage" ],
  "CreatedDate": some_timestamp
}
```

Using RDS Proxy with AWS CloudFormation

You can use RDS Proxy with AWS CloudFormation. Doing so helps you to create groups of related resources, including a proxy that can connect to a newly created Amazon RDS DB instance or Aurora DB cluster. RDS Proxy support in AWS CloudFormation involves two new registry types: `DBProxy` and `DBProxyTargetGroup`.

The following listing shows a sample AWS CloudFormation template for RDS Proxy.

```
Resources:
  DBProxy:
    Type: AWS::RDS::DBProxy
    Properties:
      DBProxyName: CanaryProxy
      EngineFamily: MYSQL
      RoleArn:
        Fn::ImportValue: SecretReaderRoleArn
      Auth:
        - {AuthScheme: SECRETS, SecretArn: !ImportValue ProxySecret, IAMAuth: DISABLED}
      VpcSubnetIds:
        Fn::Split: [",", "Fn::ImportValue": SubnetIds]

  ProxyTargetGroup:
    Type: AWS::RDS::DBProxyTargetGroup
    Properties:
      DBProxyName: CanaryProxy
      TargetGroupName: default
      DBInstanceIdentifiers:
        - Fn::ImportValue: DBInstanceName
      DependsOn: DBProxy
```

For more information about the Amazon RDS and Aurora resources that you can create using AWS CloudFormation, see [RDS resource type reference](#).

Working with option groups

Some DB engines offer additional features that make it easier to manage data and databases, and to provide additional security for your database. Amazon RDS uses option groups to enable and configure these features. An *option group* can specify features, called options, that are available for a particular Amazon RDS DB instance. Options can have settings that specify how the option works. When you associate a DB instance with an option group, the specified options and option settings are enabled for that DB instance.

Amazon RDS supports options for the following database engines:

Database engine	Relevant documentation
MariaDB	Options for MariaDB database engine (p. 616)
Microsoft SQL Server	Options for the Microsoft SQL Server database engine (p. 749)
MySQL	Options for MySQL DB instances (p. 925)
Oracle	Adding options to Oracle DB instances (p. 1126)
PostgreSQL	PostgreSQL does not use options and option groups. PostgreSQL uses extensions and modules to provide additional features. For more information, see PostgreSQL extensions supported on Amazon RDS (p. 1482) .

Option groups overview

Amazon RDS provides an empty default option group for each new DB instance. You cannot modify this default option group, but any new option group that you create derives its settings from the default option group. To apply an option to a DB instance, you must do the following:

1. Create a new option group, or copy or modify an existing option group.
2. Add one or more options to the option group.
3. Associate the option group with the DB instance.

To associate an option group with a DB instance, modify the DB instance. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Both DB instances and DB snapshots can be associated with an option group. In some cases, you might restore from a DB snapshot or perform a point-in-time restore for a DB instance. In these cases, the option group associated with the DB snapshot or DB instance is, by default, associated with the restored DB instance. You can associate a different option group with a restored DB instance. However, the new option group must contain any persistent or permanent options that were included in the original option group. Persistent and permanent options are described following.

Options require additional memory to run on a DB instance. Thus, you might need to launch a larger instance to use them, depending on your current use of your DB instance. For example, Oracle Enterprise Manager Database Control uses about 300 MB of RAM. If you enable this option for a small DB instance, you might encounter performance problems or out-of-memory errors.

Persistent and permanent options

Two types of options, persistent and permanent, require special consideration when you add them to an option group.

Persistent options can't be removed from an option group while DB instances are associated with the option group. An example of a persistent option is the TDE option for Microsoft SQL Server transparent data encryption (TDE). You must disassociate all DB instances from the option group before a persistent option can be removed from the option group. In some cases, you might restore or perform a point-in-time restore from a DB snapshot. In these cases, if the option group associated with that DB snapshot contains a persistent option, you can only associate the restored DB instance with that option group.

Permanent options, such as the TDE option for Oracle Advanced Security TDE, can never be removed from an option group. You can change the option group of a DB instance that is using the permanent option. However, the option group associated with the DB instance must include the same permanent option. In some cases, you might restore or perform a point-in-time restore from a DB snapshot. In these cases, if the option group associated with that DB snapshot contains a permanent option, you can only associate the restored DB instance with an option group with that permanent option.

For Oracle DB instances, you can copy shared DB snapshots that have the options `Timezone` or `OLS` (or both). To do so, specify a target option group that includes these options when you copy the DB snapshot. The OLS option is permanent and persistent only for Oracle DB instances running Oracle version 12.2 or higher. For more information about these options, see [Oracle time zone \(p. 1201\)](#) and [Oracle Label Security \(p. 1167\)](#).

VPC and platform considerations

When an option group is assigned to a DB instance, it is linked to the platform that the DB instance is on. That platform can either be a VPC supported by the Amazon VPC service, or EC2-Classic (non-VPC) supported by the Amazon EC2 service. For details on these two platforms, see [Amazon EC2 and Amazon Virtual Private Cloud](#).

If a DB instance is in a VPC, the option group associated with the instance is linked to that VPC. This means that you can't use the option group assigned to a DB instance if you try to restore the instance to a different VPC or a different platform. If you restore a DB instance to a different VPC or a different platform, you can do one of the following:

- Assign the default option group to the DB instance.
- Assign an option group that is linked to that VPC or platform.
- Create a new option group and assign it to the DB instance.

With persistent or permanent options, such as Oracle TDE, you must create a new option group that includes the persistent or permanent option when restoring a DB instance into a different VPC.

Option settings control the behavior of an option. For example, the Oracle Advanced Security option `NATIVE_NETWORK_ENCRYPTION` has a setting that you can use to specify the encryption algorithm for network traffic to and from the DB instance. Some options settings are optimized for use with Amazon RDS and cannot be changed.

Mutually exclusive options

Some options are mutually exclusive. You can use one or the other, but not both at the same time. The following options are mutually exclusive:

- [Oracle Enterprise Manager Database Express \(p. 1150\)](#) and [Oracle Management Agent for Enterprise Manager Cloud Control \(p. 1154\)](#).
- [Oracle native network encryption \(p. 1176\)](#) and [Oracle Secure Sockets Layer \(p. 1182\)](#).

Creating an option group

You can create a new option group that derives its settings from the default option group, and then add one or more options to the new option group. Alternatively, if you already have an existing option group, you can copy that option group with all of its options to a new option group. For more information, see [Copying an option group \(p. 215\)](#).

After you create a new option group, it has no options. To learn how to add options to the option group, see [Adding an option to an option group \(p. 216\)](#). After you have added the options you want, you can then associate the option group with a DB instance so that the options become available on the DB instance. For information about associating an option group with a DB instance, see the documentation for your specific engine listed at [Working with option groups \(p. 212\)](#).

Console

One way of creating an option group is by using the AWS Management Console.

To create a new option group by using the console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose **Create group**.
4. In the **Create option group** window, do the following:
 - a. For **Name**, type a name for the option group that is unique within your AWS account. The name can contain only letters, digits, and hyphens.
 - b. For **Description**, type a brief description of the option group. The description is used for display purposes.
 - c. For **Engine**, choose the DB engine that you want.
 - d. For **Major engine version**, choose the major version of the DB engine that you want.
5. To continue, choose **Create**. To cancel the operation instead, choose **Cancel**.

AWS CLI

To create an option group, use the AWS CLI `create-option-group` command with the following required parameters.

- `--option-group-name`
- `--engine-name`
- `--major-engine-version`
- `--option-group-description`

Example

The following example creates an option group named `testoptiongroup`, which is associated with the Oracle Enterprise Edition DB engine. The description is enclosed in quotation marks.

For Linux, macOS, or Unix:

```
aws rds create-option-group \
```

```
--option-group-name testoptiongroup \
--engine-name oracle-ee \
--major-engine-version 12.1 \
--option-group-description "Test option group"
```

For Windows:

```
aws rds create-option-group ^
--option-group-name testoptiongroup ^
--engine-name oracle-ee ^
--major-engine-version 12.1 ^
--option-group-description "Test option group"
```

RDS API

To create an option group, call the Amazon RDS API [CreateOptionGroup](#) operation. Include the following parameters:

- `OptionGroupName`
- `EngineName`
- `MajorEngineVersion`
- `OptionGroupDescription`

Copying an option group

You can use the AWS CLI or the Amazon RDS API copy an option group. Copying an option group is convenient when you have an existing option group and you want to include most of its custom parameters and values in a new option group. You can also make a copy of an option group that you use in production and then modify the copy to test other option settings.

Note

Currently, you can't copy an option group to a different AWS Region.

AWS CLI

To copy an option group, use the AWS CLI [copy-option-group](#) command. Include the following required options:

- `--source-option-group-identifier`
- `--target-option-group-identifier`
- `--target-option-group-description`

Example

The following example creates an option group named `new-option-group`, which is a local copy of the option group `my-option-group`.

For Linux, macOS, or Unix:

```
aws rds copy-option-group \
--source-option-group-identifier my-option-group \
```

```
--target-option-group-identifier new-option-group \
--target-option-group-description "My new option group"
```

For Windows:

```
aws rds copy-option-group ^
--source-option-group-identifier my-option-group ^
--target-option-group-identifier new-option-group ^
--target-option-group-description "My new option group"
```

RDS API

To copy an option group, call the Amazon RDS API [CopyOptionGroup](#) operation. Include the following required parameters.

- `SourceOptionGroupIdentifier`
- `TargetOptionGroupIdentifier`
- `TargetOptionGroupDescription`

Adding an option to an option group

You can add an option to an existing option group. After you have added the options you want, you can then associate the option group with a DB instance so that the options become available on the DB instance. For information about associating an option group with a DB instance, see the documentation for your specific DB engine listed at [Working with option groups \(p. 212\)](#).

Option group changes must be applied immediately in two cases:

- When you add an option that adds or updates a port value, such as the `OEM` option.
- When you add or remove an option group with an option that includes a port value.

In these cases, choose the **Apply Immediately** option in the console. Or you can include the `--apply-immediately` option when using the AWS CLI or set the `ApplyImmediately` parameter to `true` when using the Amazon RDS API. Options that don't include port values can be applied immediately, or can be applied during the next maintenance window for the DB instance.

Note

If you specify a security group as a value for an option in an option group, you manage the security group by modifying the option group. You can't change or remove this security group by modifying a DB instance. Also, the security group doesn't appear in the DB instance details in the AWS Management Console or in the output for the AWS CLI command `describe-db-instances`.

Console

You can use the AWS Management Console to add an option to an option group.

To add an option to an option group by using the console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose the option group that you want to modify, and then choose **Add option**.

Option groups (9)		
<input type="text"/> Filter subnet groups		
	Name	Description
<input type="checkbox"/>	carpcmysql	carpcmysql
<input checked="" type="checkbox"/>	carpcoracle	carpcoracle
<input type="checkbox"/>	default:mysql-5-5	Default option group for mysql 5.5
<input type="checkbox"/>	default:mysql-5-6	Default option group for mysql 5.6
<input type="checkbox"/>	default:mysql-5-7	Default option group for mysql 5.7

4. In the **Add option** window, do the following:

- a. Choose the option that you want to add. You might need to provide additional values, depending on the option that you select. For example, when you choose the **OEM** option, you must also type a port value and specify a security group.
- b. To enable the option on all associated DB instances as soon as you add it, for **Apply Immediately**, choose **Yes**. If you choose **No** (the default), the option is enabled for each associated DB instance during its next maintenance window.

Add Option

Option details

Option group name
carpcoracle

Option
Name of Option you want to add to this group

Port
The port number, if applicable, to use when connecting to the Option

Security Groups
A list of VPC or DB Security Groups for which this Option is enabled

default

Apply Immediately [info](#)
 Yes
 No

- When the settings are as you want them, choose **Add option**.

AWS CLI

To add an option to an option group, run the AWS CLI [add-option-to-option-group](#) command with the option that you want to add. To enable the new option immediately on all associated DB instances, include the `--apply-immediately` parameter. By default, the option is enabled for each associated DB instance during its next maintenance window. Include the following required parameter:

- `--option-group-name`

Example

The following example adds the Oracle Enterprise Manager Database Control (OEM) option to an option group named `testoptiongroup` and immediately enables it. Even if you use the default security group, you must specify that security group.

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \
```

```
--option-group-name testoptiongroup \
--options OptionName=OEM,Port=5500,DBSecurityGroupMemberships=default \
--apply-immediately
```

For Windows:

```
aws rds add-option-to-option-group ^
--option-group-name testoptiongroup ^
--options OptionName=OEM,Port=5500,DBSecurityGroupMemberships=default ^
--apply-immediately
```

Command output is similar to the following:

```
OPTIONGROUP  False  oracle-ee  12.1  arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
Test Option Group  testoptiongroup default
OPTIONS Oracle 12c EM Express  OEM      False    False   5500
DBSECURITYGROUPEMEMBERSHIPS  default authorized
```

Example

The following example adds the Oracle OEM option to an option group. It also specifies a custom port and a pair of Amazon EC2 VPC security groups to use for that port.

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \
--option-group-name testoptiongroup \
--options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2" \
--apply-immediately
```

For Windows:

```
aws rds add-option-to-option-group ^
--option-group-name testoptiongroup ^
--options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2" ^
--apply-immediately
```

Command output is similar to the following:

```
OPTIONGROUP  False  oracle-ee  12.1  arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
Test Option Group  testoptiongroup vpc-test
OPTIONS Oracle 12c EM Express  OEM      False    False   5500
VPCSECURITYGROUPEMEMBERSHIPS  active  sg-test1
VPCSECURITYGROUPEMEMBERSHIPS  active  sg-test2
```

Example

The following example adds the Oracle option NATIVE_NETWORK_ENCRYPTION to an option group and specifies the option settings. If no option settings are specified, default values are used.

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \
--option-group-name testoptiongroup \
--options '[{"OptionSettings": [{"Name": "SQLNET.ENCRYPTION_SERVER", "Value": "REQUIRED"}, {"Name": "SQLNET.ENCRYPTION_TYPES_SERVER", "Value": "AES256,AES192,DES"}]}, {"OptionName": "NATIVE_NETWORK_ENCRYPTION", "Value": "AES256,AES192,DES"}]' \
--apply-immediately
```

For Windows:

```
aws rds add-option-to-option-group ^
--option-group-name testoptiongroup ^
--options "OptionSettings=[{ "Name": "SQLNET.ENCRYPTION_SERVER", "Value": "REQUIRED"}, { "Name": "SQLNET.ENCRYPTION_TYPES_SERVER", "Value": "AES256\,AES192\,DES"}], "OptionName": "NATIVE_NETWORK_ENCRYPTION", ^
--apply-immediately
```

Command output is similar to the following:

```
OPTIONGROUP False oracle-ee 12.1 arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
Test Option Group testoptiongroup
OPTIONS Oracle Advanced Security - Native Network Encryption      NATIVE_NETWORK_ENCRYPTION
      False False
OPTIONSETTINGS
      RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40
          STATIC STRING
      RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40      Specifies
          list of encryption algorithms in order of intended use
      True True SQLNET.ENCRYPTION_TYPES_SERVER AES256,AES192,DES
OPTIONSETTINGS ACCEPTED,REJECTED,REQUESTED,REQUIRED STATIC STRING REQUESTED
          Specifies the desired encryption behavior False True SQLNET.ENCRYPTION_SERVER
          REQUIRED
OPTIONSETTINGS SHA1,MD5 STATIC STRING SHA1,MD5 Specifies list of checksumming
          algorithms in order of intended use True True SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER
          SHA1,MD5
```

RDS API

To add an option to an option group using the Amazon RDS API, call the [ModifyOptionGroup](#) operation with the option that you want to add. To enable the new option immediately on all associated DB instances, include the `ApplyImmediately` parameter and set it to `true`. By default, the option is enabled for each associated DB instance during its next maintenance window. Include the following required parameter:

- `OptionGroupName`

Listing the options and option settings for an option group

You can list all the options and option settings for an option group.

Console

You can use the AWS Management Console to list all of the options and option settings for an option group.

To list the options and option settings for an option group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose the name of the option group to display its details. The options and option settings in the option group are listed.

AWS CLI

To list the options and option settings for an option group, use the AWS CLI `describe-option-groups` command. Specify the name of the option group whose options and settings you want to view. If you don't specify an option group name, all option groups are described.

Example

The following example lists the options and option settings for all option groups.

```
aws rds describe-option-groups
```

Example

The following example lists the options and option settings for an option group named `testoptiongroup`.

```
aws rds describe-option-groups --option-group-name testoptiongroup
```

RDS API

To list the options and option settings for an option group, use the Amazon RDS API `DescribeOptionGroups` operation. Specify the name of the option group whose options and settings you want to view. If you don't specify an option group name, all option groups are described.

Modifying an option setting

After you have added an option that has modifiable option settings, you can modify the settings at any time. If you change options or option settings in an option group, those changes are applied to all DB instances that are associated with that option group. For more information on what settings are available for the various options, see the documentation for your specific engine listed at [Working with option groups \(p. 212\)](#).

Option group changes must be applied immediately in two cases:

- When you add an option that adds or updates a port value, such as the `OEM` option.
- When you add or remove an option group with an option that includes a port value.

In these cases, choose the **Apply Immediately** option in the console. Or you can include the `--apply-immediately` option when using the AWS CLI or set the `ApplyImmediately` parameter to `true` when

using the RDS API. Options that don't include port values can be applied immediately, or can be applied during the next maintenance window for the DB instance.

Note

If you specify a security group as a value for an option in an option group, you manage the security group by modifying the option group. You can't change or remove this security group by modifying a DB instance. Also, the security group doesn't appear in the DB instance details in the AWS Management Console or in the output for the AWS CLI command `describe-db-instances`.

Console

You can use the AWS Management Console to modify an option setting.

To modify an option setting by using the console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Select the option group whose option that you want to modify, and then choose **Modify option**.
4. In the **Modify option** window, from **Installed Options**, choose the option whose setting you want to modify. Make the changes that you want.
5. To enable the option as soon as you add it, for **Apply Immediately**, choose **Yes**. If you choose **No** (the default), the option is enabled for each associated DB instance during its next maintenance window.
6. When the settings are as you want them, choose **Modify Option**.

AWS CLI

To modify an option setting, use the AWS CLI `add-option-to-option-group` command with the option group and option that you want to modify. By default, the option is enabled for each associated DB instance during its next maintenance window. To apply the change immediately to all associated DB instances, include the `--apply-immediately` parameter. To modify an option setting, use the `--settings` argument.

Example

The following example modifies the port that the Oracle Enterprise Manager Database Control (OEM) uses in an option group named `testoptiongroup` and immediately applies the change.

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \
--option-group-name testoptiongroup \
--options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default \
--apply-immediately
```

For Windows:

```
aws rds add-option-to-option-group ^
--option-group-name testoptiongroup ^
--options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default ^
--apply-immediately
```

Command output is similar to the following:

```
OPTIONGROUP  False  oracle-ee  12.1  arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
  Test Option Group      testoptiongroup
OPTIONS Oracle 12c EM Express    OEM      False   False   5432
DBSECURITYGROUPMEMBERSHIPS    default  authorized
```

Example

The following example modifies the Oracle option NATIVE_NETWORK_ENCRYPTION and changes the option settings.

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \
--option-group-name testoptiongroup \
--options '[{"OptionSettings": [{"Name": "SQLNET.ENCRYPTION_SERVER", "Value": "REQUIRED"}, {"Name": "SQLNET.ENCRYPTION_TYPES_SERVER", "Value": "AES256,AES192,DES,RC4_256"}], "OptionName": "NATIVE_NETWORK_ENCRYPTION"}]' \
--apply-immediately
```

For Windows:

```
aws rds add-option-to-option-group ^
--option-group-name testoptiongroup ^
--options "OptionSettings=[{ "Name": "SQLNET.ENCRYPTION_SERVER", "Value": "REQUIRED"}, { "Name": "SQLNET.ENCRYPTION_TYPES_SERVER", "Value": "AES256\,AES192\,DES\,RC4_256"}], "OptionName": "NATIVE_NETWORK_ENCRYPTION" ^
--apply-immediately
```

Command output is similar to the following:

```
OPTIONGROUP  False  oracle-ee  12.1  arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
  Test Option Group      testoptiongroup
OPTIONS Oracle Advanced Security - Native Network Encryption      NATIVE_NETWORK_ENCRYPTION
  False   False
OPTIONSETTINGS
  RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40  STATIC
  STRING
    RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40
    Specifies list of encryption algorithms in order of intended use
      True      True      SQLNET.ENCRYPTION_TYPES_SERVER      AES256,AES192,DES,RC4_256
OPTIONSETTINGS  ACCEPTED,REJECTED,REQUESTED,REQUIRED  STATIC  STRING  REQUESTED
  Specifies the desired encryption behavior  False      True      SQLNET.ENCRYPTION_SERVER
  REQUIRED
OPTIONSETTINGS  SHA1,MD5  STATIC  STRING  SHA1,MD5  Specifies list of
  checksumming algorithms in order of intended use  True      True
  SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER  SHA1,MD5
OPTIONSETTINGS  ACCEPTED,REJECTED,REQUESTED,REQUIRED  STATIC  STRING
  REQUESTED  Specifies the desired data integrity behavior  False      True
  SQLNET.CRYPTO_CHECKSUM_SERVER  REQUESTED
```

RDS API

To modify an option setting, use the Amazon RDS API [ModifyOptionGroup](#) command with the option group and option that you want to modify. By default, the option is enabled for each associated DB instance during its next maintenance window. To apply the change immediately to all associated DB instances, include the `ApplyImmediately` parameter and set it to `true`.

Removing an option from an option group

Some options can be removed from an option group, and some cannot. A persistent option cannot be removed from an option group until all DB instances associated with that option group are disassociated. A permanent option can never be removed from an option group. For more information about what options are removable, see the documentation for your specific engine listed at [Working with option groups \(p. 212\)](#).

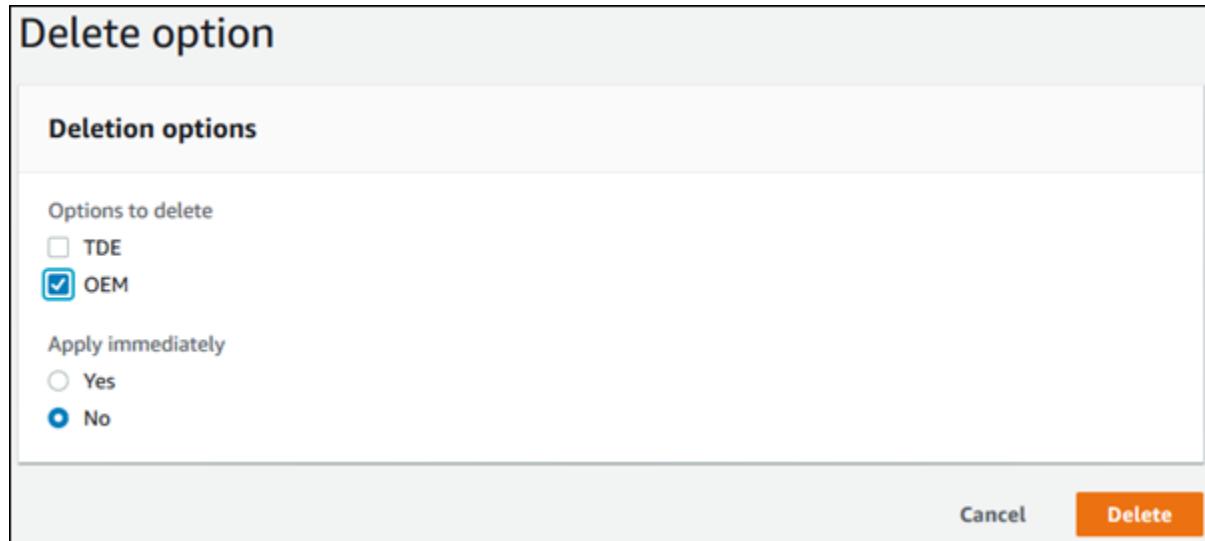
If you remove all options from an option group, Amazon RDS doesn't delete the option group. DB instances that are associated with the empty option group continue to be associated with it; they just won't have any active options. Alternatively, to remove all options from a DB instance, you can associate the DB instance with the default (empty) option group.

Console

You can use the AWS Management Console to remove an option from an option group.

To remove an option from an option group by using the console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Select the option group whose option you want to remove, and then choose **Delete option**.
4. In the **Delete option** window, do the following:
 - Select the check box for the option that you want to delete.
 - For the deletion to take effect as soon as you make it, for **Apply immediately**, choose **Yes**. If you choose **No** (the default), the option is deleted for each associated DB instance during its next maintenance window.



- When the settings are as you want them, choose **Yes, Delete**.

AWS CLI

To remove an option from an option group, use the AWS CLI `remove-option-from-option-group` command with the option that you want to delete. By default, the option is removed from each associated DB instance during its next maintenance window. To apply the change immediately, include the `--apply-immediately` parameter.

Example

The following example removes the Oracle Enterprise Manager Database Control (OEM) option from an option group named `testoptiongroup` and immediately applies the change.

For Linux, macOS, or Unix:

```
aws rds remove-option-from-option-group \
--option-group-name testoptiongroup \
--options OEM \
--apply-immediately
```

For Windows:

```
aws rds remove-option-from-option-group ^
--option-group-name testoptiongroup ^
--options OEM ^
--apply-immediately
```

Command output is similar to the following:

```
OPTIONGROUP      testoptiongroup oracle-ee    12.1      Test option group
```

RDS API

To remove an option from an option group, use the Amazon RDS API `ModifyOptionGroup` action. By default, the option is removed from each associated DB instance during its next maintenance window. To apply the change immediately, include the `ApplyImmediately` parameter and set it to `true`.

Include the following parameters:

- `OptionGroupName`
- `OptionsToRemove.OptionName`

Deleting an option group

You can delete an option group that is not associated with any Amazon RDS resource. An option group can be associated with a DB instance, a manual DB snapshot, or an automated DB snapshot.

If you try to delete an option group that is associated with an Amazon RDS resource, an error similar to the following is returned.

An error occurred (InvalidOptionGroupStateFault) when calling the DeleteOptionGroup operation: The option group 'optionGroupName' cannot be deleted because it is in use.

To find the Amazon RDS resources associated with an option group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose the name of the option group to show its details.
4. Check the **Associated Instances and Snapshots** section for the associated Amazon RDS resources.

If a DB instance is associated with the option group, modify the DB instance to use a different option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

If a manual DB snapshot is associated with the option group, modify the DB snapshot to use a different option group using the AWS CLI `modify-db-snapshot` command.

Note

You can't modify the option group of an automated DB snapshot.

Console

One way of deleting an option group is by using the AWS Management Console.

To delete an option group by using the console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose the option group.
4. Choose **Delete group**.
5. On the confirmation page, choose **Delete** to finish deleting the option group, or choose **Cancel** to cancel the deletion.

AWS CLI

To delete an option group, use the AWS CLI `delete-option-group` command with the following required parameter.

- `--option-group-name`

Example

The following example deletes an option group named `testoptiongroup`.

For Linux, macOS, or Unix:

```
aws rds delete-option-group \
--option-group-name testoptiongroup
```

For Windows:

```
aws rds delete-option-group ^
--option-group-name testoptiongroup
```

RDS API

To delete an option group, call the Amazon RDS API [DeleteOptionGroup](#) operation. Include the following parameter:

- OptionGroupName

Working with DB parameter groups

You manage your DB engine configuration by associating your DB instances with parameter groups. Amazon RDS defines parameter groups with default settings that apply to newly created DB instances.

Important

You can define your own parameter groups with customized settings. Then you can modify your DB instances to use your own parameter groups.

For information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

A *DB parameter group* acts as a container for engine configuration values that are applied to one or more DB instances.

If you create a DB instance without specifying a DB parameter group, the DB instance uses a default DB parameter group. Each default DB parameter group contains database engine defaults and Amazon RDS system defaults based on the engine, compute class, and allocated storage of the instance. You can't modify the parameter settings of a default parameter group. Instead, you create your own parameter group where you choose your own parameter settings. Not all DB engine parameters can be changed in a parameter group that you create.

If you want to use your own parameter group, you create a new parameter group and modify the parameters that you want to. You then modify your DB instance to use the new parameter group. If you update parameters within a DB parameter group, the changes apply to all DB instances that are associated with that parameter group.

You can copy an existing DB parameter group with the AWS CLI [copy-db-parameter-group](#) command. Copying a parameter group can be convenient when you want to include most of an existing DB parameter group's custom parameters and values in a new DB parameter group.

Here are some important points about working with parameters in a DB parameter group:

- When you change a dynamic parameter and save the DB parameter group, the change is applied immediately regardless of the **Apply Immediately** setting. When you change a static parameter and save the DB parameter group, the parameter change takes effect after you manually reboot the DB instance. You can reboot a DB instance using the RDS console, by calling the `reboot-db-instance` CLI command, or by calling the `RebootDbInstance` API operation. The requirement to reboot the associated DB instance after a static parameter change helps mitigate the risk of a parameter misconfiguration affecting an API call, such as calling `ModifyDBInstance` to change DB instance class or scale storage.

If a DB instance isn't using the latest changes to its associated DB parameter group, the AWS Management Console shows the DB parameter group with a status of **pending-reboot**. The **pending-reboot** parameter groups status doesn't result in an automatic reboot during the next maintenance window. To apply the latest parameter changes to that DB instance, manually reboot the DB instance.

- When you change the DB parameter group associated with a DB instance, you must manually reboot the instance before the DB instance can use the new DB parameter group. For more information about changing the DB parameter group, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
- You can specify the value for a DB parameter as an integer or as an integer expression built from formulas, variables, functions, and operators. Functions can include a mathematical log expression. For more information, see [Specifying DB parameters \(p. 240\)](#).
- Set any parameters that relate to the character set or collation of your database in your parameter group before creating the DB instance and before you create a database in your DB instance. This ensures that the default database and new databases in your DB instance use the character set and collation values that you specify. If you change character set or collation parameters for your DB instance, the parameter changes are not applied to existing databases.

You can change character set or collation values for an existing database using the `ALTER DATABASE` command, for example:

```
ALTER DATABASE database_name CHARACTER SET character_set_name COLLATE collation;
```

- Improperly setting parameters in a DB parameter group can have unintended adverse effects, including degraded performance and system instability. Always exercise caution when modifying database parameters and back up your data before modifying a DB parameter group. Try out parameter group setting changes on a test DB instance before applying those parameter group changes to a production DB instance.
- To determine the supported parameters for your DB engine, you can view the parameters in the DB parameter group used by the DB instance. For more information, see [Viewing parameter values for a DB parameter group \(p. 239\)](#).

Topics

- [Creating a DB parameter group \(p. 229\)](#)
- [Associating a DB parameter group with a DB instance \(p. 231\)](#)
- [Modifying parameters in a DB parameter group \(p. 232\)](#)
- [Resetting parameters in a DB parameter group to their default values \(p. 234\)](#)
- [Copying a DB parameter group \(p. 236\)](#)
- [Listing DB parameter groups \(p. 238\)](#)
- [Viewing parameter values for a DB parameter group \(p. 239\)](#)
- [Comparing DB parameter groups \(p. 240\)](#)
- [Specifying DB parameters \(p. 240\)](#)

Creating a DB parameter group

You can create a new DB parameter group using the AWS Management Console, the AWS CLI, or the RDS API.

Console

To create a DB parameter group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. Choose **Create parameter group**.
The **Create parameter group** window appears.
4. In the **Parameter group family** list, select a DB parameter group family.
5. In the **Type** list, select **DB Parameter Group**.
6. In the **Group name** box, enter the name of the new DB parameter group.
7. In the **Description** box, enter a description for the new DB parameter group.
8. Choose **Create**.

AWS CLI

To create a DB parameter group, use the AWS CLI [create-db-parameter-group](#) command. The following example creates a DB parameter group named *mydbparametergroup* for MySQL version 5.6 with a description of "My new parameter group."

Include the following required parameters:

- `--db-parameter-group-name`
- `--db-parameter-group-family`
- `--description`

To list all of the available parameter group families, use the following command:

```
aws rds describe-db-engine-versions --query "DBEngineVersions[].DBParameterGroupFamily"
```

Note

The output contains duplicates.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-parameter-group \  
  --db-parameter-group-name mydbparametergroup \  
  --db-parameter-group-family MySQL5.6 \  
  --description "My new parameter group"
```

For Windows:

```
aws rds create-db-parameter-group ^  
  --db-parameter-group-name mydbparametergroup ^  
  --db-parameter-group-family MySQL5.6 ^  
  --description "My new parameter group"
```

This command produces output similar to the following:

```
DBPARAMETERGROUP  mydbparametergroup  mysql5.6  My new parameter group
```

RDS API

To create a DB parameter group, use the RDS API [CreateDBParameterGroup](#) operation.

Include the following required parameters:

- `DBParameterGroupName`
- `DBParameterGroupFamily`
- `Description`

Associating a DB parameter group with a DB instance

You can create your own DB parameter groups with customized settings. You can associate a DB parameter group with a DB instance using the AWS Management Console, the AWS CLI, or the RDS API. You can do so when you create or modify a DB instance.

For information about creating a DB parameter group, see [Creating a DB parameter group \(p. 229\)](#).

For information about creating a DB instance, see [Creating an Amazon RDS DB instance \(p. 141\)](#). For information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Note

When you change the DB parameter group associated with a DB instance, you must manually reboot the instance before the DB instance can use the new DB parameter group.

Console

To associate a DB parameter group with a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
3. Choose **Modify**. The **Modify DB Instance** page appears.
4. Change the **DB parameter group** setting.
5. Choose **Continue** and check the summary of modifications.
6. (Optional) Choose **Apply immediately** to apply the changes immediately. Choosing this option can cause an outage in some cases. For more information, see [Using the Apply Immediately setting \(p. 251\)](#).
7. On the confirmation page, review your changes. If they are correct, choose **Modify DB instance** to save your changes.

Or choose **Back** to edit your changes or **Cancel** to cancel your changes.

AWS CLI

To associate a DB parameter group with a DB instance, use the AWS CLI `modify-db-instance` command with the following options:

- `--db-instance-identifier`
- `--db-parameter-group-name`

The following example associates the `mydbpg` DB parameter group with the `database-1` DB instance. The changes are applied immediately by using `--apply-immediately`. Use `--no-apply-immediately` to apply the changes during the next maintenance window. For more information, see [Using the Apply Immediately setting \(p. 251\)](#).

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
  --db-instance-identifier database-1 \
```

```
--db-parameter-group-name mydbpg \
--apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier database-1 ^
--db-parameter-group-name mydbpg ^
--apply-immediately
```

RDS API

To associate a DB parameter group with a DB instance, use the RDS API [ModifyDBInstance](#) operation with the following parameters:

- `DBInstanceName`
- `DBParameterGroupName`

Modifying parameters in a DB parameter group

You can modify parameter values in a customer-created DB parameter group; you can't change the parameter values in a default DB parameter group. Changes to parameters in a customer-created DB parameter group are applied to all DB instances that are associated with the DB parameter group.

Changes to some parameters are applied to the DB instance immediately without a reboot. Changes to other parameters are applied only after the DB instance is rebooted. The RDS console shows the status of the DB parameter group associated with a DB instance on the **Configuration** tab. For example, if the DB instance isn't using the latest changes to its associated DB parameter group, the RDS console shows the DB parameter group with a status of **pending-reboot**. To apply the latest parameter changes to that DB instance, manually reboot the DB instance.

Configuration		Instance class
DB instance id	database-2	Instance class
Engine version	14.00.3281.6.v1	db.r4.large
DB name	-	vCPU
License model	License Included	RAM
Collation	SQL_Latin1_General_CI_AS	15.25 GB
Option groups	test-se-2017	Availability
ARN	arn:aws:rds:us-west-2:123456789012:db:database-2	Master username
Resource id	db-123456789012	admin
Created time	Wed Dec 04 2019 14:22:38 GMT-0500 (Eastern Standard Time)	IAM db authentication
Parameter group	test-sqlserver-se-2017 (pending-reboot)	Not Enabled
Deletion protection	Disabled	Multi AZ
		Yes (Mirroring)
		Secondary Zone
		us-west-2d

Console

To modify a DB parameter group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. In the list, choose the parameter group that you want to modify.
4. For **Parameter group actions**, choose **Edit**.
5. Change the values of the parameters that you want to modify. You can scroll through the parameters using the arrow keys at the top right of the dialog box.

You can't change values in a default parameter group.
6. Choose **Save changes**.

AWS CLI

To modify a DB parameter group, use the AWS CLI `modify-db-parameter-group` command with the following required options:

- `--db-parameter-group-name`

- `--parameters`

The following example modifies the `max_connections` and `max_allowed_packet` values in the DB parameter group named `mydbparametergroup`.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --parameters "ParameterName=max_connections,ParameterValue=250,ApplyMethod=immediate" \
    "ParameterName=max_allowed_packet,ParameterValue=1024,ApplyMethod=immediate"
```

For Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --parameters "ParameterName=max_connections,ParameterValue=250,ApplyMethod=immediate" ^
    "ParameterName=max_allowed_packet,ParameterValue=1024,ApplyMethod=immediate"
```

The command produces output like the following:

```
DBPARAMETERGROUP  mydbparametergroup
```

RDS API

To modify a DB parameter group, use the RDS API [ModifyDBParameterGroup](#) operation with the following required parameters:

- `DBParameterGroupName`
- `Parameters`

Resetting parameters in a DB parameter group to their default values

You can reset parameter values in a customer-created DB parameter group to their default values. Changes to parameters in a customer-created DB parameter group are applied to all DB instances that are associated with the DB parameter group.

When you use the console, you can reset specific parameters to their default values, but you can't easily reset all of the parameters in the DB parameter group at once. When you use the AWS CLI or RDS API, you can reset specific parameters to their default values, and you can reset all of the parameters in the DB parameter group at once.

Changes to some parameters are applied to the DB instance immediately without a reboot. Changes to other parameters are applied only after the DB instance is rebooted. The RDS console shows the status of the DB parameter group associated with a DB instance on the **Configuration** tab. For example, if the DB instance isn't using the latest changes to its associated DB parameter group, the RDS console shows the DB parameter group with a status of **pending-reboot**. To apply the latest parameter changes to that DB instance, manually reboot the DB instance.

The screenshot shows the 'Configuration' tab selected in the top navigation bar. The main content area is titled 'Instance' and contains two columns of configuration details.

Configuration	Instance class
DB instance id database-2	Instance class db.r4.large
Engine version 14.00.3281.6.v1	vCPU 2
DB name -	RAM 15.25 GB
License model License Included	Availability
Collation SQL_Latin1_General_CI_AS	Master username admin
Option groups test-se-2017	IAM db authentication Not Enabled
ARN arn:aws:rds:us-west-2:123456789012:db:database-2	Multi AZ Yes (Mirroring)
Resource id db-12345678	Secondary Zone us-west-2d
Created time Wed Dec 04 2019 14:22:38 GMT-0500 (Eastern Standard Time)	
Parameter group test-sqlserver-se-2017 (pending-reboot)	
Deletion protection Disabled	

Note

In a default DB parameter group, parameters are always set to their default values.

Console

To reset parameters in a DB parameter group to their default values

- Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
 - In the navigation pane, choose **Parameter groups**.
 - In the list, choose the parameter group.
 - For **Parameter group actions**, choose **Edit**.
 - Choose the parameters that you want to reset to their default values. You can scroll through the parameters using the arrow keys at the top right of the dialog box.
- You can't reset values in a default parameter group.
- Choose **Reset** and then confirm by choosing **Reset parameters**.

AWS CLI

To reset some or all of the parameters in a DB parameter group, use the AWS CLI `reset-db-parameter-group` command with the following required option: `--db-parameter-group-name`.

To reset all of the parameters in the DB parameter group, specify the `--reset-all-parameters` option. To reset specific parameters, specify the `--parameters` option.

The following example resets all of the parameters in the DB parameter group named `mydbparametergroup` to their default values.

Example

For Linux, macOS, or Unix:

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --reset-all-parameters
```

For Windows:

```
aws rds reset-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --reset-all-parameters
```

The following example resets the `max_connections` and `max_allowed_packet` options to their default values in the DB parameter group named `mydbparametergroup`.

Example

For Linux, macOS, or Unix:

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" \
        "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

For Windows:

```
aws rds reset-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" ^
        "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

The command produces output like the following:

```
DBParameterGroupName mydbparametergroup
```

RDS API

To reset parameters in a DB parameter group to their default values, use the RDS API `ResetDBParameterGroup` command with the following required parameter: `DBParameterGroupName`.

To reset all of the parameters in the DB parameter group, set the `ResetAllParameters` parameter to `true`. To reset specific parameters, specify the `Parameters` parameter.

Copying a DB parameter group

You can copy custom DB parameter groups that you create. Copying a parameter group is a convenient solution when you have already created a DB parameter group and you want to include most of the

custom parameters and values from that group in a new DB parameter group. You can copy a DB parameter group by using the AWS Management Console, the AWS CLI [copy-db-parameter-group](#) command, or the RDS API [CopyDBParameterGroup](#) operation.

After you copy a DB parameter group, wait at least 5 minutes before creating your first DB instance that uses that DB parameter group as the default parameter group. Doing this allows Amazon RDS to fully complete the copy action before the parameter group is used. This is especially important for parameters that are critical when creating the default database for a DB instance. An example is the character set for the default database defined by the `character_set_database` parameter. Use the **Parameter Groups** option of the [Amazon RDS console](#) or the [describe-db-parameters](#) command to verify that your DB parameter group is created.

Note

You can't copy a default parameter group. However, you can create a new parameter group that is based on a default parameter group.

Currently, you can't copy a parameter group to a different AWS Region.

Console

To copy a DB parameter group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. In the list, choose the custom parameter group that you want to copy.
4. For **Parameter group actions**, choose **Copy**.
5. In **New DB parameter group identifier**, enter a name for the new parameter group.
6. In **Description**, enter a description for the new parameter group.
7. Choose **Copy**.

AWS CLI

To copy a DB parameter group, use the AWS CLI [copy-db-parameter-group](#) command with the following required options:

- `--source-db-parameter-group-identifier`
- `--target-db-parameter-group-identifier`
- `--target-db-parameter-group-description`

The following example creates a new DB parameter group named `mygroup2` that is a copy of the DB parameter group `mygroup1`.

Example

For Linux, macOS, or Unix:

```
aws rds copy-db-parameter-group \
--source-db-parameter-group-identifier mygroup1 \
--target-db-parameter-group-identifier mygroup2 \
--target-db-parameter-group-description "DB parameter group 2"
```

For Windows:

```
aws rds copy-db-parameter-group ^
--source-db-parameter-group-identifier mygroup1 ^
```

```
--target-db-parameter-group-identifier mygroup2 ^
--target-db-parameter-group-description "DB parameter group 2"
```

RDS API

To copy a DB parameter group, use the RDS API [CopyDBParameterGroup](#) operation with the following required parameters:

- `SourceDBParameterGroupIdentifier`
- `TargetDBParameterGroupIdentifier`
- `TargetDBParameterGroupDescription`

Listing DB parameter groups

You can list the DB parameter groups you've created for your AWS account.

Note

Default parameter groups are automatically created from a default parameter template when you create a DB instance for a particular DB engine and version. These default parameter groups contain preferred parameter settings and can't be modified. When you create a custom parameter group, you can modify parameter settings.

Console

To list all DB parameter groups for an AWS account

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.

The DB parameter groups appear in a list.

AWS CLI

To list all DB parameter groups for an AWS account, use the AWS CLI [describe-db-parameter-groups](#) command.

Example

The following example lists all available DB parameter groups for an AWS account.

```
aws rds describe-db-parameter-groups
```

The command returns a response like the following:

```
DBPARAMETERGROUP  default.mysql5.5      mysql5.5  Default parameter group for MySQL5.5
DBPARAMETERGROUP  default.mysql5.6      mysql5.6  Default parameter group for MySQL5.6
DBPARAMETERGROUP  mydbparametergroup    mysql5.6  My new parameter group
```

The following example describes the *mydbparamgroup1* parameter group.

For Linux, macOS, or Unix:

```
aws rds describe-db-parameter-groups \
```

```
--db-parameter-group-name mydbparamgroup1
```

For Windows:

```
aws rds describe-db-parameter-groups ^
--db-parameter-group-name mydbparamgroup1
```

The command returns a response like the following:

```
DBPARAMETERGROUP mydbparametergroup1 mysql5.5 My new parameter group
```

RDS API

To list all DB parameter groups for an AWS account, use the RDS API [DescribeDBParameterGroups](#) operation.

Viewing parameter values for a DB parameter group

You can get a list of all parameters in a DB parameter group and their values.

Console

To view the parameter values for a DB parameter group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
The DB parameter groups appear in a list.
3. Choose the name of the parameter group to see its list of parameters.

AWS CLI

To view the parameter values for a DB parameter group, use the AWS CLI [describe-db-parameters](#) command with the following required parameter.

- `--db-parameter-group-name`

Example

The following example lists the parameters and parameter values for a DB parameter group named `mydbparametergroup`.

```
aws rds describe-db-parameters --db-parameter-group-name mydbparametergroup
```

The command returns a response like the following:

DBPARAMETER Type	Parameter Name	Parameter Value	Source	Data Type	Apply
DBPARAMETER Is Modifiable					
DBPARAMETER	allow-suspicious-udfs		engine-default	boolean	static
	false				
DBPARAMETER	auto_increment_increment		engine-default	integer	dynamic
	true				
DBPARAMETER	auto_increment_offset		engine-default	integer	dynamic
	true				

DBPARAMETER	binlog_cache_size	32768	system	integer	dynamic
DBPARAMETER	socket	/tmp/mysql.sock	system	string	static

RDS API

To view the parameter values for a DB parameter group, use the RDS API [DescribeDBParameters](#) command with the following required parameter.

- `DBParameterGroupName`

Comparing DB parameter groups

You can use the AWS Management Console to view the differences between two parameter groups for the same DB engine and version.

To compare two parameter groups

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. In the list, choose the two parameter groups that you want to compare.
4. For **Parameter group actions**, choose **Compare**.

Note

If the items you selected aren't equivalent, you can't choose **Compare**. For example, you can't compare a MySQL 5.6 and a MySQL 5.7 parameter group. You can't compare a DB parameter group and an Aurora DB cluster parameter group.

Specifying DB parameters

DB parameter types include the following:

- integer
- Boolean
- string
- long
- double
- timestamp
- object of other defined data types
- array of values of type integer, Boolean, string, long, double, timestamp, or object

You can also specify integer and Boolean DB parameters using expressions, formulas, and functions.

For the Oracle engine, you can use the `DBInstanceClassHugePagesDefault` formula variable to specify a Boolean DB parameter. See [DB parameter formula variables \(p. 241\)](#).

For the PostgreSQL engine, you can use an expression to specify a Boolean DB parameter. See [Boolean DB parameter expressions \(p. 243\)](#).

Contents

- [DB parameter formulas \(p. 241\)](#)
 - [DB parameter formula variables \(p. 241\)](#)
 - [DB parameter formula operators \(p. 241\)](#)
- [DB parameter functions \(p. 242\)](#)
- [Boolean DB parameter expressions \(p. 243\)](#)
- [DB parameter log expressions \(p. 244\)](#)
- [DB parameter value examples \(p. 244\)](#)

DB parameter formulas

A DB parameter formula is an expression that resolves to an integer value or a Boolean value. You enclose the expression in braces: {}. You can use a formula for either a DB parameter value or as an argument to a DB parameter function.

Syntax

```
{FormulaVariable}  
{FormulaVariable*Integer}  
{FormulaVariable*Integer/Integer}  
{FormulaVariable/Integer}
```

DB parameter formula variables

Each formula variable returns an integer or a Boolean value. The names of the variables are case-sensitive.

AllocatedStorage

Returns an integer representing the size, in bytes, of the data volume.

DBInstanceClassHugePagesDefault

Returns a Boolean value. Currently, it's only supported for Oracle engines.

For more information, see [Enabling HugePages for an Oracle DB instance \(p. 1101\)](#).

DBInstanceClassMemory

Returns an integer of the number of bytes of memory allocated to the DB instance class associated with the current DB instance, less the memory used by RDS processes that manage the instance.

DBInstanceVCPU

Returns an integer representing the number of virtual central processing units (vCPUs) used by Amazon RDS to manage the instance. Currently, it's only supported for the PostgreSQL engine.

EndPointPort

Returns an integer representing the port used when connecting to the DB instance.

DB parameter formula operators

DB parameter formulas support two operators: division and multiplication.

Division operator: /

Divides the dividend by the divisor, returning an integer quotient. Decimals in the quotient are truncated, not rounded.

Syntax

```
dividend / divisor
```

The dividend and divisor arguments must be integer expressions.

Multiplication operator: *

Multiplies the expressions, returning the product of the expressions. Decimals in the expressions are truncated, not rounded.

Syntax

```
expression * expression
```

Both expressions must be integers.

DB parameter functions

You specify the arguments of DB parameter functions as either integers or formulas. Each function must have at least one argument. Specify multiple arguments as a comma-separated list. The list can't have any empty members, such as *argument1,,argument3*. Function names are case-insensitive.

IF

Returns an argument.

Currently, it's only supported for Oracle engines, and the only supported first argument is {DBInstanceClassHugePagesDefault}. For more information, see [Enabling HugePages for an Oracle DB instance \(p. 1101\)](#).

Syntax

```
IF(argument1, argument2, argument3)
```

Returns the second argument if the first argument evaluates to true. Returns the third argument otherwise.

GREATEST

Returns the largest value from a list of integers or parameter formulas.

Syntax

```
GREATEST(argument1, argument2,...argumentn)
```

Returns an integer.

LEAST

Returns the smallest value from a list of integers or parameter formulas.

Syntax

```
LEAST(argument1, argument2,...argumentn)
```

Returns an integer.

SUM

Adds the values of the specified integers or parameter formulas.

Syntax

```
SUM(argument1, argument2,...argumentn)
```

Returns an integer.

Boolean DB parameter expressions

A Boolean DB parameter expression resolves to a Boolean value of 1 or 0. The expression is enclosed in quotation marks.

Note

Boolean DB parameter expressions are only supported for the PostgreSQL engine.

Syntax

```
"expression operator expression"
```

Both expressions must resolve to integers. An expression can be the following:

- integer constant
- DB parameter formula
- DB parameter function
- DB parameter variable

Boolean DB parameter expressions support the following inequality operators:

The greater than operator: >

Syntax

```
"expression > expression"
```

The less than operator: <

Syntax

```
"expression < expression"
```

The greater than or equal to operators: >=, =>

Syntax

```
"expression >= expression"  
"expression => expression"
```

The less than or equal to operators: <=, =<

Syntax

```
"expression <= expression"
```

```
"expression =< expression"
```

Example using a Boolean DB parameter expression

The following Boolean DB parameter expression example compares the result of a parameter formula with an integer to modify the Boolean DB parameter `wal_compression` for a PostgreSQL DB instance. The parameter expression compares the number of vCPUs with the value 2. If the number of vCPUs is greater than 2, then the `wal_compression` DB parameter is set to true.

```
aws rds modify-db-parameter-group --db-parameter-group-name group-name \  
--parameters "ParameterName=wal_compression,ParameterValue=\\"{DBInstanceVCPU} > 2\\" "
```

DB parameter log expressions

You can set an integer DB parameter value to a log expression. You enclose the expression in braces: {}.

For example:

```
{log(DBInstanceClassMemory/8187281418)*1000}
```

The `log` function represents log base 2. This example also uses the `DBInstanceClassMemory` formula variable. See [DB parameter formula variables \(p. 241\)](#).

Note

Currently, you can't specify the MySQL `innodb_log_file_size` parameter with any value other than an integer.

DB parameter value examples

These examples show using formulas, functions, and expressions for the values of DB parameters.

Note

DB Parameter functions are currently supported only in the console and aren't supported in the AWS CLI.

Warning

Improperly setting parameters in a DB parameter group can have unintended adverse effects. These might include degraded performance and system instability. Use caution when modifying database parameters and back up your data before modifying your DB parameter group. Try out parameter group changes on a test DB instance, created using point-in-time-restores, before applying those parameter group changes to your production DB instances.

Example using the DB parameter function GREATEST

You can specify the `GREATEST` function in an Oracle processes parameter. Use it to set the number of user processes to the larger of either 80 or `DBInstanceClassMemory` divided by 9,868,951.

```
GREATEST({DBInstanceClassMemory/9868951},80)
```

Example using the DB parameter function LEAST

You can specify the `LEAST` function in a MySQL `max_binlog_cache_size` parameter value. Use it to set the maximum cache size a transaction can use in a MySQL instance to the lesser of 1 MB or `DBInstanceClass/256`.

```
LEAST({DBInstanceClassMemory/256},10485760)
```

Managing an Amazon RDS DB instance

Following, you can find instructions for managing and maintaining your Amazon RDS DB instance.

Topics

- [Stopping an Amazon RDS DB instance temporarily \(p. 246\)](#)
- [Starting an Amazon RDS DB instance that was previously stopped \(p. 249\)](#)
- [Modifying an Amazon RDS DB instance \(p. 250\)](#)
- [Maintaining a DB instance \(p. 264\)](#)
- [Upgrading a DB instance engine version \(p. 271\)](#)
- [Renaming a DB instance \(p. 274\)](#)
- [Rebooting a DB instance \(p. 276\)](#)
- [Working with read replicas \(p. 278\)](#)
- [Tagging Amazon RDS resources \(p. 299\)](#)
- [Working with Amazon Resource Names \(ARNs\) in Amazon RDS \(p. 309\)](#)
- [Working with storage for Amazon RDS DB instances \(p. 316\)](#)
- [Deleting a DB instance \(p. 324\)](#)

Stopping an Amazon RDS DB instance temporarily

If you use a DB instance intermittently, for temporary testing, or for a daily development activity, you can stop your Amazon RDS DB instance temporarily to save money. While your DB instance is stopped, you are charged for provisioned storage (including Provisioned IOPS) and backup storage (including manual snapshots and automated backups within your specified retention window), but not for DB instance hours. For more information, see [Billing FAQs](#).

Note

In some cases, a large amount of time is required to stop a DB instance. If you want to stop your DB instance and restart it immediately, you can reboot the DB instance. For information about rebooting a DB instance, see [Rebooting a DB instance \(p. 276\)](#).

You can stop and start DB instances that are running the following engines:

- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Stopping and starting a DB instance is supported for all DB instance classes, and in all AWS Regions.

You can stop and start a DB instance whether it is configured for a single Availability Zone or for Multi-AZ, for database engines that support Multi-AZ deployments. You can't stop an Amazon RDS for SQL Server DB instance in a Multi-AZ configuration.

Note

For a Multi-AZ deployment, a large amount of time might be required to stop a DB instance.

If you have at least one backup after a previous failover, then you can speed up the stop DB instance operation by performing a reboot with failover operation before stopping the DB instance.

When you stop a DB instance, the DB instance performs a normal shutdown and stops running. The status of the DB instance changes to **stopping** and then **stopped**. Any storage volumes remain attached to the DB instance, and their data is kept. Any data stored in the RAM of the DB instance is deleted.

Stopping a DB instance removes pending actions, except for pending actions for the DB instance's option group or DB parameter group.

Automated backups aren't created while a DB instance is stopped. Backups can be retained longer than the backup retention period if a DB instance has been stopped. RDS doesn't include time spent in the stopped state when the backup retention window is calculated.

Important

You can stop a DB instance for up to seven days. If you don't manually start your DB instance after seven days, your DB instance is automatically started so that it doesn't fall behind any required maintenance updates.

Benefits

Stopping and starting a DB instance is faster than creating a DB snapshot, and then restoring the snapshot.

When you stop a DB instance it retains its ID, Domain Name Server (DNS) endpoint, parameter group, security group, and option group. When you start a DB instance, it has the same configuration as when

you stopped it. In addition, if you stop a DB instance, Amazon RDS retains the Amazon Simple Storage Service (Amazon S3) transaction logs so you can do a point-in-time restore if necessary.

Limitations

The following are some limitations to stopping and starting a DB instance:

- You can't stop a DB instance that has a read replica, or that is a read replica.
- You can't stop an Amazon RDS for SQL Server DB instance in a Multi-AZ configuration.
- You can't modify a stopped DB instance.
- You can't delete an option group that is associated with a stopped DB instance.
- You can't delete a DB parameter group that is associated with a stopped DB instance.

Option and parameter group considerations

You can't remove persistent options (including permanent options) from an option group if there are DB instances associated with that option group. This functionality is also true of any DB instance with a state of stopping, stopped, or starting.

You can change the option group or DB parameter group that is associated with a stopped DB instance, but the change does not occur until the next time you start the DB instance. If you chose to apply changes immediately, the change occurs when you start the DB instance. Otherwise the change occurs during the next maintenance window after you start the DB instance.

Public IP address

When you stop a DB instance, it retains its DNS endpoint. If you stop a DB instance that has a public IP address, Amazon RDS releases its public IP address. When the DB instance is restarted, it has a different public IP address.

Note

You should always connect to a DB instance using the DNS endpoint, not the IP address.

Stopping a DB instance temporarily

You can stop a DB using the AWS Management Console, the AWS CLI, or the RDS API.

Console

To stop a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to stop.
3. For **Actions**, choose **Stop**.
4. (Optional) In the **Stop DB Instance** window, choose **Yes** for **Create Snapshot?** and enter the snapshot name for **Snapshot name**. Choose **Yes** if you want to create a snapshot of the DB instance before stopping it.
5. Choose **Yes, Stop Now** to stop the DB instance, or choose **Cancel** to cancel the operation.

AWS CLI

To stop a DB instance by using the AWS CLI, call the `stop-db-instance` command with the following option:

- `--db-instance-identifier` – the name of the DB instance.

Example

```
aws rds stop-db-instance --db-instance-identifier mydbinstance
```

RDS API

To stop a DB instance by using the Amazon RDS API, call the [StopDBInstance](#) operation with the following parameter:

- `DBInstanceIdentifier` – the name of the DB instance.

Starting an Amazon RDS DB instance that was previously stopped

You can stop your Amazon RDS DB instance temporarily to save money. After you stop your DB instance, you can restart it to begin using it again. For more details about stopping and starting DB instances, see [Stopping an Amazon RDS DB instance temporarily \(p. 246\)](#).

When you start a DB instance that you previously stopped, the DB instance retains the ID, Domain Name Server (DNS) endpoint, parameter group, security group, and option group. When you start a stopped instance, you are charged a full instance hour.

Console

To start a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to start.
3. For **Actions**, choose **Start**.

AWS CLI

To start a DB instance by using the AWS CLI, call the `start-db-instance` command with the following option:

- `--db-instance-identifier` – The name of the DB instance.

Example

```
aws rds start-db-instance --db-instance-identifier mydbinstance
```

RDS API

To start a DB instance by using the Amazon RDS API, call the `StartDBInstance` operation with the following parameter:

- `DBInstanceIdentifier` – The name of the DB instance.

Modifying an Amazon RDS DB instance

You can change the settings of a DB instance to accomplish tasks such as adding additional storage or changing the DB instance class. In this topic, you can find out how to modify an Amazon RDS DB instance and learn about the settings for DB instances.

We recommend that you test any changes on a test instance before modifying a production instance, so that you fully understand the impact of each change. Testing is especially important when upgrading database versions.

Most modifications to a DB instance you can either apply immediately or defer until the next maintenance window. Some modifications, such as parameter group changes, require that you manually reboot your DB instance for the change to take effect.

Important

Some modifications result in downtime because Amazon RDS must reboot your DB instance for the change to take effect. Review the impact to your database and applications before modifying your DB instance settings.

Console

To modify a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
3. Choose **Modify**. The **Modify DB Instance** page appears.
4. Change any of the settings that you want. For information about each setting, see [Settings for DB instances \(p. 251\)](#).
5. When all the changes are as you want them, choose **Continue** and check the summary of modifications.
6. (Optional) Choose **Apply immediately** to apply the changes immediately. Choosing this option can cause downtime in some cases. For more information, see [Using the Apply Immediately setting \(p. 251\)](#).
7. On the confirmation page, review your changes. If they are correct, choose **Modify DB Instance** to save your changes.

Or choose **Back** to edit your changes or **Cancel** to cancel your changes.

AWS CLI

To modify a DB instance by using the AWS CLI, call the `modify-db-instance` command. Specify the DB instance identifier and the values for the options that you want to modify. For information about each option, see [Settings for DB instances \(p. 251\)](#).

Example

The following code modifies `mydbinstance` by setting the backup retention period to 1 week (7 days). The code enables deletion protection by using `--deletion-protection`. To disable deletion protection, use `--no-deletion-protection`. The changes are applied during the next maintenance window by using `--no-apply-immediately`. Use `--apply-immediately` to apply the changes immediately. For more information, see [Using the Apply Immediately setting \(p. 251\)](#).

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--backup-retention-period 7 \
--deletion-protection \
--no-apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--backup-retention-period 7 ^
--deletion-protection ^
--no-apply-immediately
```

RDS API

To modify a DB instance by using the Amazon RDS API, call the [ModifyDBInstance](#) operation. Specify the DB instance identifier, and the parameters for the settings that you want to modify. For information about each parameter, see [Settings for DB instances \(p. 251\)](#).

Using the Apply Immediately setting

When you modify a DB instance, you can apply the changes immediately. To apply changes immediately, you choose the **Apply Immediately** option in the AWS Management Console. Or you use the `--apply-immediately` parameter when calling the AWS CLI or set the `ApplyImmediately` parameter to `true` when using the Amazon RDS API.

If you don't choose to apply changes immediately, the changes are put into the pending modifications queue. During the next maintenance window, any pending changes in the queue are applied. If you choose to apply changes immediately, your new changes and any changes in the pending modifications queue are applied.

Important

If any of the pending modifications require the DB instance to be temporarily unavailable (*downtime*), choosing the apply immediately option can cause unexpected downtime.

When you choose to apply a change immediately, any pending modifications are also applied immediately, instead of during the next maintenance window.

If you don't want a pending change to be applied in the next maintenance window, you can modify the DB instance to revert the change. You can do this by using the AWS CLI and specifying the `--apply-immediately` option.

Changes to some database settings are applied immediately, even if you choose to defer your changes. To see how the different database settings interact with the apply immediately setting, see [Settings for DB instances \(p. 251\)](#).

Settings for DB instances

In the following table, you can find details about which settings you can and can't modify, when changes can be applied, and whether the changes cause downtime for your DB instance.

You can modify a DB instance using the console, the `modify-db-instance` CLI command, or the [ModifyDBInstance](#) RDS API operation.

Console setting and description	CLI option and RDS API parameter	When the change occurs	Downtime notes	Supported DB engines
Allocated storage The storage, in gibibytes, that you want to allocate for your DB instance. You can only increase the allocated storage. You can't reduce the allocated storage. You can't modify the storage of some older DB instances, or DB instances restored from older DB snapshots. The Allocated storage setting is disabled in the console if your DB instance isn't eligible. You can check whether you can allocate more storage by using the CLI command describe-valid-db-instance-modifications . This command returns the valid storage options for your DB instance. You can't modify allocated storage if the DB instance status is storage-optimization or if the allocated storage for the DB instance has been modified in the last six hours. The maximum storage allowed depends on your DB engine and the storage type. For more information, see Amazon RDS DB instance storage (p. 40) .	CLI option: <code>--allocated-storage</code> RDS API parameter: <code>AllocatedStorage</code>	If you choose to apply the change immediately, it occurs immediately. RDS API parameter: <code>AllocatedStorage</code> If you don't choose to apply the change immediately, it occurs during the next maintenance window.	Downtime doesn't occur during this change. Performance might be degraded during the change.	All DB engines
Auto minor version upgrade Yes to enable your DB instance to receive preferred minor DB engine version upgrades automatically when they become available. Amazon RDS performs automatic minor version upgrades in the maintenance window. Otherwise, No. For more information, see Automatically upgrading the minor engine version (p. 273) .	CLI option: <code>--auto-minor-version-upgrade --no-auto-minor-version-upgrade</code> RDS API parameter: <code>AutoMinorVersionUpgrade</code>	The change occurs immediately. This setting ignores the apply immediately setting.	Downtime doesn't occur during this change.	Only MariaDB, MySQL, Oracle, and PostgreSQL
Backup retention period	CLI option:	If you choose to apply the change	Downtime occurs if you change from 0 to a nonzero value,	All DB engines

Console setting and description	CLI option and RDS API parameter	When the change occurs	Downtime notes	Supported DB engines
<p>The number of days that automatic backups are retained. To disable automatic backups, set the backup retention period to 0.</p> <p>For more information, see Working with backups (p. 328).</p> <p>Note If you use AWS Backup to manage your backups, this option doesn't appear. For information about AWS Backup, see the AWS Backup Developer Guide.</p>	CLI option: <code>--backup-retention-period</code> RDS API parameter: <code>BackupRetentionPeriod</code>	immediately, it occurs immediately. If you don't choose to apply the change immediately, and you change the setting from a nonzero value to another nonzero value, the change is applied asynchronously, as soon as possible. Otherwise, the change occurs during the next maintenance window.	or from a nonzero value to 0. This applies to both Single-AZ and Multi-AZ DB instances.	
<p>Backup window</p> <p>The time range during which automated backups of your databases occur. The backup window is a start time in Universal Coordinated Time (UTC), and a duration in hours.</p> <p>For more information, see Working with backups (p. 328).</p> <p>Note If you use AWS Backup to manage your backups, this option doesn't appear. For information about AWS Backup, see the AWS Backup Developer Guide.</p>	CLI option: <code>--preferred-backup-window</code> RDS API parameter: <code>PreferredBackupWindow</code>	The change is applied asynchronously, as soon as possible.	Downtime doesn't occur during this change.	All DB engines
<p>Certificate authority</p> <p>The certificate that you want to use for SSL/TLS connections.</p> <p>For more information, see Using SSL/TLS to encrypt a connection to a DB instance (p. 1634).</p>	CLI option: <code>--ca-certificate-identifier</code> RDS API parameter: <code>CACertificateIdentifier</code>	If you choose to apply the change immediately, it occurs immediately. If you don't choose to apply the change immediately, it occurs during the next maintenance window.	Downtime occurs during this change.	All DB engines

Console setting and description	CLI option and RDS API parameter	When the change occurs	Downtime notes	Supported DB engines
Copy tags to snapshots <p>If you have any DB instance tags, enable this option to copy them when you create a DB snapshot.</p> <p>For more information, see Tagging Amazon RDS resources (p. 299).</p>	CLI option: <code>--copy-tags-to-snapshot</code> or <code>--no-copy-tags-to-snapshot</code> RDS API parameter: <code>CopyTagsToSnapshot</code>	The change occurs immediately. This setting ignores the apply immediately setting.	Downtime doesn't occur during this change.	All DB engines
Database port <p>The port that you want to use to access the DB instance.</p> <p>The port value must not match any of the port values specified for options in the option group that is associated with the DB instance.</p> <p>For more information, see Connecting to an Amazon RDS DB instance (p. 162).</p>	CLI option: <code>--db-port-number</code> RDS API parameter: <code>DBPortNumber</code>	The change occurs immediately. This setting ignores the apply immediately setting.	The DB instance is rebooted immediately.	All DB engines
DB engine version <p>The version of the DB engine that you want to use. Before you upgrade your production DB instance, we recommend that you test the upgrade process on a test DB instance to verify its duration and to validate your applications.</p> <p>For more information, see Upgrading a DB instance engine version (p. 271).</p>	CLI option: <code>--engine-version</code> RDS API parameter: <code>EngineVersion</code>	<p>If you choose to apply the change immediately, it occurs immediately.</p> <p>If you don't choose to apply the change immediately, it occurs during the next maintenance window.</p>	Downtime occurs during this change.	All DB engines
DB instance class <p>The DB instance class that you want to use.</p> <p>For more information, see DB instance classes (p. 7).</p>	CLI option: <code>--db-instance-class</code> RDS API parameter: <code>DBInstanceClass</code>	<p>If you choose to apply the change immediately, it occurs immediately.</p> <p>If you don't choose to apply the change immediately, it occurs during the next maintenance window.</p>	Downtime occurs during this change.	All DB engines

Console setting and description	CLI option and RDS API parameter	When the change occurs	Downtime notes	Supported DB engines
DB instance identifier The new DB instance identifier. This value is stored as a lowercase string. For more information about the effects of renaming a DB instance, see Renaming a DB instance (p. 274) .	CLI option: <code>--new-db-instance-identifier</code> RDS API parameter: <code>NewDBInstanceIdentifier</code>	If you choose to apply the change immediately, it occurs immediately. If you don't choose to apply the change immediately, it occurs during the next maintenance window.	Downtime occurs during this change.	All DB engines
DB parameter group The DB parameter group that you want associated with the DB instance. For more information, see Working with DB parameter groups (p. 228) .	CLI option: <code>--db-parameter-group-name</code> RDS API parameter: <code>DBParameterGroupName</code>	The parameter group change occurs immediately.	Downtime doesn't occur during this change. However, you must manually reboot the DB instance before the new DB parameter group is used by the DB instance. For more information, see Working with DB parameter groups (p. 228) and Rebooting a DB instance (p. 276) .	All DB engines
Deletion protection Enable deletion protection to prevent your DB instance from being deleted. For more information, see Deleting a DB instance (p. 324) .	CLI option: <code>--deletion-protection --no-deletion-protection</code> RDS API parameter: <code>DeletionProtection</code>	The change occurs immediately. This setting ignores the apply immediately setting.	Downtime doesn't occur during this change.	All DB engines
Enhanced Monitoring Enable Enhanced Monitoring to enable gathering metrics in real time for the operating system that your DB instance runs on. For more information, see Using Enhanced Monitoring (p. 471) .	CLI option: <code>--monitoring-interval and --monitoring-role-arn</code> RDS API parameter: <code>MonitoringInterval</code> and <code>MonitoringRoleArn</code>	The change occurs immediately. This setting ignores the apply immediately setting.	Downtime doesn't occur during this change.	All DB engines

Console setting and description	CLI option and RDS API parameter	When the change occurs	Downtime notes	Supported DB engines
IAM DB authentication Enable IAM DB authentication to authenticate database users through IAM users and roles. For more information, see IAM database authentication for MySQL and PostgreSQL (p. 1660) .	CLI option: <code>--enable-iam-database-authentication --no-enable-iam-database-authentication</code> RDS API parameter: <code>EnableIAMDatabaseAuthentication</code>	If you choose to apply the change immediately, it occurs immediately. If you don't choose to apply the change immediately, it occurs during the next maintenance window.	Downtime doesn't occur during this change.	Only MySQL and PostgreSQL
Kerberos authentication Choose the Active Directory to move the DB instance to. The directory must exist prior to this operation. If a directory is already selected, you can specify None to remove the DB instance from its current directory. For more information, see Kerberos authentication (p. 1629) .	CLI option: <code>--domain and --domain-iam-role-name</code> RDS API parameter: <code>Domain and DomainIAMRoleName</code>	If you choose to apply the change immediately, it occurs immediately. If you don't choose to apply the change immediately, it occurs during the next maintenance window.	A brief downtime occurs during this change.	Only Microsoft SQL Server, MySQL, Oracle, and PostgreSQL
License model Choose bring-your-own-license to use your license for Oracle. Choose license-included to use the general license agreement for Microsoft SQL Server or Oracle. For more information, see Licensing Microsoft SQL Server on Amazon RDS (p. 655) and Oracle licensing options (p. 990) .	CLI option: <code>--license-model</code> RDS API parameter: <code>LicenseModel</code>	If you choose to apply the change immediately, it occurs immediately. If you don't choose to apply the change immediately, it occurs during the next maintenance window.	Downtime occurs during this change.	Only Microsoft SQL Server and Oracle
Log exports The types of database log files to publish to Amazon CloudWatch Logs. For more information, see Publishing database logs to Amazon CloudWatch Logs (p. 506) .	CLI option: <code>--cloudwatch-logs-export-configuration</code> RDS API parameter: <code>CloudwatchLogsExportConfiguration</code>	The change occurs immediately. This setting ignores the apply immediately setting.	Downtime doesn't occur during this change.	All DB engines

Console setting and description	CLI option and RDS API parameter	When the change occurs	Downtime notes	Supported DB engines
<p>Maintenance window</p> <p>The time range during which system maintenance occurs. System maintenance includes upgrades, if applicable. The maintenance window is a start time in Universal Coordinated Time (UTC), and a duration in hours.</p> <p>If you set the window to the current time, there must be at least 30 minutes between the current time and the end of the window to ensure that any pending changes are applied.</p> <p>For more information, see The Amazon RDS maintenance window (p. 268).</p>	<p>CLI option: <code>--preferred-maintenance-window</code></p> <p>RDS API parameter: <code>PreferredMaintenanceWindow</code></p>	<p>The change occurs immediately. This setting ignores the apply immediately setting.</p>	<p>If there are one or more pending actions that cause downtime, and the maintenance window is changed to include the current time, those pending actions are applied immediately and downtime occurs.</p>	All DB engines
<p>Multi-AZ deployment</p> <p>Yes to deploy your DB instance in multiple Availability Zones. Otherwise, No.</p> <p>For more information, see High availability (Multi-AZ) for Amazon RDS (p. 53).</p>	<p>CLI option: <code>--multi-az --no-multi-az</code></p> <p>RDS API parameter: <code>MultiAZ</code></p>	<p>If you choose to apply the change immediately, it occurs immediately.</p> <p>If you don't choose to apply the change immediately, it occurs during the next maintenance window.</p>	Downtime doesn't occur during this change.	All DB engines
<p>New master password</p> <p>The password for your master user. The password must contain 8–41 alphanumeric characters.</p>	<p>CLI option: <code>--master-user-password</code></p> <p>RDS API parameter: <code>MasterUserPassword</code></p>	<p>The change is applied asynchronously, as soon as possible. This setting ignores the apply immediately setting.</p>	Downtime doesn't occur during this change.	All DB engines
<p>Option group</p> <p>The option group that you want associated with the DB instance.</p> <p>For more information, see Working with option groups (p. 212).</p>	<p>CLI option: <code>--option-group-name</code></p> <p>RDS API parameter: <code>OptionGroupName</code></p>	<p>If you choose to apply the change immediately, it occurs immediately.</p> <p>If you don't choose to apply the change immediately, it occurs during the next maintenance window.</p>	Downtime doesn't occur during this change.	All DB engines

Console setting and description	CLI option and RDS API parameter	When the change occurs	Downtime notes	Supported DB engines
Performance Insights Enable Performance Insights to monitor your DB instance load so that you can analyze and troubleshoot your database performance. Performance Insights isn't available for some DB engine versions and DB instance classes. The Performance Insights section doesn't appear in the console if it isn't available for your DB instance. For more information, see Using Performance Insights on Amazon RDS (p. 412) .	CLI option: <code>--enable-performance-insights --no-enable-performance-insights</code> RDS API parameter: <code>EnablePerformanceInsights</code>	The change occurs immediately. This setting ignores the apply immediately setting.	Downtime doesn't occur during this change.	All DB engines
Performance Insights Master key The AWS KMS key identifier for the customer master key (CMK) for encryption of Performance Insights data. The key identifier is the Amazon Resource Name (ARN), AWS KMS key identifier, or the key alias for the CMK. For more information, see Enabling and disabling Performance Insights (p. 415) .	CLI option: <code>--performance-insights-kms-key-id</code> RDS API parameter: <code>PerformanceInsightsKMSKeyId</code>	The change occurs immediately. This setting ignores the apply immediately setting.	Downtime doesn't occur during this change.	All DB engines
Performance Insights Retention period The amount of time, in days, to retain Performance Insights data. Valid values are 7 or 731 (2 years). For more information, see Enabling and disabling Performance Insights (p. 415) .	CLI option: <code>--performance-insights-retention-period</code> RDS API parameter: <code>PerformanceInsightsRetentionPeriod</code>	The change occurs immediately. This setting ignores the apply immediately setting.	Downtime doesn't occur during this change.	All DB engines

Console setting and description	CLI option and RDS API parameter	When the change occurs	Downtime notes	Supported DB engines
Processor features The number of CPU cores and the number of threads per core for the DB instance class of the DB instance. For more information, see Configuring the processor for a DB instance class (p. 20) .	CLI option: <code>--processor-features</code> and <code>--use-default-processor-features</code> <code>--no-use-default-processor-features</code> RDS API parameter: <code>ProcessorFeatures</code> and <code>UseDefaultProcessorFeatures</code>	If you choose to apply the change immediately, it occurs immediately. If you don't choose to apply the change immediately, it occurs during the next maintenance window.	Downtime occurs during this change.	Only Oracle
Provisioned IOPS The new Provisioned IOPS (I/O operations per second) value for the DB instance. The setting is available only if Provisioned IOPS (SSD) is chosen for Storage type . For more information, see Provisioned IOPS SSD storage (p. 42) .	CLI option: <code>--iops</code> RDS API parameter: <code>Iops</code>	If you choose to apply the change immediately, it occurs immediately. If you don't choose to apply the change immediately, it occurs during the next maintenance window.	Downtime doesn't occur during this change.	All DB engines

Console setting and description	CLI option and RDS API parameter	When the change occurs	Downtime notes	Supported DB engines
<p>Public access</p> <p>Publicly accessible to give the DB instance a public IP address, meaning that it's accessible outside the VPC. To be publicly accessible, the DB instance also has to be in a public subnet in the VPC.</p> <p>Not publicly accessible to make the DB instance accessible only from inside the VPC.</p> <p>For more information, see Hiding a DB instance in a VPC from the internet (p. 1729).</p> <p>To connect to a DB instance from outside of its Amazon VPC, the DB instance must be publicly accessible, access must be granted using the inbound rules of the DB instance's security group, and other requirements must be met. For more information, see Can't connect to Amazon RDS DB instance (p. 1746).</p> <p>If your DB instance is isn't publicly accessible, you can also use an AWS Site-to-Site VPN connection or an AWS Direct Connect connection to access it from a private network. For more information, see Internetwork traffic privacy (p. 1643).</p>	<p>CLI option: <code>--publicly-accessible --no-publicly-accessible</code></p> <p>RDS API parameter: <code>PubliclyAccessible</code></p>	<p>The change occurs immediately. This setting ignores the apply immediately setting.</p>	<p>Downtime doesn't occur during this change.</p>	All DB engines
<p>Security group</p> <p>The VPC security group that you want associated with the DB instance.</p> <p>For more information, see Controlling access with security groups (p. 1699).</p>	<p>CLI option: <code>--vpc-security-group-ids</code></p> <p>RDS API parameter: <code>VpcSecurityGroupIds</code></p>	<p>The change is applied asynchronously, as soon as possible. This setting ignores the apply immediately setting.</p>	<p>Downtime doesn't occur during this change.</p>	All DB engines

Console setting and description	CLI option and RDS API parameter	When the change occurs	Downtime notes	Supported DB engines
<p>Storage autoscaling</p> <p>Enable storage autoscaling to enable Amazon RDS to automatically increase storage when needed to avoid having your DB instance run out of storage space.</p> <p>Use Maximum storage threshold to set the upper limit for Amazon RDS to automatically increase storage for your DB instance. The default is 1,000 GiB.</p> <p>For more information, see Managing capacity automatically with Amazon RDS storage autoscaling (p. 317).</p>	<p>CLI option: --max-allocated-storage</p> <p>RDS API parameter: MaxAllocatedStorage</p>	<p>The change occurs immediately. This setting ignores the apply immediately setting.</p>	<p>Downtime doesn't occur during this change.</p>	All DB engines

Console setting and description	CLI option and RDS API parameter	When the change occurs	Downtime notes	Supported DB engines
<p>Storage type</p> <p>The storage type that you want to use.</p> <p>After Amazon RDS begins to modify your DB instance to change the storage size or type, you can't submit another request to change the storage size or type for six hours.</p> <p>For more information, see Amazon RDS storage types (p. 40).</p>	<p>CLI option: <code>--storage-type</code></p> <p>RDS API parameter: <code>StorageType</code></p>	<p>If you choose to apply the change immediately, it occurs immediately.</p> <p>If you don't choose to apply the change immediately, it occurs during the next maintenance window.</p>	<p>The following changes all result in a brief downtime while the process starts. After that, you can use your database normally while the change takes place.</p> <ul style="list-style-type: none"> • From General Purpose (SSD) to Magnetic. • From General Purpose (SSD) to Provisioned IOPS (SSD). The downtime only happens if the DB instance is Single-AZ and you are using a custom parameter group. There is no downtime for a Multi-AZ DB instance. • From Magnetic to General Purpose (SSD). • From Magnetic to Provisioned IOPS (SSD). • From Provisioned IOPS (SSD) to Magnetic. • From Provisioned IOPS (SSD) to General Purpose (SSD). The downtime only happens if the DB instance is Single-AZ and you are using a custom parameter group. There is no downtime for a Multi-AZ DB instance. 	All DB engines

Console setting and description	CLI option and RDS API parameter	When the change occurs	Downtime notes	Supported DB engines
<p>Subnet group</p> <p>The subnet group for the DB instance. You can use this setting to move your DB instance to a different VPC. If your DB instance isn't in a VPC, you can use this setting to move your DB instance into a VPC.</p> <p>For more information, see Amazon Virtual Private Cloud VPCs and Amazon RDS (p. 1718).</p>	<p>CLI option: <code>--db-subnet-group-name</code></p> <p>RDS API parameter: <code>DBSubnetGroupName</code></p>	<p>If you choose to apply the change immediately, it occurs immediately.</p> <p>If you don't choose to apply the change immediately, it occurs during the next maintenance window.</p>	Downtime occurs during this change.	All DB engines

Maintaining a DB instance

Periodically, Amazon RDS performs maintenance on Amazon RDS resources. Maintenance most often involves updates to the DB instance's underlying hardware, underlying operating system (OS), or database engine version. Updates to the operating system most often occur for security issues and should be done as soon as possible.

Some maintenance items require that Amazon RDS take your DB instance offline for a short time. Maintenance items that require a resource to be offline include required operating system or database patching. Required patching is automatically scheduled only for patches that are related to security and instance reliability. Such patching occurs infrequently (typically once every few months) and seldom requires more than a fraction of your maintenance window.

Deferred DB instance modifications that you have chosen not to apply immediately are also applied during the maintenance window. For example, you might choose to change the DB instance class or parameter group during the maintenance window. Such modifications that you specify using the **pending reboot** setting don't show up in the **Pending maintenance** list. For information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

You can view whether a maintenance update is available for your DB instance by using the RDS console, the AWS CLI, or the Amazon RDS API. If an update is available, it is indicated in the **Maintenance** column for the DB instance on the Amazon RDS console, as shown following.

Current activity	Maintenance	VPC	Multi-AZ
0 Connections	none	vpc-2aed394c	No
0 Connections	next window	vpc-2aed394c	No
0.02 Sessions	none	vpc-2aed394c	No

If no maintenance update is available for a DB instance, the column value is **none** for it.

If a maintenance update is available for a DB instance, the following column values are possible:

- **required** – The maintenance action will be applied to the resource and can't be deferred indefinitely.
- **available** – The maintenance action is available, but it will not be applied to the resource automatically. You can apply it manually.
- **next window** – The maintenance action will be applied to the resource during the next maintenance window.
- **In progress** – The maintenance action is in the process of being applied to the resource.

If an update is available, you can take one of the actions:

- If the maintenance value is **next window**, defer the maintenance items by choosing **Defer upgrade** from **Actions**. You can't defer a maintenance action if it has already started.
- Apply the maintenance items immediately.

- Schedule the maintenance items to start during your next maintenance window.
- Take no action.

Note

Certain OS updates are marked as **required**. If you defer a required update, you get a notice from Amazon RDS indicating when the update will be performed. Other updates are marked as **available**, and these you can defer indefinitely.

To take an action, choose the DB instance to show its details, then choose **Maintenance & backups**. The pending maintenance items appear.

The screenshot shows the AWS RDS console interface. At the top, there are tabs: Connectivity & security, Monitoring, Logs & events, Configuration, Maintenance & backups (which is highlighted in orange), and Tags. Below the tabs, there's a section titled "Maintenance". It shows three status boxes: "Auto minor version upgrade" (Enabled), "Maintenance window" (mon:11:28-mon:11:58 UTC (GMT)), and "Pending maintenance next window". Under "Pending maintenance (1)", there's a table with columns: Description, Type, Status, and Apply date. A single row is listed: "Automatic minor version upgrade to postgres 9.6.11" (db-upgrade, next window, February 25th 2019, 3:28:00 am UTC-8 (local)). There are "Apply now" and "Apply at next maintenance window" buttons above the table, and a "Filter pending maintenance" search bar below it.

The maintenance window determines when pending operations start, but doesn't limit the total run time of these operations. Maintenance operations aren't guaranteed to finish before the maintenance window ends, and can continue beyond the specified end time. For more information, see [The Amazon RDS maintenance window \(p. 268\)](#).

Applying updates for a DB instance

With Amazon RDS, you can choose when to apply maintenance operations. You can decide when Amazon RDS applies updates by using the RDS console, AWS Command Line Interface (AWS CLI), or RDS API.

Console

To manage an update for a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the DB instance that has a required update.
4. For **Actions**, choose one of the following:
 - **Upgrade now**
 - **Upgrade at next window**

Note

If you choose **Upgrade at next window** and later want to delay the update, you can choose **Defer upgrade**. You can't defer a maintenance action if it has already started. To cancel a maintenance action, modify the DB instance and disable **Auto minor version upgrade**.

AWS CLI

To apply a pending update to a DB instance, use the [apply-pending-maintenance-action](#) AWS CLI command.

Example

For Linux, macOS, or Unix:

```
aws rds apply-pending-maintenance-action \
--resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db \
--apply-action system-update \
--opt-in-type immediate
```

For Windows:

```
aws rds apply-pending-maintenance-action ^
--resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db ^
--apply-action system-update ^
--opt-in-type immediate
```

Note

To defer a maintenance action, specify `undo-opt-in` for `--opt-in-type`. You can't specify `undo-opt-in` for `--opt-in-type` if the maintenance action has already started. To cancel a maintenance action, run the [modify-db-instance](#) AWS CLI command and specify `--no-auto-minor-version-upgrade`.

To return a list of resources that have at least one pending update, use the [describe-pending-maintenance-actions](#) AWS CLI command.

Example

For Linux, macOS, or Unix:

```
aws rds describe-pending-maintenance-actions \
--resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db
```

For Windows:

```
aws rds describe-pending-maintenance-actions ^
--resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db
```

You can also return a list of resources for a DB instance by specifying the `--filters` parameter of the `describe-pending-maintenance-actions` AWS CLI command. The format for the `--filters` command is `Name=filter-name,Value=resource-id,...`

The following are the accepted values for the `Name` parameter of a filter:

- `db-instance-id` – Accepts a list of DB instance identifiers or Amazon Resource Names (ARNs). The returned list only includes pending maintenance actions for the DB instances identified by these identifiers or ARNs.
- `db-cluster-id` – Accepts a list of DB cluster identifiers or ARNs for Amazon Aurora. The returned list only includes pending maintenance actions for the DB clusters identified by these identifiers or ARNs.

For example, the following example returns the pending maintenance actions for the `sample-instance1` and `sample-instance2` DB instances.

Example

For Linux, macOS, or Unix:

```
aws rds describe-pending-maintenance-actions \
--filters Name=db-instance-id,Values=sample-instance1,sample-instance2
```

For Windows:

```
aws rds describe-pending-maintenance-actions ^
--filters Name=db-instance-id,Values=sample-instance1,sample-instance2
```

RDS API

To apply an update to a DB instance, call the Amazon RDS API [ApplyPendingMaintenanceAction](#) operation.

To return a list of resources that have at least one pending update, call the Amazon RDS API [DescribePendingMaintenanceActions](#) operation.

Maintenance for Multi-AZ deployments

Running a DB instance as a Multi-AZ deployment can further reduce the impact of a maintenance event, because Amazon RDS applies operating system updates by following these steps:

1. Perform maintenance on the standby.
2. Promote the standby to primary.
3. Perform maintenance on the old primary, which becomes the new standby.

When you modify the database engine for your DB instance in a Multi-AZ deployment, then Amazon RDS upgrades both the primary and secondary DB instances at the same time. In this case, the database engine for the entire Multi-AZ deployment is shut down during the upgrade.

For more information on Multi-AZ deployments, see [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).

The Amazon RDS maintenance window

Every DB instance has a weekly maintenance window during which any system changes are applied. You can think of the maintenance window as an opportunity to control when modifications and software patching occur, in the event either are requested or required. If a maintenance event is scheduled for a given week, it is initiated during the 30-minute maintenance window you identify. Most maintenance events also complete during the 30-minute maintenance window, although larger maintenance events may take more than 30 minutes to complete.

The 30-minute maintenance window is selected at random from an 8-hour block of time per region. If you don't specify a preferred maintenance window when you create the DB instance, then Amazon RDS assigns a 30-minute maintenance window on a randomly selected day of the week.

RDS will consume some of the resources on your DB instance while maintenance is being applied. You might observe a minimal effect on performance. For a DB instance, on rare occasions, a Multi-AZ failover might be required for a maintenance update to complete.

Following, you can find the time blocks for each region from which default maintenance windows are assigned.

Region Name	Region	Time Block
US East (Ohio)	us-east-2	03:00–11:00 UTC
US East (N. Virginia)	us-east-1	03:00–11:00 UTC
US West (N. California)	us-west-1	06:00–14:00 UTC
US West (Oregon)	us-west-2	06:00–14:00 UTC
Africa (Cape Town)	af-south-1	03:00–11:00 UTC
Asia Pacific (Hong Kong)	ap-east-1	06:00–14:00 UTC
Asia Pacific (Mumbai)	ap-south-1	06:00–14:00 UTC
Asia Pacific (Osaka)	ap-northeast-3	22:00–23:59 UTC
Asia Pacific (Seoul)	ap-northeast-2	13:00–21:00 UTC
Asia Pacific (Singapore)	ap-southeast-1	14:00–22:00 UTC
Asia Pacific (Sydney)	ap-southeast-2	12:00–20:00 UTC
Asia Pacific (Tokyo)	ap-northeast-1	13:00–21:00 UTC
Canada (Central)	ca-central-1	03:00–11:00 UTC
China (Beijing)	cn-north-1	06:00–14:00 UTC
China (Ningxia)	cn-northwest-1	06:00–14:00 UTC
Europe (Frankfurt)	eu-central-1	21:00–05:00 UTC

Region Name	Region	Time Block
Europe (Ireland)	eu-west-1	22:00–06:00 UTC
Europe (London)	eu-west-2	22:00–06:00 UTC
Europe (Paris)	eu-west-3	23:59–07:29 UTC
Europe (Milan)	eu-south-1	02:00–10:00 UTC
Europe (Stockholm)	eu-north-1	23:00–07:00 UTC
Middle East (Bahrain)	me-south-1	06:00–14:00 UTC
South America (São Paulo)	sa-east-1	00:00–08:00 UTC
AWS GovCloud (US-East)	us-gov-east-1	17:00–01:00 UTC
AWS GovCloud (US-West)	us-gov-west-1	06:00–14:00 UTC

Adjusting the preferred DB instance maintenance window

The maintenance window should fall at the time of lowest usage and thus might need modification from time to time. Your DB instance will only be unavailable during this time if the system changes, such as a change in DB instance class, are being applied and require an outage, and only for the minimum amount of time required to make the necessary changes.

In the following example, you adjust the preferred maintenance window for a DB instance.

For the purpose of this example, we assume that the DB instance named *mydbinstance* exists and has a preferred maintenance window of "Sun:05:00–Sun:06:00" UTC.

Console

To adjust the preferred maintenance window

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then select the DB instance that you want to modify.
3. Choose **Modify**. The **Modify DB Instance** page appears.
4. In the **Maintenance** section, update the maintenance window.

Note

The maintenance window and the backup window for the DB instance cannot overlap. If you enter a value for the maintenance window that overlaps the backup window, an error message appears.

5. Choose **Continue**.

On the confirmation page, review your changes.

6. To apply the changes to the maintenance window immediately, select **Apply immediately**.
7. Choose **Modify DB Instance** to save your changes.

Alternatively, choose **Back** to edit your changes, or choose **Cancel** to cancel your changes.

AWS CLI

To adjust the preferred maintenance window, use the AWS CLI [modify-db-instance](#) command with the following parameters:

- `--db-instance-identifier`
- `--preferred-maintenance-window`

Example

The following code example sets the maintenance window to Tuesdays from 4:00-4:30AM UTC.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

RDS API

To adjust the preferred maintenance window, use the Amazon RDS API [ModifyDBInstance](#) operation with the following parameters:

- `DBInstanceIdentifier`
- `PreferredMaintenanceWindow`

Upgrading a DB instance engine version

Amazon RDS provides newer versions of each supported database engine so you can keep your DB instance up-to-date. Newer versions can include bug fixes, security enhancements, and other improvements for the database engine. When Amazon RDS supports a new version of a database engine, you can choose how and when to upgrade your database DB instances.

There are two kinds of upgrades: major version upgrades and minor version upgrades. In general, a *major engine version upgrade* can introduce changes that are not compatible with existing applications. In contrast, a *minor version upgrade* includes only changes that are backward-compatible with existing applications.

The version numbering sequence is specific to each database engine. For example, RDS for MySQL 5.7 and 8.0 are major engine versions and upgrading from any 5.7 version to any 8.0 version is a major version upgrade. RDS for MySQL version 5.7.22 and 5.7.23 are minor versions and upgrading from 5.7.22 to 5.7.23 is a minor version upgrade.

Important

You can't modify a DB instance when it is being upgraded. During an upgrade, the DB instance status is **upgrading**.

For more information about major and minor version upgrades for a specific DB engine, see the following documentation for your DB engine:

- [Upgrading the MariaDB DB engine \(p. 598\)](#)
- [Upgrading the Microsoft SQL Server DB engine \(p. 666\)](#)
- [Upgrading the MySQL DB engine \(p. 853\)](#)
- [Upgrading the Oracle DB engine \(p. 1209\)](#)
- [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#)

For major version upgrades, you must manually modify the DB engine version through the AWS Management Console, AWS CLI, or RDS API. For minor version upgrades, you can manually modify the engine version, or you can choose to enable auto minor version upgrades.

Topics

- [Manually upgrading the engine version \(p. 271\)](#)
- [Automatically upgrading the minor engine version \(p. 273\)](#)

Manually upgrading the engine version

To manually upgrade the engine version of a DB instance, you can use the AWS Management Console, the AWS CLI, or the RDS API.

Console

To upgrade the engine version of a DB instance by using the console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to upgrade.
3. Choose **Modify**. The **Modify DB Instance** page appears.

4. For **DB engine version**, choose the new version.
5. Choose **Continue** and check the summary of modifications.
6. To apply the changes immediately, choose **Apply immediately**. Choosing this option can cause an outage in some cases. For more information, see [Using the Apply Immediately setting \(p. 251\)](#).
7. On the confirmation page, review your changes. If they are correct, choose **Modify DB Instance** to save your changes.

Alternatively, choose **Back** to edit your changes, or choose **Cancel** to cancel your changes.

AWS CLI

To upgrade the engine version of a DB instance, use the CLI `modify-db-instance` command. Specify the following parameters:

- `--db-instance-identifier` – the name of the DB instance.
- `--engine-version` – the version number of the database engine to upgrade to.

For information about valid engine versions, use the AWS CLI `describe-db-engine-versions` command.

- `--allow-major-version-upgrade` – to upgrade the major version.
- `--no-apply-immediately` – to apply changes during the next maintenance window. To apply changes immediately, use `--apply-immediately`.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
  --db-instance-identifier mydbinstance \
  --engine-version new_version \
  --allow-major-version-upgrade \
  --no-apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --engine-version new_version ^
  --allow-major-version-upgrade ^
  --no-apply-immediately
```

RDS API

To upgrade the engine version of a DB instance, use the `ModifyDBInstance` action. Specify the following parameters:

- `DBInstanceIdentifier` – the name of the DB instance, for example `mydbinstance`.
- `EngineVersion` – the version number of the database engine to upgrade to. For information about valid engine versions, use the [DescribeDBEngineVersions](#) operation.
- `AllowMajorVersionUpgrade` – whether to allow a major version upgrade. To do so, set the value to `true`.
- `ApplyImmediately` – whether to apply changes immediately or during the next maintenance window. To apply changes immediately, set the value to `true`. To apply changes during the next maintenance window, set the value to `false`.

Automatically upgrading the minor engine version

A *minor engine version* is an update to a DB engine version within a major engine version. For example, a major engine version might be 9.6 with the minor engine versions 9.6.11 and 9.6.12 within it.

If you want Amazon RDS to upgrade the DB engine version of a database automatically, you can enable auto minor version upgrades for the database.

When Amazon RDS designates a minor engine version as the preferred minor engine version, each database that meets both of the following conditions is upgraded to the minor engine version automatically:

- The database is running a minor version of the DB engine that is lower than the preferred minor engine version.
- The database has auto minor version upgrade enabled.

You can control whether auto minor version upgrade is enabled for a DB instance when you perform the following tasks:

- [Creating a DB instance \(p. 141\)](#)
- [Modifying a DB instance \(p. 250\)](#)
- [Creating a read replica \(p. 283\)](#)
- [Restoring a DB instance from a snapshot \(p. 349\)](#)
- [Restoring a DB instance to a specific time \(p. 389\)](#)
- [Importing a DB instance from Amazon S3 \(p. 871\) \(for a MySQL backup on Amazon S3\)](#)

When you perform these tasks, you can control whether auto minor version upgrade is enabled for the DB instance in the following ways:

- Using the console, set the **Auto minor version upgrade** option.
- Using the AWS CLI, set the `--auto-minor-version-upgrade | --no-auto-minor-version-upgrade` option.
- Using the RDS API, set the `AutoMinorVersionUpgrade` parameter.

To determine whether a maintenance update, such as a DB engine version upgrade, is available for your DB instance, you can use the console, AWS CLI, or RDS API. You can also upgrade the DB engine version manually and adjust the maintenance window. For more information, see [Maintaining a DB instance \(p. 264\)](#).

Important

If you plan to migrate an RDS for PostgreSQL DB instance to an Aurora PostgreSQL DB cluster in the near future, we strongly recommend that you disable auto minor version upgrades for the DB instance early in the migration planning phase. Migration to Aurora PostgreSQL might be delayed if the RDS for PostgreSQL version isn't yet supported by Aurora PostgreSQL. For information about Aurora PostgreSQL versions, see [Engine versions for Amazon Aurora PostgreSQL](#).

Renaming a DB instance

You can rename a DB instance by using the AWS Management Console, the AWS CLI `modify-db-instance` command, or the Amazon RDS API `ModifyDBInstance` action. Renaming a DB instance can have far-reaching effects. The following is a list of considerations before you rename a DB instance.

- When you rename a DB instance, the endpoint for the DB instance changes, because the URL includes the name you assigned to the DB instance. You should always redirect traffic from the old URL to the new one.
- When you rename a DB instance, the old DNS name that was used by the DB instance is immediately deleted, although it could remain cached for a few minutes. The new DNS name for the renamed DB instance becomes effective in about 10 minutes. The renamed DB instance is not available until the new name becomes effective.
- You cannot use an existing DB instance name when renaming an instance.
- All read replicas associated with a DB instance remain associated with that instance after it is renamed. For example, suppose you have a DB instance that serves your production database and the instance has several associated read replicas. If you rename the DB instance and then replace it in the production environment with a DB snapshot, the DB instance that you renamed will still have the read replicas associated with it.
- Metrics and events associated with the name of a DB instance are maintained if you reuse a DB instance name. For example, if you promote a read replica and rename it to be the name of the previous primary DB instance, the events and metrics associated with the primary DB instance are associated with the renamed instance.
- DB instance tags remain with the DB instance, regardless of renaming.
- DB snapshots are retained for a renamed DB instance.

Note

A DB instance is an isolated database environment running in the cloud. A DB instance can host multiple databases, or a single Oracle database with multiple schemas. For information about changing a database name, see the documentation for your DB engine.

Renaming to replace an existing DB instance

The most common reasons for renaming a DB instance are that you are promoting a read replica or you are restoring data from a DB snapshot or point-in-time recovery (PITR). By renaming the database, you can replace the DB instance without having to change any application code that references the DB instance. In these cases, you would do the following:

1. Stop all traffic going to the primary DB instance. This can involve redirecting traffic from accessing the databases on the DB instance or some other way you want to use to prevent traffic from accessing your databases on the DB instance.
2. Rename the primary DB instance to a name that indicates it is no longer the primary DB instance as described later in this topic.
3. Create a new primary DB instance by restoring from a DB snapshot or by promoting a read replica, and then give the new instance the name of the previous primary DB instance.
4. Associate any read replicas with the new primary DB instance.

If you delete the old primary DB instance, you are responsible for deleting any unwanted DB snapshots of the old primary DB instance.

For information about promoting a read replica, see [Promoting a read replica to be a standalone DB instance \(p. 285\)](#).

Console

To rename a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the DB instance that you want to rename.
4. Choose **Modify**.
5. In **Settings**, enter a new name for **DB instance identifier**.
6. Choose **Continue**.
7. To apply the changes immediately, choose **Apply immediately**. Choosing this option can cause an outage in some cases. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
8. On the confirmation page, review your changes. If they are correct, choose **Modify DB Instance** to save your changes.

Alternatively, choose **Back** to edit your changes, or choose **Cancel** to cancel your changes.

AWS CLI

To rename a DB instance, use the AWS CLI command `modify-db-instance`. Provide the current `--db-instance-identifier` value and `--new-db-instance-identifier` parameter with the new name of the DB instance.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
  --db-instance-identifier DBInstanceIdentifier \
  --new-db-instance-identifier NewDBInstanceIdentifier
```

For Windows:

```
aws rds modify-db-instance ^
  --db-instance-identifier DBInstanceIdentifier ^
  --new-db-instance-identifier NewDBInstanceIdentifier
```

RDS API

To rename a DB instance, call Amazon RDS API operation `ModifyDBInstance` with the following parameters:

- `DBInstanceIdentifier` — existing name for the instance
- `NewDBInstanceIdentifier` — new name for the instance

Rebooting a DB instance

You might need to reboot your DB instance, usually for maintenance reasons. For example, if you make certain modifications, or if you change the DB parameter group associated with the DB instance, you must reboot the instance for the changes to take effect.

Note

If a DB instance isn't using the latest changes to its associated DB parameter group, the AWS Management Console shows the DB parameter group with a status of **pending-reboot**. The **pending-reboot** parameter groups status doesn't result in an automatic reboot during the next maintenance window. To apply the latest parameter changes to that DB instance, manually reboot the DB instance. For more information about parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

Rebooting a DB instance restarts the database engine service. Rebooting a DB instance results in a momentary outage, during which the DB instance status is set to *rebooting*.

If the Amazon RDS instance is configured for Multi-AZ, you can perform the reboot with a failover. An Amazon RDS event is created when the reboot is completed. If your DB instance is a Multi-AZ deployment, you can force a failover from one Availability Zone (AZ) to another when you reboot. When you force a failover of your DB instance, Amazon RDS automatically switches to a standby replica in another Availability Zone, and updates the DNS record for the DB instance to point to the standby DB instance. As a result, you need to clean up and re-establish any existing connections to your DB instance. Rebooting with failover is beneficial when you want to simulate a failure of a DB instance for testing, or restore operations to the original AZ after a failover occurs. For more information, see [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).

Note

When you force a failover from one Availability Zone to another when you reboot, the Availability Zone change might not be reflected in the AWS Management Console, and in calls to the AWS CLI and RDS API, for several minutes.

You can't reboot your DB instance if it is not in the available state. Your database can be unavailable for several reasons, such as an in-progress backup, a previously requested modification, or a maintenance-window action.

The time required to reboot your DB instance depends on the crash recovery process, database activity at the time of reboot, and the behavior of your specific DB engine. To improve the reboot time, we recommend that you reduce database activity as much as possible during the reboot process. Reducing database activity reduces rollback activity for in-transit transactions.

For a DB instance with read replicas, you can reboot the source DB instance and its read replicas independently. After a reboot completes, replication resumes automatically.

Console

To reboot a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to reboot.
3. For **Actions**, choose **Reboot**.

The **Reboot DB Instance** page appears.

4. (Optional) Choose **Reboot with failover?** to force a failover from one AZ to another.
5. Choose **Reboot** to reboot your DB instance.

Alternatively, choose **Cancel**.

AWS CLI

To reboot a DB instance by using the AWS CLI, call the [reboot-db-instance](#) command.

Example Simple reboot

For Linux, macOS, or Unix:

```
aws rds reboot-db-instance \
--db-instance-identifier mydbinstance
```

For Windows:

```
aws rds reboot-db-instance ^
--db-instance-identifier mydbinstance
```

Example Reboot with failover

To force a failover from one AZ to the other, use the `--force-failover` parameter.

For Linux, macOS, or Unix:

```
aws rds reboot-db-instance \
--db-instance-identifier mydbinstance \
--force-failover
```

For Windows:

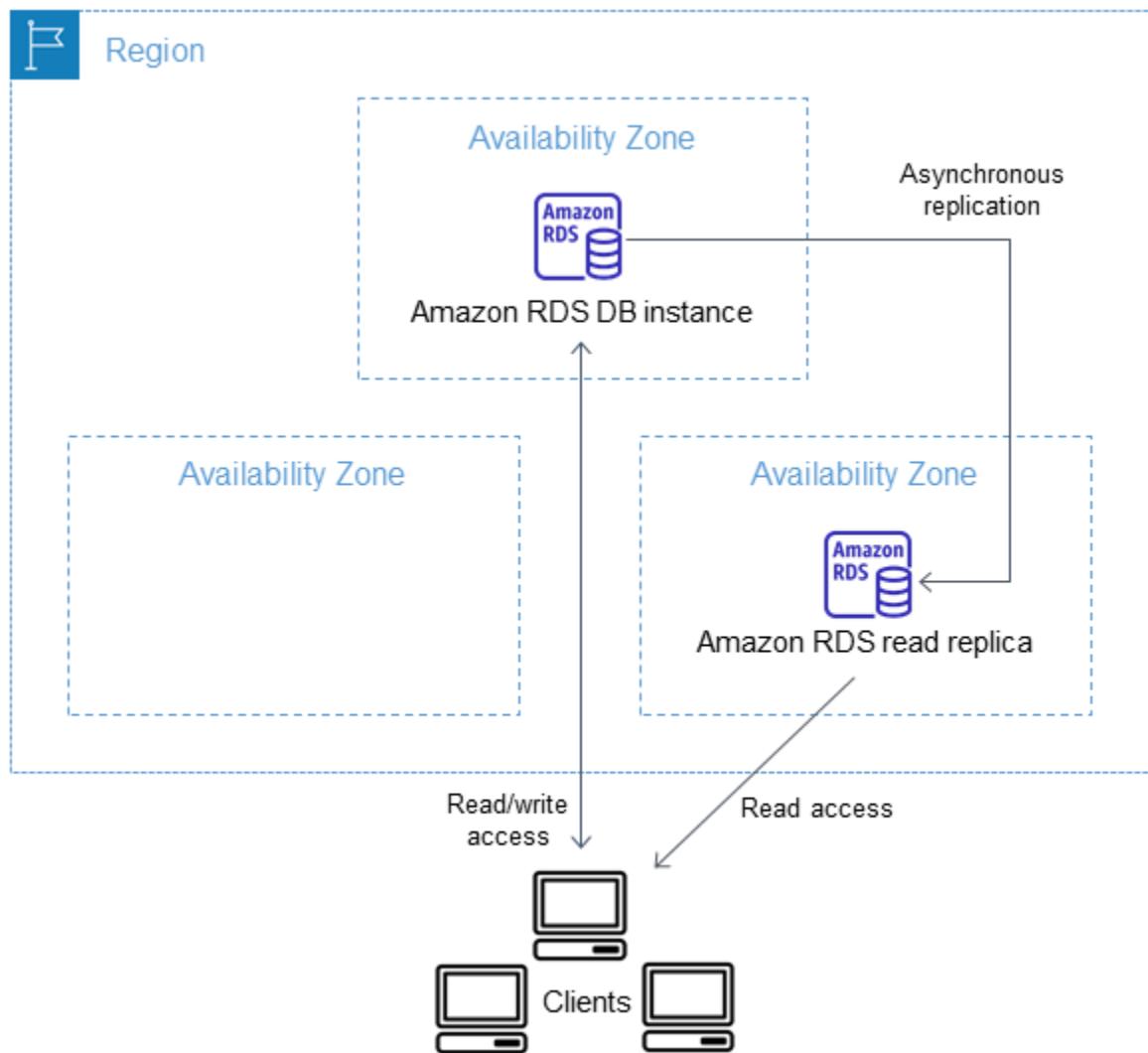
```
aws rds reboot-db-instance ^
--db-instance-identifier mydbinstance ^
--force-failover
```

RDS API

To reboot a DB instance by using the Amazon RDS API, call the [RebootDBInstance](#) operation.

Working with read replicas

Amazon RDS uses the MariaDB, Microsoft SQL Server, MySQL, Oracle, and PostgreSQL DB engines' built-in replication functionality to create a special type of DB instance called a read replica from a source DB instance. The source DB instance becomes the primary DB instance. Updates made to the primary DB instance are asynchronously copied to the read replica. You can reduce the load on your primary DB instance by routing read queries from your applications to the read replica. Using read replicas, you can elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads.



Note

The information following applies to creating Amazon RDS read replicas either in the same AWS Region as the source DB instance, or in a separate AWS Region. The information following doesn't apply to setting up replication with an instance that is running on an Amazon EC2 instance or that is on-premises.

When you create a read replica, you first specify an existing DB instance as the source. Then Amazon RDS takes a snapshot of the source instance and creates a read-only instance from the snapshot. Amazon RDS then uses the asynchronous replication method for the DB engine to update the read replica whenever there is a change to the primary DB instance. The read replica operates as a DB instance that allows only

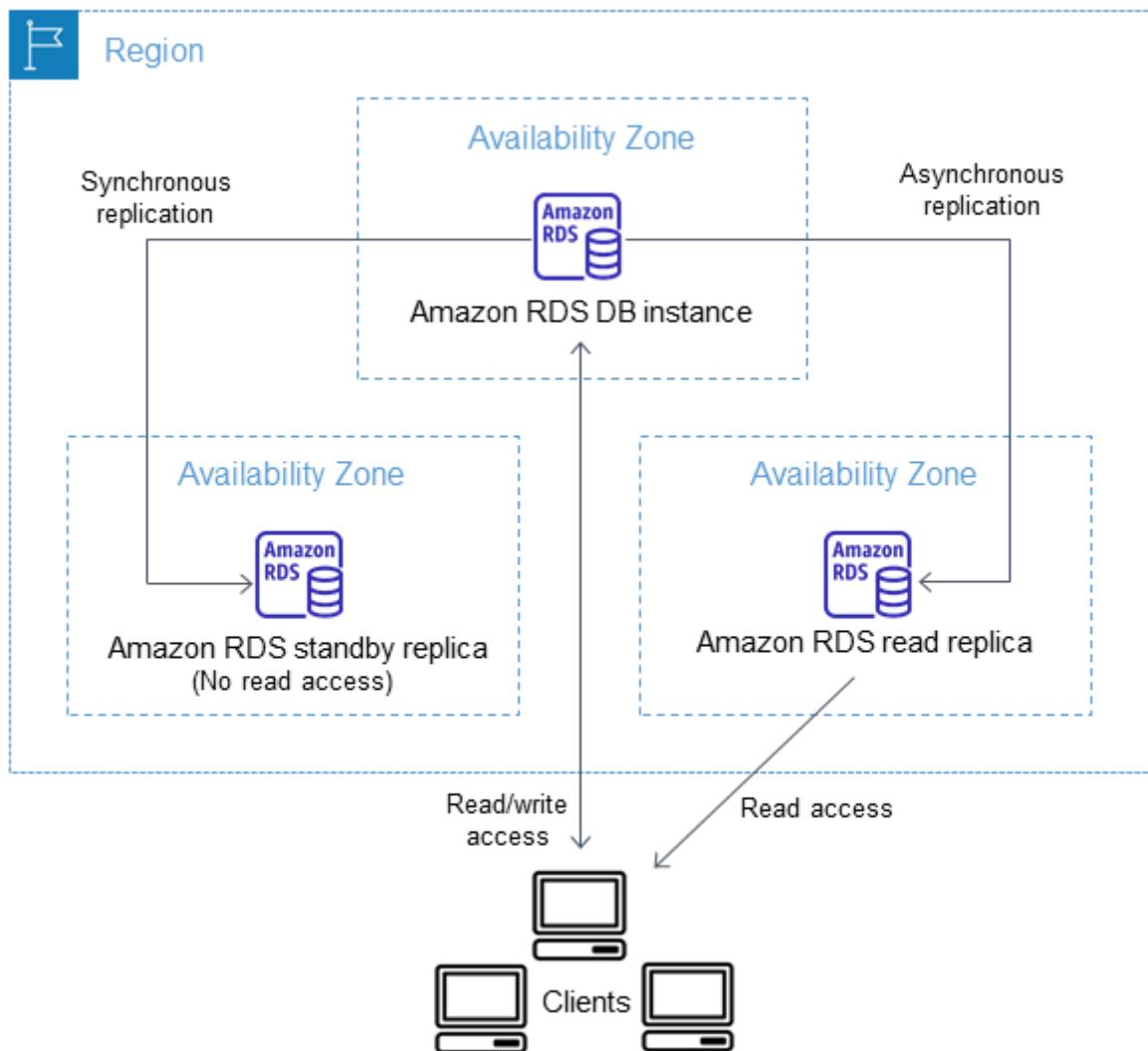
read-only connections. Applications connect to a read replica the same way they do to any DB instance. Amazon RDS replicates all databases in the source DB instance.

Note

The Oracle DB engine supports replica databases in mounted mode. A mounted replica doesn't accept user connections and so can't serve a read-only workload. The primary use for mounted replicas is cross-Region disaster recovery. For more information, see [Working with Oracle replicas for Amazon RDS \(p. 1119\)](#).

In some cases, a read replica resides in a different AWS Region from its primary DB instance. In these cases, Amazon RDS sets up a secure communications channel between the primary DB instance and the read replica. Amazon RDS establishes any AWS security configurations needed to enable the secure channel, such as adding security group entries. For more information about cross-Region read replicas, see [Creating a read replica in a different AWS Region \(p. 290\)](#).

You can configure a read replica for a DB instance that also has a standby replica configured for high availability. Replication with the standby replica is synchronous, and the standby replica can't serve read traffic.



For more information about high availability and standby replicas, see [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).

Read replicas are supported by the MariaDB, Microsoft SQL Server, MySQL, Oracle, and PostgreSQL DB engines. In this section, you can find general information about using read replicas with all of these engines. For information about using read replicas with a specific engine, see the following sections:

- [Working with MariaDB read replicas \(p. 605\)](#)
- [Working with read replicas for Microsoft SQL Server in Amazon RDS \(p. 696\)](#)
- [Working with MySQL read replicas \(p. 899\)](#)
- [Working with Oracle replicas for Amazon RDS \(p. 1119\)](#)
- [Working with PostgreSQL read replicas in Amazon RDS \(p. 1544\)](#)

Overview of Amazon RDS read replicas

Deploying one or more read replicas for a given source DB instance might make sense in a variety of scenarios, including the following:

- Scaling beyond the compute or I/O capacity of a single DB instance for read-heavy database workloads. You can direct this excess read traffic to one or more read replicas.
- Serving read traffic while the source DB instance is unavailable. In some cases, your source DB instance might not be able to take I/O requests, for example due to I/O suspension for backups or scheduled maintenance. In these cases, you can direct read traffic to your read replicas. For this use case, keep in mind that the data on the read replica might be "stale" because the source DB instance is unavailable.
- Business reporting or data warehousing scenarios where you might want business reporting queries to run against a read replica, rather than your production DB instance.
- Implementing disaster recovery. You can promote a read replica to a standalone instance as a disaster recovery solution if the primary DB instance fails.

By default, a read replica is created with the same storage type as the source DB instance. However, you can create a read replica that has a different storage type from the source DB instance based on the options listed in the following table.

Source DB instance storage type	Source DB instance storage allocation	Read replica storage type options
PIOPS	100 GiB–32 TiB	PIOPS, GP2, Standard
GP2	100 GiB–32 TiB	PIOPS, GP2, Standard
GP2	<100 GiB	GP2, Standard
Standard	100 GiB–6 TiB	PIOPS, GP2, Standard
Standard	<100 GiB	GP2, Standard

Note

When you increase the allocated storage of a read replica, it must be by at least 10 percent. If you try to increase the value by less than 10 percent, you get an error.

Amazon RDS doesn't support circular replication. You can't configure a DB instance to serve as a replication source for an existing DB instance. You can only create a new read replica from an existing DB instance. For example, if `MyDBInstance` replicates to `ReadReplica1`, you can't configure `ReadReplica1` to replicate back to `MyDBInstance`. For MariaDB and MySQL you can create a read replica from an existing read replica. For example, from `ReadReplica1`, you can create a new read

replica, such as [ReadReplica2](#). For Oracle, PostgreSQL, and SQL Server, you can't create a read replica from an existing read replica.

If you no longer need read replicas, you can explicitly delete them using the same mechanisms for deleting a DB instance. If you delete a source DB instance without deleting its read replicas in the same AWS Region, each read replica is promoted to a standalone DB instance. For information about deleting a DB instance, see [Deleting a DB instance \(p. 324\)](#). For information about read replica promotion, see [Promoting a read replica to be a standalone DB instance \(p. 285\)](#). If you have cross-Region read replicas, see [Cross-Region replication considerations \(p. 295\)](#) for considerations related to deleting the source for a cross-Region read replica.

Differences between read replicas for different DB engines

Because Amazon RDS DB engines implement replication differently, there are several significant differences you should know about, as shown in the following table.

Feature or behavior	MySQL and MariaDB	Oracle	PostgreSQL	SQL Server
What is the replication method?	Logical replication.	Physical replication.	Physical replication.	Physical replication.
How are transaction logs purged?	RDS for MySQL and RDS for MariaDB keep any binary logs that haven't been applied.	If a primary DB instance has no cross-Region read replicas, Amazon RDS for Oracle keeps a minimum of two hours of transaction logs on the source DB instance. Logs are purged from the source DB instance after two hours or after the archive log retention hours setting has passed, whichever is longer. Logs are purged from the read replica after the archive log retention hours setting has passed only if they have been successfully applied to the database. In some cases, a primary DB instance might have one or more cross-Region read replicas. If so, Amazon RDS for Oracle keeps the transaction logs on the source	PostgreSQL has the parameter <code>wal_keep_segments</code> that dictates how many write ahead log (WAL) files are kept to provide data to the read replicas. The parameter value specifies the number of logs to keep.	The Virtual Log File (VLF) of the transaction log file on the primary replica can be truncated after it is no longer required for the secondary replicas. The VLF can only be marked as inactive when the log records have been hardened in the replicas. Regardless of how fast the disk subsystems are in the primary replica, the transaction log will keep the VLFs until the slowest

Feature or behavior	MySQL and MariaDB	Oracle	PostgreSQL	SQL Server
		<p>DB instance until they have been transmitted and applied to all cross-Region read replicas.</p> <p>For information about setting archive log retention hours, see Retaining archived redo logs (p. 1067).</p>		replica has hardened it.
Can a replica be made writable?	Yes. You can enable the MySQL or MariaDB read replica to be writable.	No. An Oracle read replica is a physical copy, and Oracle doesn't allow for writes in a read replica. You can promote the read replica to make it writable. The promoted read replica has the replicated data to the point when the request was made to promote it.	No. A PostgreSQL read replica is a physical copy, and PostgreSQL doesn't allow for a read replica to be made writable.	No. A SQL Server read replica is a physical copy and also doesn't allow for writes. You can promote the read replica to make it writable. The promoted read replica has the replicated data up to the point when the request was made to promote it.
Can backups be performed on the replica?	Yes. You can enable automatic backups on a MySQL or MariaDB read replica.	No. You can't create manual snapshots of Amazon RDS for Oracle read replicas or enable automatic backups for them.	Yes, you can create a manual snapshot of a PostgreSQL read replica, but you can't enable automatic backups.	No. You can't create manual snapshots of Amazon RDS for SQL Server read replicas or enable automatic backups for them.

Feature or behavior	MySQL and MariaDB	Oracle	PostgreSQL	SQL Server
Can you use parallel replication?	Yes. MySQL version 5.6 and later and all supported MariaDB versions allow for parallel replication threads.	Yes. Redo log data is always transmitted in parallel from the primary database to all of its read replicas.	No. PostgreSQL has a single process handling replication.	Yes. Redo log data is always transmitted in parallel from the primary database to all of its read replicas.
Can you maintain a replica in a mounted rather than a read-only state?	No.	Yes. The primary use for mounted replicas is cross-Region disaster recovery. An Active Data Guard license isn't required for mounted replicas. For more information, see Working with Oracle replicas for Amazon RDS (p. 1119) .	No.	No.

Creating a read replica

You can create a read replica from an existing DB instance using the AWS Management Console, AWS CLI, or RDS API. You create a read replica by specifying `SourceDBInstanceIdentifier`, which is the DB instance identifier of the source DB instance that you want to replicate from.

When you create a read replica, Amazon RDS takes a DB snapshot of your source DB instance and begins replication. As a result, you experience a brief I/O suspension on your source DB instance while the DB snapshot occurs.

Note

The I/O suspension typically lasts about one minute. You can avoid the I/O suspension if the source DB instance is a Multi-AZ deployment, because in that case the snapshot is taken from the secondary DB instance.

An active, long-running transaction can slow the process of creating the read replica. We recommend that you wait for long-running transactions to complete before creating a read replica. If you create multiple read replicas in parallel from the same source DB instance, Amazon RDS takes only one snapshot at the start of the first create action.

When creating a read replica, there are a few things to consider. First, you must enable automatic backups on the source DB instance by setting the backup retention period to a value other than 0. This requirement also applies to a read replica that is the source DB instance for another read replica. For MySQL DB instances, automatic backups are supported only for read replicas running MySQL 5.6 and later, but not for MySQL versions 5.5. To enable automatic backups on an RDS for MySQL version 5.6 and later read replica, first create the read replica, then modify the read replica to enable automatic backups.

Note

Within an AWS Region, we strongly recommend that you create all read replicas in the same virtual private cloud (VPC) based on Amazon VPC as the source DB instance. If you create a read replica in a different VPC from the source DB instance, classless inter-domain routing (CIDR)

ranges can overlap between the replica and the RDS system. CIDR overlap makes the replica unstable, which can negatively impact applications connecting to it. If you receive an error when creating the read replica, choose a different destination DB subnet group. For more information, see [Working with a DB instance in a VPC \(p. 1727\)](#).

You can't create a read replica in a different AWS account from the source DB instance.

Console

To create a read replica from a source DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the DB instance that you want to use as the source for a read replica.
4. For **Actions**, choose **Create read replica**.
5. For **DB instance identifier**, enter a name for the read replica.
6. Choose your instance specifications. We recommend that you use the same DB instance class and storage type as the source DB instance for the read replica.
7. For **Multi-AZ deployment**, choose **Yes** to create a standby of your replica in another Availability Zone for failover support for the replica.

Note

Creating your read replica as a Multi-AZ DB instance is independent of whether the source database is a Multi-AZ DB instance.

8. To create an encrypted read replica:
 - a. Choose **Enable encryption**.
 - b. For **Master key**, choose the AWS Key Management Service (AWS KMS) key identifier of the customer master key (CMK).

Note

The source DB instance must be encrypted. To learn more about encrypting the source DB instance, see [Encrypting Amazon RDS resources \(p. 1630\)](#).

9. Choose other options, such as storage autoscaling.
10. Choose **Create read replica**.

AWS CLI

To create a read replica from a source DB instance, use the AWS CLI command `create-db-instance-read-replica`. This example also enables storage autoscaling.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-instance-read-replica \
--db-instance-identifier myreadreplica \
--source-db-instance-identifier mydbinstance \
--max-allocated-storage 1000
```

For Windows:

```
aws rds create-db-instance-read-replica ^
```

```
--db-instance-identifier myreadreplica ^
--source-db-instance-identifier mydbinstance ^
--max-allocated-storage 1000
```

RDS API

To create a read replica from a source MySQL, MariaDB, Oracle, PostgreSQL, or SQL Server DB instance, call the Amazon RDS API [CreateDBInstanceReadReplica](#) operation with the following required parameters:

- `DBInstanceIdentifier`
- `SourceDBInstanceIdentifier`

Promoting a read replica to be a standalone DB instance

You can promote a read replica into a standalone DB instance. When you promote a read replica, the DB instance is rebooted before it becomes available.



There are several reasons you might want to promote a read replica to a standalone DB instance:

- **Performing DDL operations (MySQL and MariaDB only)** – DDL operations, such as creating or rebuilding indexes, can take time and impose a significant performance penalty on your DB instance. You can perform these operations on a MySQL or MariaDB read replica once the read replica is in sync with its primary DB instance. Then you can promote the read replica and direct your applications to use the promoted instance.
- **Sharding** – Sharding embodies the "share-nothing" architecture and essentially involves breaking a large database into several smaller databases. One common way to split a database is splitting tables that are not joined in the same query onto different hosts. Another method is duplicating a table across multiple hosts and then using a hashing algorithm to determine which host receives a given update. You can create read replicas corresponding to each of your shards (smaller databases) and

promote them when you decide to convert them into standalone shards. You can then carve out the key space (if you are splitting rows) or distribution of tables for each of the shards depending on your requirements.

- **Implementing failure recovery** – You can use read replica promotion as a data recovery scheme if the primary DB instance fails. This approach complements synchronous replication, automatic failure detection, and failover.

If you are aware of the ramifications and limitations of asynchronous replication and you still want to use read replica promotion for data recovery, you can. To do this, first create a read replica and then monitor the primary DB instance for failures. In the event of a failure, do the following:

1. Promote the read replica.
2. Direct database traffic to the promoted DB instance.
3. Create a replacement read replica with the promoted DB instance as its source.

When you promote a read replica, the new DB instance that is created retains the option group and the parameter group of the former read replica. The promotion process can take several minutes or longer to complete, depending on the size of the read replica. After you promote the read replica to a new DB instance, it's just like any other DB instance. For example, you can create read replicas from the new DB instance and perform point-in-time restore operations. Because the promoted DB instance is no longer a read replica, you can't use it as a replication target. If a source DB instance has several read replicas, promoting one of the read replicas to a DB instance has no effect on the other replicas.

Backup duration is a function of the number of changes to the database since the previous backup. If you plan to promote a read replica to a standalone instance, we recommend that you enable backups and complete at least one backup prior to promotion. In addition, you can't promote a read replica to a standalone instance when it has the `Backing-up` status. If you have enabled backups on your read replica, configure the automated backup window so that daily backups don't interfere with read replica promotion.

The following steps show the general process for promoting a read replica to a DB instance:

1. Stop any transactions from being written to the primary DB instance, and then wait for all updates to be made to the read replica. Database updates occur on the read replica after they have occurred on the primary DB instance, and this replication lag can vary significantly. Use the [Replica Lag](#) metric to determine when all updates have been made to the read replica.
2. For MySQL and MariaDB only: If you need to make changes to the MySQL or MariaDB read replica, you must set the `read_only` parameter to `0` in the DB parameter group for the read replica. You can then perform all needed DDL operations, such as creating indexes, on the read replica. Actions taken on the read replica don't affect the performance of the primary DB instance.
3. Promote the read replica by using the **Promote** option on the Amazon RDS console, the AWS CLI command [promote-read-replica](#), or the [PromoteReadReplica](#) Amazon RDS API operation.

Note

The promotion process takes a few minutes to complete. When you promote a read replica, replication is stopped and the read replica is rebooted. When the reboot is complete, the read replica is available as a new DB instance.

4. (Optional) Modify the new DB instance to be a Multi-AZ deployment. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#) and [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).

Console

To promote a read replica to a standalone DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the Amazon RDS console, choose **Databases**.
The **Databases** pane appears. Each read replica shows **Replica** in the **Role** column.
3. Choose the read replica that you want to promote.
4. For **Actions**, choose **Promote**.
5. On the **Promote Read Replica** page, enter the backup retention period and the backup window for the newly promoted DB instance.
6. When the settings are as you want them, choose **Continue**.
7. On the acknowledgment page, choose **Promote Read Replica**.

AWS CLI

To promote a read replica to a standalone DB instance, use the AWS CLI [promote-read-replica](#) command.

Example

For Linux, macOS, or Unix:

```
aws rds promote-read-replica \
--db-instance-identifier myreadreplica
```

For Windows:

```
aws rds promote-read-replica ^
--db-instance-identifier myreadreplica
```

RDS API

To promote a read replica to a standalone DB instance, call the Amazon RDS API [PromoteReadReplica](#) operation with the required parameter **DBInstanceIdentifier**.

Monitoring read replication

You can monitor the status of a read replica in several ways. The Amazon RDS console shows the status of a read replica in the **Availability and durability** section of the read replica details. To view the details for a read replica, choose the name of the read replica in the list of instances in the Amazon RDS console.

Availability and durability

DB instance status

available

Replication state

replicating

Replication error

-

You can also see the status of a read replica using the AWS CLI `describe-db-instances` command or the Amazon RDS API `DescribeDBInstances` operation.

The status of a read replica can be one of the following:

- **replicating** – The read replica is replicating successfully.
- **replication degraded (SQL Server only)** – Replicas are receiving data from the primary instance, but one or more databases might be not getting updates. This can occur, for example, when a replica is in the process of setting up newly created databases.

The status doesn't transition from `replication degraded` to `error`, unless an error occurs during the degraded state.

- **error** – An error has occurred with the replication. Check the **Replication Error** field in the Amazon RDS console or the event log to determine the exact error. For more information about troubleshooting a replication error, see [Troubleshooting a MySQL read replica problem \(p. 908\)](#).
- **terminated (MariaDB, MySQL, or PostgreSQL only)** – Replication is terminated. This occurs if replication is stopped for more than 30 consecutive days, either manually or due to a replication error. In this case, Amazon RDS terminates replication between the primary DB instance and all read replicas. Amazon RDS does this to prevent increased storage requirements on the source DB instance and long failover times.

Broken replication can affect storage because the logs can grow in size and number due to the high volume of errors messages being written to the log. Broken replication can also affect failure recovery due to the time Amazon RDS requires to maintain and process the large number of logs during recovery.

- **stopped (MariaDB or MySQL only)** – Replication has stopped because of a customer-initiated request.
- **replication stop point set (MySQL only)** – A customer-initiated stop point was set using the [mysql.rds_start_replication_until \(p. 965\)](#) stored procedure and the replication is in progress.
- **replication stop point reached (MySQL only)** – A customer-initiated stop point was set using the [mysql.rds_start_replication_until \(p. 965\)](#) stored procedure and replication is stopped because the stop point was reached.

Monitoring replication lag

You can monitor replication lag in Amazon CloudWatch by viewing the `Amazon RDS ReplicaLag` metric.

For MySQL and MariaDB, the `ReplicaLag` metric reports the value of the `Seconds_Behind_Master` field of the `SHOW SLAVE STATUS` command. Common causes for replication lag for MySQL and MariaDB are the following:

- A network outage.
- Writing to tables with indexes on a read replica. If the `read_only` parameter is not set to 0 on the read replica, it can break replication.
- Using a nontransactional storage engine such as MyISAM. Replication is only supported for the InnoDB storage engine on MySQL and the XtraDB storage engine on MariaDB.

When the `ReplicaLag` metric reaches 0, the replica has caught up to the primary DB instance. If the `ReplicaLag` metric returns -1, then replication is currently not active. `ReplicaLag = -1` is equivalent to `Seconds_Behind_Master = NULL`.

For Oracle, the `ReplicaLag` metric is the sum of the `Apply_Lag` value and the difference between the current time and the apply lag's `DATUM_TIME` value. The `DATUM_TIME` value is the last time the read replica received data from its source DB instance. For more information, see [V\\$DATAGUARD_STATS](#) in the Oracle documentation.

For SQL Server, the `ReplicaLag` metric is the maximum lag of databases that have fallen behind, in seconds. For example, if you have two databases that lag 5 seconds and 10 seconds, respectively, then `ReplicaLag` is 10 seconds. The `ReplicaLag` metric returns the value of the following query.

```
select ag.name name, MAX(hdrs.secondary_lag_seconds) max_lag from
sys.dm_hadr_database_replica_state
```

For more information, see [secondary_lag_seconds](#) in the Microsoft documentation.

`ReplicaLag` returns -1 if RDS can't determine the lag, such as during replica setup, or when the read replica is in the `error` state.

Note

New databases aren't included in the lag calculation until they are accessible on the read replica.

For PostgreSQL, the `ReplicaLag` metric returns the value of the following query.

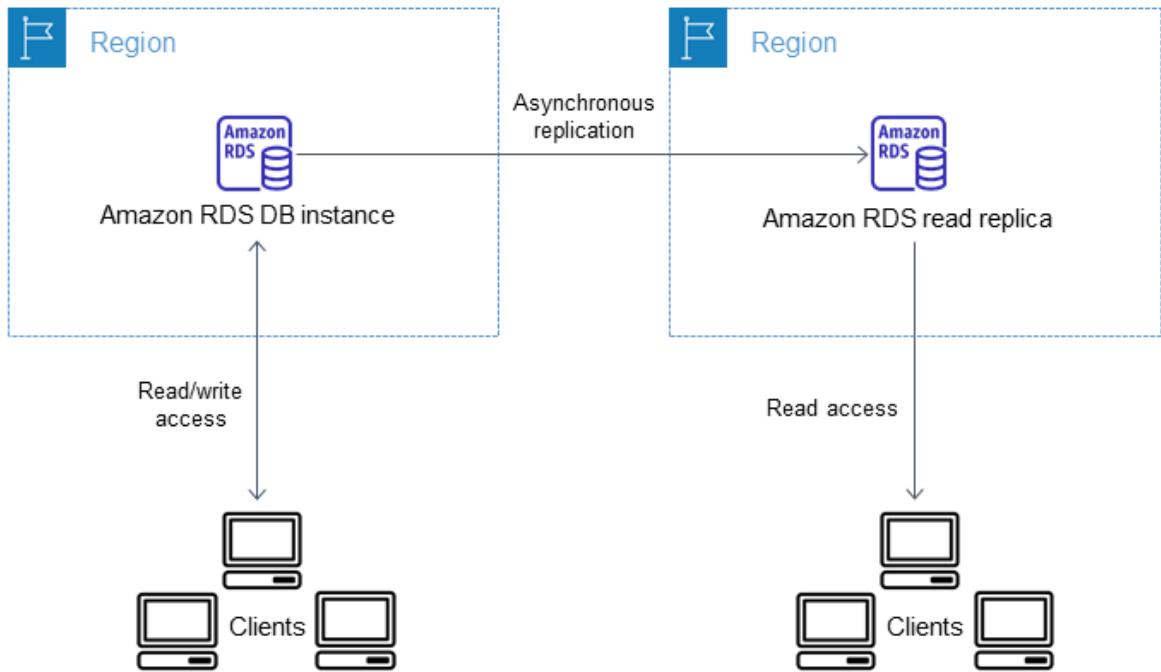
```
SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS reader_lag
```

PostgreSQL versions 9.5.2 and later use physical replication slots to manage write ahead log (WAL) retention on the source instance. For each cross-Region read replica instance, Amazon RDS creates a physical replication slot and associates it with the instance. Two Amazon CloudWatch metrics, `Oldest Replication Slot Lag` and `Transaction Logs Disk Usage`, show how far behind the most lagging replica is in terms of WAL data received and how much storage is being used for WAL data. The `Transaction Logs Disk Usage` value can substantially increase when a cross-Region read replica is lagging significantly.

For more information about monitoring a DB instance with CloudWatch, see [Monitoring Amazon RDS metrics with Amazon CloudWatch \(p. 540\)](#).

Creating a read replica in a different AWS Region

With Amazon RDS, you can create a MariaDB, MySQL, Oracle, or PostgreSQL read replica in a different AWS Region from the source DB instance. Creating a cross-Region read replica isn't supported for SQL Server on Amazon RDS.



You create a read replica in a different AWS Region to do the following:

- Improve your disaster recovery capabilities.
- Scale read operations into an AWS Region closer to your users.
- Make it easier to migrate from a data center in one AWS Region to a data center in another AWS Region.

Creating a read replica in a different AWS Region from the source instance is similar to creating a replica in the same AWS Region. You can use the AWS Management Console, run the [create-db-instance-read-replica](#) command, or call the [CreateDBInstanceReadReplica](#) API operation.

Note

To create an encrypted read replica in a different AWS Region from the source DB instance, the source DB instance must be encrypted.

Creating a cross-Region read replica

The following procedures show how to create a read replica from a source MariaDB, MySQL, Oracle, or PostgreSQL DB instance in a different AWS Region.

Console

You can create a read replica across AWS Regions using the AWS Management Console.

To create a read replica across AWS Regions with the console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the MariaDB, MySQL, Oracle, or PostgreSQL DB instance that you want to use as the source for a read replica.
4. For **Actions**, choose **Create read replica**.

5. For **DB instance identifier**, enter a name for the read replica.
6. Choose the **Destination Region**.
7. Choose the instance specifications you want to use. We recommend that you use the same DB instance class and storage type for the read replica.
8. To create an encrypted read replica in another AWS Region:
 - a. Choose **Enable encryption**.
 - b. For **Master key**, choose the AWS Key Management Service (AWS KMS) key identifier of the customer master key (CMK) of the destination AWS Region.

Note

To create an encrypted read replica, the source DB instance must be encrypted. To learn more about encrypting the source DB instance, see [Encrypting Amazon RDS resources \(p. 1630\)](#).

9. Choose other options, such as storage autoscaling.
10. Choose **Create read replica**.

AWS CLI

To create a read replica from a source MySQL, MariaDB, Oracle, or PostgreSQL DB instance in a different AWS Region, you can use the `create-db-instance-read-replica` command. In this case, you use `create-db-instance-read-replica` from the AWS Region where you want the read replica (destination Region) and specify the Amazon Resource Name (ARN) for the source DB instance. An ARN uniquely identifies a resource created in Amazon Web Services.

For example, if your source DB instance is in the US East (N. Virginia) Region, the ARN looks similar to this example:

```
arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

For information about ARNs, see [Working with Amazon Resource Names \(ARNs\) in Amazon RDS \(p. 309\)](#).

To create a read replica in a different AWS Region from the source DB instance, you can use the AWS CLI `create-db-instance-read-replica` command from the destination AWS Region. The following parameters are required for creating a read replica in another AWS Region:

- `--region` – The destination AWS Region where the read replica is created.
- `--source-db-instance-identifier` – The DB instance identifier for the source DB instance. This identifier must be in the ARN format for the source AWS Region. The AWS Region specified in `source-db-instance-identifier` must match the AWS Region specified in `--region`.
- `--db-instance-identifier` – The identifier for the read replica in the destination AWS Region.

Example of a cross-Region read replica

The following code creates a read replica in the US West (Oregon) Region from a source DB instance in the US East (N. Virginia) Region.

For Linux, macOS, or Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --region us-west-2 \
```

```
--source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

For Windows:

```
aws rds create-db-instance-read-replica ^
--db-instance-identifier myreadreplica ^
--region us-west-2 ^
--source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

The following parameters are also required for creating an encrypted read replica in another AWS Region:

- **--source-region** – The AWS Region of the source DB instance.
If **--source-region** isn't specified, you must specify a **--pre-signed-url** value. A presigned URL is a URL that contains a Signature Version 4 signed request for the `CreateDBInstanceReadReplica` operation that is called in the source AWS Region. For more information about presigned URLs, see [CreateDBInstanceReadReplica](#).
- **--kms-key-id** – The AWS KMS key identifier for the customer master key (CMK) to use to encrypt the read replica in the destination AWS Region.

Example of an encrypted cross-Region read replica

The following code creates an encrypted read replica in the US West (Oregon) Region from a source DB instance in the US East (N. Virginia) Region.

For Linux, macOS, or Unix:

```
aws rds create-db-instance-read-replica \
--db-instance-identifier myreadreplica \
--region us-west-2 \
--source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance \
--source-region us-east-1 \
--kms-key-id my-us-west-2-key
```

For Windows:

```
aws rds create-db-instance-read-replica ^
--db-instance-identifier myreadreplica ^
--region us-west-2 ^
--source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance ^
--source-region us-east-1 ^
--kms-key-id my-us-west-2-key
```

RDS API

To create a read replica from a source MySQL, MariaDB, Oracle, or PostgreSQL DB instance in a different AWS Region, you can call the Amazon RDS API function [CreateDBInstanceReadReplica](#). In this case, you call `CreateDBInstanceReadReplica` from the AWS Region where you want the read replica (destination Region) and specify the Amazon Resource Name (ARN) for the source DB instance. An ARN uniquely identifies a resource created in Amazon Web Services.

To create an encrypted read replica in a different AWS Region from the source DB instance, you can use the Amazon RDS API [CreateDBInstanceReadReplica](#) operation from the destination AWS Region. To create an encrypted read replica in another AWS Region, you must specify a value for `PreSignedURL`.

`PreSignedURL` should contain a request for the [CreateDBInstanceReadReplica](#) operation to call in the source AWS Region where the read replica is created in. To learn more about `PreSignedUrl`, see [CreateDBInstanceReadReplica](#).

For example, if your source DB instance is in the US East (N. Virginia) Region, the ARN looks similar to the following.

```
arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

For information about ARNs, see [Working with Amazon Resource Names \(ARNs\) in Amazon RDS \(p. 309\)](#).

Example

```
https://us-west-2.rds.amazonaws.com/
?Action=CreateDBInstanceReadReplica
&KmsKeyId=my-us-east-1-key
&PreSignedUrl=https%253A%252Frds.us-west-2.amazonaws.com%252F
%253FAction%253DCreateDBInstanceReadReplica
%2526DestinationRegion%253Dus-east-1
%2526KmsKeyId%253Dmy-us-east-1-key
%2526SourceDBInstanceIdentifier%253Darn%25253Aaws%25253Ards%25253Aus-
west-2%23456789012%25253Adb%25253Amydbinstance
%2526SignatureMethod%253DHmacSHA256
%2526SignatureVersion%253D4%2526SourceDBInstanceIdentifier%253Darn%25253Aaws
%25253Ards%25253Aus-west-2%25253A123456789012%25253Ainstance%25253Amydbinstance
%2526Version%253D2014-10-31
%2526X-Amz-Algorithm%253DAWS4-HMAC-SHA256
%2526X-Amz-Credential%253DAKIADQKE4SARGYLE%252F20161117%252Fus-west-2%252Frds
%252Faws4_request
%2526X-Amz-Date%253D20161117T215409Z
%2526X-Amz-Expires%253D3600
%2526X-Amz-SignedHeaders%253Dcontent-type%253Bhost%253Buser-agent%253Bx-amz-
content-sha256%253Bx-amz-date
%2526X-Amz-Signature
%253D255a0f17b4e717d3b67fad163c3ec26573b882c03a65523522cf890a67fca613
&DBInstanceIdentifier=myreadreplica
&SourceDBInstanceIdentifier=arn:aws:rds:us-east-1:123456789012:db:mydbinstance
&Version=2012-01-15
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2012-01-20T22%3A06%3A23.624Z
&AWSAccessKeyId=<AWS Access Key ID>
&Signature=<Signature>
```

How Amazon RDS does cross-Region replication

Amazon RDS uses the following process to create a cross-Region read replica. Depending on the AWS Regions involved and the amount of data in the databases, this process can take hours to complete. You can use this information to determine how far the process has proceeded when you create a cross-Region read replica:

1. Amazon RDS begins configuring the source DB instance as a replication source and sets the status to *modifying*.
2. Amazon RDS begins setting up the specified read replica in the destination AWS Region and sets the status to *creating*.
3. Amazon RDS creates an automated DB snapshot of the source DB instance in the source AWS Region. The format of the DB snapshot name is `rds:<InstanceID>-<timestamp>`, where `<InstanceID>` is the identifier of the source instance, and `<timestamp>` is the date and time the copy started.

For example, `rds:mysourceinstance-2013-11-14-09-24` was created from the instance `mysourceinstance` at 2013-11-14-09-24. During the creation of an automated DB snapshot, the source DB instance status remains *modifying*, the read replica status remains *creating*, and the DB snapshot status is *creating*. The progress column of the DB snapshot page in the console reports how far the DB snapshot creation has progressed. When the DB snapshot is complete, the status of both the DB snapshot and source DB instance are set to *available*.

4. Amazon RDS begins a cross-Region snapshot copy for the initial data transfer. The snapshot copy is listed as an automated snapshot in the destination AWS Region with a status of *creating*. It has the same name as the source DB snapshot. The progress column of the DB snapshot display indicates how far the copy has progressed. When the copy is complete, the status of the DB snapshot copy is set to *available*.
5. Amazon RDS then uses the copied DB snapshot for the initial data load on the read replica. During this phase, the read replica is in the list of DB instances in the destination, with a status of *creating*. When the load is complete, the read replica status is set to *available*, and the DB snapshot copy is deleted.
6. When the read replica reaches the available status, Amazon RDS starts by replicating the changes made to the source instance since the start of the create read replica operation. During this phase, the replication lag time for the read replica is greater than 0.

For information about replication lag time, see [Monitoring read replication \(p. 288\)](#).

Cross-Region replication considerations

All of the considerations for performing replication within an AWS Region apply to cross-Region replication. The following extra considerations apply when replicating between AWS Regions:

- You can only replicate between AWS Regions when using the following Amazon RDS DB instances:
 - MariaDB (all versions).
 - MySQL version 5.6 and later.
 - Oracle Enterprise Edition (EE) engine version 12.1.0.2.v10 and higher 12.1 versions, and all versions of 12.2, 18c, and 19c.

An Active Data Guard license is required. For information about limitations for Oracle cross-Region read replicas, see [Replica requirements for Oracle \(p. 1119\)](#).

- PostgreSQL (all versions).
- A source DB instance can have cross-Region read replicas in multiple AWS Regions.
- You can only create a cross-Region Amazon RDS read replica from a source Amazon RDS DB instance that is not a read replica of another Amazon RDS DB instance.
- You can replicate between the AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions, but not into or out of AWS GovCloud (US).
- You can expect to see a higher level of lag time for any read replica that is in a different AWS Region than the source instance. This lag time comes from the longer network channels between regional data centers.
- For cross-Region read replicas, any of the create read replica commands that specify the `--db-subnet-group-name` parameter must specify a DB subnet group from the same VPC.
- You can create a cross-Region read replica:
 - In a VPC from a source DB instance that is in a VPC in another AWS Region
 - In a VPC from a source DB instance that isn't in a VPC
 - That isn't in a VPC from a source DB instance that is in a VPC
- Due to the limit on the number of access control list (ACL) entries for a VPC, we can't guarantee more than five cross-Region read replica instances.
- The read replica uses the default DB parameter group for the specified DB engine.
- The read replica uses the default security group.

- For MariaDB, MySQL, and Oracle DB instances, when the source for a cross-Region read replica is deleted, the read replica is promoted.
- For PostgreSQL DB instances, when the source for a cross-Region read replica is deleted, the replication status of the read replica is set to terminated. The read replica isn't promoted.

Requesting a cross-Region read replica

To communicate with the source Region to request the creation of a cross-Region read replica, the requester (IAM role or IAM user) must have access to the source DB instance and the source Region.

Certain conditions in the requester's IAM policy can cause the request to fail. The following examples assume that the source DB instance is in US East (Ohio) and the read replica is created in US East (N. Virginia). These examples show conditions in the requester's IAM policy that cause the request to fail:

- The requester's policy has a condition for `aws:RequestedRegion`.

```

...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": "us-east-1"
    }
}

```

The request fails because the policy doesn't allow access to the source Region. For a successful request, specify both the source and destination Regions.

```

...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": [
            "us-east-1",
            "us-east-2"
        ]
    }
}

```

- The requester's policy doesn't allow access to the source DB instance.

```

...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "arn:aws:rds:us-east-1:123456789012:db:myreadreplica"
...
```

For a successful request, specify both the source instance and the replica.

```

...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": [
    "arn:aws:rds:us-east-1:123456789012:db:myreadreplica",
    "arn:aws:rds:us-east-2:123456789012:db:mydbinstance"
]

```

...

- The requester's policy denies aws:ViaAWSservice.

```

...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "Bool": {"aws:ViaAWSservice": "false"}
}

```

Communication with the source Region is made by RDS on the requester's behalf. For a successful request, don't deny calls made by AWS services.

- The requester's policy has a condition for aws:SourceVpc or aws:SourceVpce.

These requests might fail because when RDS makes the call to the remote Region, it isn't from the specified VPC or VPC endpoint.

If you need to use one of the previous conditions that would cause a request to fail, you can include a second statement with aws:CalledVia in your policy to make the request succeed. For example, you can use aws:CalledVia with aws:SourceVpce as shown here:

```

...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "Condition" : {
        "ForAnyValue:StringEquals" : {
            "aws:SourceVpce": "vpce-1a2b3c4d"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "rds:CreateDBInstanceReadReplica"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": [
                "rds.amazonaws.com"
            ]
        }
    }
}

```

For more information, see [Policies and permissions in IAM](#) in the *IAM User Guide*.

Authorizing the read replica

After a cross-Region DB read replica creation request returns success, RDS starts the replica creation in the background. An authorization for RDS to access the source DB instance is created. This authorization links the source DB instance to the read replica, and allows RDS to copy only to the specified read replica.

The authorization is verified by RDS using the rds:CrossRegionCommunication permission in the service-linked IAM role. If the replica is authorized, RDS communicates with the source Region and completes the replica creation.

RDS doesn't have access to DB instances that weren't authorized previously by a `CreateDBInstanceReadReplica` request. The authorization is revoked when read replica creation completes.

RDS uses the service-linked role to verify the authorization in the source Region. If you delete the service-linked role during the replication creation process, the creation fails.

For more information, see [Using service-linked roles](#) in the *IAM User Guide*.

Using AWS Security Token Service credentials

Session tokens from the global AWS Security Token Service (AWS STS) endpoint are valid only in AWS Regions that are enabled by default (commercial Regions). If you use credentials from the `assumeRole` API operation in AWS STS, use the regional endpoint if the source Region is an opt-in Region. Otherwise, the request fails. This happens because your credentials must be valid in both Regions, which is true for opt-in Regions only when the regional AWS STS endpoint is used.

To use the global endpoint, make sure that it's enabled for both Regions in the operations. Set the global endpoint to `Valid in all AWS Regions` in the AWS STS account settings.

The same rule applies to credentials in the presigned URL parameter.

For more information, see [Managing AWS STS in an AWS Region](#) in the *IAM User Guide*.

Cross-Region replication costs

The data transferred for cross-Region replication incurs Amazon RDS data transfer charges. These cross-Region replication actions generate charges for the data transferred out of the source AWS Region:

- When you create a read replica, Amazon RDS takes a snapshot of the source instance and transfers the snapshot to the read replica AWS Region.
- For each data modification made in the source databases, Amazon RDS transfers data from the source AWS Region to the read replica AWS Region.

For more information about data transfer pricing, see [Amazon RDS pricing](#).

For MySQL and MariaDB instances, you can reduce your data transfer costs by reducing the number of cross-Region read replicas that you create. For example, suppose that you have a source DB instance in one AWS Region and want to have three read replicas in another AWS Region. In this case, you create only one of the read replicas from the source DB instance. You create the other two replicas from the first read replica instead of the source DB instance.

For example, if you have `source-instance-1` in one AWS Region, you can do the following:

- Create `read-replica-1` in the new AWS Region, specifying `source-instance-1` as the source.
- Create `read-replica-2` from `read-replica-1`.
- Create `read-replica-3` from `read-replica-1`.

In this example, you are only charged for the data transferred from `source-instance-1` to `read-replica-1`. You aren't charged for the data transferred from `read-replica-1` to the other two replicas because they are all in the same AWS Region. If you create all three replicas directly from `source-instance-1`, you are charged for the data transfers to all three replicas.

Tagging Amazon RDS resources

You can use Amazon RDS tags to add metadata to your Amazon RDS resources. You can use the tags to add your own notations about database instances, snapshots, Aurora clusters, and so on. Doing so can help you to document your Amazon RDS resources. You can also use the tags with automated maintenance procedures.

In particular, you can use these tags with IAM policies to manage access to Amazon RDS resources and to control what actions can be applied to the Amazon RDS resources. You can also use these tags to track costs by grouping expenses for similarly tagged resources.

You can tag the following Amazon RDS resources:

- DB instances
- DB clusters
- Read replicas
- DB snapshots
- DB cluster snapshots
- Reserved DB instances
- Event subscriptions
- DB option groups
- DB parameter groups
- DB cluster parameter groups
- DB security groups
- DB subnet groups

Topics

- [Overview of Amazon RDS resource tags \(p. 299\)](#)
- [Using tags for access control with IAM \(p. 300\)](#)
- [Using tags to produce detailed billing reports \(p. 300\)](#)
- [Adding, listing, and removing tags \(p. 300\)](#)
- [Using the AWS Tag Editor \(p. 303\)](#)
- [Copying tags to DB instance snapshots \(p. 303\)](#)
- [Tutorial: Use tags to specify which DB instances to stop \(p. 304\)](#)
- [Using tags to enable backups in AWS Backup \(p. 306\)](#)

Overview of Amazon RDS resource tags

An Amazon RDS tag is a name-value pair that you define and associate with an Amazon RDS resource. The name is referred to as the key. Supplying a value for the key is optional. You can use tags to assign arbitrary information to an Amazon RDS resource. You can use a tag key, for example, to define a category, and the tag value might be an item in that category. For example, you might define a tag key of "project" and a tag value of "Salix", indicating that the Amazon RDS resource is assigned to the Salix project. You can also use tags to designate Amazon RDS resources as being used for test or production by using a key such as `environment=test` or `environment=production`. We recommend that you use a consistent set of tag keys to make it easier to track metadata associated with Amazon RDS resources.

Each Amazon RDS resource has a tag set, which contains all the tags that are assigned to that Amazon RDS resource. A tag set can contain as many as 50 tags, or it can be empty. If you add a tag to an Amazon

RDS resource that has the same key as an existing tag on resource, the new value overwrites the old value.

AWS does not apply any semantic meaning to your tags; tags are interpreted strictly as character strings. Amazon RDS can set tags on a DB instance or other Amazon RDS resources, depending on the settings that you use when you create the resource. For example, Amazon RDS might add a tag indicating that a DB instance is for production or for testing.

- The tag key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with "aws:" or "rds:". The string can contain only the set of Unicode letters, digits, white-space, '_', ':', ';', '/', '=', '+', '-', '@' (Java regex: "`^([\u00p{L}\u00p{Z}\u00p{N}_:=+\u00p{-@\u00p{J}])$`").
- The tag value is an optional string value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with "aws:". The string can contain only the set of Unicode letters, digits, white-space, '_', ':', ';', '/', '=', '+', '-', '@' (Java regex: "`^([\u00p{L}\u00p{Z}\u00p{N}_:=+\u00p{-@\u00p{J}])$`").

Values do not have to be unique in a tag set and can be null. For example, you can have a key-value pair in a tag set of `project=Trinity` and `cost-center=Trinity`.

You can use the AWS Management Console, the command line interface, or the Amazon RDS API to add, list, and delete tags on Amazon RDS resources. When using the command line interface or the Amazon RDS API, you must provide the Amazon Resource Name (ARN) for the Amazon RDS resource you want to work with. For more information about constructing an ARN, see [Constructing an ARN for Amazon RDS \(p. 309\)](#).

Tags are cached for authorization purposes. Because of this, additions and updates to tags on Amazon RDS resources can take several minutes before they are available.

Using tags for access control with IAM

You can use tags with IAM policies to manage access to Amazon RDS resources and to control what actions can be applied to the Amazon RDS resources.

For information on managing access to tagged resources with IAM policies, see [Identity and access management in Amazon RDS \(p. 1644\)](#).

Using tags to produce detailed billing reports

You can also use tags to track costs by grouping expenses for similarly tagged resources.

Use tags to organize your AWS bill to reflect your own cost structure. To do this, sign up to get your AWS account bill with tag key values included. Then, to see the cost of combined resources, organize your billing information according to resources with the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see [Using Cost Allocation Tags in the AWS Billing and Cost Management User Guide](#).

Note

You can add a tag to a snapshot, however, your bill will not reflect this grouping.

Adding, listing, and removing tags

The following procedures show how to perform typical tagging operations on resources related to DB instances.

Console

The process to tag an Amazon RDS resource is similar for all resources. The following procedure shows how to tag an Amazon RDS DB instance.

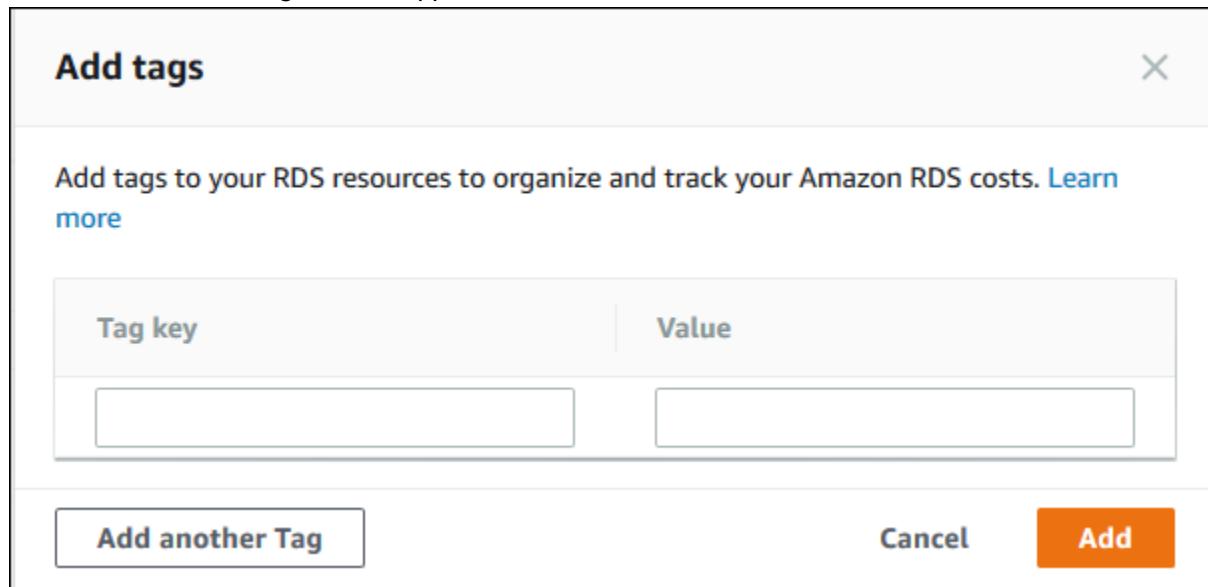
To add a tag to a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.

Note

To filter the list of DB instances in the **Databases** pane, enter a text string for **Filter databases**. Only DB instances that contain the string appear.

3. Choose the name of the DB instance that you want to tag to show its details.
4. In the details section, scroll down to the **Tags** section.
5. Choose **Add**. The **Add tags** window appears.



6. Enter a value for **Tag key** and **Value**.
7. To add another tag, you can choose **Add another Tag** and enter a value for its **Tag key** and **Value**.
Repeat this step as many times as necessary.
8. Choose **Add**.

To delete a tag from a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.

Note

To filter the list of DB instances in the **Databases** pane, enter a text string in the **Filter databases** box. Only DB instances that contain the string appear.

3. Choose the name of the DB instance to show its details.
4. In the details section, scroll down to the **Tags** section.

5. Choose the tag you want to delete.

The screenshot shows a table titled "Tags (1)". It has two columns: "Tag key" and "Value". A single row is selected, showing "workload-type" in the "Tag key" column and "other" in the "Value" column. At the top right of the table, there are three buttons: "Edit", "Delete", and "Add". Above the table, there is a search bar labeled "Filter tag key" and a page navigation area with a magnifying glass icon, the number "1", and arrows.

6. Choose **Delete**, and then choose **Delete** in the **Delete tags** window.

AWS CLI

You can add, list, or remove tags for a DB instance using the AWS CLI.

- To add one or more tags to an Amazon RDS resource, use the AWS CLI command [add-tags-to-resource](#).
- To list the tags on an Amazon RDS resource, use the AWS CLI command [list-tags-for-resource](#).
- To remove one or more tags from an Amazon RDS resource, use the AWS CLI command [remove-tags-from-resource](#).

To learn more about how to construct the required ARN, see [Constructing an ARN for Amazon RDS \(p. 309\)](#).

RDS API

You can add, list, or remove tags for a DB instance using the Amazon RDS API.

- To add a tag to an Amazon RDS resource, use the [AddTagsToResource](#) operation.
- To list tags that are assigned to an Amazon RDS resource, use the [ListTagsForResource](#).
- To remove tags from an Amazon RDS resource, use the [RemoveTagsFromResource](#) operation.

To learn more about how to construct the required ARN, see [Constructing an ARN for Amazon RDS \(p. 309\)](#).

When working with XML using the Amazon RDS API, tags use the following schema:

```
<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Trinity</Value>
    </Tag>
    <Tag>
      <Key>User</Key>
      <Value>Jones</Value>
    </Tag>
  </TagSet>
</Tagging>
```

The following table provides a list of the allowed XML tags and their characteristics. Values for Key and Value are case-dependent. For example, project=Trinity and PROJECT=Trinity are two distinct tags.

Tagging element	Description
TagSet	A tag set is a container for all tags assigned to an Amazon RDS resource. There can be only one tag set per resource. You work with a TagSet only through the Amazon RDS API.
Tag	A tag is a user-defined key-value pair. There can be from 1 to 50 tags in a tag set.
Key	A key is the required name of the tag. The string value can be from 1 to 128 Unicode characters in length and cannot be prefixed with "rds:" or "aws:". The string can only contain only the set of Unicode letters, digits, white-space, '_', '.', '/', '=', '+', '-' (Java regex: " $^([\u0000-\uFFFF][\u0000-\uFFFF]*[\u0000-\uFFFF]_::=+[\u0000-\uFFFF]*)$$ "). Keys must be unique to a tag set. For example, you cannot have a key-pair in a tag set with the key the same but with different values, such as project/Trinity and project/Xanadu.
Value	A value is the optional value of the tag. The string value can be from 1 to 256 Unicode characters in length and cannot be prefixed with "rds:" or "aws:". The string can only contain only the set of Unicode letters, digits, white-space, '_', '.', '/', '=', '+', '-' (Java regex: " $^([\u0000-\uFFFF][\u0000-\uFFFF]*[\u0000-\uFFFF]_::=+[\u0000-\uFFFF]*)$$ "). Values do not have to be unique in a tag set and can be null. For example, you can have a key-value pair in a tag set of project/Trinity and cost-center/Trinity.

Using the AWS Tag Editor

You can browse and edit the tags on your RDS resources in the AWS Management Console by using the AWS Tag editor. For more information, see [Tag Editor](#) in the *AWS Resource Groups User Guide*.

Copying tags to DB instance snapshots

When you create or restore a DB instance, you can specify that the tags from the DB instance are copied to snapshots of the DB instance. Copying tags ensures that the metadata for the DB snapshots matches that of the source DB instance and any access policies for the DB snapshot also match those of the source DB instance. Tags are not copied by default.

You can specify that tags are copied to DB snapshots for the following actions:

- Creating a DB instance.
- Restoring a DB instance.
- Creating a read replica.
- Copying a DB snapshot.

Note

If you include a value for the `--tag-key` parameter of the `create-db-snapshot` AWS CLI command (or supply at least one tag to the [CreateDBSnapshot](#) API operation) then RDS doesn't copy tags from the source DB instance to the new DB snapshot. This functionality applies even if the source DB instance has the `--copy-tags-to-snapshot` (`CopyTagsToSnapshot`) option enabled. If you take this approach, you can create a copy of a DB instance from a DB snapshot and avoid adding tags that don't apply to the new DB instance. Once you have created your DB

snapshot using the AWS CLI `create-db-snapshot` command (or the `CreateDBSnapshot` Amazon RDS API operation) you can then add tags as described later in this topic.

Tutorial: Use tags to specify which DB instances to stop

Suppose that you're creating a number of DB instances in a development or test environment. You need to keep all of these DB instances for several days. Some of the DB instances run tests overnight. Other DB instances can be stopped overnight and started again the next day. The following example shows how to assign a tag to those DB instances that are suitable to stop overnight. Then the example shows how a script can detect which DB instances have that tag and then stop those DB instances. In this example, the value portion of the key-value pair doesn't matter. The presence of the `stoppable` tag signifies that the DB instance has this user-defined property.

To specify which DB instances to stop

1. Determine the ARN of a DB instance that you want to designate as stoppable.

The commands and APIs for tagging work with ARNs. That way, they can work seamlessly across AWS Regions, AWS accounts, and different types of resources that might have identical short names. You can specify the ARN instead of the DB instance ID in CLI commands that operate on DB instances. Substitute the name of your own DB instances for `dev-test-db-instance`. In subsequent commands that use ARN parameters, substitute the ARN of your own DB instance. The ARN includes your own AWS account ID and the name of the AWS Region where your DB instance is located.

```
$ aws rds describe-db-instances --db-instance-id dev-test-db-instance \
--query "[].{DBInstance:DBInstanceArn}" --output text
arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
```

2. Add the tag `stoppable` to this DB instance.

The name for this tag is chosen by you. Using a tag like this is an alternative to devising a naming convention that encodes all the relevant information in the name of the DB instance (or other types of resources). Because this example treats the tag as an attribute that is either present or absent, it omits the `Value=` part of the `--tags` parameter.

```
$ aws rds add-tags-to-resource \
--resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance \
--tags Key=stoppable
```

3. Confirm that the tag is present in the DB instance.

These commands retrieve the tag information for the DB instance in JSON format and in plain tab-separated text.

```
$ aws rds list-tags-for-resource \
--resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
{
    "TagList": [
        {
            "Key": "stoppable",
            "Value": ""
        }
    ]
}
aws rds list-tags-for-resource \
```

```
--resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance --output
text
TAGLIST stoppable
```

4. To stop all the DB instances that are designated as stoppable, prepare a list of all your DB instances. Loop through the list and check if each DB instance is tagged with the relevant attribute.

This Linux example uses shell scripting to save the list of DB instance ARNs to a temporary file and then perform CLI commands for each DB instance.

```
$ aws rds describe-db-instances --query "[].{DBInstanceArn}" --output text >/tmp/
db_instance_arns.lst
$ for arn in $(cat /tmp/db_instance_arns.lst)
do
    match=$(aws rds list-tags-for-resource --resource-name $arn --output text | grep
stoppable)"
    if [[ ! -z "$match" ]]
    then
        echo "DB instance $arn is tagged as stoppable. Stopping it now."
# Note that you need to get the DB instance identifier from the ARN.
        dbid=$(echo $arn | sed -e 's/.*/::/')
        aws rds stop-db-instance --db-instance-identifier $dbid
    fi
done

DB instance arn:arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance is tagged as
stoppable. Stopping it now.
{
    "DBInstance": {
        "DBInstanceIdentifier": "dev-test-db-instance",
        "DBInstanceClass": "db.t3.medium",
        ...
    }
}
```

You can run a script like this at the end of each day to make sure that nonessential DB instances are stopped. You might also schedule a job using a utility such as cron to perform such a check each night, in case some DB instances were left running by mistake. In that case, you might fine-tune the command that prepares the list of DB instances to check. The following command produces a list of your DB instances, but only the ones in available state. The script can ignore DB instances that are already stopped, because they will have different status values such as stopped or stopping.

```
$ aws rds describe-db-instances \
--query '[].{DBInstanceArn:DBInstanceArn,DBInstanceStatus:DBInstanceStatus}|[?
DBInstanceStatus == `available`]|[].{DBInstanceArn:DBInstanceArn}' \
--output text
arn:aws:rds:us-east-1:123456789102:db:db-instance-2447
arn:aws:rds:us-east-1:123456789102:db:db-instance-3395
arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
arn:aws:rds:us-east-1:123456789102:db:pg2-db-instance
```

Tip

Once you're familiar with the general procedure of assigning tags and finding DB instances that have those tags, you can use the same technique to reduce costs in other ways. For example, in this scenario with DB instances used for development and testing, you might designate some DB instances to be deleted at the end of each day, or to have their DB instances changed to a small DB instance classes during times of expected low usage.

Using tags to enable backups in AWS Backup

AWS Backup is a fully managed backup service that makes it easy to centralize and automate the backup of data across AWS services in the cloud and on premises. You can manage backups of your Amazon RDS DB instances in AWS Backup.

To enable backups in AWS Backup, you use resource tagging to associate your DB instance with a backup plan.

This example assumes that you have already created a backup plan in AWS Backup. You use exactly the same tag for your DB instance that is in your backup plan, as shown in the following figure.

Backup plan tags (1)		Edit	Delete	Add
<input type="text"/> Filter by tags				« 1 » @
<input type="checkbox"/>	Tag key	Value		
<input type="checkbox"/>	BackupPlan	Test		

For more information about AWS Backup, see the [AWS Backup Developer Guide](#).

You can assign a tag to a DB instance using the AWS Management Console, the AWS CLI, or the RDS API. The following examples are for the console and CLI.

Console

To assign a tag to a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the link for the DB instance to which you want to assign a tag.
4. On the database details page, choose the **Tags** tab.
5. Under **Tags**, choose **Add tags**.
6. Under **Add tags**:
 - a. For **Tag key**, enter **BackupPlan**.
 - b. For **Value**, enter **Test**.
 - c. Choose **Add**.

The result is shown under **Tags**.

Key	Value
BackupPlan	Test

CLI

To assign a tag to a DB instance

- Use the following CLI command:

For Linux, macOS, or Unix:

```
aws rds add-tags-to-resource \
--resource-name arn:aws:rds:us-east-1:123456789012:db:new-orcl-db \
--tags Key=BackupPlan,Value=Test
```

For Windows:

```
aws rds add-tags-to-resource ^
--resource-name arn:aws:rds:us-east-1:123456789012:db:new-orcl-db ^
--tags Key=BackupPlan,Value=Test
```

The `add-tags-to-resource` CLI command returns no output.

To confirm that the DB instance is tagged

- Use the following CLI command:

For Linux, macOS, or Unix:

```
aws rds list-tags-for-resource \
--resource-name arn:aws:rds:us-east-1:123456789012:db:new-orcl-db
```

For Windows:

```
aws rds list-tags-for-resource ^
--resource-name arn:aws:rds:us-east-1:123456789012:db:new-orcl-db
```

The `list-tags-for-resource` CLI command returns the following output:

```
{  
    "TagList": [  
        {  
            "Key": "BackupPlan",  
            "Value": "Test"  
        }  
    ]  
}
```

Working with Amazon Resource Names (ARNs) in Amazon RDS

Resources created in Amazon Web Services are each uniquely identified with an Amazon Resource Name (ARN). For certain Amazon RDS operations, you must uniquely identify an Amazon RDS resource by specifying its ARN. For example, when you create an RDS DB instance read replica, you must supply the ARN for the source DB instance.

Constructing an ARN for Amazon RDS

Resources created in Amazon Web Services are each uniquely identified with an Amazon Resource Name (ARN). You can construct an ARN for an Amazon RDS resource using the following syntax.

`arn:aws:rds:<region>:<account number>:<resourcetype>:<name>`

Region Name	Region	Endpoint	Protocol	
US East (Ohio)	us-east-2	rds.us-east-2.amazonaws.com rds-fips.us-east-2.amazonaws.com	HTTPS HTTPS	
US East (N. Virginia)	us-east-1	rds.us-east-1.amazonaws.com rds-fips.us-east-1.amazonaws.com	HTTPS HTTPS	
US West (N. California)	us-west-1	rds.us-west-1.amazonaws.com rds-fips.us-west-1.amazonaws.com	HTTPS HTTPS	
US West (Oregon)	us-west-2	rds.us-west-2.amazonaws.com rds-fips.us-west-2.amazonaws.com	HTTPS HTTPS	
Africa (Cape Town)	af-south-1	rds.af-south-1.amazonaws.com	HTTPS	
Asia Pacific (Hong Kong)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS	
Asia Pacific (Mumbai)	ap-south-1	rds.ap-south-1.amazonaws.com	HTTPS	
Asia Pacific (Osaka)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com	HTTPS	
Asia Pacific (Seoul)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com	HTTPS	

Region Name	Region	Endpoint	Protocol	
Asia Pacific (Singapore)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com	HTTPS	
Asia Pacific (Sydney)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com	HTTPS	
Asia Pacific (Tokyo)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS	
Canada (Central)	ca-central-1	rds.ca-central-1.amazonaws.com rds-fips.ca-central-1.amazonaws.com	HTTPS HTTPS	
China (Beijing)	cn-north-1	rds.cn-north-1.amazonaws.com.cn	HTTPS	
China (Ningxia)	cn-northwest-1	rds.cn-northwest-1.amazonaws.com.cn	HTTPS	
Europe (Frankfurt)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS	
Europe (Ireland)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS	
Europe (London)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS	
Europe (Milan)	eu-south-1	rds.eu-south-1.amazonaws.com	HTTPS	
Europe (Paris)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS	
Europe (Stockholm)	eu-north-1	rds.eu-north-1.amazonaws.com	HTTPS	
Middle East (Bahrain)	me-south-1	rds.me-south-1.amazonaws.com	HTTPS	
South America (São Paulo)	sa-east-1	rds.sa-east-1.amazonaws.com	HTTPS	
AWS GovCloud (US-East)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com	HTTPS	
AWS GovCloud (US-West)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com	HTTPS	

The following table shows the format that you should use when constructing an ARN for a particular Amazon RDS resource type.

Resource type	ARN format
DB instance	<p>arn:aws:rds:<region>:<account>:db:<name></p> <p>For example:</p> <pre>arn:aws:rds:us-east-2:123456789012:db:my-mysql-instance-1</pre>
Event subscription	<p>arn:aws:rds:<region>:<account>:es:<name></p> <p>For example:</p> <pre>arn:aws:rds:us-east-2:123456789012:es:my-subscription</pre>
DB option group	<p>arn:aws:rds:<region>:<account>:og:<name></p> <p>For example:</p> <pre>arn:aws:rds:us-east-2:123456789012:og:my-og</pre>
DB parameter group	<p>arn:aws:rds:<region>:<account>:pg:<name></p> <p>For example:</p> <pre>arn:aws:rds:us-east-2:123456789012:pg:my-param-enable-logs</pre>
Reserved DB instance	<p>arn:aws:rds:<region>:<account>:ri:<name></p> <p>For example:</p> <pre>arn:aws:rds:us-east-2:123456789012:ri:my-reserved-postgresql</pre>
DB security group	<p>arn:aws:rds:<region>:<account>:secgrp:<name></p> <p>For example:</p> <pre>arn:aws:rds:us-east-2:123456789012:secgrp:my-public</pre>
Automated DB snapshot	<p>arn:aws:rds:<region>:<account>:snapshot:rds:<name></p> <p>For example:</p> <pre>arn:aws:rds:us-east-2:123456789012:snapshot:rds:my-mysql-db-2019-07-22-07-23</pre>
Manual DB snapshot	<p>arn:aws:rds:<region>:<account>:snapshot:<name></p> <p>For example:</p>

Resource type	ARN format
	<code>arn:aws:rds:<i>us-east-2</i>:<i>123456789012</i>:snapshot:<i>my-mysql-db-snap</i></code>
DB subnet group	<code>arn:aws:rds:<region>:<account>:subgrp:<name></code> For example: <code>arn:aws:rds:<i>us-east-2</i>:<i>123456789012</i>:subgrp:<i>my-subnet-10</i></code>

Getting an existing ARN

You can get the ARN of an RDS resource by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or RDS API.

Console

To get an ARN from the AWS Management Console, navigate to the resource you want an ARN for, and view the details for that resource. For example, you can get the ARN for a DB instance from the **Configuration** tab of the DB instance details, as shown following.

Connectivity Monitoring Logs & events Configuration

Instance

Configuration

DB instance id
oracle-instance1

Engine version
12.1.0.2.v14

Storage type
General Purpose (SSD)

IOPS
-

Storage
20 GiB

DB name
ORCL

License model
Bring Your Own License

Character set
AL32UTF8

Option groups
default:oracle-ee-12-1

ARN
arn:aws:rds:us-west-2:XXXXXXXXXX:db:oracle-instance1

Resource id

AWS CLI

To get an ARN from the AWS CLI for a particular RDS resource, you use the `describe` command for that resource. The following table shows each AWS CLI command, and the ARN property used with the command to get an ARN.

AWS CLI command	ARN property
<code>describe-event-subscriptions</code>	<code>EventSubscriptionArn</code>
<code>describe-certificates</code>	<code>CertificateArn</code>

AWS CLI command	ARN property
describe-db-parameter-groups	DBParameterGroupArn
describe-db-instances	DBInstanceArn
describe-db-security-groups	DBSecurityGroupArn
describe-db-snapshots	DBSnapshotArn
describe-events	SourceArn
describe-reserved-db-instances	ReservedDBInstanceArn
describe-db-subnet-groups	DBSubnetGroupArn
describe-option-groups	OptionGroupArn

For example, the following AWS CLI command gets the ARN for a DB instance.

Example

For Linux, macOS, or Unix:

```
aws rds describe-db-instances \
--db-instance-identifier DBInstanceIdentifier \
--region us-west-2 \
--query "[].{DBInstanceIdentifier:DBInstanceIdentifier,DBInstanceArn:DBInstanceArn}"
```

For Windows:

```
aws rds describe-db-instances ^
--db-instance-identifier DBInstanceIdentifier ^
--region us-west-2 ^
--query "[].{DBInstanceIdentifier:DBInstanceIdentifier,DBInstanceArn:DBInstanceArn}"
```

The output of that command is like the following:

```
[  
  {  
    "DBInstanceArn": "arn:aws:rds:us-west-2:account_id:db:instance_id",  
    "DBInstanceIdentifier": "instance_id"  
  }  
]
```

RDS API

To get an ARN for a particular RDS resource, you can call the following RDS API operations and use the ARN properties shown following.

RDS API operation	ARN property
DescribeEventSubscriptions	EventSubscriptionArn
DescribeCertificates	CertificateArn

RDS API operation	ARN property
DescribeDBParameterGroups	DBParameterGroupArn
DescribeDBInstances	DBInstanceArn
DescribeDBSecurityGroups	DBSecurityGroupArn
DescribeDBSchemas	DBSnapshotArn
DescribeEvents	SourceArn
DescribeReservedDBInstances	ReservedDBInstanceArn
DescribeDBSubnetGroups	DBSubnetGroupArn
DescribeOptionGroups	OptionGroupArn

Working with storage for Amazon RDS DB instances

To specify how you want your data stored in Amazon RDS, choose a storage type and provide a storage size when you create or modify a DB instance. Later, you can increase the amount or change the type of storage by modifying the DB instance. For more information about which storage type to use for your workload, see [Amazon RDS storage types \(p. 40\)](#).

Topics

- [Increasing DB instance storage capacity \(p. 316\)](#)
- [Managing capacity automatically with Amazon RDS storage autoscaling \(p. 317\)](#)
- [Modifying SSD storage settings for Provisioned IOPS \(p. 322\)](#)

Increasing DB instance storage capacity

If you need space for additional data, you can scale up the storage of an existing DB instance. To do so, you can use the Amazon RDS Management Console, the Amazon RDS API, or the AWS Command Line Interface (AWS CLI). For information about storage limits, see [Amazon RDS DB instance storage \(p. 40\)](#).

Note

Scaling storage for Amazon RDS for Microsoft SQL Server DB instances is supported only for General Purpose SSD or Provisioned IOPS SSD storage types.

To monitor the amount of free storage for your DB instance so you can respond when necessary, we recommend that you create an Amazon CloudWatch alarm. For more information on setting CloudWatch alarms, see [Using CloudWatch alarms](#).

In most cases, scaling storage doesn't require any outage and doesn't degrade performance of the server. After you modify the storage size for a DB instance, the status of the DB instance is **storage-optimization**. The DB instance is fully operational after a storage modification.

Note

You can't make further storage modifications until six (6) hours after storage optimization has completed on the instance.

However, a special case is if you have a SQL Server DB instance and haven't modified the storage configuration since November 2017. In this case, you might experience a short outage of a few minutes when you modify your DB instance to increase the allocated storage. After the outage, the DB instance is online but in the **storage-optimization** state. Performance might be degraded during storage optimization.

Note

You can't reduce the amount of storage for a DB instance after storage has been allocated. When you increase the allocated storage, it must be by at least 10 percent. If you try to increase the value by less than 10 percent, you get an error.

Console

To increase storage for a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.

3. Choose the DB instance that you want to modify.
4. Choose **Modify**.
5. Enter a new value for **Allocated storage**. It must be greater than the current value.

The screenshot shows the 'Storage type' dropdown set to 'General Purpose (SSD)'. Below it, the 'Allocated storage' input field contains '16384 GiB'. A note below the input says 'This instance supports multiple storage ranges between 20 and 16384 GiB.' with a 'See all' link. A warning box contains the text: 'Scaling your instance storage can:' followed by two bullet points: 'Deplete the initial General Purpose (SSD) I/O credits, leading to longer conversion times.' and 'Impact instance performance until operation completes.'

6. Choose **Continue** to move to the next screen.
7. Choose **Apply immediately** in the **Scheduling of modifications** section to apply the storage changes to the DB instance immediately. Or choose **Apply during the next scheduled maintenance window** to apply the changes during the next maintenance window.
8. When the settings are as you want them, choose **Modify DB instance**.

AWS CLI

To increase the storage for a DB instance, use the AWS CLI command `modify-db-instance`. Set the following parameters:

- `--allocated-storage` – Amount of storage to be allocated for the DB instance, in gibibytes.
- `--apply-immediately` – Use `--apply-immediately` to change to the new storage type immediately. Or use `--no-apply-immediately` (the default) to apply storage changes during the next maintenance window. An immediate outage occurs when the changes are applied.

For more information about storage, see [Amazon RDS DB instance storage \(p. 40\)](#).

Amazon RDS API

To increase storage for a DB instance, use the Amazon RDS API operation `ModifyDBInstance`. Set the following parameters:

- `AllocatedStorage` – Amount of storage to be allocated for the DB instance, in gibibytes.
- `ApplyImmediately` – Set this option to `True` to apply scaling changes immediately. Set this option to `False` (the default) to apply scaling changes during the next maintenance window. An immediate outage occurs when the changes are applied.

For more information about storage, see [Amazon RDS DB instance storage \(p. 40\)](#).

Managing capacity automatically with Amazon RDS storage autoscaling

If your workload is unpredictable, you can enable storage autoscaling for an Amazon RDS DB instance. To do so, you can use the Amazon RDS console, the Amazon RDS API, or the AWS CLI.

For example, you might use this feature for a new mobile gaming application that users are adopting rapidly. In this case, a rapidly increasing workload might exceed the available database storage. To avoid having to manually scale up database storage, you can use Amazon RDS storage autoscaling.

With storage autoscaling enabled, when Amazon RDS detects that you are running out of free database space it automatically scales up your storage. Amazon RDS starts a storage modification for an autoscaling-enabled DB instance when these factors apply:

- Free available space is less than 10 percent of the allocated storage.
- The low-storage condition lasts at least five minutes.
- At least six hours have passed since the last storage modification.

The additional storage is in increments of whichever of the following is greater:

- 5 GiB
- 10 percent of currently allocated storage
- Storage growth prediction for 7 hours based on the `FreeStorageSpace` metrics change in the past hour. For more information on metrics, see [Monitoring with Amazon CloudWatch](#).

The maximum storage threshold is the limit that you set for autoscaling the DB instance. You can't set the maximum storage threshold for autoscaling-enabled instances to a value greater than the maximum allocated storage.

For example, SQL Server Standard Edition on db.m5.xlarge has a default allocated storage for the instance of 20 GiB (the minimum) and a maximum allocated storage of 16,384 GiB. The default maximum storage threshold for autoscaling is 1,000 GiB. If you use this default, the instance doesn't autoscale above 1,000 GiB. This is true even though the maximum allocated storage for the instance is 16,384 GiB.

Note

We recommend that you carefully choose the maximum storage threshold based on usage patterns and customer needs. If there are any aberrations in the usage patterns, the maximum storage threshold can prevent scaling storage to an unexpectedly high value when autoscaling predicts a very high threshold. After a DB instance has been autoscaled, its allocated storage can't be reduced.

The following limitations apply to storage autoscaling:

- Autoscaling doesn't occur if the maximum storage threshold would be exceeded by the storage increment.
- Autoscaling can't completely prevent storage-full situations for large data loads, because further storage modifications can't be made until six hours after storage optimization has completed on the instance. If you perform a large data load, and autoscaling doesn't provide enough space, the database might remain in the storage-full state for several hours. This can harm the database.
- If you start a storage scaling operation at the same time that Amazon RDS starts an autoscaling operation, your storage modification takes precedence. The autoscaling operation is canceled.
- Autoscaling can't be used with magnetic storage.
- Autoscaling can't be used with the following previous-generation instance classes that have less than 6 TiB of orderable storage: db.m3.large, db.m3.xlarge, and db.m3.2xlarge.
- Autoscaling operations aren't logged by AWS CloudTrail. For more information on CloudTrail, see [Working with AWS CloudTrail and Amazon RDS \(p. 557\)](#).

Although automatic scaling helps you to increase storage on your Amazon RDS DB instance dynamically, you should still configure the initial storage for your DB instance to an appropriate size for your typical workload.

Enabling storage autoscaling for a new DB instance

When you create a new Amazon RDS DB instance, you can choose whether to enable storage autoscaling. You can also set an upper limit on the storage that Amazon RDS can allocate for the DB instance.

Note

When you clone an Amazon RDS DB instance that has storage autoscaling enabled, that setting isn't automatically inherited by the cloned instance. The new DB instance has the same amount of allocated storage as the original instance. You can turn storage autoscaling on again for the new instance if the cloned instance continues to increase its storage requirements.

Console

To enable storage autoscaling for a new DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region where you want to create the DB instance.
3. In the navigation pane, choose **Databases**.
4. Choose **Create database**. On the **Select engine** page, choose your database engine and specify your DB instance information as described in [Getting started with Amazon RDS \(p. 73\)](#).
5. In the **Storage autoscaling** section, set the **Maximum storage threshold** value for the DB instance.
6. Specify the rest of your DB instance information as described in [Getting started with Amazon RDS \(p. 73\)](#).

AWS CLI

To enable storage autoscaling for a new DB instance, use the AWS CLI command `create-db-instance`. Set the following parameter:

- `--max-allocated-storage` – Turns on storage autoscaling and sets the upper limit on storage size, in gibibytes.

To verify that Amazon RDS storage autoscaling is available for your DB instance, use the AWS CLI `describe-valid-db-instance-modifications` command. To check based on the instance class before creating an instance, use the `describe-orderable-db-instance-options` command. Check the following field in the return value:

- `SupportsStorageAutoscaling` – Indicates whether the DB instance or instance class supports storage autoscaling.

For more information about storage, see [Amazon RDS DB instance storage \(p. 40\)](#).

Amazon RDS API

To enable storage autoscaling for a new DB instance, use the Amazon RDS API operation `CreateDBInstance`. Set the following parameter:

- `MaxAllocatedStorage` – Turns on Amazon RDS storage autoscaling and sets the upper limit on storage size, in gibibytes.

To verify that Amazon RDS storage autoscaling is available for your DB instance, use the Amazon RDS API `DescribeValidDbInstanceModifications` operation for an existing instance, or the

[DescribeOrderableDBInstanceOptions](#) operation before creating an instance. Check the following field in the return value:

- `SupportsStorageAutoscaling` – Indicates whether the DB instance supports storage autoscaling.

For more information about storage, see [Amazon RDS DB instance storage \(p. 40\)](#).

Changing the storage autoscaling settings for a DB instance

You can turn storage autoscaling on for an existing Amazon RDS DB instance. You can also change the upper limit on the storage that Amazon RDS can allocate for the DB instance.

Console

To change the storage autoscaling settings for a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the DB instance that you want to modify, and choose **Modify**. The **Modify DB instance** page appears.
4. Change the storage limit in the **Autoscaling** section. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
5. When all the changes are as you want them, choose **Continue** and check your modifications.
6. On the confirmation page, review your changes. If they're correct, choose **Modify DB Instance** to save your changes. If they aren't correct, choose **Back** to edit your changes or **Cancel** to cancel your changes.

Changing the storage autoscaling limit occurs immediately. This setting ignores the **Apply immediately** setting.

AWS CLI

To change the storage autoscaling settings for a DB instance, use the AWS CLI command `modify-db-instance`. Set the following parameter:

- `--max-allocated-storage` – Sets the upper limit on storage size, in gibibytes. If the value is greater than the `--allocated-storage` parameter, storage autoscaling is turned on. If the value is the same as the `--allocated-storage` parameter, storage autoscaling is turned off.

To verify that Amazon RDS storage autoscaling is available for your DB instance, use the AWS CLI `describe-valid-db-instance-modifications` command. To check based on the instance class before creating an instance, use the `describe-orderable-db-instance-options` command. Check the following field in the return value:

- `SupportsStorageAutoscaling` – Indicates whether the DB instance supports storage autoscaling.

For more information about storage, see [Amazon RDS DB instance storage \(p. 40\)](#).

Amazon RDS API

To change the storage autoscaling settings for a DB instance, use the Amazon RDS API operation `ModifyDBInstance`. Set the following parameter:

- `MaxAllocatedStorage` – Sets the upper limit on storage size, in gibibytes.

To verify that Amazon RDS storage autoscaling is available for your DB instance, use the Amazon RDS API [DescribeValidDbInstanceModifications](#) operation for an existing instance, or the [DescribeOrderableDBInstanceOptions](#) operation before creating an instance. Check the following field in the return value:

- `SupportsStorageAutoscaling` – Indicates whether the DB instance supports storage autoscaling.

For more information about storage, see [Amazon RDS DB instance storage \(p. 40\)](#).

Turning off storage autoscaling for a DB instance

If you no longer need Amazon RDS to automatically increase the storage for an Amazon RDS DB instance, you can turn off storage autoscaling. After you do, you can still manually increase the amount of storage for your DB instance.

Console

To turn off storage autoscaling for a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the DB instance that you want to modify and choose **Modify**. The **Modify DB instance** page appears.
4. Clear the **Enable storage autoscaling** check box in the **Storage autoscaling** section. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
5. When all the changes are as you want them, choose **Continue** and check the modifications.
6. On the confirmation page, review your changes. If they're correct, choose **Modify DB Instance** to save your changes. If they aren't correct, choose **Back** to edit your changes or **Cancel** to cancel your changes.

Changing the storage autoscaling limit occurs immediately. This setting ignores the **Apply immediately** setting.

AWS CLI

To turn off storage autoscaling for a DB instance, use the AWS CLI command `modify-db-instance` and the following parameter:

- `--max-allocated-storage` – Specify a value equal to the `--allocated-storage` setting to prevent further Amazon RDS storage autoscaling for the specified DB instance.

For more information about storage, see [Amazon RDS DB instance storage \(p. 40\)](#).

Amazon RDS API

To turn off storage autoscaling for a DB instance, use the Amazon RDS API operation `ModifyDBInstance`. Set the following parameter:

- `MaxAllocatedStorage` – Specify a value equal to the `AllocatedStorage` setting to prevent further Amazon RDS storage autoscaling for the specified DB instance.

For more information about storage, see [Amazon RDS DB instance storage \(p. 40\)](#).

Modifying SSD storage settings for Provisioned IOPS

You can modify the settings for a DB instance that uses Provisioned IOPS SSD storage by using the Amazon RDS console, AWS CLI, or Amazon RDS API. Specify the storage type, allocated storage, and the amount of Provisioned IOPS that you require. You can choose from a range between 1,000 IOPS and 100 GiB of storage up to 80,000 IOPS and 64 TiB (64,000 GiB) of storage. The range depends on your database engine and instance type.

Although you can reduce the amount of IOPS provisioned for your instance, you can't reduce the amount of General Purpose SSD or magnetic storage allocated.

In most cases, scaling storage doesn't require any outage and doesn't degrade performance of the server. After you modify the storage IOPS for a DB instance, the status of the DB instance is **storage-optimization**. The DB instance is fully operational after a storage modification.

Note

You can't make further storage modifications until six (6) hours after storage optimization has completed on the instance.

Console

To change the Provisioned IOPS settings for a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.

Note

To filter the list of DB instances, for **Filter databases** enter a text string for Amazon RDS to use to filter the results. Only DB instances whose names contain the string appear.

3. Choose the DB instance with Provisioned IOPS that you want to modify.
4. Choose **Modify**.
5. On the **Modify DB Instance page**, choose Provisioned IOPS for **Storage type** and then provide a Provisioned IOPS value.

The screenshot shows the 'Modify DB Instance' page with the following fields:

- Storage type:** A dropdown menu set to "Provisioned IOPS (SSD)".
- Allocated storage:** An input field containing "16384" with a unit indicator "GiB". Below it, a note says "Minimum: 100 GiB, Maximum: 16384".
- Provisioned IOPS:** An input field containing "80000". To its left is a link "Info".

If the value you specify for either **Allocated storage** or **Provisioned IOPS** is outside the limits supported by the other parameter, a warning message is displayed. This message gives the range of values required for the other parameter.

6. Choose **Continue**.
7. To apply the changes to the DB instance immediately, choose **Apply immediately** in the **Scheduling of modifications** section. Or choose **Apply during the next scheduled maintenance window** to apply the changes during the next maintenance window.

An immediate outage occurs when the storage type changes. For more information about storage, see [Amazon RDS DB instance storage \(p. 40\)](#).

8. Review the parameters to be changed, and choose **Modify DB instance** to complete the modification.

The new value for allocated storage or for Provisioned IOPS appears in the **Status** column.

AWS CLI

To change the Provisioned IOPS setting for a DB instance, use the AWS CLI command [modify-db-instance](#). Set the following parameters:

- `--storage-type` – Set to `io1` for Provisioned IOPS.
- `--allocated-storage` – Amount of storage to be allocated for the DB instance, in gibibytes.
- `--iops` – The new amount of Provisioned IOPS for the DB instance, expressed in I/O operations per second.
- `--apply-immediately` – Use `--apply-immediately` to apply changes immediately. Use `--no-apply-immediately` (the default) to apply changes during the next maintenance window.

Amazon RDS API

To change the Provisioned IOPS settings for a DB instance, use the Amazon RDS API operation [ModifyDBInstance](#). Set the following parameters:

- `StorageType` – Set to `io1` for Provisioned IOPS.
- `AllocatedStorage` – Amount of storage to be allocated for the DB instance, in gibibytes.
- `Iops` – The new IOPS rate for the DB instance, expressed in I/O operations per second.
- `ApplyImmediately` – Set this option to `True` to apply changes immediately. Set this option to `False` (the default) to apply changes during the next maintenance window.

Deleting a DB instance

To delete a DB instance, you must do the following:

- Provide the name of the instance
- Enable or disable the option to take a final DB snapshot of the instance
- Enable or disable the option to retain automated backups

If the DB instance that you want to delete has a read replica, you should either promote the read replica or delete it. For more information, see [Promoting a read replica to be a standalone DB instance \(p. 285\)](#).

Note

When the status for a DB instance is `deleting`, its CA certificate value doesn't appear in the RDS console or in output for AWS CLI commands or RDS API operations. For more information about CA certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

Deletion protection

You can only delete instances that don't have deletion protection enabled. When you create or modify a DB instance, you have the option to enable deletion protection so that users can't delete the DB instance. Deletion protection is disabled by default for you when you use AWS CLI and API commands. Deletion protection is enabled for you when you use the AWS Management Console to create a production DB instance. However, Amazon RDS enforces deletion protection when you use the console, the CLI, or the API to delete a DB instance. To delete a DB instance that has deletion protection enabled, first modify the instance and disable deletion protection. Enabling or disabling deletion protection doesn't cause an outage.

Creating a final snapshot and retaining automated backups

When you delete a DB instance, you can choose to do one or both of the following:

- Create a final DB snapshot.
 - To be able to restore your deleted DB instance later, create a final DB snapshot. The final snapshot is retained, along with any manual snapshots that were taken.
 - To delete a DB instance quickly, you can skip creating a final DB snapshot.

Note

You can't create a final DB snapshot of your DB instance if it has the status `creating`, `failed`, `incompatible-restore`, or `incompatible-network`. For more information, see [DB instance status \(p. 404\)](#).

- Retain automated backups.
 - Your automated backups are retained for the retention period that is set on the DB instance at the time when you delete it. This set retention period occurs whether or not you choose to create a final DB snapshot.
 - If you don't choose to retain automated backups, your automated backups in the same AWS Region as the DB instance are deleted. They can't be recovered after you delete the DB instance.

Note

Automated backups that are replicated to another AWS Region are retained even if you choose not to retain automated backups. For more information, see [Replicating automated backups to another AWS Region \(p. 338\)](#).

- You typically don't need to retain automated backups if you create a final DB snapshot.

- To delete a retained automated backup, follow the instructions in [Deleting retained automated backups \(p. 333\)](#).

Important

If you skip the final DB snapshot, to restore your DB instance do one of the following:

- Use an earlier manual snapshot of the DB instance to restore the DB instance to that DB snapshot's point in time.
- Retain automated backups. You can use them to restore your DB instance during your retention period, but not after your retention period has ended.

Note

Regardless of your choice, manual DB snapshots aren't deleted. For more information on snapshots, see [Creating a DB snapshot \(p. 346\)](#).

Deleting a DB instance

You can delete a DB instance using the AWS Management Console, the AWS CLI, or the RDS API.

The time required to delete a DB instance can vary depending on the backup retention period (that is, how many backups to delete), how much data is deleted, and whether a final snapshot is taken.

Note

You can't delete a DB instance when deletion protection is enabled for it. For more information, see [Deletion protection \(p. 324\)](#).

You can disable deletion protection by modifying the DB instance. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Console

To delete a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to delete.
3. For **Actions**, choose **Delete**.
4. To create a final DB snapshot for the DB instance, choose **Create final snapshot?**.
5. If you chose to create a final snapshot, enter the **Final snapshot name**.
6. To retain automated backups, choose **Retain automated backups**.
7. Enter **delete me** in the box.
8. Choose **Delete**.

AWS CLI

To delete a DB instance by using the AWS CLI, call the `delete-db-instance` command with the following options:

- `--db-instance-identifier`
- `--final-db-snapshot-identifier` or `--skip-final-snapshot`

Example With a final snapshot and no retained automated backups

For Linux, macOS, or Unix:

```
aws rds delete-db-instance \
--db-instance-identifier mydbinstance \
--final-db-snapshot-identifier mydbinstancefinalsnapshot \
--delete-automated-backups
```

For Windows:

```
aws rds delete-db-instance ^
--db-instance-identifier mydbinstance ^
--final-db-snapshot-identifier mydbinstancefinalsnapshot ^
--delete-automated-backups
```

Example With retained automated backups and no final snapshot

For Linux, macOS, or Unix:

```
aws rds delete-db-instance \
--db-instance-identifier mydbinstance \
--skip-final-snapshot \
--no-delete-automated-backups
```

For Windows:

```
aws rds delete-db-instance ^
--db-instance-identifier mydbinstance ^
--skip-final-snapshot ^
--no-delete-automated-backups
```

RDS API

To delete a DB instance by using the Amazon RDS API, call the [DeleteDBInstance](#) operation with the following parameters:

- `DBInstanceIdentifier`
- `FinalDBSnapshotIdentifier` or `SkipFinalSnapshot`

Backing up and restoring an Amazon RDS DB instance

This section shows how to back up and restore a DB instance.

Topics

- [Working with backups \(p. 328\)](#)
- [Replicating automated backups to another AWS Region \(p. 338\)](#)
- [Creating a DB snapshot \(p. 346\)](#)
- [Restoring from a DB snapshot \(p. 349\)](#)
- [Copying a snapshot \(p. 352\)](#)
- [Sharing a DB snapshot \(p. 365\)](#)
- [Exporting DB snapshot data to Amazon S3 \(p. 373\)](#)
- [Restoring a DB instance to a specified time \(p. 389\)](#)
- [Deleting a snapshot \(p. 392\)](#)
- [Tutorial: Restore a DB instance from a DB snapshot \(p. 394\)](#)

Working with backups

Amazon RDS creates and saves automated backups of your DB instance during the backup window of your DB instance. RDS creates a storage volume snapshot of your DB instance, backing up the entire DB instance and not just individual databases. RDS saves the automated backups of your DB instance according to the backup retention period that you specify. If necessary, you can recover your database to any point in time during the backup retention period.

Automated backups follow these rules:

- Your DB instance must be in the `AVAILABLE` state for automated backups to occur. Automated backups don't occur while your DB instance is in a state other than `AVAILABLE`, for example `STORAGE_FULL`.
- Automated backups and automated snapshots don't occur while a copy is running in the same AWS Region for the same DB instance.

You can also back up your DB instance manually, by manually creating a DB snapshot. For more information about creating a DB snapshot, see [Creating a DB snapshot \(p. 346\)](#).

The first snapshot of a DB instance contains the data for the full DB instance. Subsequent snapshots of the same DB instance are incremental, which means that only the data that has changed after your most recent snapshot is saved.

You can copy both automatic and manual DB snapshots, and share manual DB snapshots. For more information about copying a DB snapshot, see [Copying a snapshot \(p. 352\)](#). For more information about sharing a DB snapshot, see [Sharing a DB snapshot \(p. 365\)](#).

Backup storage

Your Amazon RDS backup storage for each AWS Region is composed of the automated backups and manual DB snapshots for that Region. Total backup storage space equals the sum of the storage for all backups in that Region. Moving a DB snapshot to another Region increases the backup storage in the destination Region. Backups are stored in Amazon S3.

For more information about backup storage costs, see [Amazon RDS pricing](#).

If you chose to retain automated backups when you delete a DB instance, the automated backups are saved for the full retention period. If you don't choose **Retain automated backups** when you delete a DB instance, all automated backups are deleted with the DB instance. After they are deleted, the automated backups can't be recovered. If you choose to have Amazon RDS create a final DB snapshot before it deletes your DB instance, you can use that to recover your DB instance. Or you can use a previously created manual snapshot. Manual snapshots are not deleted. You can have up to 100 manual snapshots per Region.

Backup window

Automated backups occur daily during the preferred backup window. If the backup requires more time than allotted to the backup window, the backup continues after the window ends, until it finishes. The backup window can't overlap with the weekly maintenance window for the DB instance.

During the automatic backup window, storage I/O might be suspended briefly while the backup process initializes (typically under a few seconds). You might experience elevated latencies for a few minutes during backups for Multi-AZ deployments. For MariaDB, MySQL, Oracle, and PostgreSQL, I/O activity is not suspended on your primary during backup for Multi-AZ deployments, because the backup is

taken from the standby. For SQL Server, I/O activity is suspended briefly during backup for Multi-AZ deployments.

If you don't specify a preferred backup window when you create the DB instance, Amazon RDS assigns a default 30-minute backup window. This window is selected at random from an 8-hour block of time for each AWS Region. The following table lists the time blocks for each AWS Region from which the default backup windows are assigned.

Region Name	Region	Time Block
US East (Ohio)	us-east-2	03:00–11:00 UTC
US East (N. Virginia)	us-east-1	03:00–11:00 UTC
US West (N. California)	us-west-1	06:00–14:00 UTC
US West (Oregon)	us-west-2	06:00–14:00 UTC
Africa (Cape Town)	af-south-1	03:00–11:00 UTC
Asia Pacific (Hong Kong)	ap-east-1	06:00–14:00 UTC
Asia Pacific (Mumbai)	ap-south-1	16:30–00:30 UTC
Asia Pacific (Osaka)	ap-northeast-3	00:00–08:00 UTC
Asia Pacific (Seoul)	ap-northeast-2	13:00–21:00 UTC
Asia Pacific (Singapore)	ap-southeast-1	14:00–22:00 UTC
Asia Pacific (Sydney)	ap-southeast-2	12:00–20:00 UTC
Asia Pacific (Tokyo)	ap-northeast-1	13:00–21:00 UTC
Canada (Central)	ca-central-1	03:00–11:00 UTC
China (Beijing)	cn-north-1	06:00–14:00 UTC
China (Ningxia)	cn-northwest-1	06:00–14:00 UTC
Europe (Frankfurt)	eu-central-1	20:00–04:00 UTC
Europe (Ireland)	eu-west-1	22:00–06:00 UTC
Europe (London)	eu-west-2	22:00–06:00 UTC
Europe (Paris)	eu-west-3	07:29–14:29 UTC
Europe (Milan)	eu-south-1	02:00–10:00 UTC
Europe (Stockholm)	eu-north-1	23:00–07:00 UTC
Middle East (Bahrain)	me-south-1	06:00–14:00 UTC
South America (São Paulo)	sa-east-1	23:00–07:00 UTC
AWS GovCloud (US-East)	us-gov-east-1	17:00–01:00 UTC

Region Name	Region	Time Block
AWS GovCloud (US-West)	us-gov-west-1	06:00–14:00 UTC

Backup retention period

You can set the backup retention period when you create a DB instance. If you don't set the backup retention period, the default backup retention period is one day if you create the DB instance using the Amazon RDS API or the AWS CLI. The default backup retention period is seven days if you create the DB instance using the console.

After you create a DB instance, you can modify the backup retention period. You can set the backup retention period to between 0 and 35 days. Setting the backup retention period to 0 disables automated backups. Manual snapshot limits (100 per Region) do not apply to automated backups.

Automated backups aren't created while a DB instance is stopped. Backups can be retained longer than the backup retention period if a DB instance has been stopped. RDS doesn't include time spent in the stopped state when the backup retention window is calculated.

Important

An outage occurs if you change the backup retention period from 0 to a nonzero value or from a nonzero value to 0. This applies to both Single-AZ and Multi-AZ DB instances.

Enabling automated backups

If your DB instance doesn't have automated backups enabled, you can enable them at any time. You enable automated backups by setting the backup retention period to a positive nonzero value. When automated backups are enabled, your RDS instance and database is taken offline and a backup is immediately created.

Note

If you manage your backups in AWS Backup, you can't enable automated backups. For more information, see [Using AWS Backup to manage automated backups \(p. 335\)](#).

Console

To enable automated backups immediately

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
3. Choose **Modify**. The **Modify DB instance** page appears.
4. For **Backup retention period**, choose a positive nonzero value, for example 3 days.
5. Choose **Continue**.
6. Choose **Apply immediately**.
7. On the confirmation page, choose **Modify DB instance** to save your changes and enable automated backups.

AWS CLI

To enable automated backups, use the AWS CLI `modify-db-instance` command.

Include the following parameters:

- `--db-instance-identifier`
- `--backup-retention-period`
- `--apply-immediately` or `--no-apply-immediately`

In the following example, we enable automated backups by setting the backup retention period to three days. The changes are applied immediately.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
  --db-instance-identifier mydbinstance \
  --backup-retention-period 3 \
  --apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --backup-retention-period 3 ^
  --apply-immediately
```

RDS API

To enable automated backups, use the RDS API [ModifyDBInstance](#) operation with the following required parameters:

- `DBInstanceIdentifier`
- `BackupRetentionPeriod`

Retaining automated backups

When you delete a DB instance, you can retain automated backups.

Retained automated backups contain system snapshots and transaction logs from a DB instance. They also include your DB instance properties like allocated storage and DB instance class, which are required to restore it to an active instance.

You can retain automated backups for RDS instances running the MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server engines.

You can restore or remove retained automated backups using the AWS Management Console, RDS API, and AWS CLI.

Topics

- [Retention period \(p. 332\)](#)
- [Viewing retained backups \(p. 332\)](#)
- [Restoration \(p. 332\)](#)
- [Retention costs \(p. 332\)](#)
- [Limitations and recommendations \(p. 333\)](#)

Retention period

The system snapshots and transaction logs in a retained automated backup expire the same way that they expire for the source DB instance. Because there are no new snapshots or logs created for this instance, the retained automated backups eventually expire completely. Effectively, they live as long their last system snapshot would have done, based on the settings for retention period the source instance had when you deleted it. Retained automated backups are removed by the system after their last system snapshot expires.

You can remove a retained automated backup in the same way that you can delete a DB instance. You can remove retained automated backups using the console or the RDS API operation `DeleteDBInstanceAutomatedBackup`.

Final snapshots are independent of retained automated backups. We strongly suggest that you take a final snapshot even if you retain automated backups, because the retained automated backups eventually expire. The final snapshot doesn't expire.

Viewing retained backups

To view your retained automated backups, switch to the automated backups page. You can view individual snapshots associated with a retained automated backup on the database snapshots page in the console. Alternatively, you can describe individual snapshots associated with a retained automated backup. From there, you can restore a DB instance directly from one of those snapshots.

To describe your retained automated backups using the AWS CLI, use one of the following commands:

```
aws rds describe-db-instance-automated-backups --db-instance-  
identifier DBInstanceIdentifier
```

or

```
aws rds describe-db-instance-automated-backups --dbi-resource-id DbiResourceId
```

To describe your retained automated backups using the RDS API, call the `DescribeDBInstanceAutomatedBackups` action with one of the following parameters:

- `DBInstanceIdentifier`
- `DbiResourceId`

Restoration

For information on restoring DB instances from automated backups, see [Restoring a DB instance to a specified time \(p. 389\)](#).

Retention costs

The cost of a retained automated backup is the cost of total storage of the system snapshots that are associated with it. There is no additional charge for transaction logs or instance metadata. All other pricing rules for backups apply to restorable instances.

For example, suppose that your total allocated storage of running instances is 100 GB. Suppose also that you have 50 GB of manual snapshots plus 75 GB of system snapshots associated with a retained automated backup. In this case, you are charged only for the additional 25 GB of backup storage, like this: $(50 \text{ GB} + 75 \text{ GB}) - 100 \text{ GB} = 25 \text{ GB}$.

Limitations and recommendations

The following limitations apply to retained automated backups:

- The maximum number of retained automated backups in one AWS Region is 40. It's not included in the DB instances limit. You can have 40 running DB instances and an additional 40 retained automated backups at the same time.
- Retained automated backups don't contain information about parameters or option groups.
- You can restore a deleted instance to a point in time that is within the retention period at the time of delete.
- You can't modify a retained automated backup. That's because it consists of system backups, transaction logs, and the DB instance properties that existed at the time that you deleted the source instance.

Deleting retained automated backups

You can delete retained automated backups when they are no longer needed.

Console

To delete a retained automated backup

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Automated backups**.
3. On the **Retained** tab, choose the retained automated backup that you want to delete.
4. For **Actions**, choose **Delete**.
5. On the confirmation page, enter **delete me** and choose **Delete**.

AWS CLI

You can delete a retained automated backup by using the AWS CLI command `delete-db-instance-automated-backup` with the following option:

- `--dbi-resource-id` – The resource identifier for the source DB instance.

You can find the resource identifier for the source DB instance of a retained automated backup by running the AWS CLI command `describe-db-instance-automated-backups`.

Example

The following example deletes the retained automated backup with source DB instance resource identifier `db-123ABCEXAMPLE`.

For Linux, macOS, or Unix:

```
aws rds delete-db-instance-automated-backup \
--dbi-resource-id db-123ABCEXAMPLE
```

For Windows:

```
aws rds delete-db-instance-automated-backup ^
```

```
--dbi-resource-id db-123ABCEXAMPLE
```

RDS API

You can delete a retained automated backup by using the Amazon RDS API operation [DeleteDBInstanceAutomatedBackup](#) with the following parameter:

- **DbiResourceId** – The resource identifier for the source DB instance.

You can find the resource identifier for the source DB instance of a retained automated backup using the Amazon RDS API operation [DescribeDBInstanceAutomatedBackups](#).

Disabling automated backups

You might want to temporarily disable automated backups in certain situations, for example while loading large amounts of data.

Important

We highly discourage disabling automated backups because it disables point-in-time recovery. Disabling automatic backups for a DB instance deletes all existing automated backups for the instance. If you disable and then re-enable automated backups, you can restore starting only from the time you re-enabled automated backups.

Console

To disable automated backups immediately

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
3. Choose **Modify**. The **Modify DB instance** page appears.
4. For **Backup retention period**, choose **0 days**.
5. Choose **Continue**.
6. Choose **Apply immediately**.
7. On the confirmation page, choose **Modify DB instance** to save your changes and disable automated backups.

AWS CLI

To disable automated backups immediately, use the `modify-db-instance` command and set the backup retention period to 0 with `--apply-immediately`.

Example

The following example immediately disabled automatic backups.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--backup-retention-period 0 \
--apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--backup-retention-period 0 ^
--apply-immediately
```

To know when the modification is in effect, call `describe-db-instances` for the DB instance until the value for backup retention period is 0 and *mydbinstance* status is available.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

RDS API

To disable automated backups immediately, call the [ModifyDBInstance](#) operation with the following parameters:

- `DBInstanceIdentifier = mydbinstance`
- `BackupRetentionPeriod = 0`

Example

```
https://rds.amazonaws.com/
?Action=ModifyDBInstance
&DBInstanceIdentifier=mydbinstance
&BackupRetentionPeriod=0
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2009-10-14T17%3A48%3A21.746Z
&AWSAccessKeyId=<AWS Access Key ID>
&Signature=<Signature>
```

Using AWS Backup to manage automated backups

AWS Backup is a fully managed backup service that makes it easy to centralize and automate the backup of data across AWS services in the cloud and on premises. You can manage backups of your Amazon RDS DB instances in AWS Backup.

To enable backups in AWS Backup, you use resource tagging to associate your DB instance with a backup plan. For more information, see [Using tags to enable backups in AWS Backup \(p. 306\)](#).

Note

Backups managed by AWS Backup are considered manual DB snapshots, but don't count toward the DB snapshot quota for RDS. Backups that were created with AWS Backup have names ending in `awsbackup :AWS-Backup-job-number`.

For more information about AWS Backup, see the [AWS Backup Developer Guide](#).

To view backups managed by AWS Backup

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. Choose the **Backup service** tab.

Your AWS Backup backups are listed under **Backup service snapshots**.

Automated backups with unsupported MySQL storage engines

For the MySQL DB engine, automated backups are only supported for the InnoDB storage engine. Use of these features with other MySQL storage engines, including MyISAM, can lead to unreliable behavior while restoring from backups. Specifically, since storage engines like MyISAM don't support reliable crash recovery, your tables can be corrupted in the event of a crash. For this reason, we encourage you to use the InnoDB storage engine.

- To convert existing MyISAM tables to InnoDB tables, you can use the `ALTER TABLE` command, for example: `ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;`
- If you choose to use MyISAM, you can attempt to manually repair tables that become damaged after a crash by using the `REPAIR` command. For more information, see [REPAIR TABLE statement](#) in the MySQL documentation. However, as noted in the MySQL documentation, there is a good chance that you might not be able to recover all your data.
- If you want to take a snapshot of your MyISAM tables before restoring, follow these steps:
 1. Stop all activity to your MyISAM tables (that is, close all sessions).

You can close all sessions by calling the `mysql.rds_kill` command for each process that is returned from the `SHOW FULL PROCESSLIST` command.

2. Lock and flush each of your MyISAM tables. For example, the following commands lock and flush two tables named `myisam_table1` and `myisam_table2`:

```
mysql> FLUSH TABLES myisam_table, myisam_table2 WITH READ LOCK;
```

3. Create a snapshot of your DB instance. When the snapshot has completed, release the locks and resume activity on the MyISAM tables. You can release the locks on your tables using the following command:

```
mysql> UNLOCK TABLES;
```

These steps force MyISAM to flush data stored in memory to disk, which ensures a clean start when you restore from a DB snapshot. For more information on creating a DB snapshot, see [Creating a DB snapshot \(p. 346\)](#).

Automated backups with unsupported MariaDB storage engines

For the MariaDB DB engine, automated backups are only supported with the InnoDB storage engine (version 10.2 and later) and XtraDB storage engine (versions 10.0 and 10.1). Use of these features with other MariaDB storage engines, including Aria, might lead to unreliable behavior while restoring from backups. Even though Aria is a crash-resistant alternative to MyISAM, your tables can still be corrupted in the event of a crash. For this reason, we encourage you to use the XtraDB storage engine.

- To convert existing Aria tables to InnoDB tables, you can use the `ALTER TABLE` command. For example: `ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;`
- To convert existing Aria tables to XtraDB tables, you can use the `ALTER TABLE` command. For example: `ALTER TABLE table_name ENGINE=xtradb, ALGORITHM=COPY;`
- If you choose to use Aria, you can attempt to manually repair tables that become damaged after a crash by using the `REPAIR TABLE` command. For more information, see <http://mariadb.com/kb/en/mariadb/repair-table/>.

- If you want to take a snapshot of your Aria tables before restoring, follow these steps:
 1. Stop all activity to your Aria tables (that is, close all sessions).
 2. Lock and flush each of your Aria tables.
 3. Create a snapshot of your DB instance. When the snapshot has completed, release the locks and resume activity on the Aria tables. These steps force Aria to flush data stored in memory to disk, thereby ensuring a clean start when you restore from a DB snapshot.

Replicating automated backups to another AWS Region

For added disaster recovery capability, you can configure your Amazon RDS database instance to replicate snapshots and transaction logs to a destination AWS Region of your choice. When backup replication is configured for a DB instance, RDS initiates a cross-Region copy of all snapshots and transaction logs as soon as they are ready on the DB instance.

DB snapshot copy charges apply to the data transfer. After the DB snapshot is copied, standard charges apply to storage in the destination Region. For more details, see [RDS Pricing](#).

Backup replication is available for RDS DB instances running the following database engines:

- Oracle version 12.1.0.2.v10 and higher
- PostgreSQL version 9.6 and higher

Backup replication isn't supported for encrypted DB instances.

Enabling cross-Region automated backups

You can enable backup replication on new or existing DB instances using the Amazon RDS console. You can also use the `start-db-instance-automated-backups-replication` AWS CLI command or the `StartDBInstanceAutomatedBackupsReplication` RDS API operation.

Note

To be able to replicate automated backups, make sure to enable them. For more information, see [Enabling automated backups \(p. 330\)](#).

You can use the `describe-source-regions` CLI command to list the source AWS Regions that can replicate automated backups to a particular destination Region. For more information, see [Finding information about replicated backups \(p. 339\)](#).

Console

You can enable backup replication for a new or existing DB instance:

- For a new DB instance, enable it when you launch the instance. For more information, see [Settings for DB instances \(p. 145\)](#).
- For an existing DB instance, use the following procedure.

To enable backup replication for an existing DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Automated backups**.
3. On the **Current Region** tab, choose the DB instance for which you want to enable backup replication.
4. For **Actions**, choose **Manage cross-Region replication**.
5. Under **Backup replication**, choose **Enable replication to another AWS Region**.
6. Choose the **Destination Region**.
7. Choose the **Replicated backup retention period**.
8. Choose **Save**.

In the source Region, replicated backups are listed on the **Current Region** tab of the **Automated backups** page. In the destination Region, replicated backups are listed on the **Replicated backups** tab of the **Automated backups** page.

AWS CLI

Enable backup replication by using the [start-db-instance-automated-backups-replication](#) AWS CLI command.

The following CLI example replicates automated backups from a DB instance in the US West (Oregon) Region to the US East (N. Virginia) Region.

To enable backup replication

- Run one of the following commands.

For Linux, macOS, or Unix:

```
aws rds start-db-instance-automated-backups-replication \
--region us-east-1 \
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase" \
--backup-retention-period 7
```

For Windows:

```
aws rds start-db-instance-automated-backups-replication ^
--region us-east-1 ^
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase" ^
--backup-retention-period 7
```

RDS API

Enable backup replication by using the [StartDBInstanceAutomatedBackupsReplication](#) RDS API operation with the following parameters:

- Region
- SourceDBInstanceArn
- BackupRetentionPeriod

Finding information about replicated backups

You can use the following CLI commands to find information about replicated backups:

- [describe-source-regions](#)
- [describe-db-instances](#)
- [describe-db-instance-automated-backups](#)

The following `describe-source-regions` example lists the source AWS Regions from which automated backups can be replicated to the US West (Oregon) destination Region.

To show information about source Regions

- Run the following command.

```
aws rds describe-source-regions --region us-west-2
```

The output shows that backups can be replicated from Asia Pacific (Tokyo), but not from Asia Pacific (Seoul), into US West (Oregon).

```
{  
    "SourceRegions": [  
        {  
            "RegionName": "ap-northeast-1",  
            "Endpoint": "https://rds.ap-northeast-1.amazonaws.com",  
            "Status": "available",  
            "SupportsDBInstanceAutomatedBackupsReplication": true  
        },  
        {  
            "RegionName": "ap-northeast-2",  
            "Endpoint": "https://rds.ap-northeast-2.amazonaws.com",  
            "Status": "available",  
            "SupportsDBInstanceAutomatedBackupsReplication": false  
        }  
        ...  
    ]  
}
```

The following `describe-db-instances` example shows the automated backups for a DB instance.

To show the replicated backups for a DB instance

- Run one of the following commands.

For Linux, macOS, or Unix:

```
aws rds describe-db-instances \  
--db-instance-identifier mydatabase
```

For Windows:

```
aws rds describe-db-instances ^  
--db-instance-identifier mydatabase
```

The output includes the replicated backups.

```
{  
    "DBInstances": [  
        {  
            "StorageEncrypted": false,  
            "Endpoint": {  
                "HostedZoneId": "Z1PVIF0B656C1W",  
                "Port": 1521,  
                ...  
            },  
            "BackupRetentionPeriod": 7,  
            "DBInstanceAutomatedBackupsReplications": [{"DBInstanceAutomatedBackupsArn":  
"arn:aws:rds:us-east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEEXAMPLE"}]  
        }  
    ]  
}
```

The following `describe-db-instance-automated-backups` example shows the automated backups for a DB instance.

To show automated backups for a DB instance

- Run one of the following commands.

For Linux, macOS, or Unix:

```
aws rds describe-db-instance-automated-backups \
--db-instance-identifier mydatabase
```

For Windows:

```
aws rds describe-db-instance-automated-backups ^
--db-instance-identifier mydatabase
```

The output shows the source DB instance and automated backups in US West (Oregon), with backups replicated to US East (N. Virginia).

```
{
    "DBInstanceAutomatedBackups": [
        {
            "DBInstanceArn": "arn:aws:rds:us-west-2:868710585169:db:mydatabase",
            "DbiResourceId": "db-L2IJCEXJP7XQ7HOJ4SIEEXAMPLE",
            "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEEXAMPLE",
            "BackupRetentionPeriod": 7,
            "DBInstanceAutomatedBackupsReplications": [{"DBInstanceAutomatedBackupsArn": "
"arn:aws:rds:us-east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEEXAMPLE"}]
            "Region": "us-west-2",
            "DBInstanceIdentifier": "mydatabase",
            "RestoreWindow": {
                "EarliestTime": "2020-10-26T01:09:07Z",
                "LatestTime": "2020-10-31T19:09:53Z",
            }
            ...
        }
    ]
}
```

The following `describe-db-instance-automated-backups` example uses the `--db-instance-automated-backups-arn` option to show the replicated backups in the destination Region.

To show replicated backups

- Run one of the following commands.

For Linux, macOS, or Unix:

```
aws rds describe-db-instance-automated-backups \
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7HOJ4SIEEXAMPLE"
```

For Windows:

```
aws rds describe-db-instance-automated-backups ^
```

```
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEEXAMPLE"
```

The output shows the source DB instance in US West (Oregon), with replicated backups in US East (N. Virginia).

```
{  
    "DBInstanceAutomatedBackups": [  
        {  
            "DBInstanceArn": "arn:aws:rds:us-west-2:868710585169:db:mydatabase",  
            "DbiResourceId": "db-L2IJCEXJP7XQ7HOJ4SIEEXAMPLE",  
            "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEEXAMPLE",  
            "Region": "us-west-2",  
            "DBInstanceIdentifier": "mydatabase",  
            "RestoreWindow": {  
                "EarliestTime": "2020-10-26T01:09:07Z",  
                "LatestTime": "2020-10-31T19:01:23Z"  
            },  
            "AllocatedStorage": 50,  
            "BackupRetentionPeriod": 7,  
            "Status": "replicating",  
            "Port": 1521,  
            ...  
        }  
    ]  
}
```

Restoring to a specified time from a replicated backup

You can restore a DB instance to a specific point in time from a replicated backup using the Amazon RDS console. You can also use the `restore-db-instance-to-point-in-time` AWS CLI command or the `RestoreDBInstanceToPointInTime` RDS API operation.

For general information on point-in-time recovery (PITR), see [Restoring a DB instance to a specified time \(p. 389\)](#).

Console

To restore a DB instance to a specified time from a replicated backup

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose the destination Region (where backups are replicated to) from the Region selector.
3. In the navigation pane, choose **Automated backups**.
4. On the **Replicated backups** tab, choose the DB instance that you want to restore.
5. For **Actions**, choose **Restore to point in time**.
6. Choose **Latest restorable time** to restore to the latest possible time, or choose **Custom** to choose a time.

If you chose **Custom**, enter the date and time that you want to restore the instance to.

Note

Times are shown in your local time zone, which is indicated by an offset from Coordinated Universal Time (UTC). For example, UTC-5 is Eastern Standard Time/Central Daylight Time.

7. For **DB instance identifier**, enter the name of the target restored DB instance.
8. (Optional) Choose other options as needed, such as enabling autoscaling.
9. Choose **Restore to point in time**.

AWS CLI

Use the `restore-db-instance-to-point-in-time` AWS CLI command to create a new DB instance.

To restore a DB instance to a specified time from a replicated backup

- Run one of the following commands.

For Linux, macOS, or Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-automated-backups-arn "arn:aws:rds:us-
east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEEXAMPLE" \
    --target-db-instance-identifier mytargetdbinstance \
    --restore-time 2020-10-14T23:45:00.000Z
```

For Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-automated-backups-arn "arn:aws:rds:us-
east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEEXAMPLE" ^
    --target-db-instance-identifier mytargetdbinstance ^
    --restore-time 2020-10-14T23:45:00.000Z
```

RDS API

To restore a DB instance to a specified time, call the `RestoreDBInstanceToPointInTime` Amazon RDS API operation with the following parameters:

- `SourceDBInstanceAutomatedBackupsArn`
- `TargetDBInstanceIdentifier`
- `RestoreTime`

Stopping automated backup replication

You can stop backup replication for DB instances using the Amazon RDS console. You can also use the `stop-db-instance-automated-backups-replication` AWS CLI command or the `StopDBInstanceAutomatedBackupsReplication` RDS API operation.

Replicated backups are retained, subject to the backup retention period set when they were created.

Console

Stop backup replication from the **Automated backups** page in the source Region.

To stop backup replication to an AWS Region

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.

2. Choose the source Region from the **Region selector**.
3. In the navigation pane, choose **Automated backups**.
4. On the **Current Region** tab, choose the DB instance for which you want to stop backup replication.
5. For **Actions**, choose **Manage cross-Region replication**.
6. Under **Backup replication**, clear the **Enable replication to another AWS Region** check box.
7. Choose **Save**.

Replicated backups are listed on the **Retained** tab of the **Automated backups** page in the destination Region.

AWS CLI

Stop backup replication by using the `stop-db-instance-automated-backups-replication` AWS CLI command.

The following CLI example stops automated backups of a DB instance from replicating in the US West (Oregon) Region.

To stop backup replication

- Run one of the following commands.

For Linux, macOS, or Unix:

```
aws rds stop-db-instance-automated-backups-replication \
--region us-east-1 \
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase"
```

For Windows:

```
aws rds stop-db-instance-automated-backups-replication ^
--region us-east-1 ^
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase"
```

RDS API

Stop backup replication by using the `StopDBInstanceAutomatedBackupsReplication` RDS API operation with the following parameters:

- `Region`
- `SourceDBInstanceArn`

Deleting replicated backups

You can delete replicated backups for DB instances using the Amazon RDS console. You can also use the `delete-db-instance-automated-backups` AWS CLI command or the `DeleteDBInstanceAutomatedBackup` RDS API operation.

Console

Delete replicated backups in the destination Region from the **Automated backups** page.

To delete replicated backups

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose the destination Region from the **Region selector**.
3. In the navigation pane, choose **Automated backups**.
4. On the **Replicated backups** tab, choose the DB instance for which you want to delete the replicated backups.
5. For **Actions**, choose **Delete**.
6. On the confirmation page, enter **delete me** and choose **Delete**.

AWS CLI

Delete replicated backups by using the `delete-db-instance-automated-backup` AWS CLI command.

You can use the `describe-db-instances` CLI command to find the Amazon Resource Names (ARNs) of the replicated backups. For more information, see [Finding information about replicated backups \(p. 339\)](#).

To delete replicated backups

- Run one of the following commands.

For Linux, macOS, or Unix:

```
aws rds delete-db-instance-automated-backup \
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7HOJ4SIEEXAMPLE"
```

For Windows:

```
aws rds delete-db-instance-automated-backup ^
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7HOJ4SIEEXAMPLE"
```

RDS API

Delete replicated backups by using the `DeleteDBInstanceAutomatedBackup` RDS API operation with the `DBInstanceAutomatedBackupsArn` parameter.

Creating a DB snapshot

Amazon RDS creates a storage volume snapshot of your DB instance, backing up the entire DB instance and not just individual databases. Creating this DB snapshot on a Single-AZ DB instance results in a brief I/O suspension that can last from a few seconds to a few minutes, depending on the size and class of your DB instance. For MariaDB, MySQL, Oracle, and PostgreSQL, I/O activity is not suspended on your primary during backup for Multi-AZ deployments, because the backup is taken from the standby. For SQL Server, I/O activity is suspended briefly during backup for Multi-AZ deployments.

When you create a DB snapshot, you need to identify which DB instance you are going to back up, and then give your DB snapshot a name so you can restore from it later. The amount of time it takes to create a snapshot varies with the size of your databases. Since the snapshot includes the entire storage volume, the size of files, such as temporary files, also affects the amount of time it takes to create the snapshot.

Note

For PostgreSQL DB instances, data in unlogged tables might not be restored from snapshots. For more information, see [Best practices for working with PostgreSQL \(p. 137\)](#).

Unlike automated backups, manual snapshots aren't subject to the backup retention period. Snapshots don't expire.

For very long-term backups of MariaDB, MySQL, and PostgreSQL data, we recommend exporting snapshot data to Amazon S3. If the major version of your DB engine is no longer supported, you can't restore to that version from a snapshot. For more information, see [Exporting DB snapshot data to Amazon S3 \(p. 373\)](#).

You can create a DB snapshot using the AWS Management Console, the AWS CLI, or the RDS API.

Console

To create a DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. In the list of DB instances, choose the DB instance for which you want to take a snapshot.
4. For **Actions**, choose **Take snapshot**.

The **Take DB Snapshot** window appears.

5. Type the name of the snapshot in the **Snapshot Name** box.

Take DB Snapshot

This feature is currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#). 

Settings

To take a snapshot of this DB instance you must provide a name for the snapshot.

DB instance

The unique key that identifies a DB instance. This parameter isn't case-sensitive.

`mydbinstance3`

Snapshot name

The Identifier for the DB Snapshot.

[Cancel](#)

[Take Snapshot](#)

- Choose **Take Snapshot**.

AWS CLI

When you create a DB snapshot using the AWS CLI, you need to identify which DB instance you are going to back up, and then give your DB snapshot a name so you can restore from it later. You can do this by using the AWS CLI `create-db-snapshot` command with the following parameters:

- `--db-instance-identifier`
- `--db-snapshot-identifier`

In this example, you create a DB snapshot called `mydbsnapshot` for a DB instance called `mydbinstance`.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-snapshot \
--db-instance-identifier mydbinstance \
--db-snapshot-identifier mydbsnapshot
```

For Windows:

```
aws rds create-db-snapshot ^
--db-instance-identifier mydbinstance ^
--db-snapshot-identifier mydbsnapshot
```

RDS API

When you create a DB snapshot using the Amazon RDS API, you need to identify which DB instance you are going to back up, and then give your DB snapshot a name so you can restore from it later. You can do this by using the Amazon RDS API `CreateDBSnapshot` command with the following parameters:

- **DBInstanceIdentifier**
- **DBSnapshotIdentifier**

Restoring from a DB snapshot

Amazon RDS creates a storage volume snapshot of your DB instance, backing up the entire DB instance and not just individual databases. You can create a DB instance by restoring from this DB snapshot. When you restore the DB instance, you provide the name of the DB snapshot to restore from, and then provide a name for the new DB instance that is created from the restore. You can't restore from a DB snapshot to an existing DB instance; a new DB instance is created when you restore.

You can restore a DB instance and use a different storage type than the source DB snapshot. In this case, the restoration process is slower because of the additional work required to migrate the data to the new storage type. If you restore to or from magnetic storage, the migration process is the slowest. That's because magnetic storage doesn't have the IOPS capability of Provisioned IOPS or General Purpose (SSD) storage.

Note

You can't restore a DB instance from a DB snapshot that is both shared and encrypted. Instead, you can make a copy of the DB snapshot and restore the DB instance from the copy. For more information, see [Copying a snapshot \(p. 352\)](#).

Parameter group considerations

We recommend that you retain the parameter group for any DB snapshots you create, so that you can associate your restored DB instance with the correct parameter group. You can specify the parameter group when you restore the DB instance.

Security group considerations

When you restore a DB instance, the default security group is associated with the restored instance by default.

Note

- If you're using the Amazon RDS console, you can specify a custom security group to associate with the instance or create a new VPC security group.
- If you're using the AWS CLI, you can specify a custom security group to associate with the instance by including the `--vpc-security-group-ids` option in the `restore-db-instance-from-db-snapshot` command.
- If you're using the Amazon RDS API, you can include the `VpcSecurityGroupIds.VpcSecurityGroupId.N` parameter in the `RestoreDBInstanceFromDBSnapshot` action.

As soon as the restore is complete and your new DB instance is available, you can associate any custom security groups used by the snapshot you restored from. You must apply these changes by modifying the DB instance with the RDS console, the AWS CLI `modify-db-instance` command, or the `ModifyDBInstance` Amazon RDS API operation. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Option group considerations

When you restore a DB instance, the option group associated with the DB snapshot is associated with the restored DB instance after it is created. For example, if the DB snapshot you are restoring from uses Oracle Transparent Data Encryption, the restored DB instance will use the same option group.

When you assign an option group to a DB instance, the option group is also linked to the supported platform the DB instance is on, either VPC or EC2-Classic (non-VPC). If a DB instance is in a VPC, the

option group associated with the DB instance is linked to that VPC. This means that you can't use the option group assigned to a DB instance if you attempt to restore the instance into a different VPC or onto a different platform. If you restore a DB instance into a different VPC or onto a different platform, you must either assign the default option group to the instance, assign an option group that is linked to that VPC or platform, or create a new option group and assign it to the DB instance. For persistent or permanent options, when restoring a DB instance into a different VPC you must create a new option group that includes the persistent or permanent option.

Microsoft SQL Server considerations

When you restore a Microsoft SQL Server DB snapshot to a new instance, you can always restore to the same edition as your snapshot. In some cases, you can also change the edition of the DB instance. The following are the limitations when you change editions:

- The DB snapshot must have enough storage allocated for the new edition.
- Only the following edition changes are supported:
 - From Standard Edition to Enterprise Edition
 - From Web Edition to Standard Edition or Enterprise Edition
 - From Express Edition to Web Edition, Standard Edition or Enterprise Edition

If you want to change from one edition to a new edition that is not supported by restoring a snapshot, you can try using the native backup and restore feature. SQL Server verifies whether or not your database is compatible with the new edition based on what SQL Server features you have enabled on the database. For more information, see [Importing and exporting SQL Server databases \(p. 671\)](#).

Oracle considerations

If you use Oracle GoldenGate, always retain the parameter group with the compatible parameter. When you restore a DB instance from a DB snapshot, you must specify the parameter group that has a matching or greater compatible parameter value.

You can upgrade a DB snapshot while it is still a DB snapshot, before you restore it. For more information, see [Upgrading an Oracle DB snapshot \(p. 1217\)](#).

Restoring from a snapshot

You can restore a DB instance from a DB snapshot using the AWS Management Console, the AWS CLI, or the RDS API.

Console

To restore a DB instance from a DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. Choose the DB snapshot that you want to restore from.
4. For **Actions**, choose **Restore snapshot**.
5. On the **Restore snapshot** page, for **DB instance identifier**, enter the name for your restored DB instance.
6. Choose **Restore DB instance**.

AWS CLI

To restore a DB instance from a DB snapshot, use the AWS CLI command [restore-db-instance-from-db-snapshot](#).

In this example, you restore from a previously created DB snapshot named `mydbsnapshot`. You restore to a new DB instance named `mynewdbinstance`.

Example

For Linux, macOS, or Unix:

```
aws rds restore-db-instance-from-db-snapshot \
--db-instance-identifier mynewdbinstance \
--db-snapshot-identifier mydbsnapshot
```

For Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
--db-instance-identifier mynewdbinstance ^
--db-snapshot-identifier mydbsnapshot
```

This command returns output similar to the following:

```
DBINSTANCE mynewdbinstance db.m3.large MySQL      50          sa           creating  3  n
5.6.40 general-public-license
```

RDS API

To restore a DB instance from a DB snapshot, call the Amazon RDS API function [RestoreDBInstanceFromDBSnapshot](#) with the following parameters:

- `DBInstanceIdentifier`
- `DBSnapshotIdentifier`

Copying a snapshot

With Amazon RDS, you can copy automated or manual DB snapshots. After you copy a snapshot, the copy is a manual snapshot.

You can copy a snapshot within the same AWS Region, you can copy a snapshot across AWS Regions, and you can copy shared snapshots.

Limitations

The following are some limitations when you copy snapshots:

- You can't copy a snapshot to or from the China (Beijing) or China (Ningxia) Regions.
- You can copy a snapshot between AWS GovCloud (US-East) and AWS GovCloud (US-West). However, you can't copy a snapshot between these AWS GovCloud (US) Regions and commercial AWS Regions.
- If you delete a source snapshot before the target snapshot becomes available, the snapshot copy might fail. Verify that the target snapshot has a status of **AVAILABLE** before you delete a source snapshot.
- You can have up to five snapshot copy requests in progress to a single destination Region per account.
- Depending on the AWS Regions involved and the amount of data to be copied, a cross-Region snapshot copy can take hours to complete. In some cases, there might be a large number of cross-Region snapshot copy requests from a given source Region. In such cases, Amazon RDS might put new cross-Region copy requests from that source Region into a queue until some in-progress copies complete. No progress information is displayed about copy requests while they are in the queue. Progress information is displayed when the copy starts.

Snapshot retention

Amazon RDS deletes automated snapshots in several situations:

- At the end of their retention period.
- When you disable automated snapshots for a DB instance.
- When you delete a DB instance.

If you want to keep an automated snapshot for a longer period, copy it to create a manual snapshot, which is retained until you delete it. Amazon RDS storage costs might apply to manual snapshots if they exceed your default storage space.

For more information about backup storage costs, see [Amazon RDS pricing](#).

Copying shared snapshots

You can copy snapshots shared to you by other AWS accounts. In some cases, you might copy an encrypted snapshot that has been shared from another AWS account. In these cases, you must have access to the AWS KMS customer master key (CMK) that was used to encrypt the snapshot.

You can copy a shared DB snapshot across AWS Regions if the snapshot is unencrypted. However, if the shared DB snapshot is encrypted, you can only copy it in the same Region.

Note

Copying shared incremental snapshots in the same AWS Region is supported when they're unencrypted, or encrypted using the same AWS KMS key as the initial full snapshot. If you use a

different KMS key to encrypt subsequent snapshots when copying them, those shared snapshots are full snapshots.

Handling encryption

You can copy a snapshot that has been encrypted using an AWS KMS customer master key (CMK). If you copy an encrypted snapshot, the copy of the snapshot must also be encrypted. If you copy an encrypted snapshot within the same AWS Region, you can encrypt the copy with the same AWS KMS CMK as the original snapshot. Or you can specify a different CMK. If you copy an encrypted snapshot across Regions, you can't use the same AWS KMS CMK for the copy as used for the source snapshot. This is because AWS KMS CMKs are Region-specific. Instead, you must specify an AWS KMS CMK valid in the destination AWS Region.

The source snapshot remains encrypted throughout the copy process. For more information, see [Limitations of Amazon RDS encrypted DB instances \(p. 1632\)](#).

You can also encrypt a copy of an unencrypted snapshot. This way, you can quickly add encryption to a previously unencrypted DB instance.

That is, you can create a snapshot of your DB instance when you are ready to encrypt it. You then create a copy of that snapshot and specify an AWS KMS CMK to encrypt that snapshot copy. You can then restore an encrypted DB instance from the encrypted snapshot.

Incremental snapshot copying

An *incremental* snapshot contains only the data that has changed after the most recent snapshot of the same DB instance. Incremental snapshot copying is faster and results in lower storage costs than full snapshot copying.

Note

When you copy a source snapshot that is a snapshot copy itself, the new copy isn't incremental. This is because the source snapshot copy doesn't include the required metadata for incremental copies.

Whether a snapshot copy is incremental is determined by the most recently completed snapshot copy. If the most recent snapshot copy was deleted, the next copy is a full copy, not an incremental copy. If a copy is still pending when you start another copy, the second copy doesn't start until the first copy finishes.

When you copy a snapshot across AWS accounts, the copy is an incremental copy if the following conditions are met:

- The snapshot was previously copied to the destination account.
- The most recent snapshot copy still exists in the destination account.
- All copies of the snapshot in the destination account are either unencrypted, or were encrypted using the same CMK.

For shared snapshots, copying incremental snapshots across AWS accounts is only supported when they're unencrypted.

For information on copying incremental snapshots across AWS Regions, see [Full and incremental copies \(p. 356\)](#).

Cross-Region snapshot copying

You can copy DB snapshots across AWS Regions. However, there are certain constraints and considerations for cross-Region snapshot copying.

Requesting a cross-Region DB snapshot copy

To communicate with the source Region to request a cross-Region DB snapshot copy, the requester (IAM role or IAM user) must have access to the source DB snapshot and the source Region.

Certain conditions in the requester's IAM policy can cause the request to fail. The following examples assume that you're copying the DB snapshot from US East (Ohio) to US East (N. Virginia). These examples show conditions in the requester's IAM policy that cause the request to fail:

- The requester's policy has a condition for `aws:RequestedRegion`.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": "us-east-1"
    }
}
```

The request fails because the policy doesn't allow access to the source Region. For a successful request, specify both the source and destination Regions.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": [
            "us-east-1",
            "us-east-2"
        ]
    }
}
```

- The requester's policy doesn't allow access to the source DB snapshot.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "arn:aws:rds:us-east-1:123456789012:snapshot:target-snapshot"
...
```

For a successful request, specify both the source and target snapshots.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": [
    "arn:aws:rds:us-east-1:123456789012:snapshot:target-snapshot",
    "arn:aws:rds:us-east-2:123456789012:snapshot:source-snapshot"
]
...
```

- The requester's policy denies `aws:ViaAWSService`.

```
...
"Effect": "Allow",
```

```
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "Bool": {"aws:ViaAWSService": "false"}
}
```

Communication with the source Region is made by RDS on the requester's behalf. For a successful request, don't deny calls made by AWS services.

- The requester's policy has a condition for `aws:SourceVpc` or `aws:SourceVpce`.

These requests might fail because when RDS makes the call to the remote Region, it isn't from the specified VPC or VPC endpoint.

If you need to use one of the previous conditions that would cause a request to fail, you can include a second statement with `aws:CalledVia` in your policy to make the request succeed. For example, you can use `aws:CalledVia` with `aws:SourceVpce` as shown here:

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "Condition" : {
        "ForAnyValue:StringEquals" : {
            "aws:SourceVpce": "vpce-1a2b3c4d"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "rds:CopyDBSnapshot"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": [
                "rds.amazonaws.com"
            ]
        }
    }
}
```

For more information, see [Policies and permissions in IAM](#) in the *IAM User Guide*.

Authorizing the snapshot copy

After a cross-Region DB snapshot copy request returns success, RDS starts the copy in the background. An authorization for RDS to access the source snapshot is created. This authorization links the source DB snapshot to the target DB snapshot, and allows RDS to copy only to the specified target snapshot.

The authorization is verified by RDS using the `rds:CrossRegionCommunication` permission in the service-linked IAM role. If the copy is authorized, RDS communicates with the source Region and completes the copy.

RDS doesn't have access to DB snapshots that weren't authorized previously by a `CopyDBSnapshot` request. The authorization is revoked when copying completes.

RDS uses the service-linked role to verify the authorization in the source Region. If you delete the service-linked role during the copy process, the copy fails.

For more information, see [Using service-linked roles in the IAM User Guide](#).

Using AWS Security Token Service credentials

Session tokens from the global AWS Security Token Service (AWS STS) endpoint are valid only in AWS Regions that are enabled by default (commercial Regions). If you use credentials from the `assumeRole` API operation in AWS STS, use the regional endpoint if the source Region is an opt-in Region. Otherwise, the request fails. This happens because your credentials must be valid in both Regions, which is true for opt-in Regions only when the regional AWS STS endpoint is used.

To use the global endpoint, make sure that it's enabled for both Regions in the operations. Set the global endpoint to `Valid in all AWS Regions` in the AWS STS account settings.

The same rule applies to credentials in the presigned URL parameter.

For more information, see [Managing AWS STS in an AWS Region](#) in the *IAM User Guide*.

Latency and multiple copy requests

Depending on the AWS Regions involved and the amount of data to be copied, a cross-Region snapshot copy can take hours to complete.

In some cases, there might be a large number of cross-Region snapshot copy requests from a given source AWS Region. In such cases, Amazon RDS might put new cross-Region copy requests from that source AWS Region into a queue until some in-progress copies complete. No progress information is displayed about copy requests while they are in the queue. Progress information is displayed when the copying starts.

Full and incremental copies

When you copy a snapshot to a different AWS Region from the source snapshot, the first copy is a full snapshot copy, even if you copy an incremental snapshot. A full snapshot copy contains all of the data and metadata required to restore the DB instance. After the first snapshot copy, you can copy incremental snapshots of the same DB instance to the same destination Region within the same AWS account. For more information on incremental snapshots, see [Incremental snapshot copying \(p. 353\)](#).

Incremental snapshot copying across AWS Regions is supported for both unencrypted and encrypted snapshots.

When you copy a snapshot across AWS Regions, the copy is an incremental copy if the following conditions are met:

- The snapshot was previously copied to the destination Region.
- The most recent snapshot copy still exists in the destination Region.
- All copies of the snapshot in the destination Region are either unencrypted, or were encrypted using the same CMK.

Option group considerations

Option groups are specific to the AWS Region that they are created in, and you can't use an option group from one AWS Region in another AWS Region.

When you copy a snapshot across Regions, you can specify a new option group for the snapshot. We recommend that you prepare the new option group before you copy the snapshot. In the destination AWS Region, create an option group with the same settings as the original DB instance. If one already exists in the new AWS Region, you can use that one.

In some cases, you might copy a snapshot and not specify a new option group for the snapshot. In these cases, when you restore the snapshot the DB instance gets the default option group. To give the new DB instance the same options as the original, do the following:

1. In the destination AWS Region, create an option group with the same settings as the original DB instance . If one already exists in the new AWS Region, you can use that one.
2. After you restore the snapshot in the destination AWS Region, modify the new DB instance and add the new or existing option group from the previous step.

Parameter group considerations

When you copy a snapshot across Regions, the copy doesn't include the parameter group used by the original DB instance . When you restore a snapshot to create a new DB instance , that DB instance gets the default parameter group for the AWS Region it is created in. To give the new DB instance the same parameters as the original, do the following:

1. In the destination AWS Region, create a DB parameter group with the same settings as the original DB instance . If one already exists in the new AWS Region, you can use that one.
2. After you restore the snapshot in the destination AWS Region, modify the new DB instance and add the new or existing parameter group from the previous step.

Copying a DB snapshot

Use the procedures in this topic to copy a DB snapshot. For an overview of copying a snapshot, see [Copying a snapshot \(p. 352\)](#)

For each AWS account, you can copy up to five DB snapshots at a time from one AWS Region to another. If you copy a DB snapshot to another AWS Region, you create a manual DB snapshot that is retained in that AWS Region. Copying a DB snapshot out of the source AWS Region incurs Amazon RDS data transfer charges.

For more information about data transfer pricing, see [Amazon RDS pricing](#).

After the DB snapshot copy has been created in the new AWS Region, the DB snapshot copy behaves the same as all other DB snapshots in that AWS Region.

You can copy a DB snapshot using the AWS Management Console, the AWS CLI, or the RDS API.

Console

The following procedure copies an encrypted or unencrypted DB snapshot, in the same AWS Region or across Regions, by using the AWS Management Console.

To copy a DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. Select the DB snapshot that you want to copy.
4. For **Actions**, choose **Copy snapshot**.

The **Copy snapshot** page appears.

RDS > Snapshots > Copy snapshot

Copy snapshot

Settings

Source DB Snapshot
DB Snapshot Identifier for the snapshot being copied.
db1-snapshot

Destination Region [Info](#)
US West (Oregon)

New DB Snapshot Identifier
DB Snapshot Identifier for the new snapshot

Target Option Group (Optional)
No preference

Copy Tags [Info](#)

i Please note that depending on the amount of data to be copied and the Region you choose, this operation could take several hours to complete and the display on the progress bar could be delayed until setup is complete.

Encryption

Encryption [Info](#)

Enable Encryption
Choose to encrypt the copy of the source DB snapshot. Master key IDs and aliases appear in the list after they have been created using KMS. You cannot remove encryption from an encrypted DB snapshot.

Master key [Info](#)
(default) aws/rds

Account

KMS key ID

Cancel **Copy snapshot**

5. (Optional) To copy the DB snapshot to a different AWS Region, for **Destination Region**, choose the new AWS Region.

Note

The destination AWS Region must have the same database engine version available as the source AWS Region.

6. For **New DB Snapshot Identifier**, type the name of the DB snapshot copy.
7. (Optional) For **Target Option Group**, choose a new option group.

Specify this option if you are copying a snapshot from one AWS Region to another, and your DB instance uses a nondefault option group.

If your source DB instance uses Transparent Data Encryption for Oracle or Microsoft SQL Server, you must specify this option when copying across Regions. For more information, see [Option group considerations \(p. 356\)](#).

8. (Optional) Select **Copy Tags** to copy tags and values from the snapshot to the copy of the snapshot.
9. (Optional) For **Encryption**, do the following:

- a. Choose **Enable Encryption** if the DB snapshot isn't encrypted but you want to encrypt the copy.

Note

If the DB snapshot is encrypted, you must encrypt the copy, so the check box is already selected.

- b. For **Master key**, specify the AWS KMS key identifier to use to encrypt the DB snapshot copy.

10. Choose **Copy snapshot**.

AWS CLI

You can copy a DB snapshot by using the AWS CLI command `copy-db-snapshot`. If you are copying the snapshot to a new AWS Region, run the command in the new AWS Region.

The following options are used to copy a DB snapshot. Not all options are required for all scenarios. Use the descriptions and the examples that follow to determine which options to use.

- `--source-db-snapshot-identifier` – The identifier for the source DB snapshot.
 - If the source snapshot is in the same AWS Region as the copy, specify a valid DB snapshot identifier. For example, `rds:mysql-instance1-snapshot-20130805`.
 - If the source snapshot is in a different AWS Region than the copy, specify a valid DB snapshot ARN. For example, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
 - If you are copying from a shared manual DB snapshot, this parameter must be the Amazon Resource Name (ARN) of the shared DB snapshot.
 - If you are copying an encrypted snapshot this parameter must be in the ARN format for the source AWS Region, and must match the `SourceDBSnapshotIdentifier` in the `PreSignedUrl` parameter.
- `--target-db-snapshot-identifier` – The identifier for the new copy of the encrypted DB snapshot.
- `--copy-tags` – Include the `copy tags` option to copy tags and values from the snapshot to the copy of the snapshot.
- `--option-group-name` – The option group to associate with the copy of the snapshot.

Specify this option if you are copying a snapshot from one AWS Region to another, and your DB instance uses a non-default option group.

If your source DB instance uses Transparent Data Encryption for Oracle or Microsoft SQL Server, you must specify this option when copying across Regions. For more information, see [Option group considerations \(p. 356\)](#).

- **--kms-key-id** – The AWS KMS key identifier for an encrypted DB snapshot. The AWS KMS key identifier is the Amazon Resource Name (ARN), key identifier, or key alias for the AWS KMS CMK.
 - If you copy an encrypted DB snapshot from your AWS account, you can specify a value for this parameter to encrypt the copy with a new AWS KMS CMK. If you don't specify a value for this parameter, then the copy of the DB snapshot is encrypted with the same AWS KMS CMK as the source DB snapshot.
 - If you copy an encrypted DB snapshot that is shared from another AWS account, then you must specify a value for this parameter.
 - If you specify this parameter when you copy an unencrypted snapshot, the copy is encrypted.
 - If you copy an encrypted snapshot to a different AWS Region, then you must specify an AWS KMS CMK for the destination AWS Region. AWS KMS CMKs are specific to the AWS Region that they are created in, and you cannot use encryption keys from one AWS Region in another AWS Region.
- **--source-region** – The ID of the AWS Region of the source DB snapshot. If you copy an encrypted snapshot to a different AWS Region, then you must specify this option.

Example from unencrypted, to the same Region

The following code creates a copy of a snapshot, with the new name `mydbsnapshotcopy`, in the same AWS Region as the source snapshot. When the copy is made, all tags on the original snapshot are copied to the snapshot copy.

For Linux, macOS, or Unix:

```
aws rds copy-db-snapshot \  
  --source-db-snapshot-identifier mysql-instance1-snapshot-20130805 \  
  --target-db-snapshot-identifier mydbsnapshotcopy \  
  --copy-tags
```

For Windows:

```
aws rds copy-db-snapshot ^  
  --source-db-snapshot-identifier mysql-instance1-snapshot-20130805 ^  
  --target-db-snapshot-identifier mydbsnapshotcopy ^  
  --copy-tags
```

Example from unencrypted, across Regions

The following code creates a copy of a snapshot, with the new name `mydbsnapshotcopy`, in the AWS Region in which the command is run.

For Linux, macOS, or Unix:

```
aws rds copy-db-snapshot \  
  --source-db-snapshot-identifier arn:aws:rds:us-east-1:123456789012:snapshot:mysql-  
instance1-snapshot-20130805 \  
  --target-db-snapshot-identifier mydbsnapshotcopy
```

For Windows:

```
aws rds copy-db-snapshot ^  
  --source-db-snapshot-identifier arn:aws:rds:us-east-1:123456789012:snapshot:mysql-  
instance1-snapshot-20130805 ^  
  --target-db-snapshot-identifier mydbsnapshotcopy
```

Example from encrypted, across Regions

The following code example copies an encrypted DB snapshot from the US West (Oregon) Region in the US East (N. Virginia) Region. Run the command in the destination (us-east-1) Region.

For Linux, macOS, or Unix:

```
aws rds copy-db-snapshot \
--source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-
instance1-snapshot-20161115 \
--target-db-snapshot-identifier mydbsnapshotcopy \
--source-region us-west-2 \
--kms-key-id my-us-east-1-key \
--option-group-name custom-option-group-name
```

For Windows:

```
aws rds copy-db-snapshot ^
--source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-
instance1-snapshot-20161115 ^
--target-db-snapshot-identifier mydbsnapshotcopy ^
--source-region us-west-2 ^
--kms-key-id my-us-east-1-key ^
--option-group-name custom-option-group-name
```

RDS API

You can copy a DB snapshot by using the Amazon RDS API operation [CopyDBSnapshot](#). If you are copying the snapshot to a new AWS Region, perform the action in the new AWS Region.

The following parameters are used to copy a DB snapshot. Not all parameters are required for all scenarios. Use the descriptions and the examples that follow to determine which parameters to use.

- **SourceDBSnapshotIdentifier** – The identifier for the source DB snapshot.
 - If the source snapshot is in the same AWS Region as the copy, specify a valid DB snapshot identifier. For example, `rds:mysql-instance1-snapshot-20130805`.
 - If the source snapshot is in a different AWS Region than the copy, specify a valid DB snapshot ARN. For example, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
 - If you are copying from a shared manual DB snapshot, this parameter must be the Amazon Resource Name (ARN) of the shared DB snapshot.
 - If you are copying an encrypted snapshot this parameter must be in the ARN format for the source AWS Region, and must match the `SourceDBSnapshotIdentifier` in the `PreSignedUrl` parameter.
- **TargetDBSnapshotIdentifier** – The identifier for the new copy of the encrypted DB snapshot.
- **CopyTags** – Set this parameter to `true` to copy tags and values from the snapshot to the copy of the snapshot. The default is `false`.
- **OptionGroupName** – The option group to associate with the copy of the snapshot.

Specify this parameter if you are copying a snapshot from one AWS Region to another, and your DB instance uses a non-default option group.

If your source DB instance uses Transparent Data Encryption for Oracle or Microsoft SQL Server, you must specify this parameter when copying across Regions. For more information, see [Option group considerations \(p. 356\)](#).

- **KmsKeyId** – The AWS KMS key identifier for an encrypted DB snapshot. The AWS KMS key identifier is the Amazon Resource Name (ARN), key identifier, or key alias for the AWS KMS CMK.

- If you copy an encrypted DB snapshot from your AWS account, you can specify a value for this parameter to encrypt the copy with a new AWS KMS CMK. If you don't specify a value for this parameter, then the copy of the DB snapshot is encrypted with the same AWS KMS CMK as the source DB snapshot.
- If you copy an encrypted DB snapshot that is shared from another AWS account, then you must specify a value for this parameter.
- If you specify this parameter when you copy an unencrypted snapshot, the copy is encrypted.
- If you copy an encrypted snapshot to a different AWS Region, then you must specify an AWS KMS CMK for the destination AWS Region. AWS KMS CMKs are specific to the AWS Region that they are created in, and you cannot use encryption keys from one AWS Region in another AWS Region.
- **PreSignedUrl** – The URL that contains a Signature Version 4 signed request for the `CopyDBSnapshot` API operation in the source AWS Region that contains the source DB snapshot to copy.

Specify this parameter when you copy an encrypted DB snapshot from another AWS Region by using the Amazon RDS API. You can specify the source Region option instead of this parameter when you copy an encrypted DB snapshot from another AWS Region by using the AWS CLI.

The presigned URL must be a valid request for the `CopyDBSnapshot` API operation that can be run in the source AWS Region containing the encrypted DB snapshot to be copied. The presigned URL request must contain the following parameter values:

- **DestinationRegion** – The AWS Region that the encrypted DB snapshot will be copied to. This AWS Region is the same one where the `CopyDBSnapshot` operation is called that contains this presigned URL.

For example, suppose that you copy an encrypted DB snapshot from the us-west-2 Region to the us-east-1 Region. You then call the `CopyDBSnapshot` operation in the us-east-1 Region and provide a presigned URL that contains a call to the `CopyDBSnapshot` operation in the us-west-2 Region. For this example, the `DestinationRegion` in the presigned URL must be set to the us-east-1 Region.

- **KmsKeyId** – The AWS KMS key identifier for the key to use to encrypt the copy of the DB snapshot in the destination AWS Region. This is the same identifier for both the `CopyDBSnapshot` operation that is called in the destination AWS Region, and the operation contained in the presigned URL.
- **SourceDBSnapshotIdentifier** – The DB snapshot identifier for the encrypted snapshot to be copied. This identifier must be in the Amazon Resource Name (ARN) format for the source AWS Region. For example, if you are copying an encrypted DB snapshot from the us-west-2 Region, then your `SourceDBSnapshotIdentifier` looks like the following example: `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115`.

For more information on Signature Version 4 signed requests, see the following:

- [Authenticating requests: Using query parameters \(AWS signature version 4\)](#) in the Amazon Simple Storage Service API Reference
- [Signature version 4 signing process](#) in the AWS General Reference

Example from unencrypted, to the same Region

The following code creates a copy of a snapshot, with the new name `mydbsnapshotcopy`, in the same AWS Region as the source snapshot. When the copy is made, all tags on the original snapshot are copied to the snapshot copy.

```
https://rds.us-west-1.amazonaws.com/
?Action=CopyDBSnapshot
&CopyTags=true
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&SourceDBSnapshotIdentifier=mysql-instance1-snapshot-20130805
```

```
&TargetDBSnapshotIdentifier=mydbsnapshotcopy
&Version=2013-09-09
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140429/us-west-1/rds/aws4_request
&X-Amz-Date=20140429T175351Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=9164337efa99caf850e874a1cb7ef62f3cea29d0b448b9e0e7c53b288ddffed2
```

Example from unencrypted, across Regions

The following code creates a copy of a snapshot, with the new name `mydbsnapshotcopy`, in the US West (N. California) Region.

```
https://rds.us-west-1.amazonaws.com/
?Action=CopyDBSnapshot
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&SourceDBSnapshotIdentifier=arn%3Aaws%3Ards%3Aus-east-1%3A123456789012%3Asnapshot%3Amysql-
instance1-snapshot-20130805
&TargetDBSnapshotIdentifier=mydbsnapshotcopy
&Version=2013-09-09
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140429/us-west-1/rds/aws4_request
&X-Amz-Date=20140429T175351Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=9164337efa99caf850e874a1cb7ef62f3cea29d0b448b9e0e7c53b288ddffed2
```

Example from encrypted, across Regions

The following code creates a copy of a snapshot, with the new name `mydbsnapshotcopy`, in the US East (N. Virginia) Region.

```
https://rds.us-east-1.amazonaws.com/
?Action=CopyDBSnapshot
&KmsKeyId=my-us-east-1-key
&OptionGroupName=custom-option-group-name
&PreSignedUrl=https%253A%252F%252Frds.us-west-2.amazonaws.com%252F
    %253FAction%253DCopyDBSnapshot
    %2526DestinationRegion%253Dus-east-1
    %2526KmsKeyId%253Dmy-us-east-1-key
    %2526SourceDBSnapshotIdentifier%253Darn%25253Aaws%25253Ards%25253Aus-
west-2%25253A123456789012%25253Asnapshot%25253Amysql-instance1-snapshot-20161115
    %2526SignatureMethod%253DHmacSHA256
    %2526SignatureVersion%253D4
    %2526Version%253D2014-10-31
    %2526X-Amz-Algorithm%253DAWS4-HMAC-SHA256
    %2526X-Amz-Credential%253DAKIADQKE4SARGYLE%252F20161117%252Fus-west-2%252Frds
%252Faws4_request
    %2526X-Amz-Date%253D20161117T215409Z
    %2526X-Amz-Expires%253D3600
    %2526X-Amz-SignedHeaders%253Dcontent-type%253Bhost%253Buser-agent%253Bx-amz-
content-sha256%253Bx-amz-date
    %2526X-Amz-Signature
%253D255a0f17b4e717d3b67fad163c3ec26573b882c03a65523522cf890a67fca613
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &SourceDBSnapshotIdentifier=arn%3Aaws%3Ards%3Aus-west-2%3A123456789012%3Asnapshot
%3Amysql-instance1-snapshot-20161115
    &TargetDBSnapshotIdentifier=mydbsnapshotcopy
    &Version=2014-10-31
    &X-Amz-Algorithm=AWS4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20161117/us-east-1/rds/aws4_request
    &X-Amz-Date=20161117T221704Z
```

```
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=da4f2da66739d2e722c85fcfd225dc27bba7e2b8dbea8d8612434378e52adccf
```

Sharing a DB snapshot

Using Amazon RDS, you can share a manual DB snapshot in the following ways:

- Sharing a manual DB snapshot, whether encrypted or unencrypted, enables authorized AWS accounts to copy the snapshot.
- Sharing an unencrypted manual DB snapshot enables authorized AWS accounts to directly restore a DB instance from the snapshot instead of taking a copy of it and restoring from that. However, you can't restore a DB instance from a DB snapshot that is both shared and encrypted. Instead, you can make a copy of the DB snapshot and restore the DB instance from the copy.

Note

To share an automated DB snapshot, create a manual DB snapshot by copying the automated snapshot, and then share that copy. This process also applies to AWS Backup-generated resources.

For more information on copying a snapshot, see [Copying a snapshot \(p. 352\)](#). For more information on restoring a DB instance from a DB snapshot, see [Restoring from a DB snapshot \(p. 349\)](#).

You can share a manual snapshot with up to 20 other AWS accounts. You can also share an unencrypted manual snapshot as public, which makes the snapshot available to all AWS accounts. Take care when sharing a snapshot as public so that none of your private information is included in any of your public snapshots.

You can use the following AWS CLI command (Unix only) to find the public snapshots for your AWS account in a particular AWS Region:

```
aws rds describe-db-snapshots --snapshot-type public --include-public | grep account_number
```

The output returned is similar to the following example if you have public snapshots:

```
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mysnapshot1",
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mysnapshot2",
```

Note

You might see duplicate entries for `DBSnapshotIdentifier` or `SourceDBSnapshotIdentifier`.

The following limitations apply when sharing manual snapshots with other AWS accounts:

- When you restore a DB instance from a shared snapshot using the AWS Command Line Interface (AWS CLI) or Amazon RDS API, you must specify the Amazon Resource Name (ARN) of the shared snapshot as the snapshot identifier.
- You cannot share a DB snapshot that uses an option group with permanent or persistent options, except for Oracle DB instances that have the `Timezone` or `OLS` option (or both).

A *permanent option* cannot be removed from an option group. Option groups with persistent options cannot be removed from a DB instance once the option group has been assigned to the DB instance.

The following table lists permanent and persistent options and their related DB engines.

Option name	Persistent	Permanent	DB engine
TDE	Yes	No	Microsoft SQL Server Enterprise Edition

Option name	Persistent	Permanent	DB engine
TDE	Yes	Yes	Oracle Enterprise Edition
Timezone	Yes	Yes	Oracle Enterprise Edition
			Oracle Standard Edition
			Oracle Standard Edition One
			Oracle Standard Edition Two

For Oracle DB instances, you can copy shared DB snapshots that have the `Timezone` or `OLS` option (or both). To do so, specify a target option group that includes these options when you copy the DB snapshot. The `OLS` option is permanent and persistent only for Oracle DB instances running Oracle version 12.2 or higher. For more information about these options, see [Oracle time zone \(p. 1201\)](#) and [Oracle Label Security \(p. 1167\)](#).

Sharing an encrypted snapshot

You can share DB snapshots that have been encrypted "at rest" using the AES-256 encryption algorithm, as described in [Encrypting Amazon RDS resources \(p. 1630\)](#). To do this, you must take the following steps:

1. Share the AWS Key Management Service (AWS KMS) customer master key (CMK) that was used to encrypt the snapshot with any accounts that you want to be able to access the snapshot.

You can share AWS KMS CMKs with another AWS account by adding the other account to the AWS KMS key policy. For details on updating a key policy, see [Key policies](#) in the *AWS KMS Developer Guide*. For an example of creating a key policy, see [Allowing access to an AWS KMS customer master key \(CMK\) \(p. 366\)](#) later in this topic.

2. Use the AWS Management Console, AWS CLI, or Amazon RDS API to share the encrypted snapshot with the other accounts.

These restrictions apply to sharing encrypted snapshots:

- You can't share encrypted snapshots as public.
- You can't share Oracle or Microsoft SQL Server snapshots that are encrypted using Transparent Data Encryption (TDE).
- You can't share a snapshot that has been encrypted using the default AWS KMS CMK of the AWS account that shared the snapshot.

Allowing access to an AWS KMS customer master key (CMK)

For another AWS account to copy an encrypted DB snapshot shared from your account, the account that you share your snapshot with must have access to the AWS KMS customer master key (CMK) that encrypted the snapshot. To allow another AWS account access to an AWS KMS CMK, update the key policy for the AWS KMS CMK with the ARN of the AWS account that you are sharing to as a `Principal` in the AWS KMS key policy, and then allow the `kms:CreateGrant` action.

After you have given an AWS account access to your AWS KMS CMK, to copy your encrypted snapshot, that AWS account must create an AWS Identity and Access Management (IAM) user if it doesn't already have one. In addition, that AWS account must also attach an IAM policy to that IAM user that allows the IAM user to copy an encrypted DB snapshot using your AWS KMS CMK. The account must be an IAM user and cannot be a root AWS account identity due to AWS KMS security restrictions.

In the following key policy example, user 111122223333 is the owner of the AWS KMS CMK, and user 444455556666 is the account that the key is being shared with. This updated key policy gives the AWS account access to the AWS KMS CMK by including the ARN for the root AWS account identity for user 444455556666 as a Principal for the policy, and by allowing the kms:CreateGrant action.

```
{
    "Id": "key-policy-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {"AWS": [
                "arn:aws:iam::111122223333:user/KeyUser",
                "arn:aws:iam::444455556666:root"
            ]},
            "Action": [
                "kms:CreateGrant",
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow attachment of persistent resources",
            "Effect": "Allow",
            "Principal": {"AWS": [
                "arn:aws:iam::111122223333:user/KeyUser",
                "arn:aws:iam::444455556666:root"
            ]},
            "Action": [
                "kms:CreateGrant",
                "kms>ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": "*",
            "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
        }
    ]
}
```

Creating an IAM policy to enable copying of the encrypted snapshot

Once the external AWS account has access to your AWS KMS customer master key (CMK), the owner of that AWS account can create a policy that allows an IAM user created for that account to copy an encrypted snapshot encrypted with that AWS KMS CMK.

The following example shows a policy that can be attached to an IAM user for AWS account 444455556666 that enables the IAM user to copy a shared snapshot from AWS account 111122223333 that has been encrypted with the AWS KMS CMK c989c1dd-a3f2-4a5d-8d96-e793d082ab26 in the us-west-2 region.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUseOfTheKey",
            "Effect": "Allow",
            "Action": [
                "kms:CopyGrant"
            ],
            "Resource": [
                "arn:aws:kms:us-west-2:111122223333:key/c989c1dd-a3f2-4a5d-8d96-e793d082ab26"
            ]
        }
    ]
}
```

```
"Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey",
    "kms:CreateGrant",
    "kms:RetireGrant"
],
"Resource": ["arn:aws:kms:us-west-2:111122223333:key/c989c1dd-a3f2-4a5d-8d96-
e793d082ab26"]
},
{
    "Sid": "AllowAttachmentOfPersistentResources",
    "Effect": "Allow",
    "Action": [
        "kms:CreateGrant",
        "kms>ListGrants",
        "kms:RevokeGrant"
    ],
    "Resource": ["arn:aws:kms:us-west-2:111122223333:key/c989c1dd-a3f2-4a5d-8d96-
e793d082ab26"],
    "Condition": {
        "Bool": {
            "kms:GrantIsForAWSResource": true
        }
    }
}
]
```

For details on updating a key policy, see [Key policies in the AWS KMS Developer Guide](#).

Sharing a snapshot

You can share a DB snapshot using the AWS Management Console, the AWS CLI, or the RDS API.

Console

Using the Amazon RDS console, you can share a manual DB snapshot with up to 20 AWS accounts. You can also use the console to stop sharing a manual snapshot with one or more accounts.

To share a manual DB snapshot by using the Amazon RDS console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. Select the manual snapshot that you want to share.
4. For **Actions**, choose **Share Snapshot**.
5. Choose one of the following options for **DB snapshot visibility**.
 - If the source is unencrypted, choose **Public** to permit all AWS accounts to restore a DB instance from your manual DB snapshot, or choose **Private** to permit only AWS accounts that you specify to restore a DB instance from your manual DB snapshot.

Warning

If you set **DB snapshot visibility** to **Public**, all AWS accounts can restore a DB instance from your manual DB snapshot and have access to your data. Do not share any manual DB snapshots that contain private information as **Public**.

- If the source is encrypted, **DB snapshot visibility** is set as **Private** because encrypted snapshots can't be shared as public.
6. For **AWS Account ID**, type the AWS account identifier for an account that you want to permit to restore a DB instance from your manual snapshot, and then choose **Add**. Repeat to include additional AWS account identifiers, up to 20 AWS accounts.

If you make an error when adding an AWS account identifier to the list of permitted accounts, you can delete it from the list by choosing **Delete** at the right of the incorrect AWS account identifier.

Snapshot permissions

Preferences

You are sharing an unencrypted DB snapshot. When you share an unencrypted DB snapshot, you give the other account permission to make a copy of the DB snapshot and to restore a database from your DB snapshot.

DB snapshot
testoracletags-snap

DB snapshot visibility
 Private
 Public

AWS account ID

AWS account ID	<input type="button" value="Delete"/>
Please add AWS account ID	

7. After you have added identifiers for all of the AWS accounts that you want to permit to restore the manual snapshot, choose **Save** to save your changes.

To stop sharing a manual DB snapshot with an AWS account

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. Select the manual snapshot that you want to stop sharing.
4. Choose **Actions**, and then choose **Share Snapshot**.
5. To remove permission for an AWS account, choose **Delete** for the AWS account identifier for that account from the list of authorized accounts.

Snapshot permissions

Preferences

You are sharing an unencrypted DB snapshot. When you share an unencrypted DB snapshot, you give the other account permission to make a copy of the DB snapshot and to restore a database from your DB snapshot.

DB snapshot

testoracletags-snap

DB snapshot visibility

- Private
- Public

AWS account ID

Add

AWS account ID

Delete

Delete

Cancel

Save

6. Choose **Save** to save your changes.

AWS CLI

To share a DB snapshot, use the `aws rds modify-db-snapshot-attribute` command. Use the `--values-to-add` parameter to add a list of the IDs for the AWS accounts that are authorized to restore the manual snapshot.

Example of sharing a snapshot with a single account

The following example enables AWS account identifier 123456789012 to restore the DB snapshot named db7-snapshot.

For Linux, macOS, or Unix:

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier db7-snapshot \
--attribute-name restore \
--values-to-add 123456789012
```

For Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier db7-snapshot ^
--attribute-name restore ^
--values-to-add 123456789012
```

Example of sharing a snapshot with multiple accounts

The following example enables two AWS account identifiers, 111122223333 and 444455556666, to restore the DB snapshot named manual-snapshot1.

For Linux, macOS, or Unix:

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
--values-to-add {"111122223333","444455556666"}
```

For Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier manual-snapshot1 ^
--attribute-name restore ^
--values-to-add "[\"111122223333\", \"444455556666\"]"
```

Note

When using the Windows command prompt, you must escape double quotes ("") in JSON code by prefixing them with a backslash (\).

To remove an AWS account identifier from the list, use the --values-to-remove parameter.

Example of stopping snapshot sharing

The following example prevents AWS account ID 444455556666 from restoring the snapshot.

For Linux, macOS, or Unix:

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
--values-to-remove 444455556666
```

For Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier manual-snapshot1 ^
--attribute-name restore ^
--values-to-remove 444455556666
```

To list the AWS accounts enabled to restore a snapshot, use the [describe-db-snapshot-attributes](#) AWS CLI command.

RDS API

You can also share a manual DB snapshot with other AWS accounts by using the Amazon RDS API. To do so, call the [ModifyDBSnapshotAttribute](#) operation. Specify `restore` for `AttributeName`, and use the `ValuesToAdd` parameter to add a list of the IDs for the AWS accounts that are authorized to restore the manual snapshot.

To make a manual snapshot public and restorable by all AWS accounts, use the value `all`. However, take care not to add the `all` value for any manual snapshots that contain private information that you don't want to be available to all AWS accounts. Also, don't specify `all` for encrypted snapshots, because making such snapshots public isn't supported.

To remove sharing permission for an AWS account, use the [ModifyDBSnapshotAttribute](#) operation with `AttributeName` set to `restore` and the `ValuesToRemove` parameter. To mark a manual snapshot as private, remove the value `all` from the values list for the `restore` attribute.

To list all of the AWS accounts permitted to restore a snapshot, use the [DescribeDBSnapshotAttributes](#) API operation.

Exporting DB snapshot data to Amazon S3

You can export DB snapshot data to an Amazon S3 bucket. After the data is exported, you can analyze the exported data directly through tools like Amazon Athena or Amazon Redshift Spectrum. The export process runs in the background and doesn't affect the performance of your active DB instance.

When you export a DB snapshot, Amazon RDS extracts data from the snapshot and stores it in an Amazon S3 bucket in your account. The data is stored in an Apache Parquet format that is compressed and consistent.

You can export all types of DB snapshots—including manual snapshots, automated system snapshots, and snapshots created by the AWS Backup service. By default, all data in the snapshot is exported. However, you can choose to export specific sets of databases, schemas, or tables.

Amazon RDS supports exporting snapshots in all AWS Regions except the following:

- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

The following table shows the engine versions that are supported for exporting snapshot data to Amazon S3.

MariaDB	MySQL	PostgreSQL
10.3	8.0.13 and higher	11.2 and higher
10.2.12 and higher	5.7.24 and higher	10.7 and higher
10.1.26 and higher	5.6.40 and higher	9.6.6–9.6.9, 9.6.12 and higher
10.0.32 and higher		

For complete lists of engine versions supported by Amazon RDS, see the following:

- [MariaDB on Amazon RDS versions \(p. 576\)](#)
- [MySQL on Amazon RDS versions \(p. 828\)](#)
- [Supported PostgreSQL database versions \(p. 1461\)](#)

Topics

- [Limitations \(p. 374\)](#)
- [Overview of exporting snapshot data \(p. 374\)](#)
- [Setting up access to an Amazon S3 bucket \(p. 374\)](#)
- [Exporting a snapshot to an Amazon S3 bucket \(p. 377\)](#)
- [Monitoring snapshot exports \(p. 379\)](#)
- [Canceling a snapshot export task \(p. 380\)](#)
- [Troubleshooting PostgreSQL permissions errors \(p. 381\)](#)
- [File naming convention \(p. 382\)](#)
- [Data conversion when exporting to an Amazon S3 bucket \(p. 382\)](#)

Limitations

Exporting DB snapshot data to S3 has the following limitations:

- Exporting snapshots from DB instances that use magnetic storage isn't supported.
- If a database, schema, or table has characters in its name other than the following, partial export isn't supported. However, you can export the entire DB snapshot.
 - Latin letters (A–Z)
 - Digits (0–9)
 - Dollar symbol (\$)
 - Underscore (_)
- Some characters aren't supported in database table column names. Tables with the following characters in column names are skipped during export:

```
, ; { } ( ) \n \t =
```

- If the data contains a huge value close to or greater than 500 MB, the export fails.

Overview of exporting snapshot data

You use the following process to export DB snapshot data to an Amazon S3 bucket. For more details, see the following sections.

1. Identify the snapshot to export.

Use an existing automated or manual snapshot, or create a manual snapshot of a DB instance.

2. Set up access to the Amazon S3 bucket.

A *bucket* is a container for Amazon S3 objects or files. To provide the information to access a bucket, take the following steps:

- a. Identify the S3 bucket where the snapshot is to be exported to. The S3 bucket must be in the same AWS Region as the snapshot. For more information, see [Identifying the Amazon S3 bucket for export \(p. 375\)](#).
 - b. Create an AWS Key Management Service (AWS KMS) customer master key (CMK) for the server-side encryption. The AWS KMS CMK is used by the snapshot export task to set up AWS KMS server-side encryption when writing the export data to S3. For more information, see [Encrypting Amazon RDS resources \(p. 1630\)](#).
 - c. Create an AWS Identity and Access Management (IAM) role that grants the snapshot export task access to the S3 bucket. For more information, see [Providing access to an Amazon S3 bucket using an IAM role \(p. 375\)](#).
3. Export the snapshot to Amazon S3 using the console or the `start-export-task` CLI command. For more information, see [Exporting a snapshot to an Amazon S3 bucket \(p. 377\)](#).
 4. To access your exported data in the Amazon S3 bucket, see [Uploading, downloading, and managing objects in the Amazon Simple Storage Service Console User Guide](#).

Setting up access to an Amazon S3 bucket

To export DB snapshot data to an Amazon S3 file, you first give the snapshot permission to access the Amazon S3 bucket. You then create an IAM role to allow the Amazon RDS service to write to the Amazon S3 bucket.

Topics

- [Identifying the Amazon S3 bucket for export \(p. 375\)](#)
- [Providing access to an Amazon S3 bucket using an IAM role \(p. 375\)](#)

Identifying the Amazon S3 bucket for export

Identify the Amazon S3 bucket to export the DB snapshot to. Use an existing S3 bucket or create a new S3 bucket.

Note

The S3 bucket to export to must be in the same AWS Region as the snapshot.

For more information about working with Amazon S3 buckets, see the following in the *Amazon Simple Storage Service Console User Guide*:

- [How do I view the properties for an S3 bucket?](#)
- [How do I enable default encryption for an Amazon S3 bucket?](#)
- [How do I create an S3 bucket?](#)

Providing access to an Amazon S3 bucket using an IAM role

Before you export DB snapshot data to Amazon S3, give the snapshot export tasks write-access permission to the Amazon S3 bucket.

To do this, create an IAM policy that provides access to the bucket. Then create an IAM role and attach the policy to the role. You later assign the IAM role to your snapshot export task.

Important

If you plan to use the AWS Management Console to export your snapshot, you can choose to create the IAM policy and the role automatically when you export the snapshot. For instructions, see [Exporting a snapshot to an Amazon S3 bucket \(p. 377\)](#).

To give DB snapshot tasks access to Amazon S3

1. Create an IAM policy. This policy provides the bucket and object permissions that allow your snapshot export task to access Amazon S3.

Include in the policy the following required actions to allow the transfer of files from Amazon RDS to an S3 bucket:

- `s3:PutObject*`
- `s3:GetObject*`
- `s3>ListBucket`
- `s3>DeleteObject*`
- `s3:GetBucketLocation`

Include in the policy the following resources to identify the S3 bucket and objects in the bucket. The following list of resources shows the Amazon Resource Name (ARN) format for accessing Amazon S3.

- `arn:aws:s3:::your-s3-bucket`
- `arn:aws:s3:::your-s3-bucket/*`

For more information on creating an IAM policy for Amazon RDS, see [Creating and using an IAM policy for IAM database access \(p. 1664\)](#). See also [Tutorial: Create and attach your first customer managed policy](#) in the *IAM User Guide*.

The following AWS CLI command creates an IAM policy named `ExportPolicy` with these options. It grants access to a bucket named `your-s3-bucket`.

Note

After you create the policy, note the ARN of the policy. You need the ARN for a subsequent step when you attach the policy to an IAM role.

```
aws iam create-policy --policy-name ExportPolicy --policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListBucket",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": [  
                "arn:aws:s3:::*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:PutObject*",  
                "s3:GetObject*",  
                "s3:CopyObject*",  
                "s3>DeleteObject*"  
            ],  
            "Resource": [  
                "arn:aws:s3:::your-s3-bucket",  
                "arn:aws:s3:::your-s3-bucket/*"  
            ]  
        }  
    ]  
}'
```

2. Create an IAM role. You do this so that Amazon RDS can assume this IAM role on your behalf to access your Amazon S3 buckets. For more information, see [Creating a role to delegate permissions to an IAM user](#) in the *IAM User Guide*.

The following example shows using the AWS CLI command to create a role named `rds-s3-export-role`.

```
aws iam create-role --role-name rds-s3-export-role --assume-role-policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "export.rds.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}'
```

3. Attach the IAM policy that you created to the IAM role that you created.

The following AWS CLI command attaches the policy created earlier to the role named `rds-s3-export-role`. Replace `your-policy-arn` with the policy ARN that you noted in an earlier step.

```
aws iam attach-role-policy --policy-arn your-policy-arn --role-name rds-s3-export-role
```

Exporting a snapshot to an Amazon S3 bucket

You can have up to five concurrent DB snapshot export tasks in progress per account.

Note

Exporting RDS snapshots can take a while depending on your database type and size. The export task first restores and scales the entire database before extracting the data to Amazon S3. The task's progress during this phase displays as **Starting**. When the task switches to exporting data to S3, progress displays as **In progress**.

The time it takes for the export to complete depends on the data stored in the database. For example, tables with well distributed numeric primary key or index columns will export the fastest. Tables that don't contain a column suitable for partitioning and tables with only one index on a string-based column will take longer because the export uses a slower single threaded process.

You can export a DB snapshot to Amazon S3 using the AWS Management Console, the AWS CLI, or the RDS API.

If you use a Lambda function to export a snapshot, add the `kms:DescribeKey` action to the Lambda function policy. For more information, see [AWS Lambda permissions](#).

Console

The **Export to Amazon S3** console option appears only for snapshots that can be exported to Amazon S3. A snapshot might not be available for export because of the following reasons:

- The DB engine isn't supported for S3 export.
- The DB instance version isn't supported for S3 export.
- S3 export isn't supported in the AWS Region where the snapshot was created.

To export a DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. From the tabs, choose the type of snapshot that you want to export.
4. In the list of snapshots, choose the snapshot that you want to export.
5. For **Actions**, choose **Export to Amazon S3**.

The **Export to Amazon S3** window appears.

6. For **Export identifier**, enter a name to identify the export task. This value is also used for the name of the file created in the S3 bucket.
7. Choose the amount of data to be exported:
 - Choose **All** to export all data in the snapshot.
 - Choose **Partial** to export specific parts of the snapshot. To identify which parts of the snapshot to export, enter one or more tables for **Identifiers**.

8. For **S3 bucket**, choose the bucket to export to.

To assign the exported data to a folder path in the S3 bucket, enter the optional path for **S3 prefix**.

9. For **IAM role**, either choose a role that grants you write access to your chosen S3 bucket, or create a new role.
 - If you created a role by following the steps in [Providing access to an Amazon S3 bucket using an IAM role \(p. 375\)](#), choose that role.
 - If you didn't create a role that grants you write access to your chosen S3 bucket, choose **Create a new role** to create the role automatically. Next, enter a name for the role in **IAM role name**.
10. For **Master key**, enter the ARN for the key to use for encrypting the exported data.
11. Choose **Export to Amazon S3**.

AWS CLI

To export a DB snapshot to Amazon S3 using the AWS CLI, use the [start-export-task](#) command with the following required options:

- `--export-task-identifier`
- `--source-arn`
- `--s3-bucket-name`
- `--iam-role-arn`
- `--kms-key-id`

In the following examples, the snapshot export task is named `my_snapshot_export`, which exports a snapshot to an S3 bucket named `my_export_bucket`.

Example

For Linux, macOS, or Unix:

```
aws rds start-export-task \
--export-task-identifier my_snapshot_export \
--source-arn arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot_name \
--s3-bucket-name my_export_bucket \
--iam-role-arn iam_role \
--kms-key-id master_key
```

For Windows:

```
aws rds start-export-task ^
--export-task-identifier my_snapshot_export ^
--source-arn arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot_name ^
--s3-bucket-name my_export_bucket ^
--iam-role-arn iam_role ^
--kms-key-id master_key
```

Sample output follows.

```
{  
    "Status": "STARTING",  
    "IamRoleArn": "iam_role",  
    "ExportTime": "2019-08-12T01:23:53.109Z",  
    "S3Bucket": "my_export_bucket",  
    "PercentProgress": 0,  
    "KmsKeyId": "master_key",  
}
```

```
    "ExportTaskIdentifier": "my_snapshot_export",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-11-13T19:46:00.173Z",
    "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot_name"
}
```

To provide a folder path in the S3 bucket for the snapshot export, include the `--s3-prefix` option in the [start-export-task](#) command.

RDS API

To export a DB snapshot to Amazon S3 using the Amazon RDS API, use the [StartExportTask](#) operation with the following required parameters:

- `ExportTaskIdentifier`
- `SourceArn`
- `S3BucketName`
- `IamRoleArn`
- `KmsKeyId`

Monitoring snapshot exports

You can monitor DB snapshot exports using the AWS Management Console, the AWS CLI, or the RDS API.

Console

To monitor DB snapshot exports

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. To view the list of snapshot exports, choose the **Exports in Amazon S3** tab.
4. To view information about a specific snapshot export, choose the export task.

AWS CLI

To monitor DB snapshot exports using the AWS CLI, use the [describe-export-tasks](#) command.

The following example shows how to display current information about all of your snapshot exports.

Example

```
aws rds describe-export-tasks

{
    "ExportTasks": [
        {
            "Status": "CANCELED",
            "TaskEndTime": "2019-11-01T17:36:46.961Z",
            "S3Prefix": "something",
            "ExportTime": "2019-10-24T20:23:48.364Z",
            "S3Bucket": "examplebucket",
            "PercentProgress": 0,
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/K7MDENG/bPxRfiCYEXAMPLEKEY",
            "ExportTaskIdentifier": "anewtest",
```

```

    "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-10-25T19:10:58.885Z",
    "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:parameter-groups-
test"
},
{
    "Status": "COMPLETE",
    "TaskEndTime": "2019-10-31T21:37:28.312Z",
    "WarningMessage": "{\"skippedTables\":[],\"skippedObjectives\":[],\"general\":
[{\\"reason\\":\\"FAILED_TO_EXTRACT_TABLES_LIST_FOR_DATABASE\\\"}]}",
    "S3Prefix": "",
    "ExportTime": "2019-10-31T06:44:53.452Z",
    "S3Bucket": "examplebucket1",
    "PercentProgress": 100,
    "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/2Zp9Utk/
h3yCo8nzbEXAMPLEKEY",
    "ExportTaskIdentifier": "thursday-events-test",
    "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
    "TotalExtractedDataInGB": 263,
    "TaskStartTime": "2019-10-31T20:58:06.998Z",
    "SourceArn":
"arn:aws:rds:AWS_Region:123456789012:snapshot:rds:example-1-2019-10-31-06-44"
},
{
    "Status": "FAILED",
    "TaskEndTime": "2019-10-31T02:12:36.409Z",
    "FailureCause": "The S3 bucket edgcuc-export isn't located in the current AWS
Region. Please, review your S3 bucket name and retry the export.",
    "S3Prefix": "",
    "ExportTime": "2019-10-30T06:45:04.526Z",
    "S3Bucket": "examplebucket2",
    "PercentProgress": 0,
    "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/2Zp9Utk/
h3yCo8nzbEXAMPLEKEY",
    "ExportTaskIdentifier": "wednesday-afternoon-test",
    "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-10-30T22:43:40.034Z",
    "SourceArn":
"arn:aws:rds:AWS_Region:123456789012:snapshot:rds:example-1-2019-10-30-06-45"
}
]
}

```

To display information about a specific snapshot export, include the `--export-task-identifier` option with the `describe-export-tasks` command. To filter the output, include the `--Filters` option. For more options, see the [describe-export-tasks](#) command.

RDS API

To display information about DB snapshot exports using the Amazon RDS API, use the [DescribeExportTasks](#) operation.

To track completion of the export workflow or to trigger another workflow, you can subscribe to Amazon Simple Notification Service topics. For more information on Amazon SNS, see [Using Amazon RDS event notification \(p. 487\)](#).

Canceling a snapshot export task

You can cancel a DB snapshot export task using the AWS Management Console, the AWS CLI, or the RDS API.

Note

Cancelling a snapshot export task doesn't remove any data that was exported to Amazon S3. For information about how to delete the data using the console, see [How do I delete objects from an S3 bucket?](#) To delete the data using the CLI, use the `delete-object` command.

Console

To cancel a snapshot export task

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. Choose the **Exports in Amazon S3** tab.
4. Choose the snapshot export task that you want to cancel.
5. Choose **Cancel**.
6. Choose **Cancel export task** on the confirmation page.

AWS CLI

To cancel a snapshot export task using the AWS CLI, use the `cancel-export-task` command. The command requires the `--export-task-identifier` option.

Example

```
aws rds cancel-export-task --export-task-identifier my_export
{
    "Status": "CANCELING",
    "S3Prefix": "",
    "ExportTime": "2019-08-12T01:23:53.109Z",
    "S3Bucket": "examplebucket",
    "PercentProgress": 0,
    "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "ExportTaskIdentifier": "my_export",
    "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-11-13T19:46:00.173Z",
    "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:export-example-1"
}
```

RDS API

To cancel a snapshot export task using the Amazon RDS API, use the `CancelExportTask` operation with the `ExportTaskIdentifier` parameter.

Troubleshooting PostgreSQL permissions errors

When exporting PostgreSQL databases to Amazon S3, you might see a `PERMISSIONS_DO_NOT_EXIST` error stating that certain tables were skipped. This is usually caused by the superuser, which you specify when creating the DB instance, not having permissions to access those tables.

To fix this error, run the following command:

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA schema_name TO superuser_name
```

For more information on superuser privileges, see [Master user account privileges \(p. 1712\)](#).

File naming convention

Exported data for specific tables is stored in the format `base_prefix/files`, where the base prefix is the following:

```
export_identifier/database_name/schema_name.table_name/
```

For example:

```
export-1234567890123-459/rdststdb/rdststdb.DataInsert_7ADB5D19965123A2/
```

There are two conventions for how files are named:

- `part-partition_index-random_uuid.format-based_extension`
- `partition_index/part-00000-random_uuid.format-based_extension`

For example:

```
part-00000-c5a881bb-58ff-4ee6-1111-b41ecff340a3-c000.gz.parquet
part-00001-d7a881cc-88cc-5ab7-2222-c41ecab340a4-c000.gz.parquet
part-00002-f5a991ab-59aa-7fa6-3333-d41eccd340a7-c000.gz.parquet
```

```
1/part-00000-c5a881bb-58ff-4ee6-1111-b41ecff340a3-c000.gz.parquet
2/part-00000-d7a881cc-88cc-5ab7-2222-c41ecab340a4-c000.gz.parquet
3/part-00000-f5a991ab-59aa-7fa6-3333-d41eccd340a7-c000.gz.parquet
```

The file naming convention is subject to change. Therefore, when reading target tables we recommend that you read everything inside the base prefix for the table.

Data conversion when exporting to an Amazon S3 bucket

When you export a DB snapshot to an Amazon S3 bucket, Amazon RDS converts data to, exports data in, and stores data in the Parquet format. For more information about Parquet, see the [Apache Parquet](#) website.

Parquet stores all data as one of the following primitive types:

- BOOLEAN
- INT32
- INT64
- INT96
- FLOAT
- DOUBLE
- BYTE_ARRAY – A variable-length byte array, also known as binary
- FIXED_LEN_BYTE_ARRAY – A fixed-length byte array used when the values have a constant size

The Parquet data types are few to reduce the complexity of reading and writing the format. Parquet provides logical types for extending primitive types. A *logical type* is implemented as an annotation with the data in a `LogicalType` metadata field. The logical type annotation explains how to interpret the primitive type.

When the `STRING` logical type annotates a `BYTE_ARRAY` type, it indicates that the byte array should be interpreted as a UTF-8 encoded character string. After an export task completes, Amazon RDS notifies you if any string conversion occurred. The underlying data exported is always the same as the data from the source. However, due to the encoding difference in UTF-8, some characters might appear different from the source when read in tools such as Athena.

For more information, see [Parquet logical type definitions](#) in the Parquet documentation.

Topics

- [MySQL and MariaDB data type mapping to Parquet \(p. 383\)](#)
- [PostgreSQL data type mapping to Parquet \(p. 385\)](#)

MySQL and MariaDB data type mapping to Parquet

The following table shows the mapping from MySQL and MariaDB data types to Parquet data types when data is converted and exported to Amazon S3.

Source data type	Parquet primitive type	Logical type annotation	Conversion notes
Numeric data types			
BIGINT	INT64		
BIGINT UNSIGNED	FIXED_LEN_BYTE_ARRAY(9)DECIMAL(20,0)		Parquet supports only signed types, so the mapping requires an additional byte (8 plus 1) to store the BIGINT_UNSIGNED type.
BIT	BYTE_ARRAY		
DECIMAL	INT32	DECIMAL(p,s)	If the source value is less than 2^{31} , it's stored as INT32.
	INT64	DECIMAL(p,s)	If the source value is 2^{31} or greater, but less than 2^{63} , it's stored as INT64.
	FIXED_LEN_BYTE_ARRAY(ND)DECIMAL(p,s)		If the source value is 2^{63} or greater, it's stored as FIXED_LEN_BYTE_ARRAY(N).
	BYTE_ARRAY	STRING	Parquet doesn't support Decimal precision greater than 38. The Decimal value is converted to a string in a BYTE_ARRAY type and encoded as UTF8.
DOUBLE	DOUBLE		
FLOAT	DOUBLE		

Source data type	Parquet primitive type	Logical type annotation	Conversion notes
INT	INT32		
INT UNSIGNED	INT64		
MEDIUMINT	INT32		
MEDIUMINT UNSIGNED	INT64		
NUMERIC	INT32	DECIMAL(p,s)	If the source value is less than 2^{31} , it's stored as INT32.
	INT64	DECIMAL(p,s)	If the source value is 2^{31} or greater, but less than 2^{63} , it's stored as INT64.
	FIXED_LEN_ARRAY(N)	DECIMAL(p,s)	If the source value is 2^{63} or greater, it's stored as FIXED_LEN_BYTE_ARRAY(N).
	BYTE_ARRAY	STRING	Parquet doesn't support Numeric precision greater than 38. This Numeric value is converted to a string in a BYTE_ARRAY type and encoded as UTF8.
SMALLINT	INT32		
SMALLINT UNSIGNED	INT32		
TINYINT	INT32		
TINYINT UNSIGNED	INT32		
String data types			
BINARY	BYTE_ARRAY		
BLOB	BYTE_ARRAY		
CHAR	BYTE_ARRAY		
ENUM	BYTE_ARRAY	STRING	
LINESTRING	BYTE_ARRAY		
LONGBLOB	BYTE_ARRAY		
LONGTEXT	BYTE_ARRAY	STRING	
MEDIUMBLOB	BYTE_ARRAY		
MEDIUMTEXT	BYTE_ARRAY	STRING	
MULTILINESTRING	BYTE_ARRAY		
SET	BYTE_ARRAY	STRING	

Source data type	Parquet primitive type	Logical type annotation	Conversion notes
TEXT	BYTE_ARRAY	STRING	
TINYBLOB	BYTE_ARRAY		
TINYTEXT	BYTE_ARRAY	STRING	
VARBINARY	BYTE_ARRAY		
VARCHAR	BYTE_ARRAY	STRING	
Date and time data types			
DATE	BYTE_ARRAY	STRING	A date is converted to a string in a BYTE_ARRAY type and encoded as UTF8.
DATETIME	INT64	TIMESTAMP_MICROS	
TIME	BYTE_ARRAY	STRING	A TIME type is converted to a string in a BYTE_ARRAY and encoded as UTF8.
TIMESTAMP	INT64	TIMESTAMP_MICROS	
YEAR	INT32		
Geometric data types			
GEOMETRY	BYTE_ARRAY		
GEOMETRYCOLLECTION	BYTE_ARRAY		
MULTIPOINT	BYTE_ARRAY		
MULTIPOLYGON	BYTE_ARRAY		
POINT	BYTE_ARRAY		
POLYGON	BYTE_ARRAY		
JSON data type			
JSON	BYTE_ARRAY	STRING	

PostgreSQL data type mapping to Parquet

The following table shows the mapping from PostgreSQL data types to Parquet data types when data is converted and exported to Amazon S3.

PostgreSQL data type	Parquet primitive type	Logical type annotation	Mapping notes
Numeric data types			

PostgreSQL data type	Parquet primitive type	Logical type annotation	Mapping notes
BIGINT	INT64		
BIGSERIAL	INT64		
DECIMAL	BYTE_ARRAY	STRING	A DECIMAL type is converted to a string in a BYTE_ARRAY type and encoded as UTF8. This conversion is to avoid complications due to data precision and data values that are not a number (NaN).
DOUBLE PRECISION	DOUBLE		
INTEGER	INT32		
MONEY	BYTE_ARRAY	STRING	
REAL	FLOAT		
SERIAL	INT32		
SMALLINT	INT32	INT_16	
SMALLSERIAL	INT32	INT_16	
String and related data types			
ARRAY	BYTE_ARRAY	STRING	An array is converted to a string and encoded as BINARY (UTF8). This conversion is to avoid complications due to data precision, data values that are not a number (NaN), and time data values.
BIT	BYTE_ARRAY	STRING	
BIT VARYING	BYTE_ARRAY	STRING	
BYTEA	BINARY		
CHAR	BYTE_ARRAY	STRING	
CHAR(N)	BYTE_ARRAY	STRING	
ENUM	BYTE_ARRAY	STRING	
NAME	BYTE_ARRAY	STRING	
TEXT	BYTE_ARRAY	STRING	

PostgreSQL data type	Parquet primitive type	Logical type annotation	Mapping notes
TEXT SEARCH	BYTE_ARRAY	STRING	
VARCHAR(N)	BYTE_ARRAY	STRING	
XML	BYTE_ARRAY	STRING	
Date and time data types			
DATE	BYTE_ARRAY	STRING	
INTERVAL	BYTE_ARRAY	STRING	
TIME	BYTE_ARRAY	STRING	
TIME WITH TIME ZONE	BYTE_ARRAY	STRING	
TIMESTAMP	BYTE_ARRAY	STRING	
TIMESTAMP WITH TIME ZONE	BYTE_ARRAY	STRING	
Geometric data types			
BOX	BYTE_ARRAY	STRING	
CIRCLE	BYTE_ARRAY	STRING	
LINE	BYTE_ARRAY	STRING	
LINESEGMENT	BYTE_ARRAY	STRING	
PATH	BYTE_ARRAY	STRING	
POINT	BYTE_ARRAY	STRING	
POLYGON	BYTE_ARRAY	STRING	
JSON data types			
JSON	BYTE_ARRAY	STRING	
JSONB	BYTE_ARRAY	STRING	
Other data types			
BOOLEAN	BOOLEAN		
CIDR	BYTE_ARRAY	STRING	Network data type
COMPOSITE	BYTE_ARRAY	STRING	
DOMAIN	BYTE_ARRAY	STRING	
INET	BYTE_ARRAY	STRING	Network data type
MACADDR	BYTE_ARRAY	STRING	
OBJECT IDENTIFIER	N/A		
PG_LSN	BYTE_ARRAY	STRING	

PostgreSQL data type	Parquet primitive type	Logical type annotation	Mapping notes
RANGE	BYTE_ARRAY	STRING	
UUID	BYTE_ARRAY	STRING	

Restoring a DB instance to a specified time

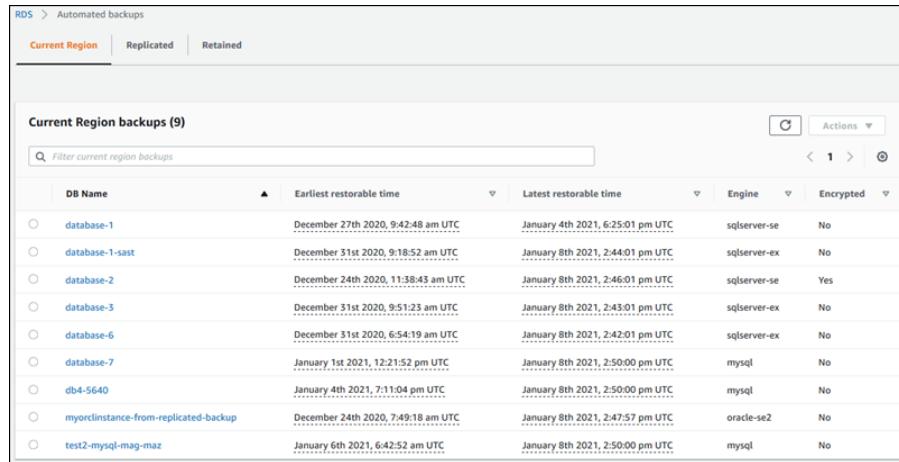
You can restore a DB instance to a specific point in time, creating a new DB instance.

When you restore a DB instance to a point in time, the default DB security group is applied to the new DB instance. If you need custom DB security groups applied to your DB instance, you must apply them explicitly using the AWS Management Console, the AWS CLI `modify-db-instance` command, or the Amazon RDS API `ModifyDBInstance` operation after the DB instance is available.

Restored DB instances are automatically associated with the default parameter and option groups. However, you can apply a custom parameter group and option group by specifying them during a restore.

RDS uploads transaction logs for DB instances to Amazon S3 every 5 minutes. To see the latest restorable time for a DB instance, use the AWS CLI `describe-db-instances` command and look at the value returned in the `LatestRestorableTime` field for the DB instance. To see the latest restorable time for each DB instance in the Amazon RDS console, choose **Automated backups**.

You can restore to any point in time within your backup retention period. To see the earliest restorable time for each DB instance, choose **Automated backups** in the Amazon RDS console.



DB Name	Earliest restorable time	Latest restorable time	Engine	Encrypted
database-1	December 27th 2020, 9:42:48 am UTC	January 4th 2021, 6:25:01 pm UTC	sqlserver-se	No
database-1-sast	December 31st 2020, 9:18:52 am UTC	January 8th 2021, 2:44:01 pm UTC	sqlserver-ex	No
database-2	December 24th 2020, 11:38:43 am UTC	January 8th 2021, 2:46:01 pm UTC	sqlserver-se	Yes
database-3	December 31st 2020, 9:51:23 am UTC	January 8th 2021, 2:43:01 pm UTC	sqlserver-ex	No
database-6	December 31st 2020, 6:54:19 am UTC	January 8th 2021, 2:42:01 pm UTC	sqlserver-ex	No
database-7	January 1st 2021, 12:21:52 pm UTC	January 8th 2021, 2:50:00 pm UTC	mysql	No
db4-5640	January 4th 2021, 7:11:04 pm UTC	January 8th 2021, 2:50:00 pm UTC	mysql	No
myordinstance-from-replicated-backup	December 24th 2020, 7:49:18 am UTC	January 8th 2021, 2:47:57 pm UTC	oracle-se2	No
test2-mysql-mag-maz	January 6th 2021, 6:42:52 am UTC	January 8th 2021, 2:50:00 pm UTC	mysql	No

Note

We recommend that you restore to the same or similar DB instance size—and IOPS if using Provisioned IOPS storage—as the source DB instance. You might get an error if, for example, you choose a DB instance size with an incompatible IOPS value.

Some of the database engines used by Amazon RDS have special considerations when restoring from a point in time:

- When you restore an Oracle DB instance to a point in time, you can specify a different Oracle DB engine, license model, and DBName (SID) to be used by the new DB instance.
- When you restore a SQL Server DB instance to a point in time, each database within that instance is restored to a point in time within 1 second of each other database within the instance. Transactions that span multiple databases within the instance might be restored inconsistently.
- For a SQL Server DB instance, the `OFFLINE`, `EMERGENCY`, and `SINGLE_USER` modes aren't supported. Setting any database into one of these modes causes the latest restorable time to stop moving ahead for the whole instance.
- Some actions, such as changing the recovery model of a SQL Server database, can break the sequence of logs that are used for point-in-time recovery. In some cases, Amazon RDS can detect this issue and the latest restorable time is prevented from moving forward. In other cases, such as when a SQL Server database uses the `BULK_LOGGED` recovery model, the break in log sequence isn't detected. It

might not be possible to restore a SQL Server DB instance to a point in time if there is a break in the log sequence. For these reasons, Amazon RDS doesn't support changing the recovery model of SQL Server databases.

Note

You can also use AWS Backup to manage backups of Amazon RDS DB instances. If your DB instance is associated with a backup plan in AWS Backup, that backup plan is used for point-in-time recovery. Backups that were created with AWS Backup have names ending in `awsbackup:AWS-Backup-job-number`. For information about AWS Backup, see the [AWS Backup Developer Guide](#).

You can restore a DB instance to a point in time using the AWS Management Console, the AWS CLI, or the RDS API.

Console

To restore a DB instance to a specified time

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Automated backups**.
3. Choose the DB instance that you want to restore.
4. For **Actions**, choose **Restore to point in time**.

The **Restore to point in time** window appears.

5. Choose **Latest restorable time** to restore to the latest possible time, or choose **Custom** to choose a time.

If you chose **Custom**, enter the date and time to which you want to restore the instance.

Note

Times are shown in your local time zone, which is indicated by an offset from Coordinated Universal Time (UTC). For example, UTC-5 is Eastern Standard Time/Central Daylight Time.

6. For **DB instance identifier**, enter the name of the target restored DB instance. The name must be unique.
7. Choose other options as needed, such as DB instance class, storage, and whether you want to use storage autoscaling.
8. Choose **Restore to point in time**.

AWS CLI

To restore a DB instance to a specified time, use the AWS CLI command `restore-db-instance-to-point-in-time` to create a new DB instance. This example also enables storage autoscaling.

Example

For Linux, macOS, or Unix:

```
aws rds restore-db-instance-to-point-in-time \
--source-db-instance-identifier mysourcedbinstance \
--target-db-instance-identifier mytargetdbinstance \
--restore-time 2017-10-14T23:45:00.000Z \
--max-allocated-storage 1000
```

For Windows:

```
aws rds restore-db-instance-to-point-in-time ^
--source-db-instance-identifier mysourcedbinstance ^
--target-db-instance-identifier mytargetdbinstance ^
--restore-time 2017-10-14T23:45:00.000Z ^
--max-allocated-storage 1000
```

RDS API

To restore a DB instance to a specified time, call the Amazon RDS API [RestoreDBInstanceToPointInTime](#) operation with the following parameters:

- `SourceDBInstanceIdentifier`
- `TargetDBInstanceIdentifier`
- `RestoreTime`

Deleting a snapshot

You can delete DB snapshots managed by Amazon RDS when you no longer need them.

Note

To delete backups managed by AWS Backup, use the AWS Backup console. For information about AWS Backup, see the [AWS Backup Developer Guide](#).

Deleting a DB snapshot

You can delete a manual, shared, or public DB snapshot using the AWS Management Console, the AWS CLI, or the RDS API.

To delete a shared or public snapshot, you must sign in to the AWS account that owns the snapshot.

If you have automated DB snapshots that you want to delete without deleting the DB instance, change the backup retention period for the DB instance to 0. The automated snapshots are deleted when the change is applied. You can apply the change immediately if you don't want to wait until the next maintenance period. After the change is complete, you can then re-enable automatic backups by setting the backup retention period to a number greater than 0. For information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

If you deleted a DB instance, you can delete its automated DB snapshots by removing the automated backups for the DB instance. For information about automated backups, see [Working with backups \(p. 328\)](#).

Console

To delete a DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.

The **Manual snapshots** list appears.

3. Choose the DB snapshot that you want to delete.
4. For **Actions**, choose **Delete Snapshot**.
5. Choose **Delete** on the confirmation page.

AWS CLI

You can delete a DB snapshot by using the AWS CLI command `delete-db-snapshot`.

The following options are used to delete a DB snapshot.

- `--db-snapshot-identifier` – The identifier for the DB snapshot.

Example

The following code deletes the `mydbsnapshot` DB snapshot.

For Linux, macOS, or Unix:

```
aws rds delete-db-snapshot \
--db-snapshot-identifier mydbsnapshot
```

For Windows:

```
aws rds delete-db-snapshot ^
--db-snapshot-identifier mydbsnapshot
```

RDS API

You can delete a DB snapshot by using the Amazon RDS API operation [DeleteDBSnapshot](#).

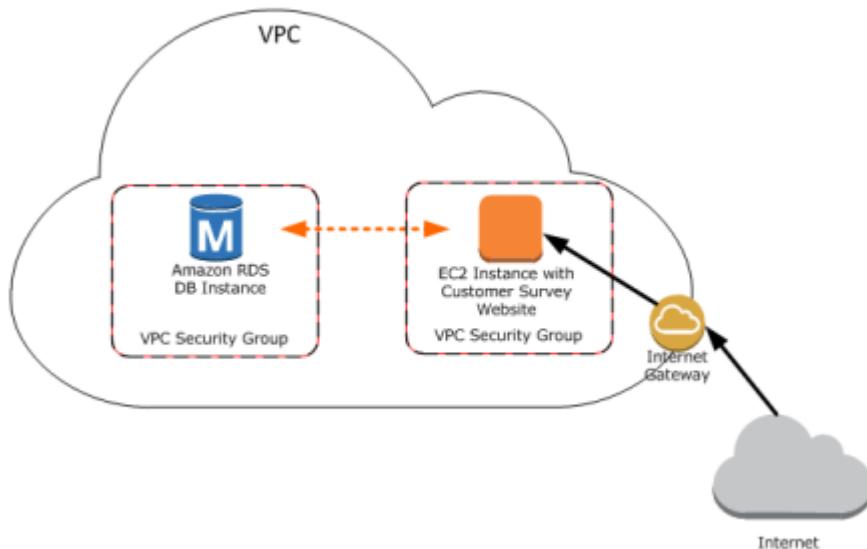
The following parameters are used to delete a DB snapshot.

- **DBSnapshotIdentifier** – The identifier for the DB snapshot.

Tutorial: Restore a DB instance from a DB snapshot

A common scenario when working with Amazon RDS is to have a DB instance that you work with occasionally but that you don't need full time. For example, you might have a quarterly customer survey that uses an Amazon Elastic Compute Cloud (Amazon EC2) instance to host a customer survey website and a DB instance that is used to store the survey results. One way to save money on such a scenario is to take a DB snapshot of the DB instance after the survey is completed, delete the DB instance, and then restore the DB instance when you need to conduct the survey again.

In the following illustration, you can see a possible scenario where an EC2 instance hosting a customer survey website is in the same Amazon Virtual Private Cloud (Amazon VPC) as a DB instance that retains the customer survey data. Note that each instance has its own security group; the EC2 instance security group allows access from the internet while the DB instance security group allows access only to and from the EC2 instance. When the survey is done, the EC2 instance can be stopped and the DB instance can be deleted after a final DB snapshot is created. When you need to conduct another survey, you can restart the EC2 instance and restore the DB instance from the DB snapshot.



For information about how to set up the needed VPC security groups for this scenario that allows the EC2 instance to connect with the DB instance, see [A DB instance in a VPC accessed by an EC2 instance in the same VPC \(p. 1720\)](#).

You must create a DB snapshot before you can restore a DB instance from one. When you restore the DB instance, you provide the name of the DB snapshot to restore from, and then provide a name for the new DB instance that is created from the restore operation. You cannot restore from a DB snapshot to an existing DB instance; a new DB instance is created when you restore.

Prerequisites for restoring a DB instance from a DB snapshot

Some settings on the restored DB instance are reset when the instance is restored, so you must retain the original resources to be able to restore the DB instance to its previous settings. For example, when you restore a DB instance from a DB snapshot, the default DB parameter and a default security group are associated with the restored instance. That association means that the default security group does not allow access to the DB instance, and no custom parameter settings are available in the default parameter group. You need to retain the DB parameter group and security group associated with the DB instance that was used to create the DB snapshot.

The following are required before you can restore a DB instance from a DB snapshot:

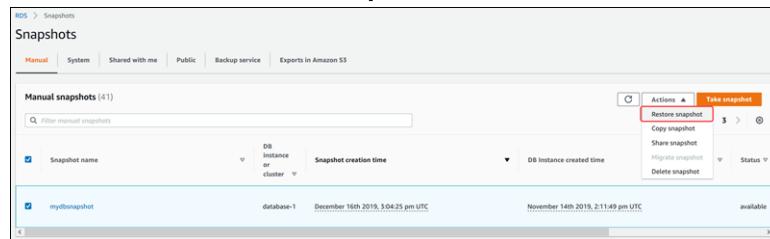
- You must have created a DB snapshot of a DB instance before you can restore a DB instance from that DB snapshot. For more information about creating a DB snapshot, see [Creating a DB snapshot \(p. 346\)](#).
- You must retain the parameter group and security group associated with the DB instance you created the DB snapshot from.
- You need to determine the correct option group for the restored DB instance:
 - The option group associated with the DB snapshot that you restore from is associated with the restored DB instance once it is created. For example, if the DB snapshot you restore from uses Oracle Transparent Data Encryption (TDE), the restored DB instance uses the same option group, which had the TDE option.
 - You cannot use the option group associated with the original DB instance if you attempt to restore that instance into a different VPC or into a different platform. This restriction occurs because when an option group is assigned to a DB instance, it is also linked to the platform that the DB instance is on, either VPC or EC2-Classic (non-VPC). If a DB instance is in a VPC, the option group associated with the instance is linked to that VPC.
 - If you restore a DB instance into a different VPC or onto a different platform, you must either assign the default option group to the instance, assign an option group that is linked to that VPC or platform, or create a new option group and assign it to the DB instance. Note that with persistent or permanent options, such as Oracle TDE, you must create a new option group that includes the persistent or permanent option when restoring a DB instance into a different VPC. For more information about working with option groups, see [Working with option groups \(p. 212\)](#).

Restoring a DB instance from a DB snapshot

You can use the procedure following to restore from a snapshot in the AWS Management Console.

To restore a DB instance from a DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. Choose the DB snapshot that you want to restore from.
4. For **Actions**, choose **Restore snapshot**.



The **Restore snapshot** page appears.

5. For **DB Instance identifier** under **Settings**, enter the unique name that you want to use for the restored DB instance.

If you're restoring from a DB instance that you deleted after you made the DB snapshot, you can use the name of that DB instance.

6. Choose additional settings as needed.
7. Choose **Restore DB Instance**.

Modifying a restored DB instance

As soon as the restore operation is complete, you should associate the custom security group used by the instance you restored from with any applicable custom DB parameter group that you might have. Only the default DB parameter and security groups are associated with the restored instance. If you want to restore the functionality of the DB instance to that of the DB instance that the snapshot was created from, you must modify the DB instance to use the security group and parameter group used by the previous DB instance.

You must apply any changes explicitly using the RDS console's **Modify** command, the `ModifyDBInstance` API, or the `aws rds modify-db-instance` command line tool, once the DB instance is available. We recommend that you retain parameter groups for any DB snapshots you have so that you can associate a restored instance with the correct parameter file.

You can modify other settings on the restored DB instance. For example, you can use a different storage type than the source DB snapshot. In this case the restoration process is slower because of the additional work required to migrate the data to the new storage type. In the case of restoring to or from Magnetic (Standard) storage, the migration process is the slowest, because Magnetic storage does not have the IOPS capability of Provisioned IOPS or General Purpose (SSD) storage.

The next steps assume that your DB instance is in a VPC. If your DB instance is not in a VPC, use the AWS Management Console to locate the DB security group you need for the DB instance.

To modify a restored DB instance to have the settings of the original DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the name of the DB instance created when you restored from the DB snapshot to display its details. Choose the **Connectivity** tab. The security group assigned to the DB instance might not allow access. If there are no inbound rules, no permissions exist that allow inbound access.

The screenshot shows the 'Connectivity' tab of the AWS RDS console. It displays three main sections: 'Endpoint & port', 'Networking', and 'Security'. In the 'Security' section, a red box highlights the 'VPC security groups' field, which lists '(active)' and 'rds-ca-2015'. Below this, it shows 'Public accessibility' set to 'Yes'.

Connectivity		
Endpoint & port	Networking	Security
Endpoint restored-db-instance.cahj48jpj34l.us-west-2.rds.amazonaws.com	Availability zone us-west-2a	VPC security groups (active)
Port 3306	VPC	Public accessibility Yes
	Subnet group default	Certificate authority rds-ca-2015
	Subnets	Certificate authority d... Mar 5th, 2020

Security group (1)

Filter security group rules

Security group	Type
[Redacted]	No applicable security group roles

4. Choose **Modify**.
5. In the **Network & Security** section, choose the security group that you want to use for your DB instance. If you need to add rules to create a new security group to use with an EC2 instance, see [A DB instance in a VPC accessed by an EC2 instance in the same VPC \(p. 1720\)](#) for more information.

You can also remove a security group by choosing the X associated with it.

The screenshot shows the 'Network & Security' configuration dialog. A red oval highlights the 'Select security groups' dropdown, which contains 'default ([]) ([]) X'. Other fields shown include 'Subnet group' (set to 'default'), 'Certificate authority' (set to 'rds-ca-2015'), and 'Public accessibility info' (set to 'Yes').

Network & Security

Subnet group
Use this field to move the DB instance to a new subnet group in another VPC. [Learn more.](#)

default

Security group
List of DB security groups to associate with this DB instance.

Select security groups

default ([]) ([]) X

Certificate authority
Certificate authority for this DB instance

rds-ca-2015

Public accessibility info

Yes
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

No
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

6. Choose **Continue**, and then choose **Apply immediately**.
7. Choose **Modify DB Instance**.

After the instance status is available, choose the DB instance name to display its details. Choose the **Connectivity** tab, and confirm that the new security group has been applied, making the DB instance authorized for access.

The screenshot shows the 'Connectivity' tab selected in the top navigation bar of the AWS RDS console. The page displays three main sections: 'Endpoint & port', 'Networking', and 'Security'. The 'Security' section is highlighted with a red oval, specifically focusing on the 'VPC security groups' field which lists 'rds-launch-wizard-9 (active)'. Other visible details include the endpoint 'restored-db-instance.cahj48jp34l.us-west-2.rds.amazonaws.com' and port '3306' under 'Endpoint & port'; 'us-west-2a' under 'Availability zone'; and 'Subnet group default' under 'Networking'.

Monitoring an Amazon RDS DB instance

This section shows you how to monitor Amazon RDS.

Topics

- [Overview of monitoring Amazon RDS \(p. 400\)](#)
- [DB instance status \(p. 404\)](#)
- [Using Amazon RDS recommendations \(p. 407\)](#)
- [Using Performance Insights on Amazon RDS \(p. 412\)](#)
- [Using Enhanced Monitoring \(p. 471\)](#)
- [Using Amazon RDS event notification \(p. 487\)](#)
- [Viewing Amazon RDS events \(p. 503\)](#)
- [Accessing Amazon RDS database log files \(p. 504\)](#)
- [Monitoring Amazon RDS metrics with Amazon CloudWatch \(p. 540\)](#)
- [Publishing database engine logs to Amazon CloudWatch Logs \(p. 547\)](#)
- [Getting CloudWatch Events and Amazon EventBridge events for Amazon RDS \(p. 551\)](#)
- [Working with AWS CloudTrail and Amazon RDS \(p. 557\)](#)

Overview of monitoring Amazon RDS

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon RDS and your AWS solutions. To more easily debug multi-point failures, we recommend that you collect monitoring data from all parts of your AWS solution.

Monitoring plan

Before you start monitoring Amazon RDS, create a monitoring plan. This plan should answer the following questions:

- What are your monitoring goals?
- Which resources will you monitor?
- How often will you monitor these resources?
- Which monitoring tools will you use?
- Who will perform the monitoring tasks?
- Whom should be notified when something goes wrong?

Performance baseline

To achieve your monitoring goals, you need to establish a baseline. To do this, measure performance under different load conditions at various times in your Amazon RDS environment. You can monitor metrics such as the following:

- Network throughput
- Client connections
- I/O for read, write, or metadata operations
- Burst credit balances for your DB instances

We recommend that you store historical performance data for Amazon RDS. Using the stored data, you can compare current performance against past trends. You can also distinguish normal performance patterns from anomalies, and devise techniques to address issues.

Performance guidelines

In general, acceptable values for performance metrics depend on what your application is doing relative to your baseline. Investigate consistent or trending variances from your baseline. The following metrics are often the source of performance issues:

- **High CPU or RAM consumption** – High values for CPU or RAM consumption might be appropriate, if they're in keeping with your goals for your application (like throughput or concurrency) and are expected.
- **Disk space consumption** – Investigate disk space consumption if space used is consistently at or above 85 percent of the total disk space. See if it is possible to delete data from the instance or archive data to a different system to free up space.
- **Network traffic** – For network traffic, talk with your system administrator to understand what expected throughput is for your domain network and internet connection. Investigate network traffic if throughput is consistently lower than expected.
- **Database connections** – If you see high numbers of user connections and also decreases in instance performance and response time, consider constraining database connections. The best number of

user connections for your DB instance varies based on your instance class and the complexity of the operations being performed. To determine the number of database connections, associate your DB instance with a parameter group where the `User_Connections` parameter is set to a value other than 0 (unlimited). You can either use an existing parameter group or create a new one. For more information, see [Working with DB parameter groups \(p. 228\)](#).

- **IOPS metrics** – The expected values for IOPS metrics depend on disk specification and server configuration, so use your baseline to know what is typical. Investigate if values are consistently different than your baseline. For best IOPS performance, make sure that your typical working set fits into memory to minimize read and write operations.

When performance falls outside your established baseline, you might need to make changes to optimize your database availability for your workload. For example, you might need to change the instance class of your DB instance. Or you might need to change the number of DB instances and read replicas that are available for clients.

Monitoring tools

AWS provides various tools that you can use to monitor Amazon RDS. You can configure some of these tools to do the monitoring for you, and other tools require manual intervention.

Automated monitoring tools

We recommend that you automate monitoring tasks as much as possible.

Amazon RDS reporting tools

You can use the following automated tools to watch Amazon RDS and report when something is wrong:

- **Amazon RDS instance status** — View details about the current status of your instance by using the Amazon RDS console, the AWS CLI command, or the RDS API.
- **Amazon RDS recommendations** — Respond to automated recommendations for database resources, such as DB instances, read replicas, and DB parameter groups. For more information, see [Using Amazon RDS recommendations \(p. 407\)](#).
- **Amazon RDS Performance Insights** — Assess the load on your database, and determine when and where to take action. For more information, see [Using Performance Insights on Amazon RDS \(p. 412\)](#).
- **Amazon RDS Enhanced Monitoring** — Look at metrics in real time for the operating system. For more information, see [Using Enhanced Monitoring \(p. 471\)](#).
- **Amazon RDS events** — Subscribe to Amazon RDS events to be notified when changes occur with a DB instance, DB snapshot, DB parameter group, or DB security group. For more information, see [Using Amazon RDS event notification \(p. 487\)](#).
- **Amazon RDS database logs** — View, download, or watch database log files using the Amazon RDS console or Amazon RDS API operations. You can also query some database log files that are loaded into database tables. For more information, see [Accessing Amazon RDS database log files \(p. 504\)](#).

Integrated monitoring tools

Amazon RDS integrates with Amazon CloudWatch, Amazon EventBridge, and AWS CloudTrail for additional monitoring capabilities.

- **Amazon CloudWatch** – This service monitors your AWS resources and the applications you run on AWS in real time. You can use the following Amazon CloudWatch features with Amazon RDS:
 - **Amazon CloudWatch metrics** – Amazon RDS automatically sends metrics to CloudWatch every minute for each active database. You don't get additional charges for Amazon RDS metrics

in CloudWatch. For more information, see [Monitoring Amazon RDS metrics with Amazon CloudWatch \(p. 540\)](#).

- **Amazon CloudWatch alarms** – You can watch a single Amazon RDS metric over a specific time period. You can then perform one or more actions based on the value of the metric relative to a threshold that you set. For more information, see [Monitoring Amazon RDS metrics with Amazon CloudWatch \(p. 540\)](#).
- **Amazon CloudWatch Logs** – Most DB engines enable you to monitor, store, and access your database log files in CloudWatch Logs. For more information, see [Amazon CloudWatch Logs User Guide](#).
- **Amazon EventBridge** – is a serverless event bus service that makes it easy to connect your applications with data from a variety of sources. EventBridge delivers a stream of real-time data from your own applications, Software-as-a-Service (SaaS) applications, and AWS services and routes that data to targets such as Lambda. This enables you to monitor events that happen in services, and build event-driven architectures. For more information, see [Getting CloudWatch Events and Amazon EventBridge events for Amazon RDS \(p. 551\)](#).
- **AWS CloudTrail** – You can view a record of actions taken by a user, role, or an AWS service in Amazon RDS. CloudTrail captures all API calls for Amazon RDS as events. These captures include calls from the Amazon RDS console and from code calls to the Amazon RDS API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon RDS. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. For more information, see [Working with AWS CloudTrail and Amazon RDS \(p. 557\)](#).

Manual monitoring tools

You need to manually monitor those items that the CloudWatch alarms don't cover. The Amazon RDS, CloudWatch, AWS Trusted Advisor and other AWS console dashboards provide an at-a-glance view of the state of your AWS environment. We recommend that you also check the log files on your DB instance.

- From the Amazon RDS console, you can monitor the following items for your resources:
 - The number of connections to a DB instance
 - The amount of read and write operations to a DB instance
 - The amount of storage that a DB instance is currently using
 - The amount of memory and CPU being used for a DB instance
 - The amount of network traffic to and from a DB instance
- From the Trusted Advisor dashboard, you can review the following cost optimization, security, fault tolerance, and performance improvement checks:
 - Amazon RDS Idle DB Instances
 - Amazon RDS Security Group Access Risk
 - Amazon RDS Backups
 - Amazon RDS Multi-AZ

For more information on these checks, see [Trusted Advisor best practices \(checks\)](#).

- CloudWatch home page shows:
 - Current alarms and status
 - Graphs of alarms and resources
 - Service health status

In addition, you can use CloudWatch to do the following:

- Create [customized dashboards](#) to monitor the services that you care about.
- Graph metric data to troubleshoot issues and discover trends.
- Search and browse all your AWS resource metrics.

- Create and edit alarms to be notified of problems.

DB instance status

The status of a DB instance indicates the health of the DB instance. You can view the status of a DB instance by using the Amazon RDS console, the AWS CLI command [describe-db-instances](#), or the API operation [DescribeDBInstances](#).

Note

Amazon RDS also uses another status called *maintenance status*, which is shown in the **Maintenance** column of the Amazon RDS console. This value indicates the status of any maintenance patches that need to be applied to a DB instance. Maintenance status is independent of DB instance status. For more information on *maintenance status*, see [Applying updates for a DB instance \(p. 266\)](#).

Find the possible status values for DB instances in the following table, which also shows how you are billed for each status. It shows if you will be billed for the DB instance and storage, billed only for storage, or not billed. For all DB instance statuses, you are always billed for backup usage.

DB instance status	Billed	Description
available	Billed	The DB instance is healthy and available.
backing-up	Billed	The DB instance is currently being backed up.
backtracking	Billed	The DB instance is currently being backtracked. This status only applies to Aurora MySQL.
configuring-enhanced-monitoring	Billed	Enhanced Monitoring is being enabled or disabled for this DB instance.
configuring-iam-database-auth	Billed	AWS Identity and Access Management (IAM) database authentication is being enabled or disabled for this DB instance.
configuring-log-exports	Billed	Publishing log files to Amazon CloudWatch Logs is being enabled or disabled for this DB instance.
converting-to-vpc	Billed	The DB instance is being converted from a DB instance that is not in an Amazon Virtual Private Cloud (Amazon VPC) to a DB instance that is in an Amazon VPC.
creating	Not billed	The DB instance is being created. The DB instance is inaccessible while it is being created.
deleting	Not billed	The DB instance is being deleted.
failed	Not billed	The DB instance has failed and Amazon RDS can't recover it. Perform a point-in-time restore to the latest restorable time of the DB instance to recover the data.
inaccessible-encryption-credentials	Not billed	The AWS KMS customer master key (CMK) used to encrypt or decrypt the DB instance can't be accessed.
incompatible-network	Not billed	Amazon RDS is attempting to perform a recovery action on a DB instance but can't do so because the VPC is in a state that prevents the action from being completed. This status can occur if, for example, all available IP addresses in a subnet are in use and Amazon RDS can't get an IP address for the DB instance.

DB instance status	Billed	Description
incompatible-option-group	Billed	Amazon RDS attempted to apply an option group change but can't do so, and Amazon RDS can't roll back to the previous option group state. For more information, check the Recent Events list for the DB instance. This status can occur if, for example, the option group contains an option such as TDE and the DB instance doesn't contain encrypted information.
incompatible-parameters	Billed	Amazon RDS can't start the DB instance because the parameters specified in the DB instance's DB parameter group aren't compatible with the DB instance. Revert the parameter changes or make them compatible with the DB instance to regain access to your DB instance. For more information about the incompatible parameters, check the Recent Events list for the DB instance.
incompatible-restore	Not billed	Amazon RDS can't do a point-in-time restore. Common causes for this status include using temp tables, using MyISAM tables with MySQL, or using Aria tables with MariaDB.
insufficient-capacity		Amazon RDS can't create your instance because sufficient capacity isn't currently available. To create your DB instance in the same AZ with the same instance type, delete your DB instance, wait a few hours, and try to create again. Alternatively, create a new instance using a different instance class or AZ.
maintenance	Billed	Amazon RDS is applying a maintenance update to the DB instance. This status is used for instance-level maintenance that RDS schedules well in advance.
modifying	Billed	The DB instance is being modified because of a customer request to modify the DB instance.
moving-to-vpc	Billed	The DB instance is being moved to a new Amazon Virtual Private Cloud (Amazon VPC).
rebooting	Billed	The DB instance is being rebooted because of a customer request or an Amazon RDS process that requires the rebooting of the DB instance.
resetting-master-credentials	Billed	The master credentials for the DB instance are being reset because of a customer request to reset them.
renaming	Billed	The DB instance is being renamed because of a customer request to rename it.
restore-error	Billed	The DB instance encountered an error attempting to restore to a point-in-time or from a snapshot.
starting	Billed for storage	The DB instance is starting.
stopped	Billed for storage	The DB instance is stopped.

DB instance status	Billed	Description
stopping	Billed for storage	The DB instance is being stopped.
storage-full	Billed	The DB instance has reached its storage capacity allocation. This is a critical status, and we recommend that you fix this issue immediately. To do so, scale up your storage by modifying the DB instance. To avoid this situation, set Amazon CloudWatch alarms to warn you when storage space is getting low.
storage-optimization	Billed	Your DB instance is being modified to change the storage size or type. The DB instance is fully operational. However, while the status of your DB instance is storage-optimization , you can't request any changes to the storage of your DB instance. The storage optimization process is usually short, but can sometimes take up to and even beyond 24 hours.
upgrading	Billed	The database engine version is being upgraded.

Using Amazon RDS recommendations

Amazon RDS provides automated recommendations for database resources, such as DB instances, read replicas, and DB parameter groups. These recommendations provide best practice guidance by analyzing DB instance configuration, usage, and performance data.

You can find examples of these recommendations in the following table.

Type	Description	Recommendation	Additional information
Engine version outdated	Your DB instance is not running the latest minor engine version.	We recommend that you upgrade to the latest version because it contains the latest security fixes and other improvements.	Upgrading a DB instance engine version (p. 271)
Pending maintenance available	You have pending maintenance available on your DB instance.	We recommend that you perform the pending maintenance available on your DB instance. Updates to the operating system most often occur for security issues and should be done as soon as possible.	Maintaining a DB instance (p. 264)
Automated backups disabled	Your DB instance has automated backups disabled.	We recommend that you enable automated backups on your DB instance. Automated backups enable point-in-time recovery of your DB instance. You receive backup storage up to the storage size of your DB instance at no additional charge.	Working with backups (p. 328)
Magnetic volumes in use	Your DB instance is using magnetic storage.	Magnetic storage is not recommended for most DB instances. We recommend switching to General Purpose (SSD) storage or provisioned IOPS storage.	Amazon RDS DB instance storage (p. 40)
EC2-Classic platform in use	Your DB instance is using the legacy EC2-Classic platform.	We recommend moving your DB instance to the EC2-VPC platform for better network access control. Amazon VPC provides a virtual network that is logically isolated from other virtual networks in the AWS Cloud.	Determining whether you are using the EC2-VPC or EC2-Classic platform (p. 1718)
Enhanced Monitoring disabled	Your DB instance doesn't have Enhanced Monitoring enabled.	We recommend enabling Enhanced Monitoring. Enhanced Monitoring provides real-time operating system metrics for monitoring and troubleshooting.	Using Enhanced Monitoring (p. 471)
Encryption disabled	Your DB instance doesn't have encryption enabled.	We recommend enabling encryption. You can encrypt your existing Amazon RDS DB instances by restoring from an encrypted snapshot.	Encrypting Amazon RDS resources (p. 1630)
Previous generation	Your DB instance is running on a	Previous-generation DB instance classes have been replaced by DB	DB instance classes (p. 7)

Type	Description	Recommendation	Additional information
DB instance class in use	previous-generation DB instance class.	instance classes with better price, better performance, or both. We recommend running your DB instance on a later generation DB instance class.	
Huge pages not used for an Oracle DB instance	The <code>use_large_pages</code> parameter is not set to <code>ONLY</code> in the DB parameter group used by your DB instance.	For increased database scalability, we recommend setting <code>use_large_pages</code> to <code>ONLY</code> in the DB parameter group used by your DB instance.	Enabling HugePages for an Oracle DB instance (p. 1101)
Nondefault custom memory parameters	Your DB parameter group sets memory parameters that diverge too much from the default values.	Settings that diverge too much from the default values can cause poor performance and errors. We recommend setting custom memory parameters to their default values in the DB parameter group used by the DB instance.	Working with DB parameter groups (p. 228)
Change buffering enabled for a MySQL DB instance	Your DB parameter group has change buffering enabled.	Change buffering allows a MySQL DB instance to defer some writes necessary to maintain secondary indexes. This configuration can improve performance slightly, but it can create a large delay in crash recovery. During crash recovery, the secondary index must be brought up to date. So, the benefits of change buffering are outweighed by the potentially very long crash recovery events. We recommend disabling change buffering.	Best practices for configuring parameters for Amazon RDS for MySQL, part 1: Parameters related to performance on the AWS Database Blog
Query cache enabled for a MySQL DB instance	Your DB parameter group has query cache parameter enabled.	The query cache can cause the DB instance to appear to stall when changes require the cache to be purged. Most workloads don't benefit from a query cache. The query cache was removed from MySQL version 8.0. We recommend that you disable the query cache parameter.	Best practices for configuring parameters for Amazon RDS for MySQL, part 1: Parameters related to performance on the AWS Database Blog
Logging to table	Your DB parameter group sets logging output to <code>TABLE</code> .	Setting logging output to <code>TABLE</code> uses more storage than setting this parameter to <code>FILE</code> . To avoid reaching the storage limit, we recommend setting the logging output parameter to <code>FILE</code> .	Accessing MySQL database log files (p. 519)

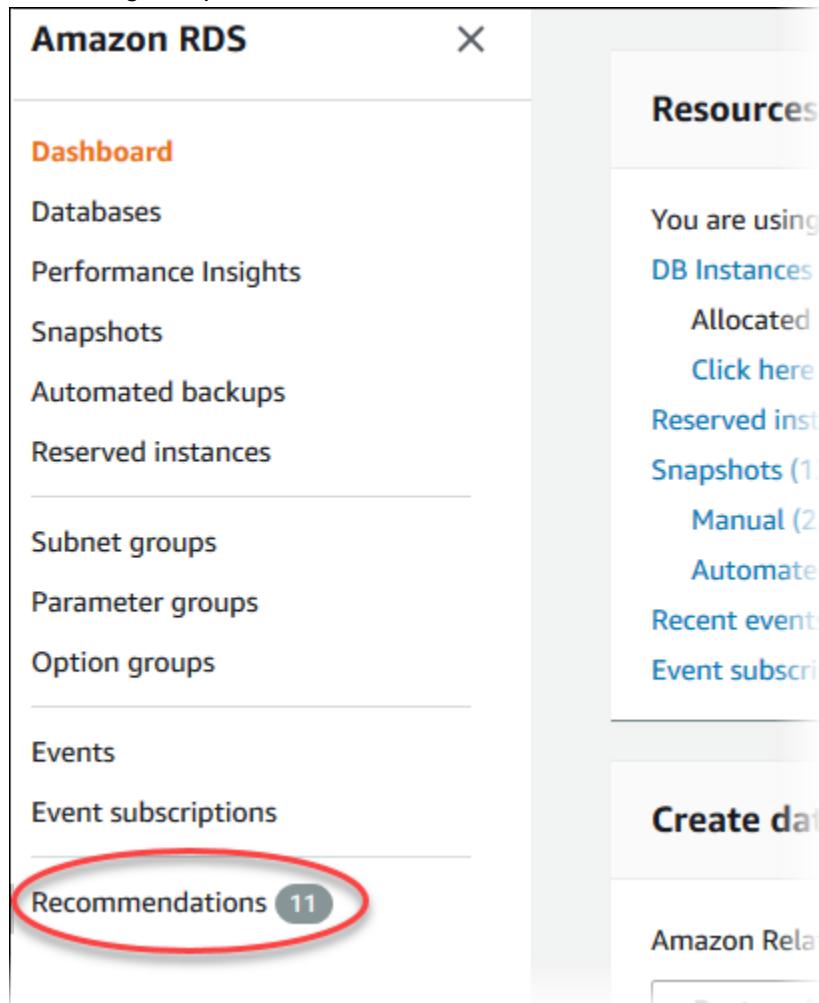
Amazon RDS generates recommendations for a resource when the resource is created or modified. Amazon RDS also periodically scans your resources and generates recommendations.

Responding to Amazon RDS recommendations

You can find recommendations in the AWS Management Console. You can perform the recommended action immediately, schedule it for the next maintenance window, or dismiss it.

To respond to Amazon RDS recommendations

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Recommendations**.



The Recommendations page appears.

The screenshot shows a 'Recommendations' page with a header bar containing four tabs: Active (2), Dismissed (0), Scheduled (0), and Applied (1). The Active tab is selected. Below the tabs, there are two main sections: 'Engine version outdated for DB instances (1)' and 'Enhanced monitoring disabled (1)'. Each section contains a brief description and a 'Info' link.

3. On the **Recommendations** page, choose one of the following:

- **Active** – Shows the current recommendations that you can apply, dismiss, or schedule.
- **Dismissed** – Shows the recommendations that have been dismissed. When you choose **Dismissed**, you can apply these dismissed recommendations.
- **Scheduled** – Shows the recommendations that are scheduled but not yet applied. These recommendations will be applied in the next scheduled maintenance window.
- **Applied** – Shows the recommendations that are currently applied.

From any list of recommendations, you can open a section to view the recommendations in that section.

The screenshot shows the 'Engine version outdated for DB instances (1)' section from the previous screenshot. It includes a 'DB instances' heading, a search bar, and a table with columns for Resource and Recommendation. One row in the table is for 'database-1', which has a note about upgrading to mysql version 5.6.41. There are also buttons for Dismiss, Schedule, and Apply now.

To configure preferences for displaying recommendations in each section, choose the **Preferences** icon.

Recommendations

Active (2) Dismissed (0) Scheduled (0) Applied (1)

▼ Engine version outdated for DB instances (1)
DB instances that are not running the latest minor engine version. [Info](#)

DB instances

Filter recommendations

<input type="checkbox"/>	Resource	Recommendation	Actions
<input type="checkbox"/>	database-1	Your DB instance is running mysql version 5.6.34. We recommend that you upgrade to version 5.6.41 because it contains the latest security fixes and other improvements.	Dismiss Schedule Apply now

► Enhanced monitoring disabled (1)
DB instances that don't have Enhanced Monitoring enabled. [Info](#)

From the **Preferences** window that appears, you can set display options. These options include the visible columns and the number of recommendations to display on the page.

4. Manage your active recommendations:

- Choose **Active** and open one or more sections to view the recommendations in them.
- Choose one or more recommendations and choose **Apply now** (to apply them immediately), **Schedule** (to apply them in next maintenance window), or **Dismiss**.

If the **Apply now** button appears for a recommendation but is unavailable (grayed out), the DB instance is not available. You can apply recommendations immediately only if the DB instance status is **available**. For example, you can't apply recommendations immediately to the DB instance if its status is **modifying**. In this case, wait for the DB instance to be available and then apply the recommendation.

If the **Apply now** button doesn't appear for a recommendation, you can't apply the recommendation using the **Recommendations** page. You can modify the DB instance to apply the recommendation manually.

For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Note

When you choose **Apply now**, a brief DB instance outage might result.

Using Performance Insights on Amazon RDS

Performance Insights expands on existing Amazon RDS monitoring features to illustrate your database's performance and help you analyze any issues that affect it. With the Performance Insights dashboard, you can visualize the database load and filter the load by waits, SQL statements, hosts, or users.

Topics

- [Overview of Performance Insights \(p. 412\)](#)
- [Enabling and disabling Performance Insights \(p. 415\)](#)
- [Accessing Performance Insights \(p. 419\)](#)
- [Monitoring with the Performance Insights dashboard \(p. 421\)](#)
- [Customizing the Performance Insights dashboard \(p. 444\)](#)
- [Retrieving data with the Performance Insights API \(p. 454\)](#)
- [Performance Insights metrics published to Amazon CloudWatch \(p. 467\)](#)
- [Logging Performance Insights calls by using AWS CloudTrail \(p. 468\)](#)

Overview of Performance Insights

By default, Performance Insights is enabled in the console create wizard for Amazon RDS engines. If you have more than one database on a DB instance, Performance Insights aggregates performance data.

Topics

- [DB load \(p. 412\)](#)
- [Maximum CPU \(p. 414\)](#)
- [DB engine support for Performance Insights \(p. 414\)](#)
- [AWS Region support for Performance Insights \(p. 415\)](#)

DB load

The central metric for Performance Insights is DB Load. The DB Load metric is collected every second.

Active Sessions

An *active session* is a connection that has submitted work to the DB engine and is waiting for a response. The DB load represents the *average active sessions* (AAS) for the DB engine. For example, if you submit a SQL query to the DB engine, the database session is active while the engine is processing the query.

To obtain the AAS, Performance Insights samples the number of sessions concurrently running a query. The AAS is the total number of sessions divided by the total number of samples. The following table shows 5 consecutive samples of a running query.

Sample	Number of sessions running query	AAS	Calculation
1	2	2	2 sessions / 1 sample
2	0	1	2 sessions / 2 samples
3	4	2	6 sessions / 3 samples
4	0	1.5	6 sessions / 4 samples

Sample	Number of sessions running query	AAS	Calculation
5	4	2	10 sessions / 5 samples

Related to AAS is the *average active executions (AAE)* per second. To calculate the AAE, Performance Insights divides the total execution time of a query by the time interval. The following table shows the AAE calculation for the same query in the preceding table.

Elapsed time (s)	Total execution time (s)	AAE	Calculation
60	120	2	120 execution seconds / 60 elapsed seconds
120	120	1	120 execution seconds / 120 elapsed seconds
180	380	2.11	360 execution seconds / 180 elapsed seconds
240	380	1.58	360 execution seconds / 240 elapsed seconds
300	600	2	600 execution seconds / 300 elapsed seconds

In most cases, the AAS and AAE for a query are approximately the same. However, because the inputs to the calculations are different data sources, the calculations often vary slightly.

Dimensions

The `DB_Load` metric has subcomponents called *dimensions*. You can think of dimensions as categories for the different characteristics of the `DB_Load` metric. When you are diagnosing performance issues, the most useful dimensions are wait events and top SQL.

Wait events

A *wait event* causes a SQL statement to wait for a specific event to happen before it can continue running. For example, a SQL statement might wait until a locked resource is unlocked. By combining `DB_Load` with wait events, you can get a complete picture of the session state. Wait events vary by DB engine:

- For information about all MariaDB and MySQL wait events, see [Wait Event Summary Tables](#) in the MySQL documentation.
- For information about all PostgreSQL wait events, see [PostgreSQL Wait Events](#) in the PostgreSQL documentation.
- For information about all Oracle wait events, see [Descriptions of Wait Events](#) in the Oracle documentation.
- For information about all SQL Server wait events, see [Types of Waits](#) in the SQL Server documentation.

Note

For Oracle, background processes sometimes do work without an associated SQL statement. In these cases, Performance Insights reports the type of background process concatenated with a

colon and the wait class associated with that background process. Types of background process include LGWR, ARC0, PMON, and so on.

For example, when the archiver is performing I/O, the Performance Insights report for it is similar to `ARC1 : System I/O`. Occasionally, the background process type is also missing, and Performance Insights only reports the wait class, for example `:System I/O`.

Top SQL

Whereas wait events show bottlenecks, top SQL shows which queries are contributing the most to DB load. For example, many queries might be currently running on the database, but a single query might consume 99% of the DB load. In this case, the high load might indicate a problem with the query.

By default, the Performance Insights console displays top SQL queries that are contributing to the database load. The console also shows relevant statistics for each statement. To diagnose performance problems for a specific statement, you can examine its execution plan.

Maximum CPU

In the dashboard, the **Database load** chart collects, aggregates, and displays session information. To see whether active sessions are exceeding the maximum CPU, look at their relationship to the **Max vCPU** line. The **Max vCPU** value is determined by the number of vCPU (virtual CPU) cores for your DB instance.

If the load is often above the **Max vCPU** line, and the primary wait state is CPU, the system CPU is overloaded. In these cases, you might want to throttle connections to the instance, tune any SQL queries with a high CPU load, or consider a larger instance class. High and consistent instances of any wait state indicate that there might be bottlenecks or resource contention issues to resolve. This can be true even if the load doesn't cross the **Max vCPU** line.

You can find an overview of Performance Insights in the following video.

[Using Performance Insights to Analyze Performance of Amazon Aurora PostgreSQL](#)

DB engine support for Performance Insights

Following, you can find the DB engines that support Performance Insights.

DB Engine	Supported DB Engine Versions
Amazon RDS for MariaDB	All 10.5 versions, all 10.4 versions, 10.3.13 and higher 10.3 versions, and 10.2.21 and higher 10.2 versions. Not supported for MariaDB version 10.0 or 10.1. Not supported for MariaDB version 10.3.13 DB instances in the Europe (Frankfurt) and Europe (Stockholm) AWS Regions. Not supported on the following DB instance classes: db.t2.micro, db.t2.small, db.t3.micro, and db.t3.small.
Amazon RDS for MySQL	8.0.17 and higher 8.0 versions, version 5.7.22 and higher 5.7 versions, and version 5.6.41 and higher 5.6 versions. Not supported for version 5.5. Not supported on the following DB instance classes: db.t2.micro, db.t2.small, db.t3.micro, and db.t3.small.
Amazon RDS for Microsoft SQL Server	All versions except SQL Server 2008.
Amazon RDS for PostgreSQL	Versions 10, 11, 12, and 13.

DB Engine	Supported DB Engine Versions
	is
Amazon RDS for Oracle	All versions.

AWS Region support for Performance Insights

Performance Insights for Amazon RDS isn't supported in the following AWS Regions:

- Middle East (Bahrain) Region
- Europe (Milan) Region
- Africa (Cape Town) Region
- Asia Pacific (Osaka)
- AWS GovCloud (US) Regions

Important

This guide describes using Amazon RDS Performance Insights with non-Aurora DB engines. For information about using Amazon RDS Performance Insights with Amazon Aurora, see the [Using Amazon RDS Performance Insights](#) in the *Amazon Aurora User Guide*.

Enabling and disabling Performance Insights

To use Performance Insights, enable it on your DB instance. If needed, you can disable it later. Enabling and disabling Performance Insights doesn't cause downtime, a reboot, or a failover.

The Performance Insights agent consumes limited CPU and memory on the DB host. When the DB load is high, the agent limits the performance impact by collecting data less frequently.

Console

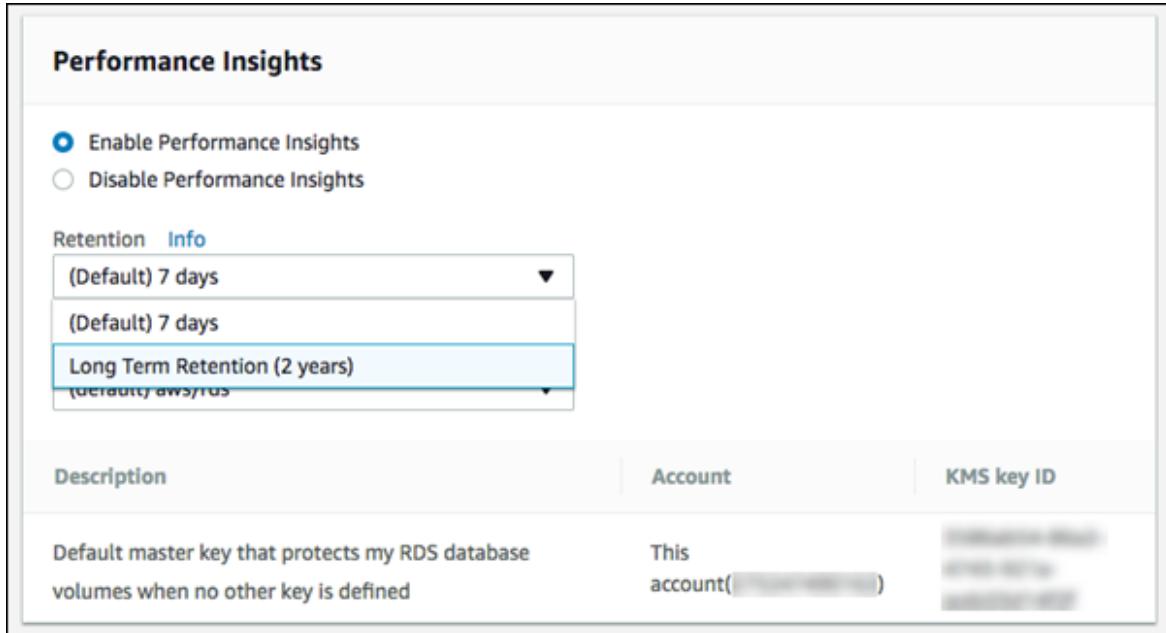
In the console, you can enable or disable Performance Insights when you create or modify a new DB instance.

[Enabling or disabling Performance Insights when creating an instance](#)

When you create a new DB instance, enable Performance Insights by choosing **Enable Performance Insights** in the **Performance Insights** section. Or choose **Disable Performance Insights**.

To create a DB instance, follow the instructions for your DB engine in [Creating an Amazon RDS DB instance \(p. 141\)](#).

The following screenshot shows the **Performance Insights** section.



If you choose **Enable Performance Insights**, you have the following options:

- **Retention** – The amount of time to retain Performance Insights data. Choose either 7 days (the default) or 2 years.
- **Master key** – Specify your AWS Key Management Service (AWS KMS) customer master key (CMK). Performance Insights encrypts all potentially sensitive data using your AWS KMS CMK. Data is encrypted in flight and at rest. For more information, see [Encrypting Amazon RDS resources \(p. 1630\)](#).

[Enabling or disabling Performance Insights when modifying an instance](#)

In the console, you can modify a DB instance to enable or disable Performance Insights using the console.

To enable or disable Performance Insights for a DB instance using the console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose **Databases**.
3. Choose a DB instance, and choose **Modify**.
4. In the **Performance Insights** section, choose either **Enable Performance Insights** or **Disable Performance Insights**.

If you choose **Enable Performance Insights**, you have the following options:

- **Retention** – The amount of time to retain Performance Insights data. Choose either 7 days (the default) or 2 years.
 - **Master key** – Specify your AWS Key Management Service (AWS KMS) customer master key (CMK). Performance Insights encrypts all potentially sensitive data using your AWS KMS CMK. Data is encrypted in flight and at rest. For more information, see [Encrypting Amazon RDS resources \(p. 1630\)](#).
5. Choose **Continue**.
 6. For **Scheduling of Modifications**, choose one of the following:

- **Apply during the next scheduled maintenance window** – Wait to apply the **Performance Insights** modification until the next maintenance window.
 - **Apply immediately** – Apply the **Performance Insights** modification as soon as possible.
7. Choose **Modify instance**.

AWS CLI

When you use the [create-db-instance](#) AWS CLI command, enable Performance Insights by specifying `--enable-performance-insights`. Or disable Performance Insights by specifying `--no-enable-performance-insights`.

You can also specify these values using the following AWS CLI commands:

- [create-db-instance-read-replica](#)
- [modify-db-instance](#)
- [restore-db-instance-from-s3](#)

The following procedure describes how to enable or disable Performance Insights for a DB instance using the AWS CLI.

To enable or disable Performance Insights for a DB instance using the AWS CLI

- Call the [modify-db-instance](#) AWS CLI command and supply the following values:
 - `--db-instance-identifier` – The name of the DB instance.
 - `--enable-performance-insights` to enable or `--no-enable-performance-insights` to disable

The following example enables Performance Insights for `sample-db-instance`.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
  --db-instance-identifier sample-db-instance \
  --enable-performance-insights
```

For Windows:

```
aws rds modify-db-instance ^
  --db-instance-identifier sample-db-instance ^
  --enable-performance-insights
```

When you enable Performance Insights, you can optionally specify the amount of time, in days, to retain Performance Insights data with the `--performance-insights-retention-period` option. Valid values are 7 (the default) or 731 (2 years).

The following example enables Performance Insights for `sample-db-instance` and specifies that Performance Insights data is retained for two years.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
  --db-instance-identifier sample-db-instance \
```

```
--enable-performance-insights \
--performance-insights-retention-period 731
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier sample-db-instance ^
--enable-performance-insights ^
--performance-insights-retention-period 731
```

RDS API

When you create a new DB instance using the [CreateDBInstance](#) operation Amazon RDS API operation, enable Performance Insights by setting `EnablePerformanceInsights` to `True`. To disable Performance Insights, set `EnablePerformanceInsights` to `False`.

You can also specify the `EnablePerformanceInsights` value using the following API operations:

- [ModifyDBInstance](#)
- [CreateDBInstanceReadReplica](#)
- [RestoreDBInstanceFromS3](#)

When you enable Performance Insights, you can optionally specify the amount of time, in days, to retain Performance Insights data with the `PerformanceInsightsRetentionPeriod` parameter. Valid values are 7 (the default) or 731 (2 years).

Enabling the Performance Schema for Performance Insights on Amazon RDS for MariaDB or MySQL

When the Performance Schema is enabled for Amazon RDS for MariaDB or MySQL, Performance Insights provides more detailed information. For example, Performance Insights displays DB load categorized by detailed wait events. When Performance Schema isn't enabled, Performance Insights displays DB load categorized by the list state of the MySQL process.

You have the following options for enabling the Performance Schema:

- Allow Performance Insights to manage required parameters automatically.

When you create an Amazon RDS for MariaDB or MySQL DB instance with Performance Insights enabled, Performance Schema is enabled automatically. In this case, Performance Insights automatically manages your parameters.

Important

In this scenario, Performance Insights changes schema-related parameters on the DB instance. These changes aren't visible in the parameter group associated with the DB instance. However, these changes are visible in the output of the `SHOW GLOBAL VARIABLES` command.

- Set the required parameters yourself.

For Performance Insights to list wait events, you must set all parameters as shown in the following table.

Parameter Name	Parameter Value
<code>performance_schema</code>	1 (the <code>Source</code> column has the value <code>engine-default</code>)

Parameter Name	Parameter Value
performance-schema-consumer-events-waits-current	ON
performance-schema-instrument	wait/%=ON
performance-schema-consumer-global-instrumentation	ON
performance-schema-consumer-thread-instrumentation	ON

For more information, see [Performance Schema Command Options](#) and [Performance Schema Option and Variable Reference](#) in the MySQL documentation.

Enabling the Performance Schema manually

Performance Schema is *not* enabled when both the following conditions are true:

- The `performance_schema` parameter is set to 0 or 1.
- The **Source** column for the `performance_schema` parameter is set to `user`.

To enable the Performance Schema manually

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose **Parameter groups**.
3. Select the name of the parameter group for your DB instance.
4. Choose **Edit parameters**.
5. Enter `perf` in the search bar.
6. Select the `performance_schema` parameter.
A screenshot of the AWS RDS Parameter Group configuration screen. A blue rectangular box highlights the 'performance_schema' parameter. To its left is a checked checkbox icon. The parameter name 'performance_schema' is displayed to the right of the checkbox.
7. Choose **Reset**.
8. Choose **Reset parameters**.
9. Restart the DB instance.

For more information about modifying instance parameters, see [Modifying parameters in a DB parameter group \(p. 232\)](#). For more information about the dashboard, see [Monitoring with the Performance Insights dashboard \(p. 421\)](#). For more information about the MySQL performance schema, see [MySQL 8.0 Reference Manual](#).

Accessing Performance Insights

To access Performance Insights, you must have the appropriate permissions from AWS Identity and Access Management (IAM). There are two options available for granting access:

1. Attach the `AmazonRDSFullAccess` managed policy to an IAM user or role.
2. Create a custom IAM policy and attach it to an IAM user or role.

AmazonRDSFullAccess managed policy

AmazonRDSFullAccess is an AWS-managed policy that grants access to all of the Amazon RDS API operations. The policy also grants access to related services that are used by the Amazon RDS console—for example, event notifications using Amazon SNS.

In addition, AmazonRDSFullAccess contains all the permissions needed for using Performance Insights. If you attach this policy to an IAM user or role, the recipient can use Performance Insights, along with other console features.

Using a custom IAM policy

For users who don't have full access with the AmazonRDSFullAccess policy, you can grant access to Performance Insights by creating or modifying a user-managed IAM policy. When you attach the policy to an IAM user or role, the recipient can use Performance Insights.

To create a custom policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**.
3. Choose **Create policy**.
4. On the **Create Policy** page, choose the JSON tab.
5. Copy and paste the following.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "pi:*",  
            "Resource": "arn:aws:pi:*:metrics/rds/*"  
        }  
    ]  
}
```

6. Choose **Review policy**.
7. Provide a name for the policy and optionally a description, and then choose **Create policy**.

You can now attach the policy to an IAM user or role. The following procedure assumes that you already have an IAM user available for this purpose.

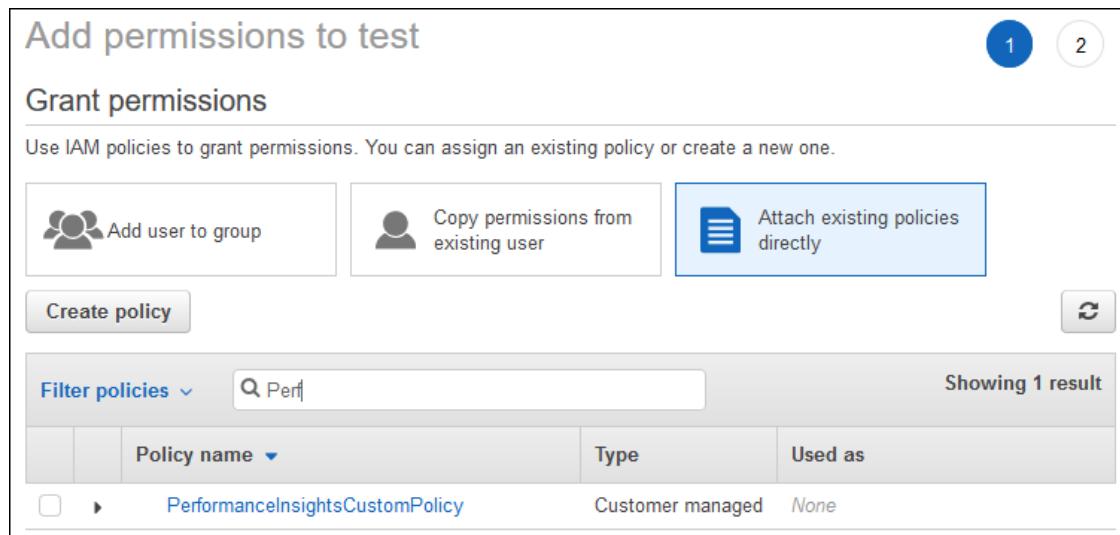
To attach the policy to an IAM user

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Users**.
3. Choose an existing user from the list.

Important

To use Performance Insights, make sure that you have access to Amazon RDS in addition to the custom policy. For example, the AmazonRDSReadOnlyAccess predefined policy provides read-only access to Amazon RDS. For more information, see [Managing access using policies \(p. 1646\)](#).

4. On the **Summary** page, choose **Add permissions**.
5. Choose **Attach existing policies directly**. For **Search**, type the first few characters of your policy name, as shown following.



6. Choose your policy, and then choose **Next: Review**.
7. Choose **Add permissions**.

Monitoring with the Performance Insights dashboard

The Performance Insights dashboard contains database performance information to help you analyze and troubleshoot performance issues. On the main dashboard page, you can view information about the database load. You can also drill into details for a particular wait state, SQL query, host, or user.

Opening the Performance Insights dashboard

To see the Performance Insights dashboard, use the following procedure.

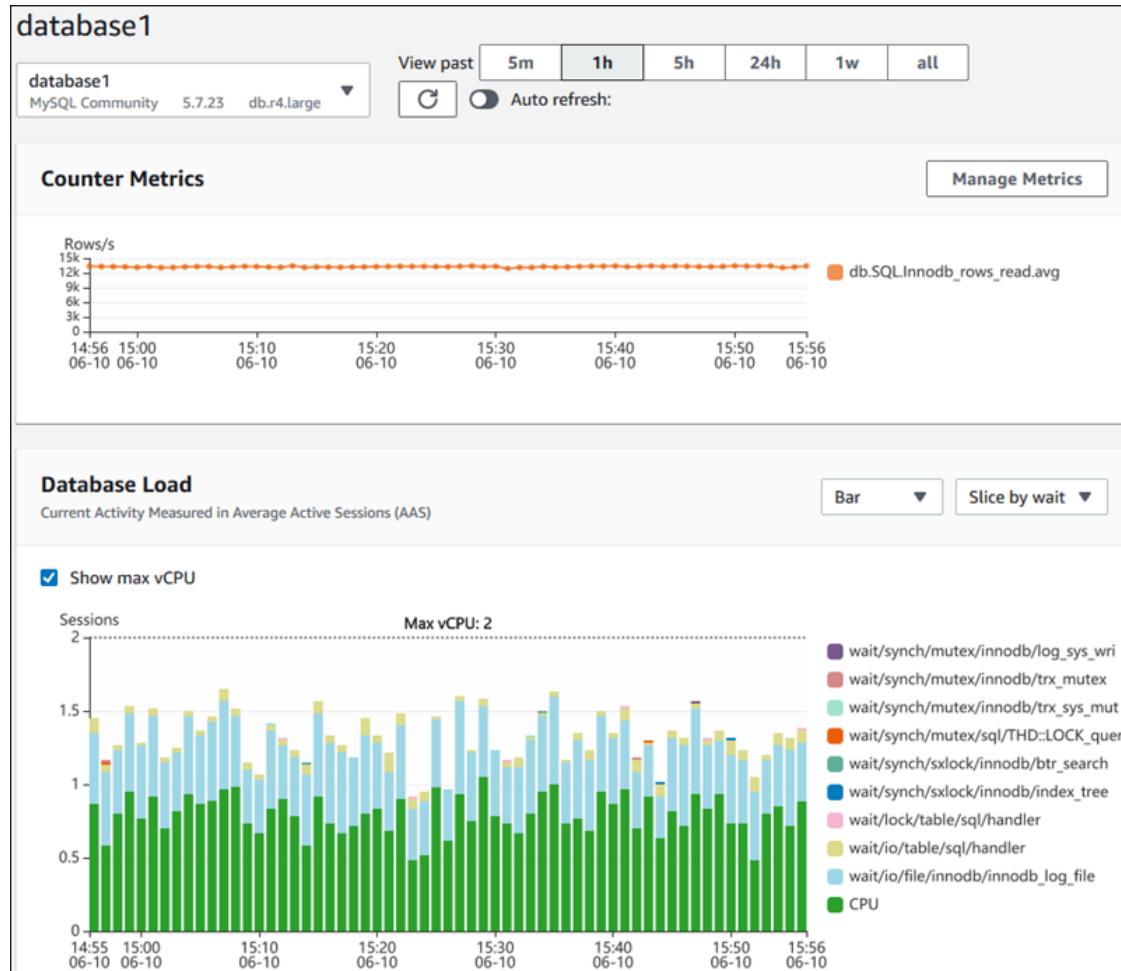
To view the Performance Insights dashboard in the AWS Management Console

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Performance Insights**.
3. Choose a DB instance. The Performance Insights dashboard is displayed for that DB instance.

For DB instances with Performance Insights enabled, you can also reach the dashboard by choosing the **Sessions** item in the list of DB instances. Under **Current activity**, the **Sessions** item shows the database load in average active sessions over the last five minutes. The bar graphically shows the load. When the bar is empty, the DB instance is idle. As the load increases, the bar fills with blue. When the load passes the number of virtual CPUs (vCPUs) on the DB instance class, the bar turns red, indicating a potential bottleneck.

Databases		<input checked="" type="radio"/> Group resources		Modify	Actions ▾	Restore from S3	Create database
<input type="text"/> Filter databases							
<input type="checkbox"/>	DB identifier		▲	Engine	▼	CPU	Current activity
<input type="checkbox"/>	database1			MySQL Community		<div style="width: 45.51%;">45.51%</div>	<div style="width: 1.34%;">1.34 Sessions</div>
<input type="checkbox"/>	database2			Oracle Enterprise Edition		<div style="width: 55.41%;">55.41%</div>	<div style="width: 3.48%;">3.48 Sessions</div>
<input type="checkbox"/>	database3			Oracle Enterprise Edition		<div style="width: 1.02%;">1.02%</div>	<div style="width: 0%;">0 Connections</div>

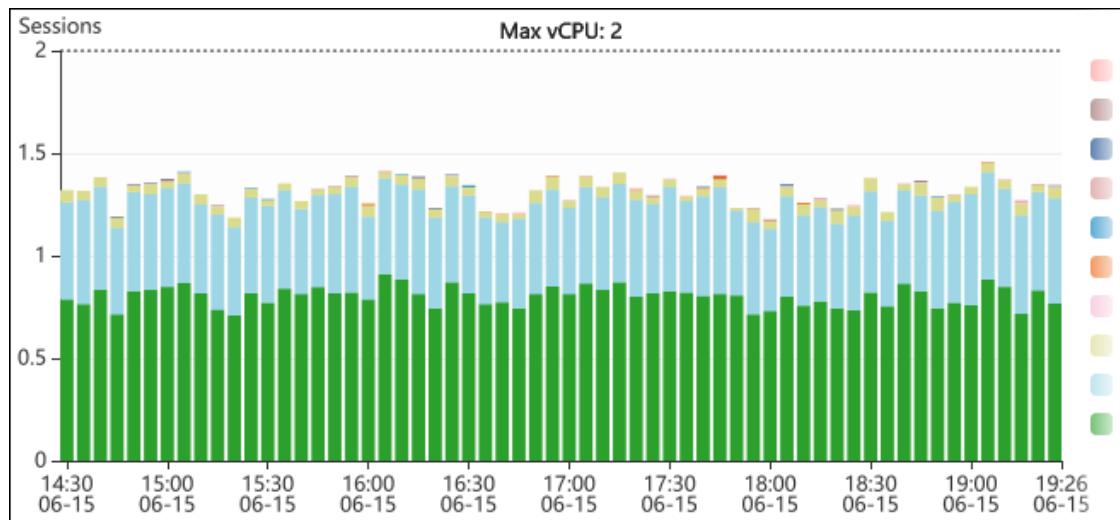
The following screenshot shows the dashboard for a DB instance. By default, the Performance Insights dashboard shows data for the last hour.



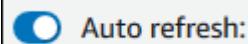
4. (Optional) Choose a different time interval by selecting a button in the upper right. For example, to change the interval to 5 hours, select **5h**.



In the following screenshot, the DB load interval is 5 hours.



5. (Optional) To refresh your data automatically, enable **Auto refresh**.



The Performance Insight dashboard automatically refreshes with new data. The refresh rate depends on the amount of data displayed:

- 5 minutes refreshes every 5 seconds.
- 1 hour refreshes every minute.
- 5 hours refreshes every minute.
- 24 hours refreshes every 5 minutes.
- 1 week refreshes every hour.

Performance Insights dashboard components

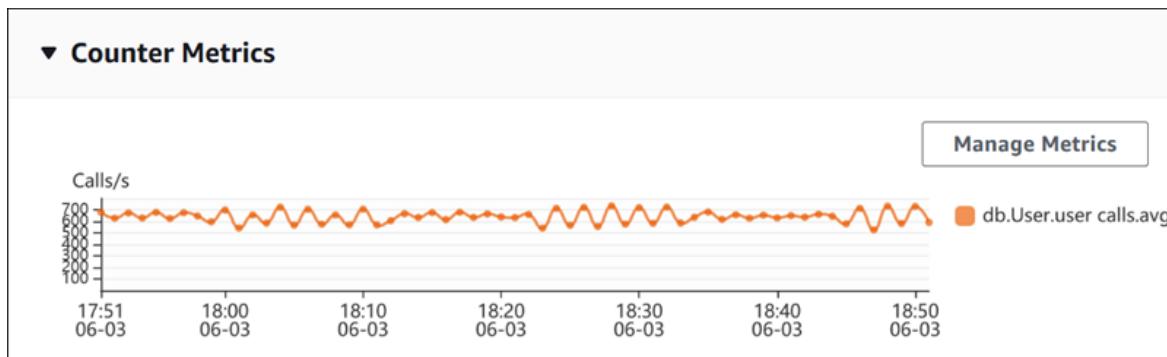
The dashboard is divided into three parts:

1. **Counter Metrics** – Shows data for specific performance counter metrics.
2. **DB Load Chart** – Shows how the DB load compares to DB instance capacity as represented by the **Max vCPU** line.
3. **Top items** – Shows the top waits, SQL, hosts, and users contributing to DB load.

Counter Metrics chart

The **Counter Metrics** chart displays data for performance counters. The default metrics depend on the DB engine.

- MySQL and MariaDB – db.SQL.Innodb_rows_read.avg
- Oracle – db.User.user_calls.avg
- Microsoft SQL Server – db.Databases.Active_Transactions(_Total).avg
- PostgreSQL – db.Transactions.xact_commit.avg



Change the performance counters by choosing **Manage Metrics**. You can select multiple **OS metrics** or **Database metrics**, as shown in the following screenshot. To see details for any metric, hover over the metric name.

This screenshot shows the 'Select metrics shown on the graph' dialog box. It includes a search bar labeled 'Find metrics', a tab for 'Database metrics (1)' which is selected, and a 'Clear all selections' button. The main area lists various metrics grouped by category:

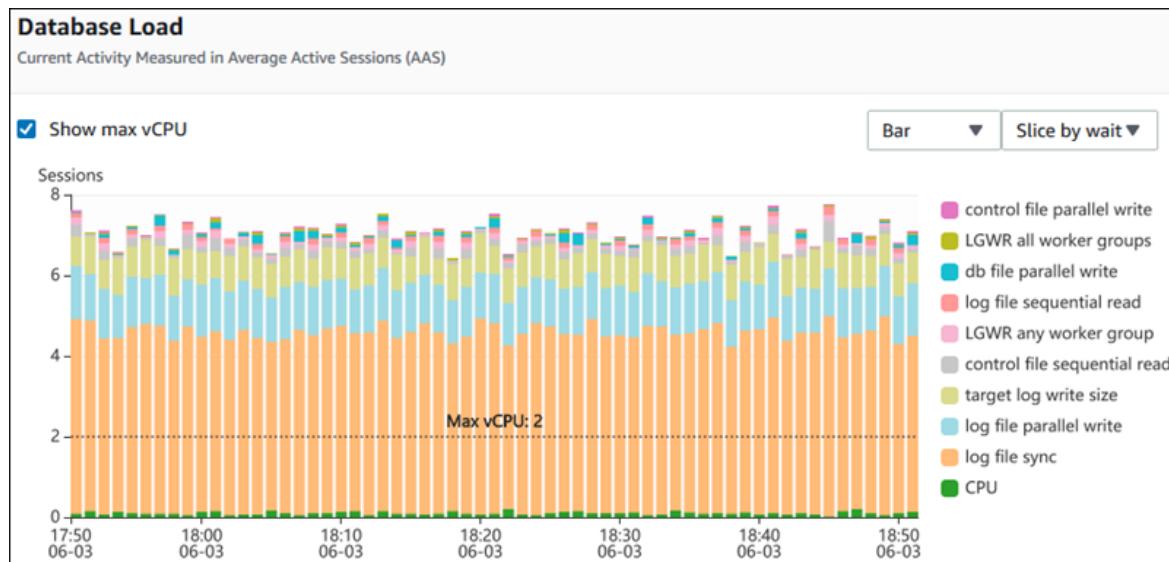
- User**:
 - CPU used by this session
 - user commits
 - bytes sent via SQL*Net to client
 - SQL*Net roundtrips to/from client
 - logons cumulative
 - user calls
 - bytes received via SQL*Net from client
 - user rollbacks
- Redo**:
 - redo size
- Cache**:
 - physical read bytes
 - physical reads
 - consistent gets
 - db block gets
 - consistent gets from cache
 - DBWR checkpoints
 - db block gets from cache
- SQL**:
 - parse count (total)
 - sorts (memory)
 - parse count (hard)
 - sorts (disk)
 - table scan rows gotten
 - sorts (rows)

At the bottom right are 'Cancel' and 'Update graph' buttons.

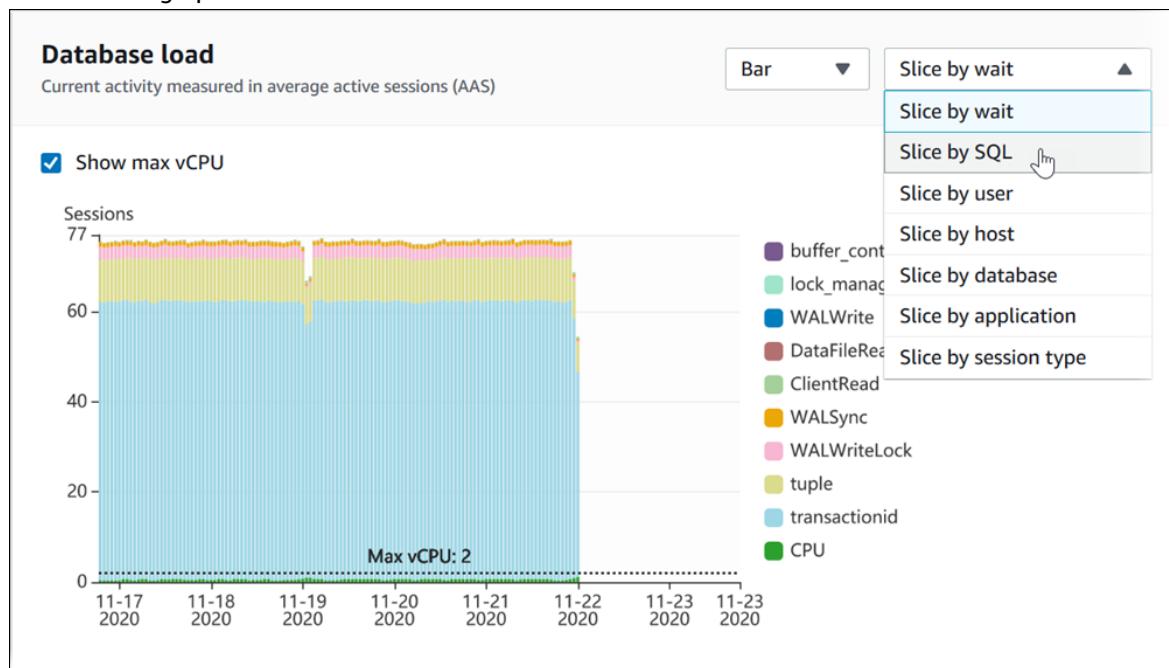
For more information, see [Customizing the Performance Insights dashboard \(p. 444\)](#).

Average Active Sessions chart

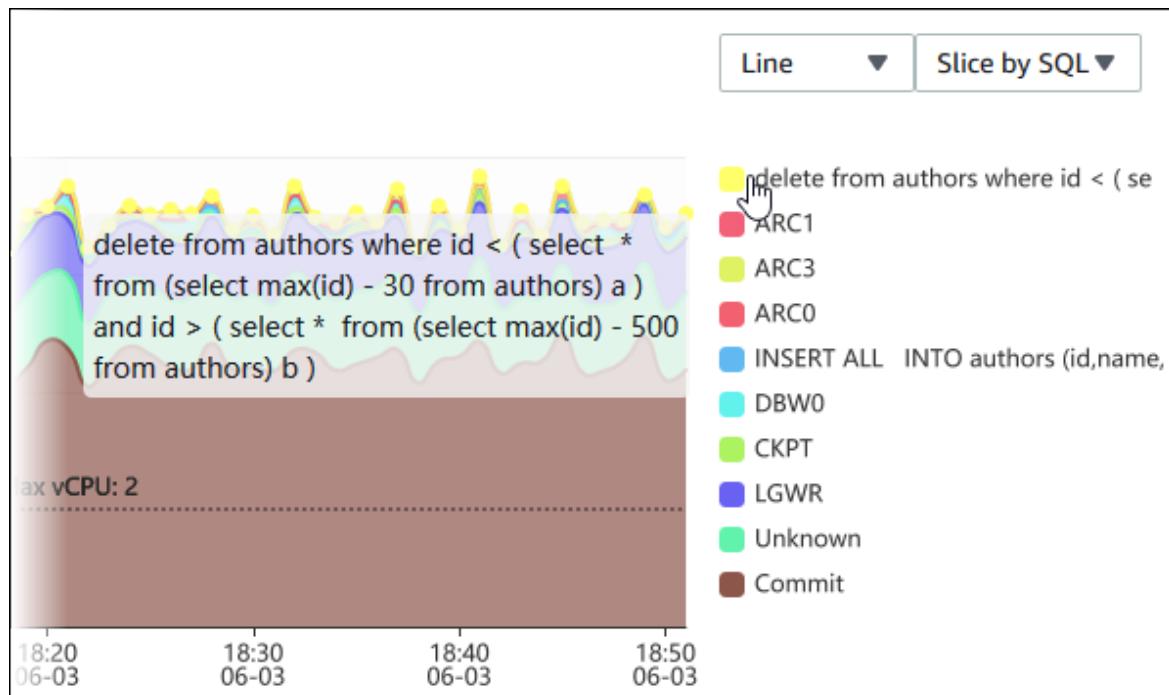
The **Database Load** chart shows how the database load compares to DB instance capacity as represented by the **Max vCPU** line. By default, load is shown as active sessions grouped by wait states in a bar graph.



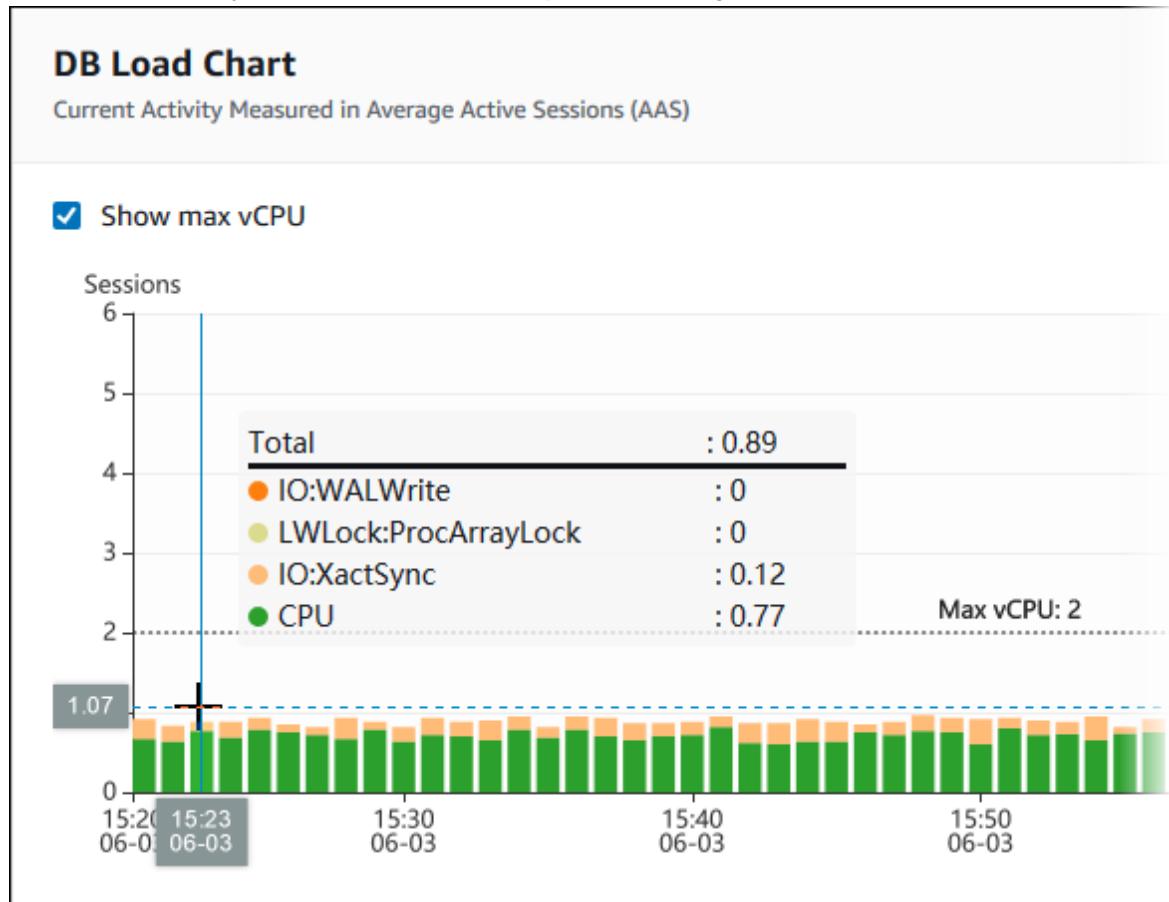
You can choose to display load as active sessions grouped by waits, SQL, users, or hosts. You can also choose a line graph.



To see details about a DB load item such as a SQL statement, hover over the item name.



To see details for any item for the selected time period in the legend, hover over that item.



Top load table

The Top load table shows the top items contributing to database load. You can choose any of the following dimension tabs:

- Top SQL – The SQL statements that are currently running
- Top waits – The event for which the database backend is waiting
- Top hosts – The host name of the connected client
- Top users – The user logged in to the database
- Top databases – The name of the database to which the client is connected (PostgreSQL, MySQL, and MariaDB only)
- Top applications (PostgreSQL only) – The name of the application that is connected to the database
- Top session types (PostgreSQL only) – The type of the current session

By default, the console displays top SQL queries that are contributing to the database load. Every line in the table shows relevant statistics for the SQL statement:

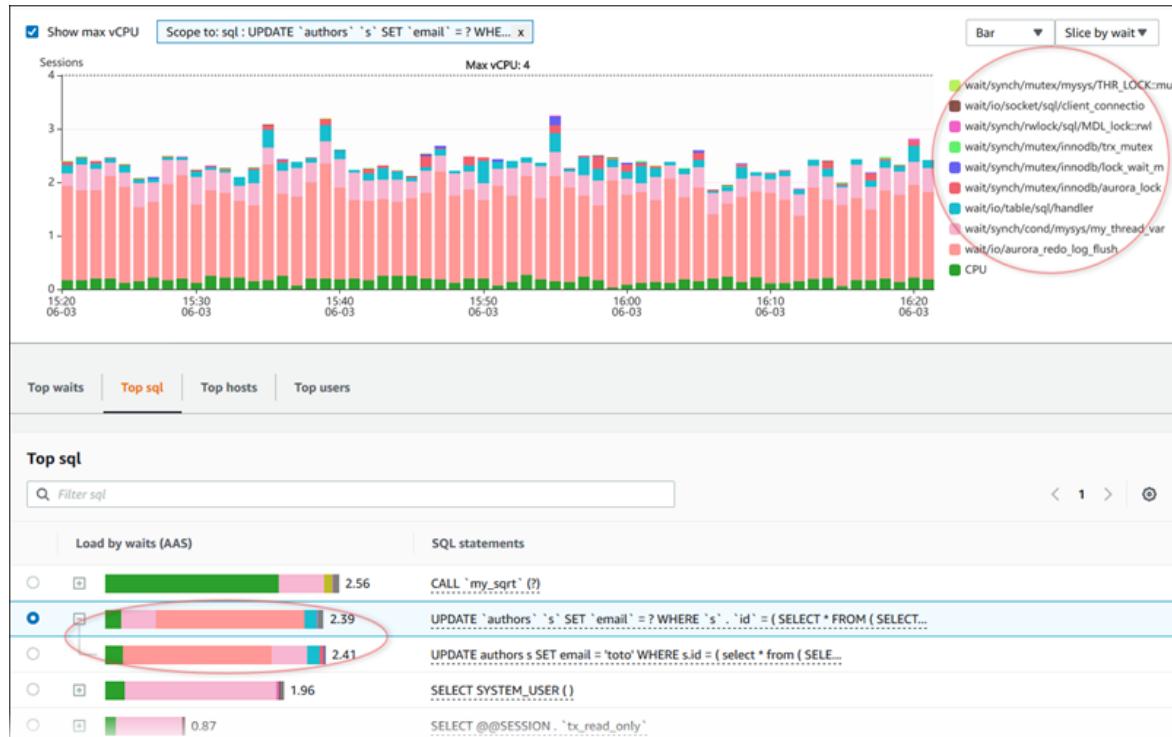
Top SQL			
<input type="text"/> Filter sql			
Load by waits (AAS)	SQL statements	calls/sec	rows/sec
0.88	select minute_rollups(?)	0.06	0.06
0.53	select count(*) from authors where id < (select max(id) - 31 from authors) and...	33.68	101.04
0.17	WITH cte AS (SELECT id FROM authors LIMIT ?) UPDATE ...	33.68	33.68
0.08	delete from authors where id < (select * from (select max(id) - ? from authors...)	33.68	303.13
0.07	INSERT INTO authors (id,name,email) VALUES (nextval(?) ,? , (nextval(?) ,?))	33.68	303.13
0.06	select count(*) from authors where id < (select max(id) - 31 from authors) and...	0.00	0.00

To see the components of a query, select the query, and then choose the +. A *SQL digest* is a composite of multiple actual queries that are structurally similar but that possibly have different literal values. In the following screenshot, the selected query is a digest. The digest replaces hardcoded values with a question mark.

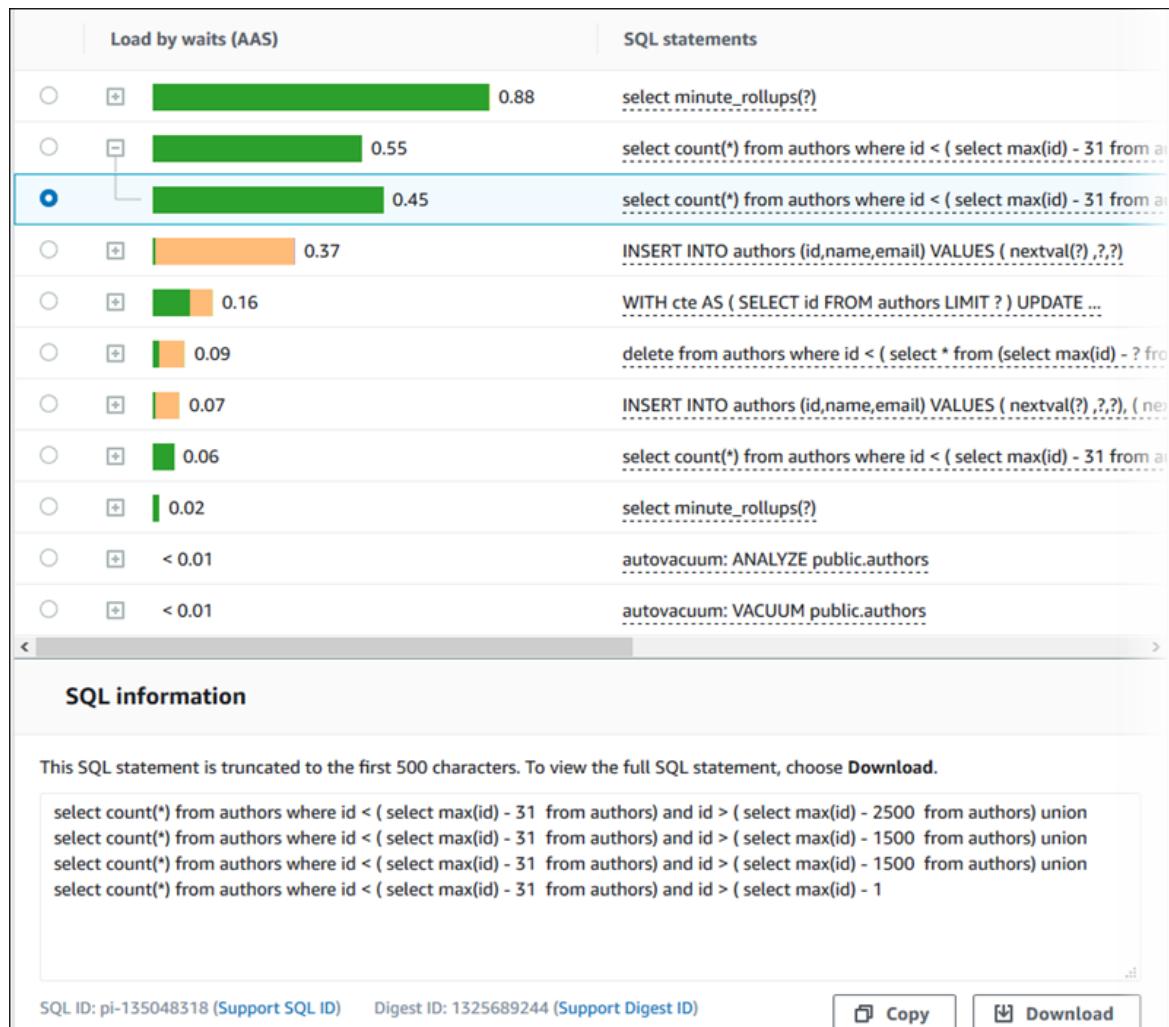
Note

A SQL digest groups similar SQL statements, but does not redact sensitive information.

In **Top sql**, the **Load by waits (AAS)** column illustrates the percentage of the database load associated with each top load item. This column reflects the load for that item by whatever grouping is currently selected in the **DB Load Chart**. For example, you might group the **DB Load Chart** chart by wait states. You examine SQL queries in the top load items table. In this case, the **DB Load by Waits** bar is sized, segmented, and color-coded to show how much of a given wait state that query is contributing to. It also shows which wait states are affecting the selected query.



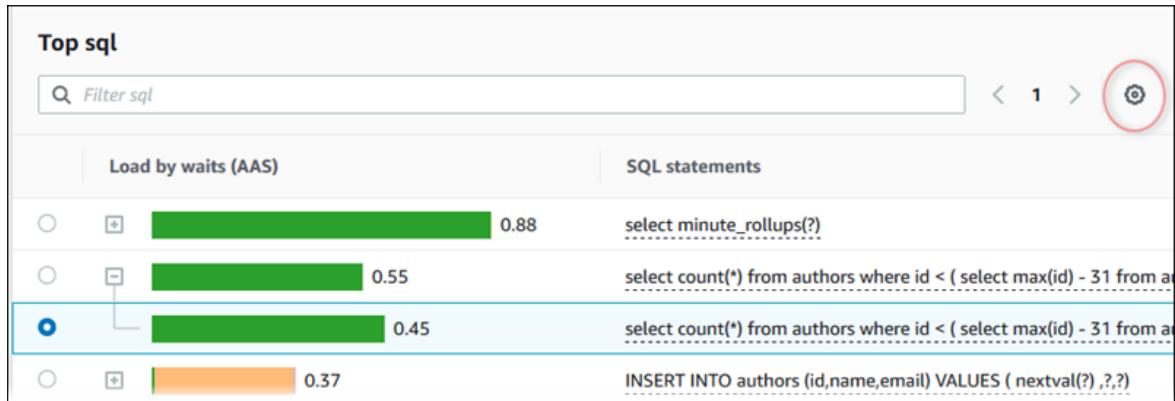
In the **Top sql** table, you can open a statement to view its information. The information appears in the bottom pane.



In the **Top sql** tab, you can view the following types of identifiers (IDs) that are associated with SQL statements:

- **Support SQL ID** – A hash value of the SQL ID. This value is only for referencing a SQL ID when you are working with AWS Support. AWS Support doesn't have access to your actual SQL IDs and SQL text.
- **Support Digest ID** – A hash value of the digest ID. This value is only for referencing a digest ID when you are working with AWS Support. AWS Support doesn't have access to your actual digest IDs and SQL text.

You can control the statistics displayed in the **Top sql** tab by choosing the **Preferences** icon.



When you choose the **Preferences** icon, the **Preferences** window opens.

The Preferences window allows you to set page size and column visibility. The 'All resources' option is selected under 'Page size'. Under 'Columns', most metrics are enabled (indicated by blue switches), except for 'local blk dirty/sec (local_blk_dirtied_per_sec)' which is disabled (gray switch).

Column	Status
Load by waits (AAS)	Enabled
SQL statements	Enabled
calls/sec (calls_per_sec)	Enabled
rows/sec (rows_per_sec)	Enabled
AAE (total_time_per_sec)	Enabled
blk hits/sec (shared_blk_hit_per_sec)	Enabled
blk reads/sec (shared_blk_read_per_sec)	Enabled
blk dirty/sec (shared_blk_dirtied_per_sec)	Enabled
blk writes/sec (shared_blk_written_per_sec)	Enabled
local blk hits/sec (local_blk_hit_per_sec)	Enabled
local blk reads/sec (local_blk_read_per_sec)	Enabled
local blk dirty/sec (local_blk_dirtied_per_sec)	Disabled

Enable the statistics that you want to have visible in the **Top sql** tab, use your mouse to scroll to the bottom of the window, and then choose **Continue**.

Analyzing DB load using the Performance Insights dashboard

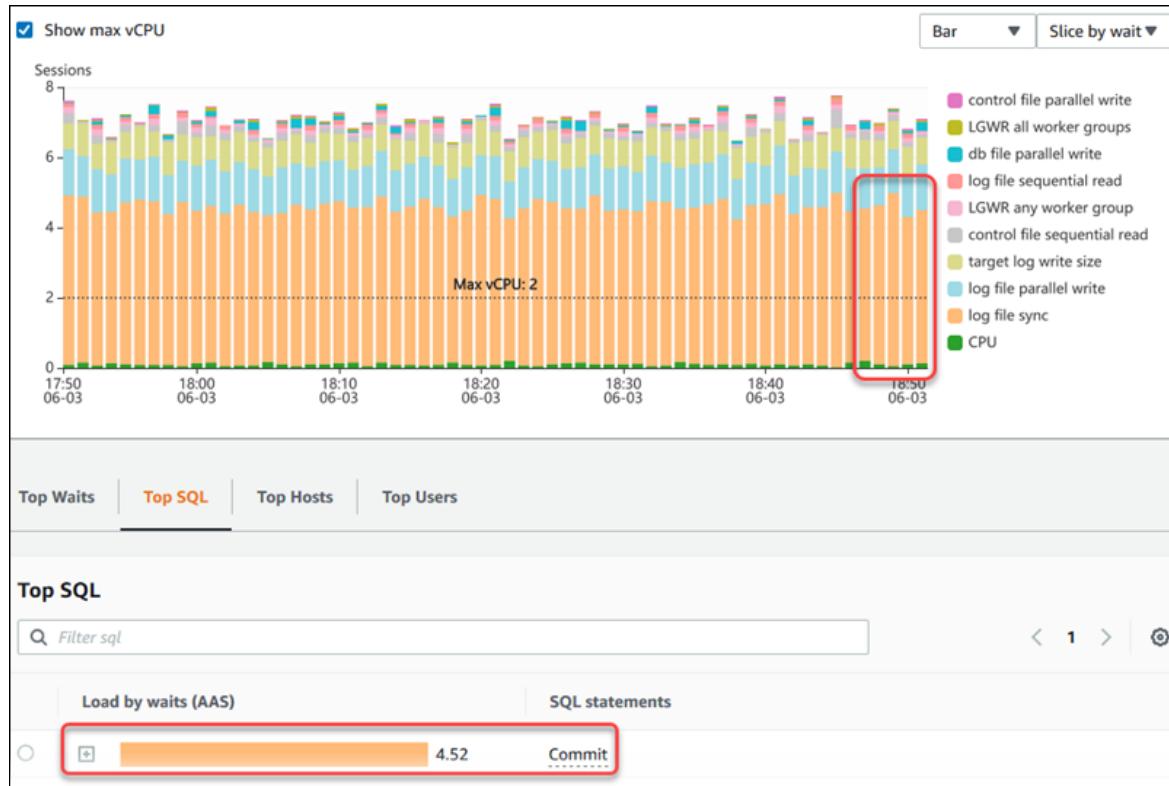
If the **Average Active Sessions** chart shows a bottleneck, you can find out where the load is coming from. To do so, look at the top load items table below the **Average Active Sessions** chart. Choose a particular item, like a SQL query or a user, to drill down into that item and see details about it.

DB load grouped by waits and top SQL queries is the default Performance Insights dashboard view. This combination typically provides the most insight into performance issues. DB load grouped by waits shows if there are any resource or concurrency bottlenecks in the database. In this case, the **SQL** tab of the top load items table shows which queries are driving that load.

Your typical workflow for diagnosing performance issues is as follows:

1. Review the **Average Active Sessions** chart and see if there are any incidents of database load exceeding the **Max CPU** line.
2. If there is, look at the **Average Active Sessions** chart and identify which wait state or states are primarily responsible.
3. Identify the digest queries causing the load by seeing which of the queries the **SQL** tab on the top load items table are contributing most to those wait states. You can identify these by the **DB Load by Wait** column.
4. Choose one of these digest queries in the **SQL** tab to expand it and see the child queries that it is composed of.

For example, in the dashboard following, **log file sync** waits account for most of the DB load. The **LGWR all worker groups** wait is also high. The **Top sql** chart shows what is causing the **log file sync** waits: frequent COMMIT statements. In this case, committing less frequently will reduce DB load.



Analyzing statistics for running queries

In Amazon RDS Performance Insights, you can find statistics on running queries in the **Top SQL** section. Performance Insights collects statistics only for the most common queries. Typically, these match the top queries by load shown in the Performance Insights dashboard.

Topics

- [Statistics for MariaDB and MySQL \(p. 432\)](#)
- [Statistics for Oracle \(p. 435\)](#)
- [Statistics for PostgreSQL \(p. 438\)](#)

Statistics for MariaDB and MySQL

Performance Insights collects SQL digest statistics from the `events_statements_summary_by_digest` table. This table is managed by the database and doesn't have an eviction policy. If the table becomes full, new SQL queries aren't tracked. To address this issue, Performance Insights automatically truncates the table when it's nearly full.

Performance Insights automatically truncates the table only if your parameter group doesn't have an explicitly set value for the `performance_schema` parameter. You can examine the `performance_schema` parameter, and if the value of source is `user`, then you set a value. If you want Performance Insights to truncate the table automatically, then reset the value for the `performance_schema` parameter. You can view the source of a parameter value by viewing the parameter in the AWS Management Console or by running the AWS CLI [describe-db-parameters](#) command. The following message is shown in the AWS Management Console when the table is full:

Performance Insights is unable to collect SQL Digest statistics on new queries because the table `events_statements_summary_by_digest` is full.
Please truncate `events_statements_summary_by_digest` table to clear the issue. Check the User Guide for more details.

The following SQL statistics are available for MariaDB and MySQL DB instances.

Metric	Unit
<code>db.sql_tokenized.stats.count_star_per_sec</code>	Calls per second
<code>db.sql_tokenized.stats.sum_timer_wait_per_sec</code>	Average active executions per second (AAE)
<code>db.sql_tokenized.stats.sum_select_full_join_per_sec</code>	Select full join per second
<code>db.sql_tokenized.stats.sum_select_range_check_per_sec</code>	Select range check per second
<code>db.sql_tokenized.stats.sum_select_scan_per_sec</code>	Select scan per second
<code>db.sql_tokenized.stats.sum_sort_merge_passes_per_sec</code>	Sort merge passes per second
<code>db.sql_tokenized.stats.sum_sort_scan_per_sec</code>	Sort scans per second
<code>db.sql_tokenized.stats.sum_sort_range_per_sec</code>	Sort ranges per second
<code>db.sql_tokenized.stats.sum_sort_rows_per_sec</code>	Sort rows per second
<code>db.sql_tokenized.stats.sum_rows_affected_per_sec</code>	Rows affected per second
<code>db.sql_tokenized.stats.sum_rows_examined_per_sec</code>	Rows examined per second

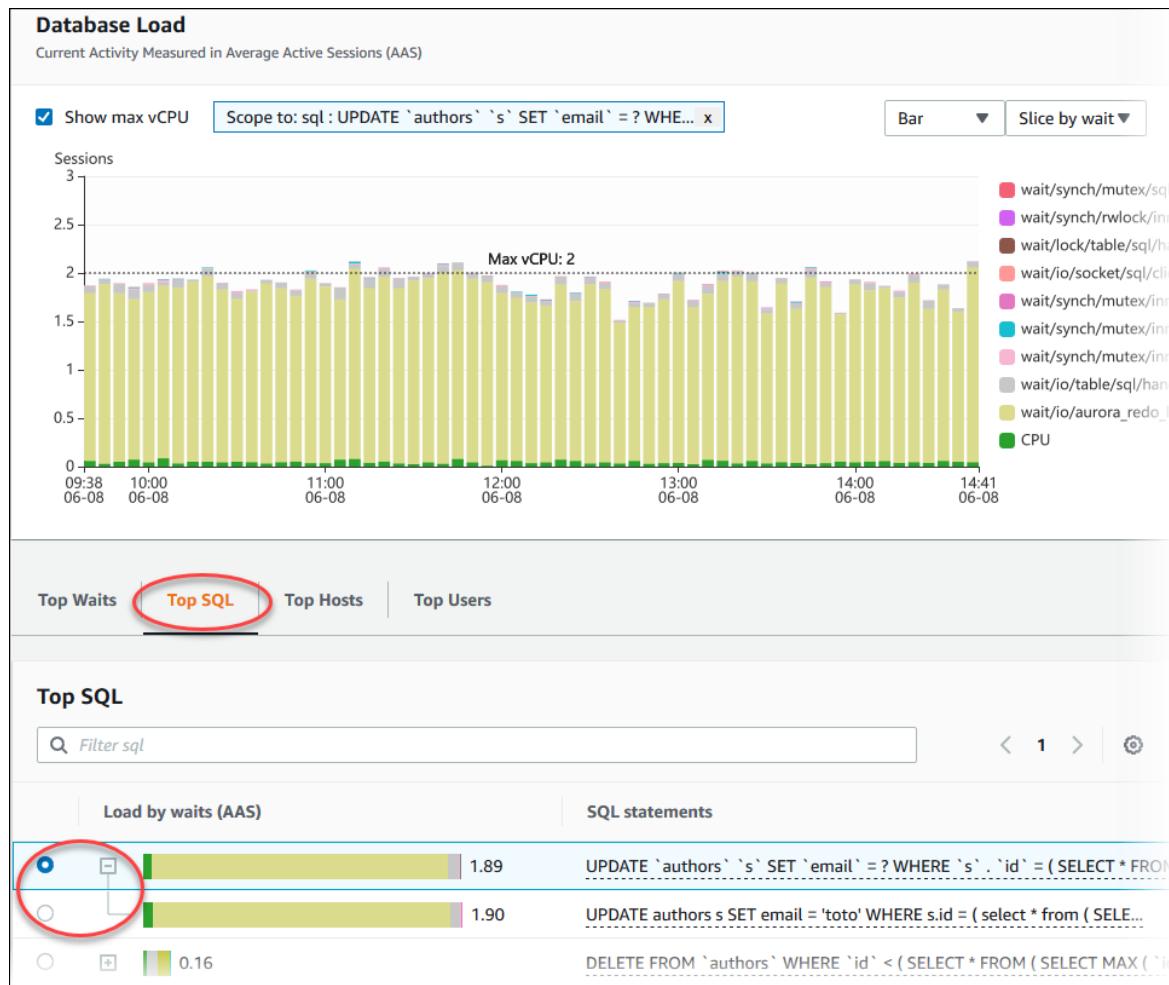
Metric	Unit
db.sql_tokenized.stats.sum_rows_sent_per_sec	Rows sent per second
db.sql_tokenized.stats.sum_created_tmp_disk_tables	Created temporary disk tables per second
db.sql_tokenized.stats.sum_created_tmp_tables_per_sec	Created temporary tables per second
db.sql_tokenized.stats.sum_lock_time_per_sec	Lock time per second (in ms)

The following metrics provide per call statistics for a SQL statement.

Metric	Unit
db.sql_tokenized.stats.sum_timer_wait_per_call	Average latency per call (in ms)
db.sql_tokenized.stats.sum_select_full_join_per_call	Select full joins per call
db.sql_tokenized.stats.sum_select_range_check_per_call	Select range check per call
db.sql_tokenized.stats.sum_select_scan_per_call	Select scans per call
db.sql_tokenized.stats.sum_sort_merge_passes_per_call	Sort merge passes per call
db.sql_tokenized.stats.sum_sort_scan_per_call	Sort scans per call
db.sql_tokenized.stats.sum_sort_range_per_call	Sort ranges per call
db.sql_tokenized.stats.sum_sort_rows_per_call	Sort rows per call
db.sql_tokenized.stats.sum_rows_affected_per_call	Rows affected per call
db.sql_tokenized.stats.sum_rows_examined_per_call	Rows examined per call
db.sql_tokenized.stats.sum_rows_sent_per_call	Rows sent per call
db.sql_tokenized.stats.sum_created_tmp_disk_tables	Created temporary disk tables per call
db.sql_tokenized.stats.sum_created_tmp_tables_per_call	Created temporary tables per call
db.sql_tokenized.stats.sum_lock_time_per_call	Lock time per call (in ms)

Analyzing MariaDB and MySQL Metrics for running SQL statements

Using the AWS Management Console, you can view the metrics for a running SQL query by choosing the **SQL** tab and expanding the query.



Choose which statistics to display by choosing the gear icon in the upper-right corner of the chart.

The following screenshot shows the preferences for MariaDB and MySQL DB instances.

Preferences

Page size

All resources

Wrap lines
Check to see all the text and wrap the lines

Columns

Load by waits (AAS)	<input checked="" type="button"/>
SQL statements	<input checked="" type="button"/>
Support ID	<input type="button"/>
ID	<input type="button"/>
calls/sec (count_star_per_sec)	<input type="button"/>
AAE (sum_timer_wait_per_sec)	<input type="button"/>
select full join/sec (sum_select_full_join_per_sec)	<input type="button"/>
select range check/sec (sum_select_range_check_per_sec)	<input type="button"/>

Statistics for Oracle

The following SQL statistics are available for Oracle DB instances.

Metric	Unit
db.sql.stats.executions_per_sec	Number of executions per second
db.sql.stats.elapsed_time_per_sec	Average active executions (AAE)
db.sql.stats.rows_processed_per_sec	Rows processed per second
db.sql.stats.buffer_gets_per_sec	Buffer gets per second
db.sql.stats.physical_read_requests_per_sec	Physical reads per second
db.sql.stats.physical_write_requests_per_sec	Physical writes per second
db.sql.stats.total_sharable_mem_per_sec	Total shareable memory per second (in bytes)
db.sql.stats.cpu_time_per_sec	CPU time per second (in ms)

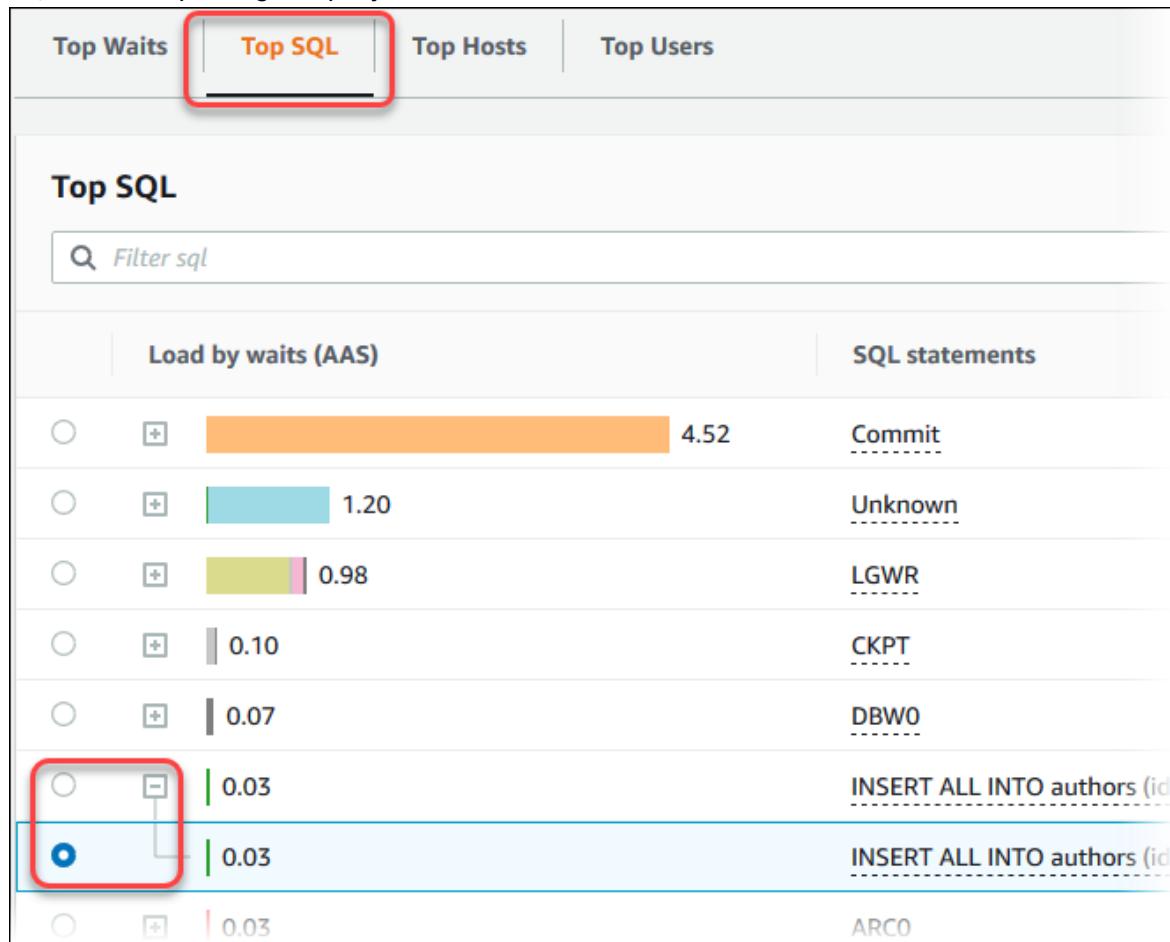
The following metrics provide per call statistics for a SQL statement.

Metric	Unit
db.sql.stats.elapsed_time_per_exec	Elapsed time per executions (in ms)

Metric	Unit
db.sql.stats.rows_processed_per_exec	Rows processed per execution
db.sql.stats.buffer_gets_per_exec	Buffer gets per execution
db.sql.stats.physical_read_requests_per_exec	Physical reads per execution
db.sql.stats.physical_write_requests_per_exec	Physical writes per execution
db.sql.stats.total_sharable_mem_per_exec	Total shareable memory per execution (in bytes)
db.sql.stats.cpu_time_per_exec	CPU time per execution (in ms)

Analyzing Oracle metrics for running SQL statements

Using the AWS Management Console, you can view the metrics for a running SQL query by choosing the **SQL** tab and expanding the query.



Choose which statistics to display by choosing the gear icon in the upper-right corner of the chart.

The following screenshot shows the preferences for Oracle DB instances.

Preferences

Page size

All resources

Wrap lines
Check to see all the text and wrap the lines

Columns

Load by waits (AAS)	<input checked="" type="checkbox"/>
SQL statements	<input checked="" type="checkbox"/>
Support ID	<input type="checkbox"/>
ID	<input type="checkbox"/>
executions/sec (executions_per_sec)	<input checked="" type="checkbox"/>
AAE (elapsed_time_per_sec)	<input type="checkbox"/>
rows processed/sec (rows_processed_per_sec)	<input type="checkbox"/>
buffer gets/sec (buffer_gets_per_sec)	<input type="checkbox"/>
physical reads/sec (physical_read_requests_per_sec)	<input type="checkbox"/>
physical writes/sec (physical_write_requests_per_sec)	<input type="checkbox"/>
total shareable memory (bytes)/sec (total_sharable_mem_per_sec)	<input type="checkbox"/>

The following screenshot shows the statistics for a SQL statement.

SQL statements	executions/sec	elapsed time (ms)
Commit	-	-
Unknown	-	-
LGWR	-	-
CKPT	-	-
DBW0	-	-
INSERT ALL INTO authors (id,name,email) VALUES (serial.nextval , 'Priya' , p@g...	-	-
INSERT ALL INTO authors (id,name,email) VALUES (serial.nextval , 'Priya' , p@g...	73.38	0.56
ARC0	-	-

Statistics for PostgreSQL

To view SQL Digest statistics, the `pg_stat_statements` library must be loaded. For PostgreSQL DB instances that are compatible with PostgreSQL 11 or later, this library is loaded by default. For PostgreSQL DB instances that are compatible with PostgreSQL 10 or earlier, you enable this library manually. To enable it manually, add `pg_stat_statements` to `shared_preload_libraries` in the DB parameter group associated with the DB instance. Then reboot your DB instance. For more information, see [Working with DB parameter groups \(p. 228\)](#).

Note

Performance Insights can only collect statistics for queries in `pg_stat_activity` that aren't truncated. By default, PostgreSQL databases truncate queries longer than 1,024 bytes. To increase the query size, change the `track_activity_query_size` parameter in the DB parameter group associated with your DB instance. When you change this parameter, a DB instance reboot is required.

The following SQL Digest statistics are available for PostgreSQL DB instances.

Metric	Unit
<code>db.sql_tokenized.stats.calls_per_sec</code>	Calls per second
<code>db.sql_tokenized.stats.rows_per_sec</code>	Rows per second
<code>db.sql_tokenized.stats.total_time_per_sec</code>	Average active executions per second (AAE)
<code>db.sql_tokenized.stats.shared_blk_hit_per_sec</code>	Block hits per second
<code>db.sql_tokenized.stats.shared_blk_read_per_sec</code>	Block reads per second
<code>db.sql_tokenized.stats.shared_blk_dirtied_per_sec</code>	Blocks dirtied per second
<code>db.sql_tokenized.stats.shared_blk_written_per_sec</code>	Block writes per second
<code>db.sql_tokenized.stats.local_blk_hit_per_sec</code>	Local block hits per second
<code>db.sql_tokenized.stats.local_blk_read_per_sec</code>	Local block reads per second
<code>db.sql_tokenized.stats.local_blk_dirtied_per_sec</code>	Local block dirty per second
<code>db.sql_tokenized.stats.local_blk_written_per_sec</code>	Local block writes per second
<code>db.sql_tokenized.stats.temp_blk_written_per_sec</code>	Temporary writes per second
<code>db.sql_tokenized.stats.temp_blk_read_per_sec</code>	Temporary reads per second
<code>db.sql_tokenized.stats.blk_read_time_per_sec</code>	Average concurrent reads per second
<code>db.sql_tokenized.stats.blk_write_time_per_sec</code>	Average concurrent writes per second

The following metrics provide per call statistics for a SQL statement.

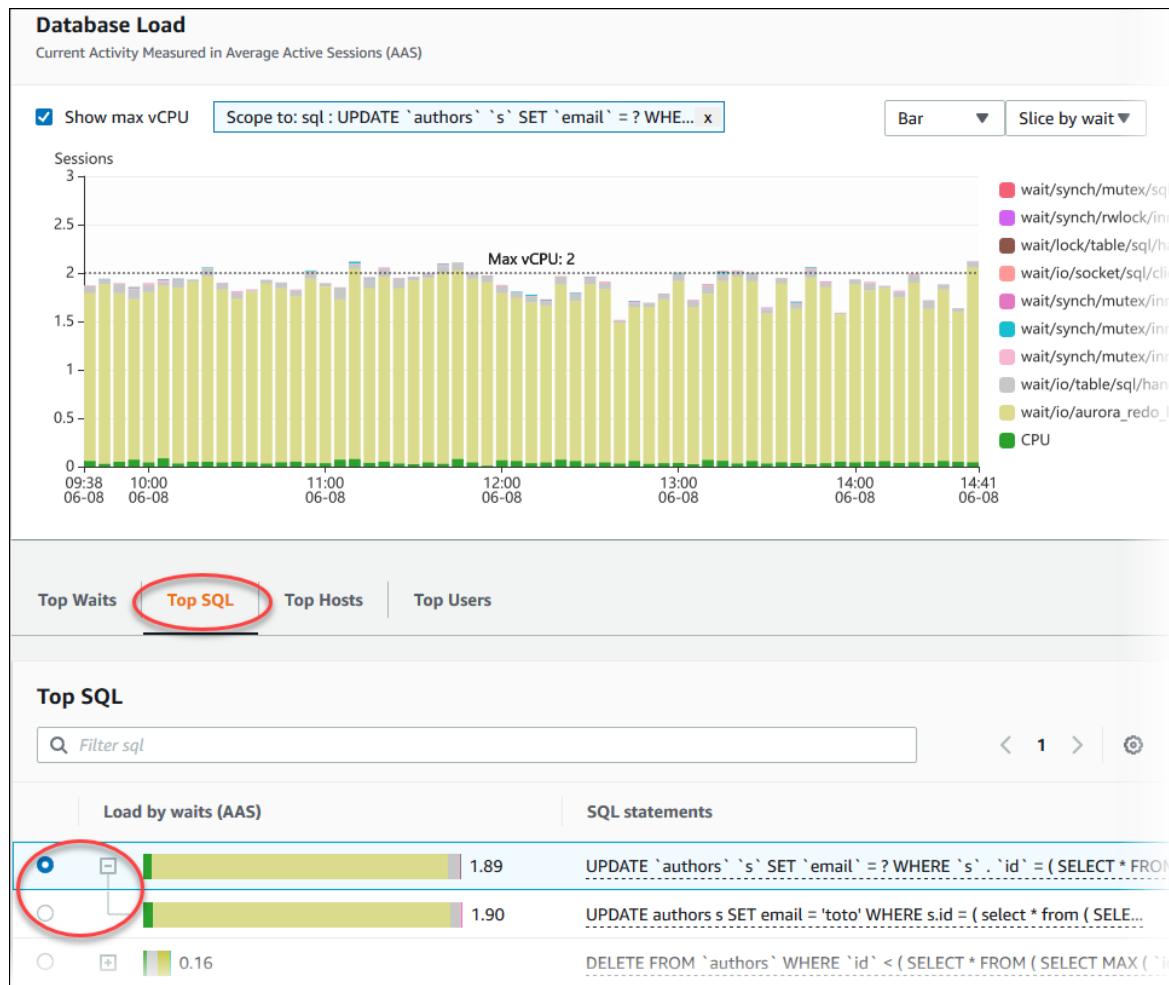
Metric	Unit
<code>db.sql_tokenized.stats.rows_per_call</code>	Rows per call
<code>db.sql_tokenized.stats.avg_latency_per_call</code>	Average latency per call (in ms)
<code>db.sql_tokenized.stats.shared_blk_hit_per_call</code>	Block hits per call

Metric	Unit
db.sql_tokenized.stats.shared_blk_reads_per_call	Block reads per call
db.sql_tokenized.stats.shared_blk_writes_per_call	Block writes per call
db.sql_tokenized.stats.shared_blk_dirtied_per_call	Blocks dirtied per call
db.sql_tokenized.stats.local_blk_hits_per_call	Local block hits per call
db.sql_tokenized.stats.local_blk_reads_per_call	Local block reads per call
db.sql_tokenized.stats.local_blk_dirtied_per_call	Local block dirty per call
db.sql_tokenized.stats.local_blk_writes_per_call	Local block writes per call
db.sql_tokenized.stats.temp_blk_writes_per_call	Temporary block writes per call
db.sql_tokenized.stats.temp_blk_reads_per_call	Temporary block reads per call
db.sql_tokenized.stats.blk_read_time_per_call	Read time per call (in ms)
db.sql_tokenized.stats.blk_write_time_per_call	Write time per call (in ms)

For more information about these metrics, see [pg_stat_statements](#) in the PostgreSQL documentation.

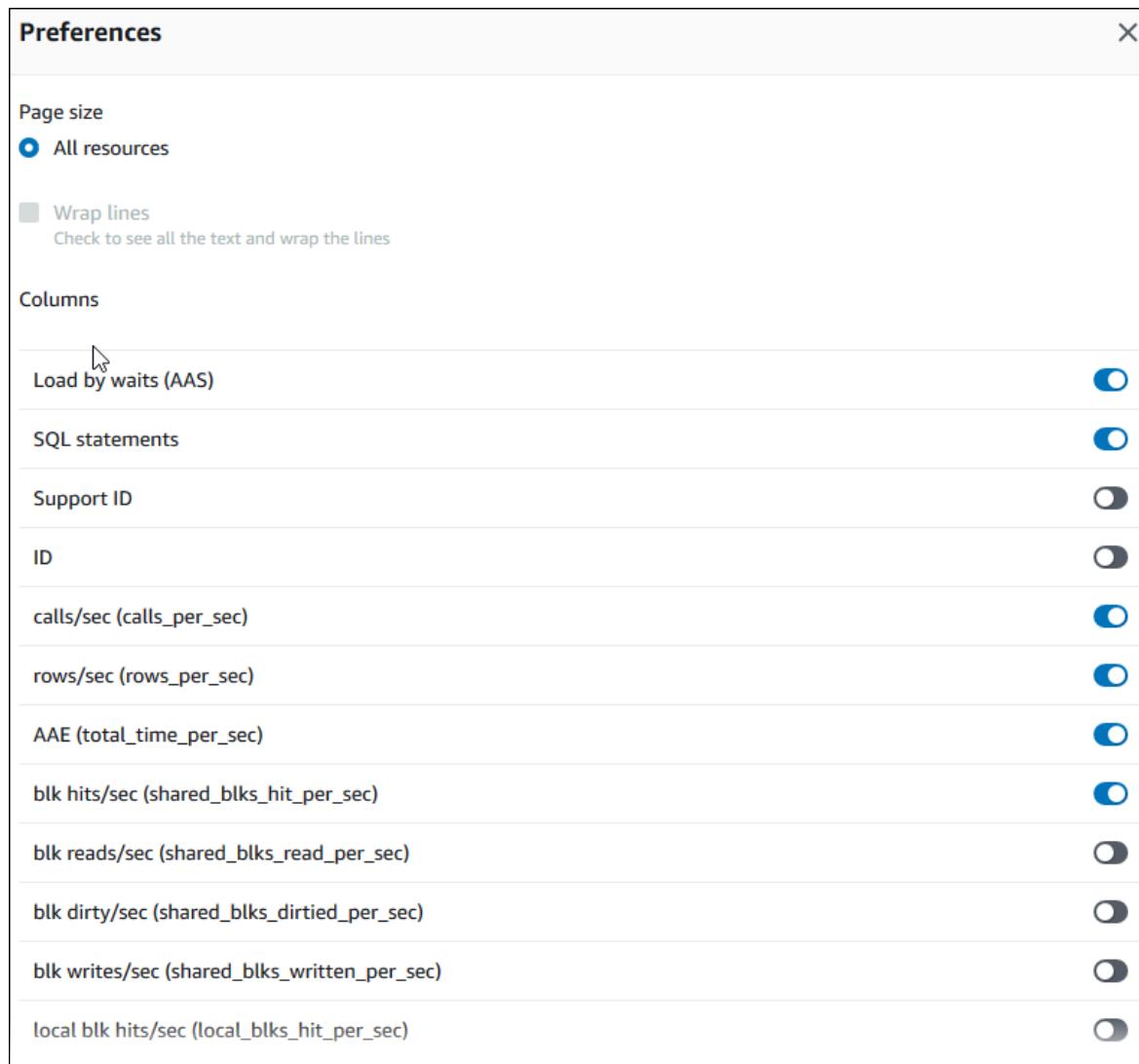
Analyzing PostgreSQL metrics for running SQL statements

Using the AWS Management Console, you can view the metrics for a running SQL query by choosing the **SQL** tab.



Choose which statistics to display by choosing the gear icon in the upper-right corner of the chart.

The following screenshot shows the preferences for PostgreSQL.



Viewing more SQL text in the Performance Insights dashboard

By default, each row in the **Top sql** table shows 500 bytes of SQL text for each SQL statement. When a SQL statement is larger than 500 bytes, you can view more of the SQL statement by opening the statement in the Performance Insights dashboard. The dashboard displays text up to the following per-engine limits:

- Amazon RDS for Microsoft SQL Server – 4,096 characters
- Amazon RDS for MySQL and MariaDB – 1,024 bytes
- Amazon RDS for Oracle – 1,000 bytes

You can copy the text that is displayed on the dashboard. If you view a child SQL statement, you can also choose **Download**.

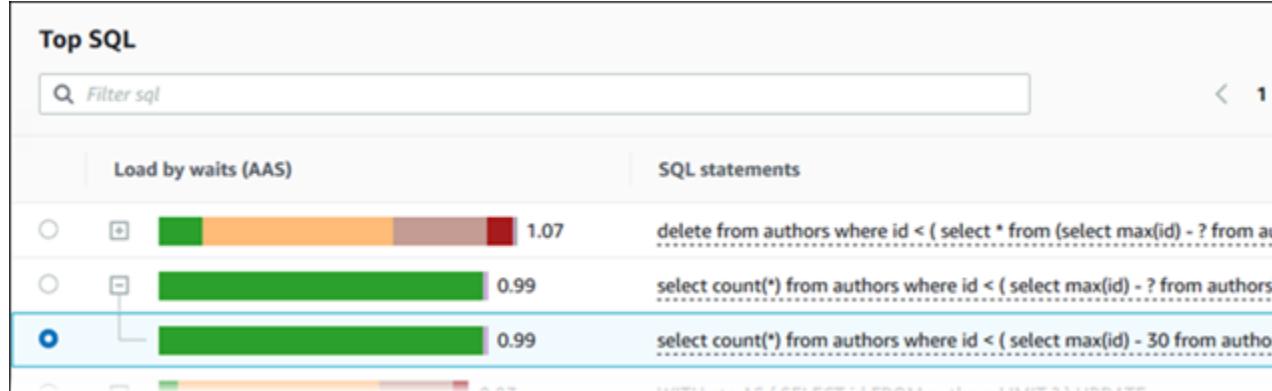
Amazon RDS for PostgreSQL handles text differently. Using the Performance Insights dashboard, you can view and download up to 500 bytes. To access more than 500 bytes, set the size limit with the DB instance parameter `track_activity_query_size`. The maximum value is 102,400 bytes. To view

or download text over 500 bytes, use the AWS Management Console, not the Performance Insights CLI or API. For more information, see [Setting the SQL text limit for Amazon RDS for PostgreSQL DB instances \(p. 442\)](#).

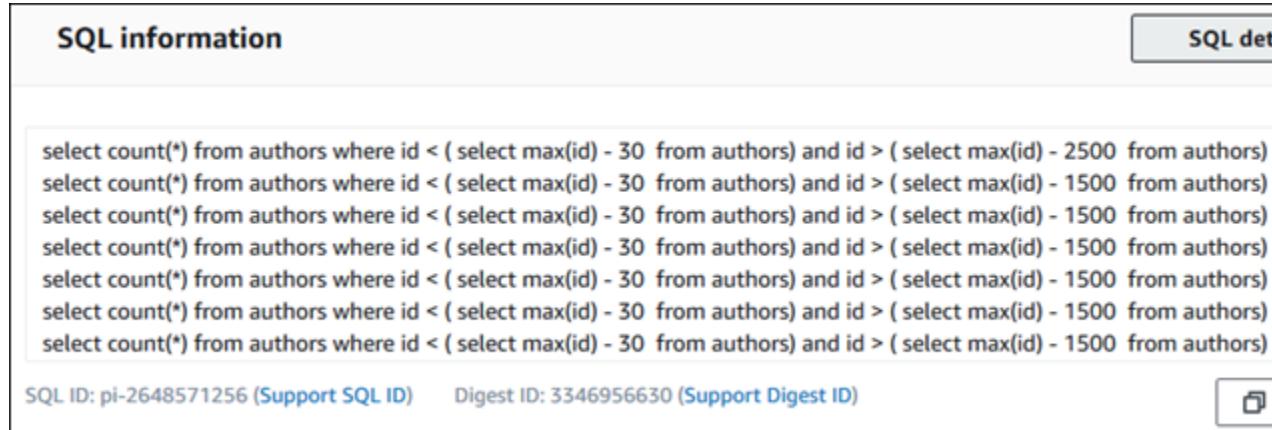
To view more SQL text in the Performance Insights dashboard

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Performance Insights**.
3. Choose a DB instance. The Performance Insights dashboard is displayed for that DB instance.

SQL statements with text larger than 500 bytes look similar to the following image.



4. Examine the SQL information section to view more of the SQL text.



The Performance Insights dashboard can display up to 4,096 bytes for each SQL statement.

5. (Optional) Choose **Copy** to copy the displayed SQL statement, or choose **Download** to download the SQL statement to view the SQL text up to the DB engine limit.

Note

To copy or download the SQL statement, disable pop-up blockers.

Setting the SQL text limit for Amazon RDS for PostgreSQL DB instances

For Amazon RDS for PostgreSQL DB instances, you can control the limit for the SQL text that can be shown on the Performance Insights dashboard.

To do so, modify the `track_activity_query_size` DB instance parameter. The default setting for the `track_activity_query_size` parameter is 1,024 bytes.

You can increase the number of bytes to increase the SQL text size visible in the Performance Insights dashboard. The limit for the parameter is 102,400 bytes. For more information about the `track_activity_query_size` DB instance parameter, see [Run-time Statistics](#) in the PostgreSQL documentation.

To modify the parameter, change the parameter setting in the parameter group that is associated with the Amazon RDS for PostgreSQL DB instance.

If the Amazon RDS for PostgreSQL DB instance is using the default parameter group, complete the following steps:

1. Create a new DB instance parameter group for the appropriate DB engine and DB engine version.
2. Set the parameter in the new parameter group.
3. Associate the new parameter group with the DB instance.

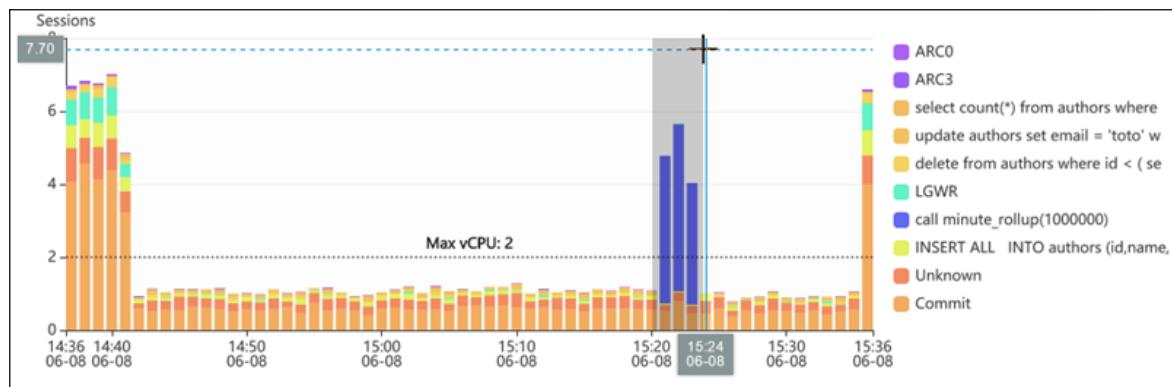
For information about setting a DB instance parameter, see [Modifying parameters in a DB parameter group \(p. 232\)](#).

Zooming In on the DB Load chart

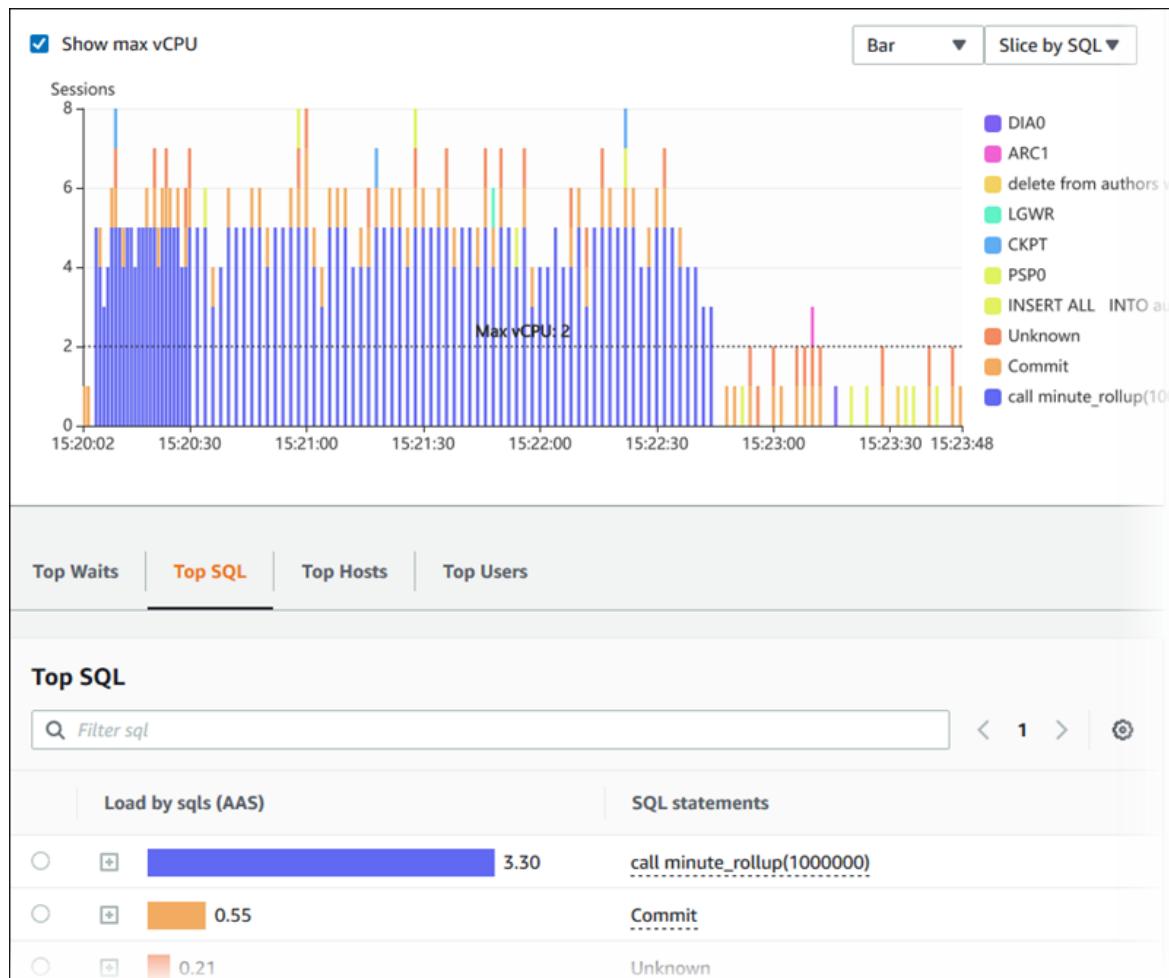
You can use other features of the Performance Insights user interface to help analyze performance data.

Click-and-Drag Zoom In

In the Performance Insights interface, you can choose a small portion of the load chart and zoom in on the detail.



To zoom in on a portion of the load chart, choose the start time and drag to the end of the time period you want. When you do this, the selected area is highlighted. When you release the mouse, the load chart zooms in on the selected AWS Region, and the **Top items** table is recalculated.



Customizing the Performance Insights dashboard

With counter metrics, you can customize the Performance Insights dashboard to include up to 10 additional graphs. These graphs that show a selection of dozens of operating system and database performance metrics. This information can be correlated with database load to help identify and analyze performance problems.

Topics

- [Performance Insights operating system counters \(p. 444\)](#)
- [Performance Insights counters for Amazon RDS for MariaDB and MySQL \(p. 447\)](#)
- [Performance Insights counters for Amazon RDS for Microsoft SQL Server \(p. 450\)](#)
- [Performance Insights counters for Amazon RDS for Oracle \(p. 451\)](#)
- [Performance Insights counters for Amazon RDS for PostgreSQL \(p. 452\)](#)

Performance Insights operating system counters

The following operating system counters are available with Performance Insights for Aurora PostgreSQL. You can find definitions for these metrics in [Viewing Enhanced Monitoring by using CloudWatch Logs \(p. 478\)](#).

Counter	Type	Metric
active	memory	os.memory.active
buffers	memory	os.memory.buffers
cached	memory	os.memory.cached
dirty	memory	os.memory.dirty
free	memory	os.memory.free
hugePagesFree	memory	os.memory.hugePagesFree
hugePagesRsvd	memory	os.memory.hugePagesRsvd
hugePagesSize	memory	os.memory.hugePagesSize
hugePagesSurp	memory	os.memory.hugePagesSurp
hugePagesTotal	memory	os.memory.hugePagesTotal
inactive	memory	os.memory.inactive
mapped	memory	os.memory.mapped
pageTables	memory	os.memory.pageTables
slab	memory	os.memory.slab
total	memory	os.memory.total
writeback	memory	os.memory.writeback
guest	cpuUtilization	os.cpuUtilization.guest
idle	cpuUtilization	os.cpuUtilization.idle
irq	cpuUtilization	os.cpuUtilization.irq
nice	cpuUtilization	os.cpuUtilization.nice
steal	cpuUtilization	os.cpuUtilization.steal
system	cpuUtilization	os.cpuUtilization.system
total	cpuUtilization	os.cpuUtilization.total
user	cpuUtilization	os.cpuUtilization.user
wait	cpuUtilization	os.cpuUtilization.wait
avgQueueLen	diskIO	os.diskIO.avgQueueLen
avgReqSz	diskIO	os.diskIO.avgReqSz
await	diskIO	os.diskIO.await
readIOsPS	diskIO	os.diskIO.readIOsPS
readKb	diskIO	os.diskIO.readKb
readKbPS	diskIO	os.diskIO.readKbPS

Counter	Type	Metric
rrqmPS	diskIO	os.diskIO.rrqmPS
tps	diskIO	os.diskIO.tps
util	diskIO	os.diskIO.util
writelOsPS	diskIO	os.diskIO.writelOsPS
writeKb	diskIO	os.diskIO.writeKb
writeKbPS	diskIO	os.diskIO.writeKbPS
wrqmPS	diskIO	os.diskIO.wrqmPS
blocked	tasks	os.tasks.blocked
running	tasks	os.tasks.running
sleeping	tasks	os.tasks.sleeping
stopped	tasks	os.tasks.stopped
total	tasks	os.tasks.total
zombie	tasks	os.tasks.zombie
one	loadAverageMinute	os.loadAverageMinute.one
fifteen	loadAverageMinute	os.loadAverageMinute.fifteen
five	loadAverageMinute	os.loadAverageMinute.five
cached	swap	os.swap.cached
free	swap	os.swap.free
in	swap	os.swap.in
out	swap	os.swap.out
total	swap	os.swap.total
maxFiles	fileSys	os.fileSys.maxFiles
usedFiles	fileSys	os.fileSys.usedFiles
usedFilePercent	fileSys	os.fileSys.usedFilePercent
usedPercent	fileSys	os.fileSys.usedPercent
used	fileSys	os.fileSys.used
total	fileSys	os.fileSys.total
rx	network	os.network.rx
tx	network	os.network.tx
numVCPU	general	os.general.numVCPU

Performance Insights counters for Amazon RDS for MariaDB and MySQL

The following database counters are available with Performance Insights for Amazon RDS for MariaDB and MySQL.

Topics

- [Native counters for RDS for MariaDB and RDS for MySQL \(p. 447\)](#)
- [Non-native counters for Amazon RDS for MariaDB and MySQL \(p. 448\)](#)

Native counters for RDS for MariaDB and RDS for MySQL

For definitions of these native metrics, see [Server Status Variables](#) in the MySQL documentation.

Counter	Type	Unit	Metric
Com_analyze	SQL	Queries per second	db.SQL.Com_analyze
Com_optimize	SQL	Queries per second	db.SQL.Com_optimize
Com_select	SQL	Queries per second	db.SQL.Com_select
Connections	SQL	The number of connection attempts per minute (successful or not) to the MySQL server	db.SQL.Connections
Innodb_rows_deleted	SQL	Rows per second	db.SQL.Innodb_rows_deleted
Innodb_rows_inserted	SQL	Rows per second	db.SQL.Innodb_rows_inserted
Innodb_rows_read	SQL	Rows per second	db.SQL.Innodb_rows_read
Innodb_rows_updated	SQL	Rows per second	db.SQL.Innodb_rows_updated
Select_full_join	SQL	Queries per second	db.SQL.Select_full_join
Select_full_range_join	SQL	Queries per second	db.SQL.Select_full_range_join
Select_range	SQL	Queries per second	db.SQL.Select_range
Select_range_check	SQL	Queries per second	db.SQL.Select_range_check
Select_scan	SQL	Queries per second	db.SQL.Select_scan
Slow_queries	SQL	Queries per second	db.SQL.Slow_queries
Sort_merge_passes	SQL	Queries per second	db.SQL.Sort_merge_passes
Sort_range	SQL	Queries per second	db.SQL.Sort_range
Sort_rows	SQL	Queries per second	db.SQL.Sort_rows
Sort_scan	SQL	Queries per second	db.SQL.Sort_scan
Questions	SQL	Queries per second	db.SQL.Questions
Innodb_row_lock_time	Locks	Milliseconds (average)	db.Locks.Innodb_row_lock_time

Counter	Type	Unit	Metric
Table_locks_immediate	Locks	Requests per second	db.Locks.Table_locks_immediate
Table_locks_waited	Locks	Requests per second	db.Locks.Table_locks_waited
Aborted_clients	Users	Connections	db.Users.Aborted_clients
Aborted_connects	Users	Connections	db.Users.Aborted_connects
Threads_created	Users	Connections	db.Users.Threads_created
Threads_running	Users	Connections	db.Users.Threads_running
Innodb_data_writes	I/O	Operations per second	db.IO.Innodb_data_writes
Innodb_dblwr_writes	I/O	Operations per second	db.IO.Innodb_dblwr_writes
Innodb_log_write_requests	I/O	Operations per second	db.IO.Innodb_log_write_requests
Innodb_log_writes	I/O	Operations per second	db.IO.Innodb_log_writes
Innodb_pages_written	I/O	Pages per second	db.IO.Innodb_pages_written
Created_tmp_disk_tables	Temp	Tables per second	db.Temp.Created_tmp_disk_tables
Created_tmp_tables	Temp	Tables per second	db.Temp.Created_tmp_tables
Innodb_buffer_pool_pages_cached	Cache	Pages	db.Cache.Innodb_buffer_pool_pages_cached
Innodb_buffer_pool_pages_total	Cache	Pages	db.Cache.Innodb_buffer_pool_pages_total
Innodb_buffer_pool_read_requests	Cache	Pages per second	db.Cache.Innodb_buffer_pool_read_requests
Innodb_buffer_pool_reads	Cache	Pages per second	db.Cache.Innodb_buffer_pool_reads
Opened_tables	Cache	Tables	db.Cache.Opened_tables
Opened_table_definitions	Cache	Tables	db.Cache.Opened_table_definitions
Qcache_hits	Cache	Queries	db.Cache.Qcache_hits

Non-native counters for Amazon RDS for MariaDB and MySQL

Non-native counter metrics are counters defined by Amazon RDS. A non-native metric can be a metric that you get with a specific query. A non-native metric also can be a derived metric, where two or more native counters are used in calculations for ratios, hit rates, or latencies.

Counter	Type	Metric	Description	Definition
innodb_buffer_pool_hits		db.Cache.innodb_buffer_pool_hits	The number of reads that InnoDB could satisfy from the buffer pool.	innodb_buffer_pool_read_requests - innodb_buffer_pool_reads
innodb_buffer_pool_hit_rate		db.Cache.innodb_buffer_pool_hit_rate	The percentage of reads that InnoDB could satisfy from the buffer pool.	100 * innodb_buffer_pool_read_requests / (innodb_buffer_pool_read_requests + innodb_buffer_pool_reads)

Counter	Type	Metric	Description	Definition
innodb_buffer_pool_usage	Code	db.Cache.innodb_buffer_pool_usage	The percentage of the InnoDB buffer pool that contains data (pages). Note When using compressed tables, this value can vary. For more information, see the information about Innodb_buffer_pool_pages_data and Innodb_buffer_pool_pages_total in Server Status Variables in the MySQL documentation.	Innodb_buffer_pool_pages_data / Innodb_buffer_pool_pages_total * 100.0
query_cache_hits	Code	db.Cache.query_cache_hits	MySQL result set cache (query cache) hit ratio.	Qcache_hits / (QCache_hits + Com_select) * 100
innodb_datafile_writes_to_disk	Code	db.IO.innodb_datafile_writes_to_disk	The number of InnoDB data file writes to disk, excluding double write and redo logging write operations.	Innodb_data_writes - Innodb_log_writes - Innodb_dblwr_writes
innodb_rows_change	Code	db.SQL.innodb_rows_change	The total number of InnoDB row operations.	db.SQL.Innodb_rows_inserted + db.SQL.Innodb_rows_deleted + db.SQL.Innodb_rows_updated
active_transactions	Transactions	db.Transaction.active_transactions	The total number of active transactions.	SELECT COUNT(1) AS active_transactions FROM INFORMATION_SCHEMA.INNODB_TRX
innodb_deadlocks	Locks	db.Locks.innodb_deadlocks	The total number of deadlocks.	SELECT COUNT AS innodb_deadlocks FROM INFORMATION_SCHEMA.INNODB_METRIC WHERE NAME='lock_deadlocks'
innodb_lock_timeouts	Locks	db.Locks.innodb_lock_timeouts	The total number of locks that timed out.	SELECT COUNT AS innodb_lock_timeouts FROM INFORMATION_SCHEMA.INNODB_METRIC WHERE NAME='lock_timeouts'

Counter	Type	Metric	Description	Definition
innodb_row_lock_waits	Locks	db.Locks.innodb	The total number of row locks that resulted in a wait.	SELECT COUNT AS innodb_row_lock_waits FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_row_lock_waits'

Performance Insights counters for Amazon RDS for Microsoft SQL Server

The following database counters are available with Performance Insights for RDS for Microsoft SQL Server.

Native counters for RDS for Microsoft SQL Server

You can find definitions for these native metrics in [Use SQL Server Objects](#) in the Microsoft SQL Server documentation.

Counter	Type	Unit	Metric
Forwarded Records	Access Methods	Records per second	db.Access Methods.Forwarded Records
Page Splits	Access Methods	Splits per second	db.Access Methods.Page Splits
Buffer cache hit ratio	Buffer Manager	Ratio	db.Buffer Manager.Buffer cache hit ratio
Page life expectancy	Buffer Manager	Expectancy in seconds	db.Buffer Manager.Page life expectancy
Page lookups	Buffer Manager	Lookups per second	db.Buffer Manager.Page lookups
Page reads	Buffer Manager	Reads per second	db.Buffer Manager.Page reads
Page writes	Buffer Manager	Writes per second	db.Buffer Manager.Page writes
Active Transactions	Databases	Transactions	db.Databases.Active Transactions (_Total)
Log Bytes Flushed	Databases	Bytes flushed per second	db.Databases.Log Bytes Flushed (_Total)
Log Flush Waits	Databases	Waits per second	db.Databases.Log Flush Waits (_Total)
Log Flushes	Databases	Flushes per second	db.Databases.Log Flushes (_Total)

Counter	Type	Unit	Metric
Write Transactions	Databases	Transactions per second	db.Databases.Write Transactions (_Total)
Processes blocked	General Statistics	Processes blocked	db.General Statistics.Processes blocked
User Connections	General Statistics	Connections	db.General Statistics.User Connections
Latch Waits	Latches	Waits per second	db.Latches.Latch Waits
Number of Deadlocks	Locks	Deadlocks per second	db.Locks.Number of Deadlocks (_Total)
Memory Grants Pending	Memory Manager	Memory grants	db.Memory Manager.Memory Grants Pending
Batch Requests	SQL Statistics	Requests per second	db.SQL Statistics.Batch Requests
SQL Compilations	SQL Statistics	Compilations per second	db.SQL Statistics.SQL Compilations
SQL Re-Compilations	SQL Statistics	Re-compilations per second	db.SQL Statistics.SQL Re-Compilations

Performance Insights counters for Amazon RDS for Oracle

The following database counters are available with Performance Insights for RDS for Oracle.

Native counters for RDS for Oracle

You can find definitions for these native metrics in [Statistics Descriptions](#) in the Oracle documentation.

Note

For the CPU used by this session counter metric, the unit has been transformed from the native centiseconds to active sessions to make the value easier to use. For example, CPU send in the DB Load chart represents the demand for CPU. The counter metric CPU used by this session represents the amount of CPU used by Oracle sessions. You can compare CPU send to the CPU used by this session counter metric. When demand for CPU is higher than CPU used, sessions are waiting for CPU time.

Counter	Type	Unit	Metric
CPU used by this session	User	Active sessions	db.User.CPU used by this session
SQL*Net roundtrips to/from client	User	Roundtrips per second	db.User.SQL*Net roundtrips to/from client
Bytes received via SQL*Net from client	User	Bytes per second	db.User.bytes received via SQL*Net from client

Counter	Type	Unit	Metric
User commits	User	Commits per second	db.User.user commits
Logons cumulative	User	Logons per second	db.User.logons cumulative
User calls	User	Calls per second	db.User.user calls
Bytes sent via SQL*Net to client	User	Bytes per second	db.User.bytes sent via SQL*Net to client
User rollbacks	User	Rollbacks per second	db.User.user rollbacks
Redo size	Redo	Bytes per second	db.Redo.redo size
Parse count (total)	SQL	Parses per second	db.SQL.parse count (total)
Parse count (hard)	SQL	Parses per second	db.SQL.parse count (hard)
Table scan rows gotten	SQL	Rows per second	db.SQL.table scan rows gotten
Sorts (memory)	SQL	Sorts per second	db.SQL.sorts (memory)
Sorts (disk)	SQL	Sorts per second	db.SQL.sorts (disk)
Sorts (rows)	SQL	Sorts per second	db.SQL.sorts (rows)
Physical read bytes	Cache	Bytes per second	db.Cache.physical read bytes
DB block gets	Cache	Blocks per second	db.Cache.db block gets
DBWR checkpoints	Cache	Checkpoints per minute	db.Cache.DBWR checkpoints
Physical reads	Cache	Reads per second	db.Cache.physical reads
Consistent gets from cache	Cache	Gets per second	db.Cache.consistent gets from cache
DB block gets from cache	Cache	Gets per second	db.Cache.db block gets from cache
Consistent gets	Cache	Gets per second	db.Cache.consistent gets

Performance Insights counters for Amazon RDS for PostgreSQL

The following database counters are available with Performance Insights for Amazon RDS for PostgreSQL.

Topics

- [Native counters for Amazon RDS for PostgreSQL \(p. 453\)](#)
- [Non-native counters for Amazon RDS for PostgreSQL \(p. 454\)](#)

Native counters for Amazon RDS for PostgreSQL

You can find definitions for these native metrics in [Viewing Statistics](#) in the PostgreSQL documentation.

Counter	Type	Unit	Metric
blks_hit	Cache	Blocks per second	db.Cache.blks_hit
buffers_alloc	Cache	Blocks per second	db.Cache.buffers_alloc
buffers_checkpoint	Checkpoint	Blocks per second	db.Checkpoint.buffers_checkpoint
checkpoint_sync_time	Checkpoint	Milliseconds per checkpoint	db.Checkpoint.checkpoint_sync_time
checkpoint_write_time	Checkpoint	Milliseconds per checkpoint	db.Checkpoint.checkpoint_write_time
checkpoints_req	Checkpoint	Checkpoints per minute	db.Checkpoint.checkpoints_req
checkpoints_timed	Checkpoint	Checkpoints per minute	db.Checkpoint.checkpoints_timed
maxwritten_clean	Checkpoint	Bgwriter clean stops per minute	db.Checkpoint.maxwritten_clean
deadlocks	Concurrency	Deadlocks per minute	db.Concurrency.deadlocks
blk_read_time	I/O	Milliseconds	db.IO.blk_read_time
blks_read	I/O	Blocks per second	db.IO.blks_read
buffers_backend	I/O	Blocks per second	db.IO.buffers_backend
buffers_backend_fsync	I/O	Blocks per second	db.IO.buffers_backend_fsync
buffers_clean	I/O	Blocks per second	db.IO.buffers_clean
tup_deleted	SQL	Tuples per second	db.SQL.tup_deleted
tup_fetched	SQL	Tuples per second	db.SQL.tup_fetched
tup_inserted	SQL	Tuples per second	db.SQL.tup_inserted
tup_returned	SQL	Tuples per second	db.SQL.tup_returned
tup_updated	SQL	Tuples per second	db.SQL.tup_updated
temp_bytes	Temp	Bytes per second	db.Temp.temp_bytes
temp_files	Temp	Files per minute	db.Temp.temp_files
active_transactions	Transactions	Transactions	db.Transactions.active_transactions
blocked_transactions	Transactions	Transactions	db.Transactions.blocked_transactions
max_used_xact_ids	Transactions	Transactions	db.Transactions.max_used_xact_ids
xact_commit	Transactions	Commits per second	db.Transactions.xact_commit
xact_rollback	Transactions	Rollbacks per second	db.Transactions.xact_rollback
numbackends	User	Connections	db.User.numbackends

Counter	Type	Unit	Metric
archived_count	Write-ahead log (WAL)	Files per minute	db.WAL.archived_count
archive_failed_count	WAL	Files per minute	db.WAL.archive_failed_count

Non-native counters for Amazon RDS for PostgreSQL

Non-native counter metrics are counters defined by Amazon RDS. A non-native metric can be a metric that you get with a specific query. A non-native metric also can be a derived metric, where two or more native counters are used in calculations for ratios, hit rates, or latencies.

Counter	Type	Metric	Description	Definition
checkpoint_sync_time	Checkpoint	db.Checkpoint.checkpoint_sync_time	The total time spent in milliseconds that has been spent in the portion of checkpoint processing where files are synchronized to disk.	checkpoint_sync_time / (checkpoints_timed + checkpoints_req)
checkpoint_write_time	Checkpoint	db.Checkpoint.checkpoint_write_time	The total time spent in milliseconds that has been spent in the portion of checkpoint processing where files are written to disk.	checkpoint_write_time / (checkpoints_timed + checkpoints_req)
read_latency	I/O	db.IO.read_latency	The time spent reading data file blocks by backends in this instance.	blk_read_time / blks_read

Retrieving data with the Performance Insights API

When Performance Insights is enabled for supported engine types, the API provides visibility into instance performance. Amazon CloudWatch Logs provides the authoritative source for vended monitoring metrics for AWS services.

Performance Insights offers a domain-specific view of database load measured as average active sessions (AAS). This metric appears to API consumers as a two-dimensional time-series dataset. The time dimension of the data provides DB load data for each time point in the queried time range. Each time point decomposes overall load in relation to the requested dimensions, such as `SQL`, `Wait-event`, `User`, or `Host`, measured at that time point.

Amazon RDS Performance Insights monitors your Amazon RDS DB instance so that you can analyze and troubleshoot database performance. One way to view Performance Insights data is in the AWS Management Console. Performance Insights also provides a public API so that you can query your own data. You can use the API to offload data into a database, add Performance Insights data to existing monitoring dashboards, or to build monitoring tools. To use the Performance Insights API, enable Performance Insights on one of your Amazon RDS DB instances. For information about enabling Performance Insights, see [Enabling and disabling Performance Insights \(p. 415\)](#).

The Performance Insights API provides the following operations.

Performance Insights Operation	AWS CLI Command	Description
DescribeDimensionKeys	<code>aws pi describe-dimension-keys</code>	Retrieves the top N dimension keys for a metric for a specific time period.
GetResourceMetrics	<code>aws pi get-resource-metrics</code>	Retrieves Performance Insights metrics for a set of data sources, over a time period. You can provide specific dimension groups and dimensions, and provide aggregation and filtering criteria for each group.

For more information about the Performance Insights API, see the [Amazon RDS Performance Insights API Reference](#).

AWS CLI for Performance Insights

You can view Performance Insights data using the AWS CLI. You can view help for the AWS CLI commands for Performance Insights by entering the following on the command line.

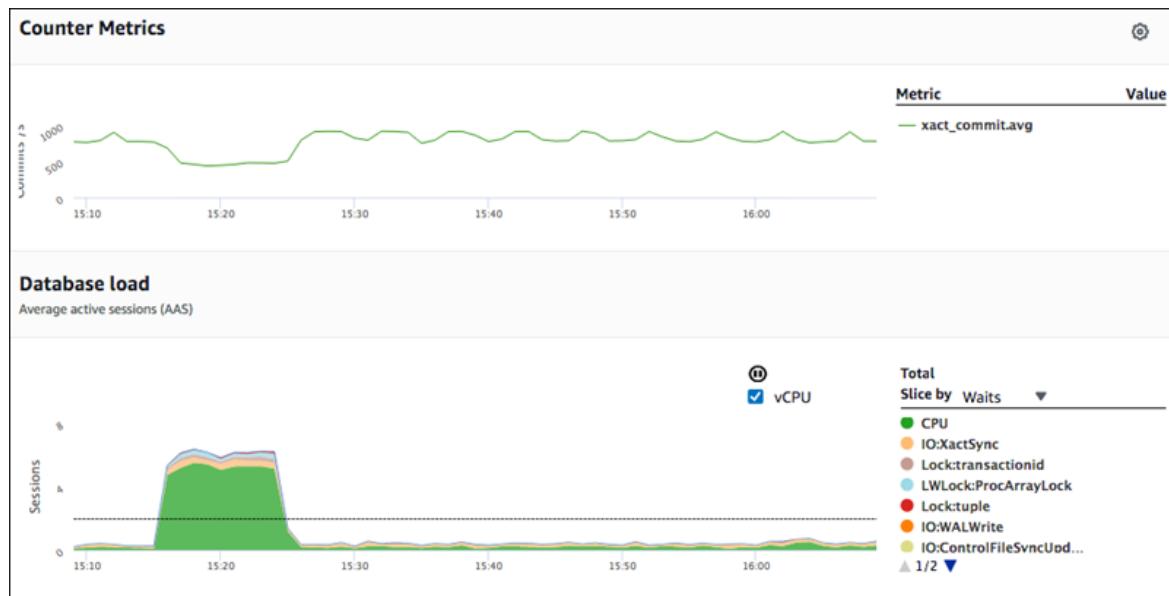
```
aws pi help
```

If you don't have the AWS CLI installed, see [Installing the AWS Command Line Interface](#) in the *AWS CLI User Guide* for information about installing it.

Retrieving time-series metrics

The GetResourceMetrics operation retrieves one or more time-series metrics from the Performance Insights data. GetResourceMetrics requires a metric and time period, and returns a response with a list of data points.

For example, the AWS Management Console uses GetResourceMetrics in two places in the Performance Insights dashboard. GetResourceMetrics is used to populate the **Counter Metrics** chart and in the **Database Load** chart, as seen in the following image.



All the metrics returned by `GetResourceMetrics` are standard time-series metrics with one exception. The exception is `db.load`, which is the core metric in Performance Insights. This metric is displayed in the **Database Load** chart. The `db.load` metric is different from the other time-series metrics because you can break it into subcomponents called dimensions. In the previous image, `db.load` is broken down and grouped by the waits states that make up the `db.load`.

Note

`GetResourceMetrics` can also return the `db.sampleload` metric, but the `db.load` metric is appropriate in most cases.

For information about the counter metrics returned by `GetResourceMetrics`, see [Customizing the Performance Insights dashboard \(p. 444\)](#).

The following calculations are supported for the metrics:

- Average – The average value for the metric over a period of time. Append `.avg` to the metric name.
- Minimum – The minimum value for the metric over a period of time. Append `.min` to the metric name.
- Maximum – The maximum value for the metric over a period of time. Append `.max` to the metric name.
- Sum – The sum of the metric values over a period of time. Append `.sum` to the metric name.
- Sample count – The number of times the metric was collected over a period of time. Append `.sample_count` to the metric name.

For example, assume that a metric is collected for 300 seconds (5 minutes), and that the metric is collected one time each minute. The values for each minute are 1, 2, 3, 4, and 5. In this case, the following calculations are returned:

- Average – 3
- Minimum – 1
- Maximum – 5
- Sum – 15
- Sample count – 5

For information about using the `get-resource-metrics` AWS CLI command, see [get-resource-metrics](#).

For the `--metric-queries` option, specify one or more queries that you want to get results for. Each query consists of a mandatory `Metric` and optional `GroupBy` and `Filter` parameters. The following is an example of a `--metric-queries` option specification.

```
{  
    "Metric": "string",  
    "GroupBy": {  
        "Group": "string",  
        "Dimensions": ["string", ...],  
        "Limit": integer  
    },  
    "Filter": {"string": "string"  
        ...}  
}
```

AWS CLI examples for Performance Insights

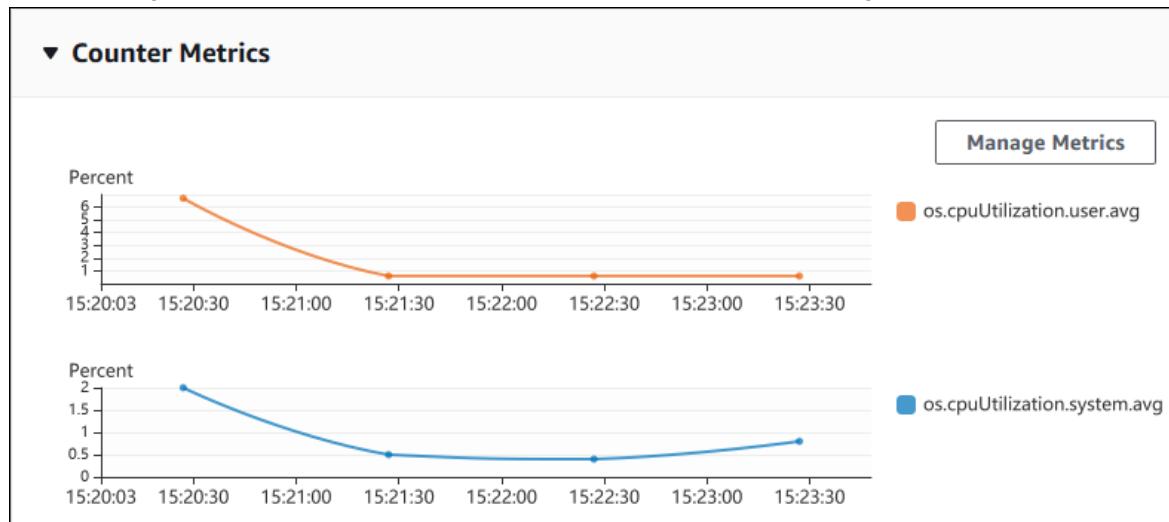
The following are several examples that use the AWS CLI for Performance Insights.

Topics

- [Retrieving counter metrics \(p. 457\)](#)
- [Retrieving the DB load average for top wait events \(p. 460\)](#)
- [Retrieving the DB load average for top SQL \(p. 462\)](#)
- [Retrieving the DB Load Average Filtered by SQL \(p. 464\)](#)

Retrieving counter metrics

The following screenshot shows two counter metrics charts in the AWS Management Console.



The following example shows how to gather the same data that the AWS Management Console uses to generate the two counter metric charts.

For Linux, macOS, or Unix:

```
aws pi get-resource-metrics \
--service-type RDS \
--identifier db-ID \
--start-time 2018-10-30T00:00:00Z \
--end-time 2018-10-30T01:00:00Z \
--period-in-seconds 60 \
--metric-queries '[{"Metric": "os.cpuUtilization.user.avg" },
 {"Metric": "os.cpuUtilization.idle.avg"}]'
```

For Windows:

```
aws pi get-resource-metrics ^
--service-type RDS ^
--identifier db-ID ^
--start-time 2018-10-30T00:00:00Z ^
--end-time 2018-10-30T01:00:00Z ^
--period-in-seconds 60 ^
--metric-queries '[{"Metric": "os.cpuUtilization.user.avg" },
 {"Metric": "os.cpuUtilization.idle.avg"}]'
```

You can also make a command easier to read by specifying a file for the --metrics-query option. The following example uses a file called query.json for the option. The file has the following contents.

```
[  
 {  
     "Metric": "os.cpuUtilization.user.avg"  
 },  
 {  
     "Metric": "os.cpuUtilization.idle.avg"  
 }]
```

Run the following command to use the file.

For Linux, macOS, or Unix:

```
aws pi get-resource-metrics \
--service-type RDS \
--identifier db-ID \
--start-time 2018-10-30T00:00:00Z \
--end-time 2018-10-30T01:00:00Z \
--period-in-seconds 60 \
--metric-queries file://query.json
```

For Windows:

```
aws pi get-resource-metrics ^
--service-type RDS ^
--identifier db-ID ^
--start-time 2018-10-30T00:00:00Z ^
--end-time 2018-10-30T01:00:00Z ^
```

```
--period-in-seconds 60 ^
--metric-queries file://query.json
```

The preceding example specifies the following values for the options:

- `--service-type` – RDS for Amazon RDS
- `--identifier` – The resource ID for the DB instance
- `--start-time` and `--end-time` – The ISO 8601 DateTime values for the period to query, with multiple supported formats

It queries for a one-hour time range:

- `--period-in-seconds` – 60 for a per-minute query
- `--metric-queries` – An array of two queries, each just for one metric.

The metric name uses dots to classify the metric in a useful category, with the final element being a function. In the example, the function is `avg` for each query. As with Amazon CloudWatch, the supported functions are `min`, `max`, `total`, and `avg`.

The response looks similar to the following.

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1540857600.0,
    "AlignedEndTime": 1540861200.0,
    "MetricList": [
        { //A list of key/datapoints
            "Key": {
                "Metric": "os.cpuUtilization.user.avg" //Metric1
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of items
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 4.0
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 4.0
                },
                {
                    "Timestamp": 1540857780.0, //Minute 3
                    "Value": 10.0
                }
                //... 60 datapoints for the os.cpuUtilization.user.avg metric
            ]
        },
        {
            "Key": {
                "Metric": "os.cpuUtilization.idle.avg" //Metric2
            },
            "DataPoints": [
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 12.0
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 10.0
                }
            ]
        }
    ]
}
```

```

        "Value": 13.5
    },
    //... 60 datapoints for the os.cpuUtilization.idle.avg metric
]
}
] //end of MetricList
} //end of response

```

The response has an `Identifier`, `AlignedStartTime`, and `AlignedEndTime`. Because the `--period-in-seconds` value was 60, the start and end times have been aligned to the minute. If the `--period-in-seconds` was 3600, the start and end times would have been aligned to the hour.

The `MetricList` in the response has a number of entries, each with a `Key` and a `DataPoints` entry. Each `DataPoint` has a `Timestamp` and a `Value`. Each `DataPoints` list has 60 data points because the queries are for per-minute data over an hour, with `Timestamp1/Minute1`, `Timestamp2/Minute2`, and so on, up to `Timestamp60/Minute60`.

Because the query is for two different counter metrics, there are two elements in the response `MetricList`.

Retrieving the DB load average for top wait events

The following example is the same query that the AWS Management Console uses to generate a stacked area line graph. This example retrieves the `db.load.avg` for the last hour with load divided according to the top seven wait events. The command is the same as the command in [Retrieving counter metrics \(p. 457\)](#). However, the `query.json` file has the following contents.

```
[
{
    "Metric": "db.load.avg",
    "GroupBy": { "Group": "db.wait_event", "Limit": 7 }
}
]
```

Run the following command.

For Linux, macOS, or Unix:

```
aws pi get-resource-metrics \
--service-type RDS \
--identifier db-ID \
--start-time 2018-10-30T00:00:00Z \
--end-time 2018-10-30T01:00:00Z \
--period-in-seconds 60 \
--metric-queries file://query.json
```

For Windows:

```
aws pi get-resource-metrics ^
--service-type RDS ^
--identifier db-ID ^
--start-time 2018-10-30T00:00:00Z ^
--end-time 2018-10-30T01:00:00Z ^
--period-in-seconds 60 ^
--metric-queries file://query.json
```

The example specifies the metric of `db.load.avg` and a `GroupBy` of the top seven wait events. For details about valid values for this example, see [DimensionGroup](#) in the *Performance Insights API Reference*.

The response looks similar to the following.

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1540857600.0,
    "AlignedEndTime": 1540861200.0,
    "MetricList": [
        { //A list of key/datapoints
            "Key": {
                //A Metric with no dimensions. This is the total db.load.avg
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of items
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 0.5166666666666667
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 0.3833333333333336
                },
                {
                    "Timestamp": 1540857780.0, //Minute 3
                    "Value": 0.2666666666666666
                }
                //... 60 datapoints for the total db.load.avg key
            ]
        },
        {
            "Key": {
                //Another key. This is db.load.avg broken down by CPU
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.name": "CPU",
                    "db.wait_event.type": "CPU"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 0.35
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 0.15
                }
                //... 60 datapoints for the CPU key
            ]
        },
        //... In total we have 8 key/datapoints entries, 1) total, 2-8) Top Wait Events
    ] //end of MetricList
} //end of response
}
```

In this response, there are eight entries in the `MetricList`. There is one entry for the total `db.load.avg`, and seven entries each for the `db.load.avg` divided according to one of the top seven wait events. Unlike in the first example, because there was a grouping dimension, there must be one key for each grouping of the metric. There can't be only one key for each metric, as in the basic counter metric use case.

Retrieving the DB load average for top SQL

The following example groups db.wait_events by the top 10 SQL statements. There are two different groups for SQL statements:

- db.sql – The full SQL statement, such as `select * from customers where customer_id = 123`
- db.sql_tokenized – The tokenized SQL statement, such as `select * from customers where customer_id = ?`

When analyzing database performance, it can be useful to consider SQL statements that only differ by their parameters as one logic item. So, you can use db.sql_tokenized when querying. However, especially when you are interested in explain plans, sometimes it's more useful to examine full SQL statements with parameters, and query grouping by db.sql. There is a parent-child relationship between tokenized and full SQL, with multiple full SQL (children) grouped under the same tokenized SQL (parent).

The command in this example is the similar to the command in [Retrieving the DB load average for top wait events \(p. 460\)](#). However, the query.json file has the following contents.

```
[  
  {  
    "Metric": "db.load.avg",  
    "GroupBy": { "Group": "db.sql_tokenized", "Limit": 10 }  
  }  
]
```

The following example uses db.sql_tokenized.

For Linux, macOS, or Unix:

```
aws pi get-resource-metrics \  
  --service-type RDS \  
  --identifier db-ID \  
  --start-time 2018-10-29T00:00:00Z \  
  --end-time 2018-10-30T00:00:00Z \  
  --period-in-seconds 3600 \  
  --metric-queries file://query.json
```

For Windows:

```
aws pi get-resource-metrics ^  
  --service-type RDS ^  
  --identifier db-ID ^  
  --start-time 2018-10-29T00:00:00Z ^  
  --end-time 2018-10-30T00:00:00Z ^  
  --period-in-seconds 3600 ^  
  --metric-queries file://query.json
```

This example queries over 24 hours, with a one hour period-in-seconds.

The example specifies the metric of db.load.avg and a GroupBy of the top seven wait events. For details about valid values for this example, see [DimensionGroup](#) in the *Performance Insights API Reference*.

The response looks similar to the following.

```
{
    "AlignedStartTime": 1540771200.0,
    "AlignedEndTime": 1540857600.0,
    "Identifier": "db-XXX",

    "MetricList": [ //11 entries in the MetricList
        {
            "Key": { //First key is total
                "Metric": "db.load.avg"
            }
            "DataPoints": [ //Each DataPoints list has 24 per-hour Timestamps and a value
                {
                    "Value": 1.6964980544747081,
                    "Timestamp": 1540774800.0
                },
                //... 24 datapoints
            ]
        },
        {
            "Key": { //Next key is the top tokenized SQL
                "Dimensions": {
                    "db.sql_tokenized.statement": "INSERT INTO authors (id,name,email)
VALUES\n( nextval('') ,?,?)",
                    "db.sql_tokenized.db_id": "pi-2372568224",
                    "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE"
                },
                "Metric": "db.load.avg"
            },
            "DataPoints": [ //... 24 datapoints
            ]
        },
        // In total 11 entries, 10 Keys of top tokenized SQL, 1 total key
    ] //End of MetricList
} //End of response
}
```

This response has 11 entries in the MetricList (1 total, 10 top tokenized SQL), with each entry having 24 per-hour DataPoints.

For tokenized SQL, there are three entries in each dimensions list:

- db.sql_tokenized.statement – The tokenized SQL statement.
- db.sql_tokenized.db_id – Either the native database ID used to refer to the SQL, or a synthetic ID that Performance Insights generates for you if the native database ID isn't available. This example returns the pi-2372568224 synthetic ID.
- db.sql_tokenized.id – The ID of the query inside Performance Insights.

In the AWS Management Console, this ID is called the Support ID. It's named this because the ID is data that AWS Support can examine to help you troubleshoot an issue with your database. AWS takes the security and privacy of your data extremely seriously, and almost all data is stored encrypted with your AWS KMS customer master key (CMK). Therefore, nobody inside AWS can look at this data. In the example preceding, both the tokenized.statement and the tokenized.db_id are stored encrypted. If you have an issue with your database, AWS Support can help you by referencing the Support ID.

When querying, it might be convenient to specify a Group in GroupBy. However, for finer-grained control over the data that's returned, specify the list of dimensions. For example, if all that is needed is the db.sql_tokenized.statement, then a Dimensions attribute can be added to the query.json file.

```
[
{
```

```

        "Metric": "db.load.avg",
        "GroupBy": {
            "Group": "db.sql_tokenized",
            "Dimensions": ["db.sql_tokenized.statement"],
            "Limit": 10
        }
    }
]

```

Retrieving the DB Load Average Filtered by SQL



The preceding image shows that a particular query is selected, and the top average active sessions stacked area line graph is scoped to that query. Although the query is still for the top seven overall wait events, the value of the response is filtered. The filter causes it to take into account only sessions that are a match for the particular filter.

The corresponding API query in this example is similar to the command in [Retrieving the DB load average for top SQL \(p. 462\)](#). However, the query.json file has the following contents.

```

[
{
    "Metric": "db.load.avg",
    "GroupBy": { "Group": "db.wait_event", "Limit": 5 },
    "Filter": { "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }
}
]

```

For Linux, macOS, or Unix:

```

aws pi get-resource-metrics \
--service-type RDS \
--identifier db-ID \
--start-time 2018-10-30T00:00:00Z \
--end-time 2018-10-30T01:00:00Z \
--period-in-seconds 60 \
--metric-queries file://query.json

```

For Windows:

```
aws pi get-resource-metrics ^
--service-type RDS ^
--identifier db-ID ^
--start-time 2018-10-30T00:00:00Z ^
--end-time 2018-10-30T01:00:00Z ^
--period-in-seconds 60 ^
--metric-queries file://query.json
```

The response looks similar to the following.

```
{
  "Identifier": "db-XXX",
  "AlignedStartTime": 1556215200.0,
  "MetricList": [
    {
      "Key": {
        "Metric": "db.load.avg"
      },
      "DataPoints": [
        {
          "Timestamp": 1556218800.0,
          "Value": 1.4878117913832196
        },
        {
          "Timestamp": 1556222400.0,
          "Value": 1.192823803967328
        }
      ]
    },
    {
      "Key": {
        "Metric": "db.load.avg",
        "Dimensions": {
          "db.wait_event.type": "io",
          "db.wait_event.name": "wait/io/aurora_redo_log_flush"
        }
      },
      "DataPoints": [
        {
          "Timestamp": 1556218800.0,
          "Value": 1.1360544217687074
        },
        {
          "Timestamp": 1556222400.0,
          "Value": 1.058051341890315
        }
      ]
    },
    {
      "Key": {
        "Metric": "db.load.avg",
        "Dimensions": {
          "db.wait_event.type": "io",
          "db.wait_event.name": "wait/io/table/sql/handler"
        }
      },
      "DataPoints": [
        {
          "Timestamp": 1556218800.0,
          "Value": 0.16241496598639457
        }
      ]
    }
  ]
}
```

```

        "Timestamp": 1556222400.0,
        "Value": 0.05163360560093349
    }
]
},
{
    "Key": {
        "Metric": "db.load.avg",
        "Dimensions": {
            "db.wait_event.type": "synch",
            "db.wait_event.name": "wait/synch/mutex/innodb/
aurora_lock_thread_slot_futex"
        }
    },
    "DataPoints": [
        {
            "Timestamp": 1556218800.0,
            "Value": 0.11479591836734694
        },
        {
            "Timestamp": 1556222400.0,
            "Value": 0.013127187864644107
        }
    ]
},
{
    "Key": {
        "Metric": "db.load.avg",
        "Dimensions": {
            "db.wait_event.type": "CPU",
            "db.wait_event.name": "CPU"
        }
    },
    "DataPoints": [
        {
            "Timestamp": 1556218800.0,
            "Value": 0.05215419501133787
        },
        {
            "Timestamp": 1556222400.0,
            "Value": 0.05805134189031505
        }
    ]
},
{
    "Key": {
        "Metric": "db.load.avg",
        "Dimensions": {
            "db.wait_event.type": "synch",
            "db.wait_event.name": "wait/synch/mutex/innodb/lock_wait_mutex"
        }
    },
    "DataPoints": [
        {
            "Timestamp": 1556218800.0,
            "Value": 0.017573696145124718
        },
        {
            "Timestamp": 1556222400.0,
            "Value": 0.002333722287047841
        }
    ]
},
],
"AlignedEndTime": 1556222400.0
} //end of response

```

In this response, all values are filtered according to the contribution of tokenized SQL AKIAIOSFODNN7EXAMPLE specified in the query.json file. The keys also might follow a different order than a query without a filter, because it's the top five wait events that affected the filtered SQL.

Performance Insights metrics published to Amazon CloudWatch

Performance Insights automatically publishes metrics to Amazon CloudWatch. The same data can be queried from Performance Insights, but having the metrics in CloudWatch makes it easy to add CloudWatch alarms. It also makes it easy to add the metrics to existing CloudWatch Dashboards.

Metric	Description
DBLoad	The number of active sessions for the DB engine. Typically, you want the data for the average number of active sessions. In Performance Insights, this data is queried as db.load.avg.
DBLoadCPU	The number of active sessions where the wait event type is CPU. In Performance Insights, this data is queried as db.load.avg, filtered by the wait event type CPU.
DBLoadNonCPU	The number of active sessions where the wait event type is not CPU.

Note

These metrics are published to CloudWatch only if there is load on the DB instance.

You can examine these metrics using the CloudWatch console, the AWS CLI, or the CloudWatch API.

For example, you can get the statistics for the DBLoad metric by running the [get-metric-statistics](#) command.

```
aws cloudwatch get-metric-statistics \
--region us-west-2 \
--namespace AWS/RDS \
--metric-name DBLoad \
--period 60 \
--statistics Average \
--start-time 1532035185 \
--end-time 1532036185 \
--dimensions Name=DBInstanceIdentifier,Value=db-loadtest-0
```

This example generates output similar to the following.

```
{
  "Datapoints": [
    {
      "Timestamp": "2018-07-19T21:30:00Z",
      "Unit": "None",
      "Average": 2.1
    },
    ...
  ]
}
```

```
{  
  "Timestamp": "2018-07-19T21:34:00Z",  
  "Unit": "None",  
  "Average": 1.7  
},  
{  
  "Timestamp": "2018-07-19T21:35:00Z",  
  "Unit": "None",  
  "Average": 2.8  
},  
{  
  "Timestamp": "2018-07-19T21:31:00Z",  
  "Unit": "None",  
  "Average": 1.5  
},  
{  
  "Timestamp": "2018-07-19T21:32:00Z",  
  "Unit": "None",  
  "Average": 1.8  
},  
{  
  "Timestamp": "2018-07-19T21:29:00Z",  
  "Unit": "None",  
  "Average": 3.0  
},  
{  
  "Timestamp": "2018-07-19T21:33:00Z",  
  "Unit": "None",  
  "Average": 2.4  
}  
],  
"Label": "DBLoad"  
}
```

For more information about CloudWatch, see [What is Amazon CloudWatch?](#) in the *Amazon CloudWatch User Guide*.

Logging Performance Insights calls by using AWS CloudTrail

Performance Insights runs with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Performance Insights. CloudTrail captures all API calls for Performance Insights as events. This capture includes calls from the Amazon RDS console and from code calls to the Performance Insights API operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Performance Insights. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the data collected by CloudTrail, you can determine certain information. This information includes the request that was made to Performance Insights, the IP address the request was made from, who made the request, and when it was made. It also includes additional details.

To learn more about CloudTrail, see the [AWS CloudTrail User Guide](#).

Working with Performance Insights information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Performance Insights, that activity is recorded in a CloudTrail event along with other AWS service events in the CloudTrail console in **Event history**. You can view, search, and download recent events in your AWS

account. For more information, see [Viewing Events with CloudTrail Event History](#) in *AWS CloudTrail User Guide*.

For an ongoing record of events in your AWS account, including events for Performance Insights, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all AWS Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following topics in *AWS CloudTrail User Guide*:

- [Overview for Creating a Trail](#)
- [CloudTrail Supported Services and Integrations](#)
- [Configuring Amazon SNS Notifications for CloudTrail](#)
- [Receiving CloudTrail Log Files from Multiple Regions](#) and [Receiving CloudTrail Log Files from Multiple Accounts](#)

All Performance Insights operations are logged by CloudTrail and are documented in the [Performance Insights API Reference](#). For example, calls to the `DescribeDimensionKeys` and `GetResourceMetrics` operations generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the [CloudTrail userIdentity Element](#).

Performance Insights log file entries

A *trail* is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An *event* represents a single request from any source. Each event includes information about the requested operation, the date and time of the operation, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `GetResourceMetrics` operation.

```
{  
    "eventVersion": "1.05",  
    "userIdentity": {  
        "type": "IAMUser",  
        "principalId": "AKIAIOSFODNN7EXAMPLE",  
        "arn": "arn:aws:iam::123456789012:user/johndoe",  
        "accountId": "123456789012",  
        "accessKeyId": "AKIAI44QH8DHBEEXAMPLE",  
        "userName": "johndoe"  
    },  
    "eventTime": "2019-12-18T19:28:46Z",  
    "eventSource": "pi.amazonaws.com",  
    "eventName": "GetResourceMetrics",  
    "awsRegion": "us-east-1",  
    "sourceIPAddress": "72.21.198.67",  
    "userAgent": "aws-cli/1.16.240 Python/3.7.4 Darwin/18.7.0 botocore/1.12.230",  
    "requestParameters": {
```

```
"identifier": "db-YTDU5J5V66X7CXSCVDFD2V3SZM",
"metricQueries": [
    {
        "metric": "os.cpuUtilization.user.avg"
    },
    {
        "metric": "os.cpuUtilization.idle.avg"
    }
],
"startTime": "Dec 18, 2019 5:28:46 PM",
"periodInSeconds": 60,
"endTime": "Dec 18, 2019 7:28:46 PM",
"serviceType": "RDS"
},
"responseElements": null,
"requestID": "9fffbe15c-96b5-4fe6-bed9-9fccff1a0525",
"eventID": "08908de0-2431-4e2e-ba7b-f5424f908433",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

Using Enhanced Monitoring

Amazon RDS provides metrics in real time for the operating system (OS) that your DB instance runs on. You can view the metrics for your DB instance using the console. Also, you can consume the Enhanced Monitoring JSON output from Amazon CloudWatch Logs in a monitoring system of your choice.

By default, Enhanced Monitoring metrics are stored for 30 days in the CloudWatch Logs, which are different from typical CloudWatch metrics. To modify the amount of time the metrics are stored in the CloudWatch Logs, change the retention for the `RDSOSMetrics` log group in the CloudWatch console. For more information, see [Change log data retention in CloudWatch logs](#) in the *Amazon CloudWatch Logs User Guide*.

Because Enhanced Monitoring metrics are stored in the CloudWatch logs instead of in Cloudwatch metrics, the cost of Enhanced Monitoring depends on several factors:

- You are only charged for Enhanced Monitoring that exceeds the free tier provided by Amazon CloudWatch Logs.

For more information about pricing, see [Amazon CloudWatch pricing](#).
- A smaller monitoring interval results in more frequent reporting of OS metrics and increases your monitoring cost.
- Usage costs for Enhanced Monitoring are applied for each DB instance that Enhanced Monitoring is enabled for. Monitoring a large number of DB instances is more expensive than monitoring only a few.
- DB instances that support a more compute-intensive workload have more OS process activity to report and higher costs for Enhanced Monitoring.

Enhanced Monitoring availability

Enhanced Monitoring is available for the following database engines:

- MariaDB
- Microsoft SQL Server
- MySQL version 5.5 or later
- Oracle
- PostgreSQL

Enhanced Monitoring is available for all DB instance classes except for the db.m1.small instance class.

Differences between CloudWatch and Enhanced Monitoring metrics

CloudWatch gathers metrics about CPU utilization from the hypervisor for a DB instance, and Enhanced Monitoring gathers its metrics from an agent on the instance. As a result, you might find differences between the measurements, because the hypervisor layer performs a small amount of work. The differences can be greater if your DB instances use smaller instance classes, because then there are likely more virtual machines (VMs) that are managed by the hypervisor layer on a single physical instance. Enhanced Monitoring metrics are useful when you want to see how different processes or threads on a DB instance use the CPU.

Setting up and enabling Enhanced Monitoring

To use Enhanced Monitoring, you must create an IAM role, and then enable Enhanced Monitoring.

Creating an IAM role for Enhanced Monitoring

Enhanced Monitoring requires permission to act on your behalf to send OS metric information to CloudWatch Logs. You grant Enhanced Monitoring permissions using an AWS Identity and Access Management (IAM) role.

Creating the IAM role when you enable Enhanced Monitoring

When you enable Enhanced Monitoring in the RDS console, Amazon RDS can create the required IAM role for you. The role is named `rds-monitoring-role`. RDS uses this role for the specified DB instance or read replica.

To create the IAM role when enabling Enhanced Monitoring

1. Follow the steps in [Enabling and disabling Enhanced Monitoring \(p. 472\)](#).
2. Set **Monitoring Role** to **Default** in the step where you choose a role.

Creating the IAM role before you enable Enhanced Monitoring

You can create the required role before you enable Enhanced Monitoring. When you enable Enhanced Monitoring, specify your new role's name. You must create this required role if you enable Enhanced Monitoring using the AWS CLI or the RDS API.

The user that enables Enhanced Monitoring must be granted the `PassRole` permission. For more information, see Example 2 in [Granting a user permissions to pass a role to an AWS service](#) in the *IAM User Guide*.

To create an IAM role for Amazon RDS enhanced monitoring

1. Open the [IAM console](#) at <https://console.aws.amazon.com>.
2. In the navigation pane, choose **Roles**.
3. Choose **Create role**.
4. Choose the **AWS service** tab, and then choose **RDS** from the list of services.
5. Choose **RDS - Enhanced Monitoring**, and then choose **Next: Permissions**.
6. Ensure that the **Attached permissions policy** page shows `AmazonRDSEnhancedMonitoringRole`, and then choose **Next: Tags**.
7. On the **Add tags** page, choose **Next: Review**.
8. For **Role Name**, enter a name for your role. For example, enter `emaccess`.

The trusted entity for your role is the AWS service `monitoring.rds.amazonaws.com`.

9. Choose **Create role**.

Enabling and disabling Enhanced Monitoring

You can enable and disable Enhanced Monitoring using the AWS Management Console, AWS CLI, or RDS API.

Console

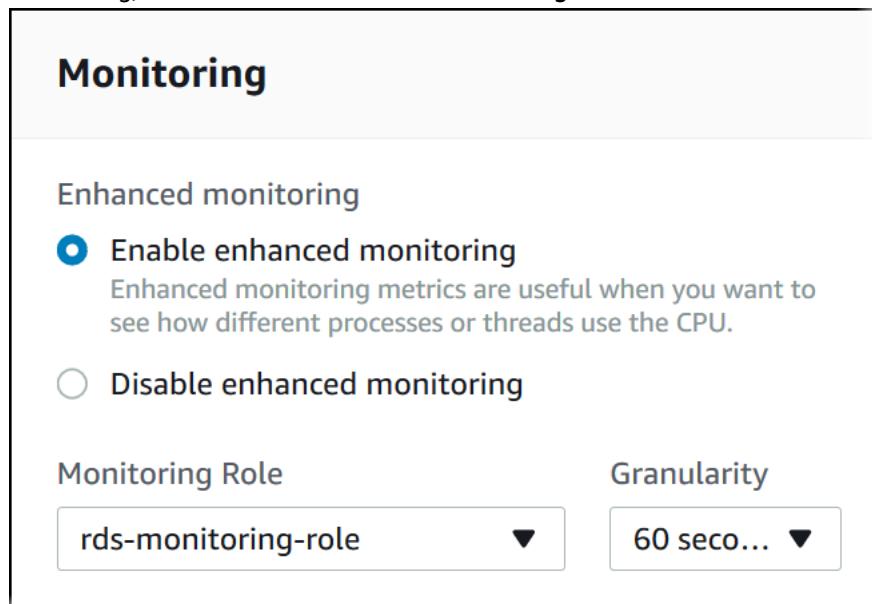
You can enable Enhanced Monitoring when you create a DB instance or read replica, or when you modify a DB instance. If you modify a DB instance to enable Enhanced Monitoring, you don't need to reboot your DB instance for the change to take effect.

You can enable Enhanced Monitoring in the RDS console when you do one of the following actions:

- **Create a DB instance** – You can enable Enhanced Monitoring in the **Monitoring** section under **Additional configuration**.
- **Create a read replica** – You can enable Enhanced Monitoring in the **Monitoring** section.
- **Modify a DB instance** – You can enable Enhanced Monitoring in the **Monitoring** section.

To enable Enhanced Monitoring by using the RDS console

1. Scroll to the **Monitoring** section.
2. Choose **Enable enhanced monitoring** for your DB instance or read replica. To disable Enhanced Monitoring, choose **Disable enhanced monitoring**.



3. Set the **Monitoring Role** property to the IAM role that you created to permit Amazon RDS to communicate with Amazon CloudWatch Logs for you, or choose **Default** to have RDS create a role for you named `rds-monitoring-role`.
4. Set the **Granularity** property to the interval, in seconds, between points when metrics are collected for your DB instance or read replica. The **Granularity** property can be set to one of the following values: 1, 5, 10, 15, 30, or 60.

Note

The fastest that the RDS console refreshes is every 5 seconds. If you set the granularity to 1 second in the RDS console, you still see updated metrics only every 5 seconds. You can retrieve 1-second metric updates by using CloudWatch Logs.

AWS CLI

To enable Enhanced Monitoring using the AWS CLI, in the following commands, set the `--monitoring-interval` option to a value other than 0 and set the `--monitoring-role-arn` option to the role you created in [Creating an IAM role for Enhanced Monitoring \(p. 472\)](#).

- `create-db-instance`
- `create-db-instance-read-replica`
- `modify-db-instance`

The `--monitoring-interval` option specifies the interval, in seconds, between points when Enhanced Monitoring metrics are collected. Valid values for the option are 0, 1, 5, 10, 15, 30, and 60.

To disable Enhanced Monitoring using the AWS CLI, set the `--monitoring-interval` option to 0 in the these commands.

Example

The following example enables Enhanced Monitoring for a DB instance:

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --monitoring-interval 30 \
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

For Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --monitoring-interval 30 ^
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

RDS API

To enable Enhanced Monitoring using the RDS API, set the `MonitoringInterval` parameter to a value other than 0 and set the `MonitoringRoleArn` parameter to the role you created in [Creating an IAM role for Enhanced Monitoring \(p. 472\)](#). Set these parameters in the following actions:

- [CreateDBInstance](#)
- [CreateDBInstanceReadReplica](#)
- [ModifyDBInstance](#)

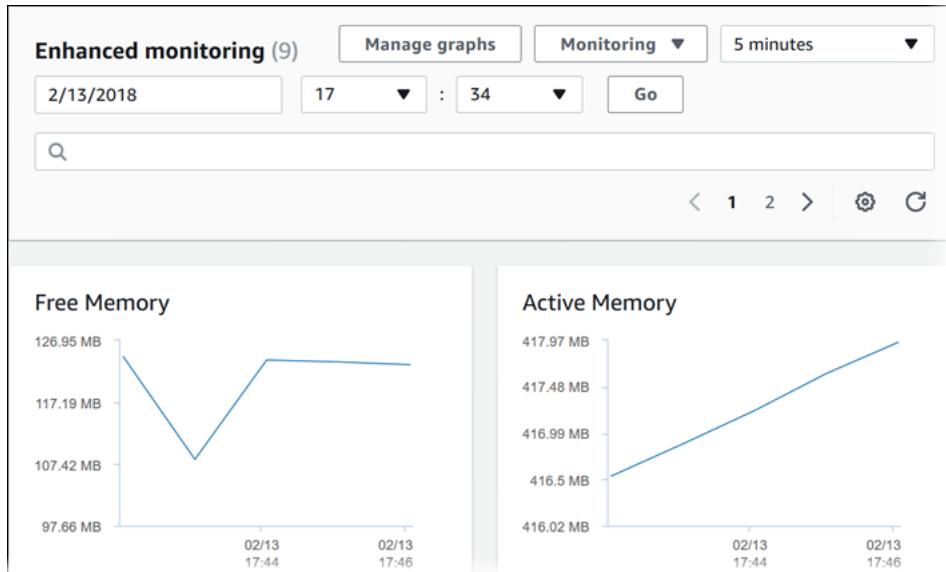
The `MonitoringInterval` parameter specifies the interval, in seconds, between points when Enhanced Monitoring metrics are collected. Valid values are 0, 1, 5, 10, 15, 30, and 60.

To disable Enhanced Monitoring using the RDS API, set `MonitoringInterval` to 0.

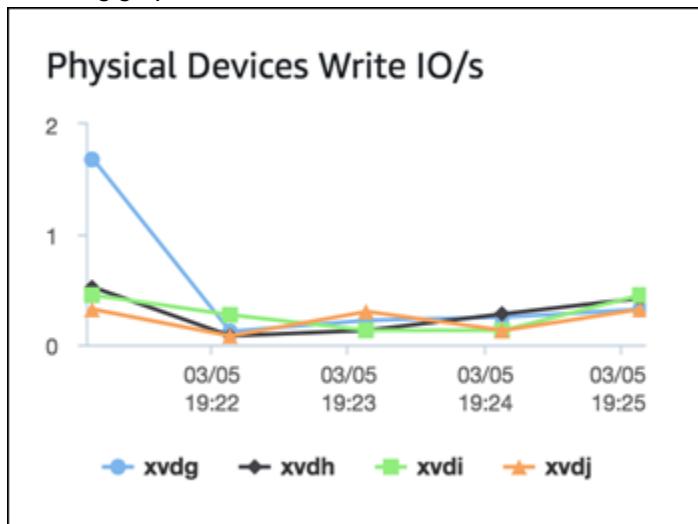
Viewing Enhanced Monitoring

You can view OS metrics reported by Enhanced Monitoring in the RDS console by choosing **Enhanced monitoring for Monitoring**.

The Enhanced Monitoring page is shown following.



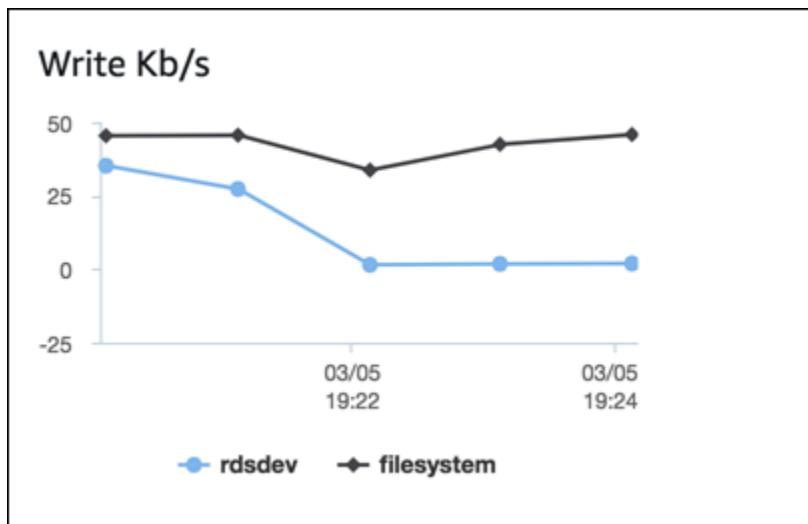
Some DB instances use more than one disk for the DB instance's data storage volume. On those DB instances, the **Physical Devices** graphs show metrics for each one of the disks. For example, the following graph shows metrics for four disks.



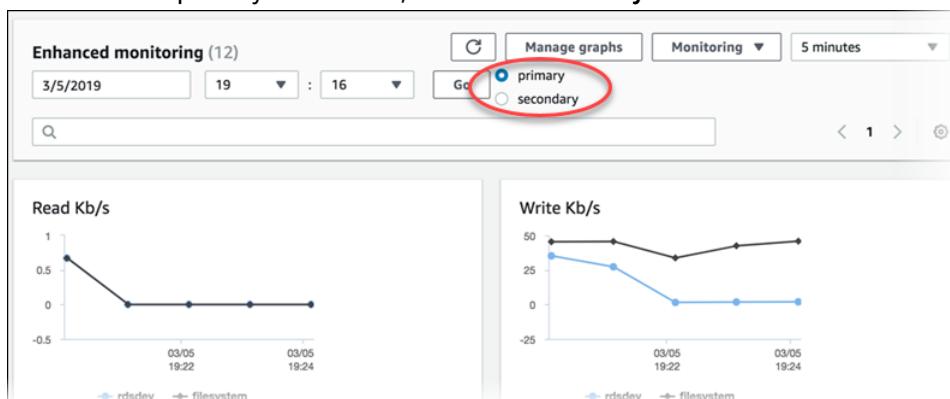
Note

Currently, **Physical Devices** graphs are not available for Microsoft SQL Server DB instances.

When you are viewing aggregated **Disk I/O** and **File system** graphs, the **rdsdev** device relates to the **/rdsdbdata** file system, where all database files and logs are stored. The **filesystem** device relates to the **/** file system (also known as root), where files related to the operating system are stored.



If the DB instance is a Multi-AZ deployment, you can view the OS metrics for the primary DB instance and its Multi-AZ standby replica. In the **Enhanced monitoring** view, choose **primary** to view the OS metrics for the primary DB instance, or choose **secondary** to view the OS metrics for the standby replica.



For more information about Multi-AZ deployments, see [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).

Note

Currently, viewing OS metrics for a Multi-AZ standby replica is not supported for MariaDB or Microsoft SQL Server DB instances.

If you want to see details for the processes running on your DB instance, choose **OS process list** for **Monitoring**.

The **Process List** view is shown following.

Process List					
<input type="text"/> Filter process list					
NAME	VIRT	RES	CPU%	MEM%	VMLIMIT
postgres [3181] ^t	283.55 MB	17.11 MB	0.02	1.72	
postgres:					
rdsadmin	384.7 MB	9.51 MB	0.02	0.95	
rdsadmin					
localhost(40156)					
idle [2953] ^t					

The Enhanced Monitoring metrics shown in the **Process list** view are organized as follows:

- **RDS child processes** – Shows a summary of the RDS processes that support the DB instance, for example `mysqld` for MySQL DB instances. Process threads appear nested beneath the parent process. Process threads show CPU utilization only as other metrics are the same for all threads for the process. The console displays a maximum of 100 processes and threads. The results are a combination of the top CPU consuming and memory consuming processes and threads. If there are more than 50 processes and more than 50 threads, the console displays the top 50 consumers in each category. This display helps you identify which processes are having the greatest impact on performance.
- **RDS processes** – Shows a summary of the resources used by the RDS management agent, diagnostics monitoring processes, and other AWS processes that are required to support RDS DB instances.
- **OS processes** – Shows a summary of the kernel and system processes, which generally have minimal impact on performance.

Note

The **Process list** view might sometimes contain internal, undocumented processes.

The items listed for each process are:

- **VIRT** – Displays the virtual size of the process.
- **RES** – Displays the actual physical memory being used by the process.
- **CPU%** – Displays the percentage of the total CPU bandwidth being used by the process.
- **MEM%** – Displays the percentage of the total memory being used by the process.

The monitoring data that is shown in the RDS console is retrieved from Amazon CloudWatch Logs. You can also retrieve the metrics for a DB instance as a log stream from CloudWatch Logs. For more information, see [Viewing Enhanced Monitoring by using CloudWatch Logs \(p. 478\)](#).

Enhanced Monitoring metrics are not returned during the following:

- A failover of the DB instance.
- Changing the instance class of the DB instance (scale compute).

Enhanced Monitoring metrics are returned during a reboot of a DB instance because only the database engine is rebooted. Metrics for the operating system are still reported.

Viewing Enhanced Monitoring by using CloudWatch Logs

After you have enabled Enhanced Monitoring for your DB instance, you can view the metrics for your DB instance using CloudWatch Logs, with each log stream representing a single DB instance being monitored. The log stream identifier is the resource identifier (`DbiResourceId`) for the DB instance.

To view Enhanced Monitoring log data

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, choose the region that your DB instance is in. For more information, see [Regions and endpoints](#) in the *Amazon Web Services General Reference*.
3. Choose **Logs** in the navigation pane.
4. Choose **RDSOSMetrics** from the list of log groups.

In a Multi-AZ deployment, log files with `-secondary` appended to the name are for the Multi-AZ standby replica.

The screenshot shows the CloudWatch Log Groups interface. The top navigation bar includes 'CloudWatch' and 'Log Groups'. Below it, there are three buttons: 'Search Log Group', 'Create Log Stream', and 'Delete Log Stream'. A 'Filter' input field is set to 'Log Stream Name Prefix'. The main area displays a table of log streams. The first row has a checkbox and the text 'Log Streams'. The second row contains a checkbox, the log stream name 'db-SORJBHOPBSMWGRI5EJW3KMYOTU-secondary', and the 'Last Event Time' '2019-03-05 12:12 UTC-8'. The third row contains a checkbox, the log stream name 'db-SORJBHOPBSMWGRI5EJW3KMYOTU', and the 'Last Event Time' '2019-03-05 12:07 UTC-8'. A red box highlights the log stream name in the second row.

5. Choose the log stream that you want to view from the list of log streams.

Available OS metrics

The following tables list the OS metrics available using Amazon CloudWatch Logs.

Metrics for MariaDB, MySQL, Oracle, and PostgreSQL DB instances

Group	Metric	Console name	Description
General	engine	Not applicable	The database engine for the DB instance.
	instanceID	Not applicable	The DB instance identifier.
	instanceResouceID	Not applicable	An immutable identifier for the DB instance that is unique to an AWS Region, also used as the log stream identifier.
	numVCPU	Not applicable	The number of virtual CPUs for the DB instance.

Group	Metric	Console name	Description
	timestamp	Not applicable	The time at which the metrics were taken.
	uptime	Not applicable	The amount of time that the DB instance has been active.
	version	Not applicable	The version of the OS metrics' stream JSON format.
cpuUtilization	guest	CPU Guest	The percentage of CPU in use by guest programs.
	idle	CPU Idle	The percentage of CPU that is idle.
	irq	CPU IRQ	The percentage of CPU in use by software interrupts.
	nice	CPU Nice	The percentage of CPU in use by programs running at lowest priority.
	steal	CPU Steal	The percentage of CPU in use by other virtual machines.
	system	CPU System	The percentage of CPU in use by the kernel.
	total	CPU Total	The total percentage of the CPU in use. This value includes the nice value.
	user	CPU User	The percentage of CPU in use by user programs.
	wait	CPU Wait	The percentage of CPU unused while waiting for I/O access.
diskIO	avgQueueLen	Avg Queue Size	The number of requests waiting in the I/O device's queue.
	avgReqSz	Ave Request Size	The average request size, in kilobytes.
	await	Disk I/O Await	The number of milliseconds required to respond to requests, including queue time and service time.
	device	Not applicable	The identifier of the disk device in use.
	readIOsPS	Read IO/s	The number of read operations per second.
	readKb	Read Total	The total number of kilobytes read.
	readKbPS	Read Kb/s	The number of kilobytes read per second.
	readLatency	Read Latency	The elapsed time between the submission of a read I/O request and its completion, in milliseconds. This metric is only available for Amazon Aurora.
	readThroughput	Read Throughput	The amount of network throughput used by requests to the DB cluster, in bytes per second. This metric is only available for Amazon Aurora.
	rrqmPS	Rrqms	The number of merged read requests queued per second.

Group	Metric	Console name	Description
physicalDeviceIOPs	tps	TPS	The number of I/O transactions per second.
	util	Disk I/O Util	The percentage of CPU time during which requests were issued.
	writeIOsPS	Write IO/s	The number of write operations per second.
	writeKb	Write Total	The total number of kilobytes written.
	writeKbPS	Write Kb/s	The number of kilobytes written per second.
	writeLatency	Write Latency	The average elapsed time between the submission of a write I/O request and its completion, in milliseconds. This metric is only available for Amazon Aurora.
	writeThroughput	Write Throughput	The amount of network throughput used by responses from the DB cluster, in bytes per second. This metric is only available for Amazon Aurora.
	wrqmPS	Wrqms	The number of merged write requests queued per second.
	physicalDeviceReqQueueLen	Physical Devices Avg Queue Size	The number of requests waiting in the I/O device's queue.
	avgReqSz	Physical Devices Ave Request Size	The average request size, in kilobytes.
	await	Physical Devices Disk I/O Await	The number of milliseconds required to respond to requests, including queue time and service time.
	device	Not applicable	The identifier of the disk device in use.
	readIOsPS	Physical Devices Read IO/s	The number of read operations per second.
	readKb	Physical Devices Read Total	The total number of kilobytes read.
	readKbPS	Physical Devices Read Kb/s	The number of kilobytes read per second.
	rrqmPS	Physical Devices Rrqms	The number of merged read requests queued per second.
	tps	Physical Devices TPS	The number of I/O transactions per second.

Group	Metric	Console name	Description
	util	Physical Devices Disk I/O Util	The percentage of CPU time during which requests were issued.
	writeIOPSPS	Physical Devices Write IO/s	The number of write operations per second.
	writeKb	Physical Devices Write Total	The total number of kilobytes written.
	writeKbPS	Physical Devices Write Kb/s	The number of kilobytes written per second.
	wrqmPS	Physical Devices Wrqms	The number of merged write requests queued per second.
fileSys	maxFiles	Max Inodes	The maximum number of files that can be created for the file system.
	mountPoint	Not applicable	The path to the file system.
	name	Not applicable	The name of the file system.
	total	Total Filesystem	The total number of disk space available for the file system, in kilobytes.
	used	Used Filesystem	The amount of disk space used by files in the file system, in kilobytes.
	usedFilePercent	Used %	The percentage of available files in use.
	usedFiles	Used Inodes	The number of files in the file system.
	usedPercent	Used Inodes %	The percentage of the file-system disk space in use.
loadAverage	fifteen	Load Avg 15 min	The number of processes requesting CPU time over the last 15 minutes.
	five	Load Avg 5 min	The number of processes requesting CPU time over the last 5 minutes.
	one	Load Avg 1 min	The number of processes requesting CPU time over the last minute.
memory	active	Active Memory	The amount of assigned memory, in kilobytes.
	buffers	Buffered Memory	The amount of memory used for buffering I/O requests prior to writing to the storage device, in kilobytes.

Group	Metric	Console name	Description
memory	cached	Cached Memory	The amount of memory used for caching file system-based I/O.
	dirty	Dirty Memory	The amount of memory pages in RAM that have been modified but not written to their related data block in storage, in kilobytes.
	free	Free Memory	The amount of unassigned memory, in kilobytes.
	hugePagesFree	Huge Pages Free	The number of free huge pages. Huge pages are a feature of the Linux kernel.
	hugePagesRsvd	Huge Pages Rsvd	The number of committed huge pages.
	hugePagesSize	Huge Pages Size	The size for each huge pages unit, in kilobytes.
	hugePagesSurp	Huge Pages Surp	The number of available surplus huge pages over the total.
	hugePagesTotal	Huge Pages Total	The total number of huge pages.
	inactive	Inactive Memory	The amount of least-frequently used memory pages, in kilobytes.
	mapped	Mapped Memory	The total amount of file-system contents that is memory mapped inside a process address space, in kilobytes.
	pageTables	Page Tables	The amount of memory used by page tables, in kilobytes.
	slab	Slab Memory	The amount of reusable kernel data structures, in kilobytes.
	total	Total Memory	The total amount of memory, in kilobytes.
	writeback	Writeback Memory	The amount of dirty pages in RAM that are still being written to the backing storage, in kilobytes.
network	interface	Not applicable	The identifier for the network interface being used for the DB instance.
	rx	RX	The number of bytes received per second.
	tx	TX	The number of bytes uploaded per second.
processList	cpuUsedPc	CPU %	The percentage of CPU used by the process.
	id	Not applicable	The identifier of the process.
	memoryUsedPc	MEM%	The percentage of memory used by the process.
	name	Not applicable	The name of the process.

Group	Metric	Console name	Description
	parentID	Not applicable	The process identifier for the parent process of the process.
	rss	RES	The amount of RAM allocated to the process, in kilobytes.
	tgid	Not applicable	The thread group identifier, which is a number representing the process ID to which a thread belongs. This identifier is used to group threads from the same process.
	vss	VIRT	The amount of virtual memory allocated to the process, in kilobytes.
swap	swap	Swap	The amount of swap memory available, in kilobytes.
	swap_in	Swaps in	The amount of memory, in kilobytes, swapped in from disk.
	swap_out	Swaps out	The amount of memory, in kilobytes, swapped out to disk.
	free	Free Swap	The amount of swap memory free, in kilobytes.
	committed	Committed Swap	The amount of swap memory, in kilobytes, used as cache memory.
tasks	blocked	Tasks Blocked	The number of tasks that are blocked.
	running	Tasks Running	The number of tasks that are running.
	sleeping	Tasks Sleeping	The number of tasks that are sleeping.
	stopped	Tasks Stopped	The number of tasks that are stopped.
	total	Tasks Total	The total number of tasks.
	zombie	Tasks Zombie	The number of child tasks that are inactive with an active parent task.

Metrics for Microsoft SQL Server DB instances

Group	Metric	Console name	Description
General	engine	Not applicable	The database engine for the DB instance.
	instanceID	Not applicable	The DB instance identifier.
	instanceResourceIdentifier	Not applicable	An immutable identifier for the DB instance that is unique to an AWS Region, also used as the log stream identifier.
	numVCpus	Not applicable	The number of virtual CPUs for the DB instance.

Group	Metric	Console name	Description
	timestamp	Not applicable	The time at which the metrics were taken.
	uptime	Not applicable	The amount of time that the DB instance has been active.
	version	Not applicable	The version of the OS metrics' stream JSON format.
cpuUtilization	idle	CPU Idle	The percentage of CPU that is idle.
	kern	CPU Kernel	The percentage of CPU in use by the kernel.
	user	CPU User	The percentage of CPU in use by user programs.
disks	name	Not applicable	The identifier for the disk.
	totalKb	Total Disk Space	The total space of the disk, in kilobytes.
	usedKb	Used Disk Space	The amount of space used on the disk, in kilobytes.
	usedPc	Used Disk Space %	The percentage of space used on the disk.
	availKb	Available Disk Space	The space available on the disk, in kilobytes.
	availPc	Available Disk Space %	The percentage of space available on the disk.
	rdCountPS	Reads/s	The number of read operations per second
	rdBytesPS	Read Kb/s	The number of bytes read per second.
	wrCountPS	Write IO/s	The number of write operations per second.
	wrBytesPS	Write Kb/s	The amount of bytes written per second.
memory	commitTotKb	Commit Total	The amount of pagefile-backed virtual address space in use, that is, the current commit charge. This value is composed of main memory (RAM) and disk (pagefiles).
	commitLimitKb	Maximum Commit	The maximum possible value for the commitTotKb metric. This value is the sum of the current pagefile size plus the physical memory available for pageable contents, excluding RAM that is assigned to nonpageable areas.
	commitPeakKb	Commit Peak	The largest value of the commitTotKb metric since the operating system was last started.
	kernTotKb	Total Kernel Memory	The sum of the memory in the paged and nonpaged kernel pools, in kilobytes.
	kernPagedKb	Paged Kernel Memory	The amount of memory in the paged kernel pool, in kilobytes.

Group	Metric	Console name	Description
	kernNonpagedKb	Nonpaged Kernel Memory	The amount of memory in the nonpaged kernel pool, in kilobytes.
	pageSize	Page Size	The size of a page, in bytes.
	physTotKb	Total Memory	The amount of physical memory, in kilobytes.
	physAvailKb	Available Memory	The amount of available physical memory, in kilobytes.
	sqlServerTotKb	SQL Server Total Memory	The amount of memory committed to SQL Server, in kilobytes.
	sysCacheKb	System Cache	The amount of system cache memory, in kilobytes.
network	interface	Not applicable	The identifier for the network interface being used for the DB instance.
	rdBytesPS	Network Read Kb/s	The number of bytes received per second.
	wrBytesPS	Network Write Kb/s	The number of bytes sent per second.
processList	cpuUsedPc	Used %	The percentage of CPU used by the process.
	memUsedPc	MEM%	The percentage of total memory used by the process.
	name	Not applicable	The name of the process.
	pid	Not applicable	The identifier of the process. This value is not present for processes that are owned by Amazon RDS.
	ppid	Not applicable	The process identifier for the parent of this process. This value is only present for child processes.
	tid	Not applicable	The thread identifier. This value is only present for threads. The owning process can be identified by using the pid value.
	workingSetKb	Not applicable	The amount of memory in the private working set plus the amount of memory that is in use by the process and can be shared with other processes, in kilobytes.
	workingSetPrivKb	Not applicable	The amount of memory that is in use by a process, but can't be shared with other processes, in kilobytes.
	workingSetShareableKb	Not applicable	The amount of memory that is in use by a process and can be shared with other processes, in kilobytes.

Group	Metric	Console name	Description
	virtKb	Not applicable	The amount of virtual address space the process is using, in kilobytes. Use of virtual address space doesn't necessarily imply corresponding use of either disk or main memory pages.
system	handles	Handles	The number of handles that the system is using.
	processes	Processes	The number of processes running on the system.
	threads	Threads	The number of threads running on the system.

Using Amazon RDS event notification

Topics

- [Amazon RDS event categories and event messages \(p. 488\)](#)
- [Subscribing to Amazon RDS event notification \(p. 494\)](#)
- [Listing Amazon RDS event notification subscriptions \(p. 496\)](#)
- [Modifying an Amazon RDS event notification subscription \(p. 497\)](#)
- [Adding a source identifier to an Amazon RDS event notification subscription \(p. 499\)](#)
- [Removing a source identifier from an Amazon RDS event notification subscription \(p. 500\)](#)
- [Listing the Amazon RDS event notification categories \(p. 501\)](#)
- [Deleting an Amazon RDS event notification subscription \(p. 502\)](#)

Amazon RDS uses the Amazon Simple Notification Service (Amazon SNS) to provide notification when an Amazon RDS event occurs. These notifications can be in any notification form supported by Amazon SNS for an AWS Region, such as an email, a text message, or a call to an HTTP endpoint.

Amazon RDS groups these events into categories that you can subscribe to so that you can be notified when an event in that category occurs. You can subscribe to an event category for a DB instance, DB snapshot, DB parameter group, or DB security group. For example, if you subscribe to the Backup category for a given DB instance, you are notified whenever a backup-related event occurs that affects the DB instance. If you subscribe to a configuration change category for a DB security group, you are notified when the DB security group is changed. You also receive notification when an event notification subscription changes.

Event notifications are sent to the addresses that you provide when you create the subscription. You might want to create several different subscriptions, such as one subscription receiving all event notifications and another subscription that includes only critical events for your production DB instances. You can easily turn off notification without deleting a subscription by choosing **No** for **Enabled** in the Amazon RDS console or by setting the `Enabled` parameter to `false` using the AWS CLI or Amazon RDS API.

Important

Amazon RDS doesn't guarantee the order of events sent in an event stream. The event order is subject to change.

Note

For more information on using text messages with SNS, see [Mobile text messaging \(SMS\)](#) in the [Amazon Simple Notification Service Developer Guide](#).

Amazon RDS uses the ARN of an Amazon SNS topic to identify each subscription. The Amazon RDS console creates the ARN for you when you create the subscription. If you use the CLI or API, you create the ARN by using the Amazon SNS console or the Amazon SNS API when you create a subscription.

Billing for Amazon RDS event notification is through the Amazon Simple Notification Service (Amazon SNS). Amazon SNS fees apply when using event notification. For more information on Amazon SNS billing, see [Amazon Simple Notification Service pricing](#).

The process for subscribing to Amazon RDS event notification is as follows:

1. Create an Amazon RDS event notification subscription by using the Amazon RDS console, AWS CLI, or API.
2. Amazon RDS sends an approval email or SMS message to the addresses you submitted with your subscription. To confirm your subscription, choose the link in the notification you were sent.
3. When you have confirmed the subscription, the status of your subscription is updated in the Amazon RDS console's **My Event Subscriptions** section.
4. You then begin to receive event notifications.

Note

When Amazon SNS sends a notification to a subscribed HTTP or HTTPS endpoint, the POST message sent to the endpoint has a message body that contains a JSON document. For more information, see [Amazon SNS message and JSON formats](#) in the *Amazon Simple Notification Service Developer Guide*.

You can use AWS Lambda to process event notifications from a DB instance. For more information, see [Using AWS Lambda with Amazon RDS](#) in the *AWS Lambda Developer Guide*.

The following section lists all categories and events that you can be notified of. It also provides information about subscribing to and working with Amazon RDS event subscriptions.

Amazon RDS event categories and event messages

Amazon RDS generates a significant number of events in categories that you can subscribe to using the Amazon RDS Console, AWS CLI, or the API. Each category applies to a source type, which can be one of the following:

- DB instance
- DB security group
- DB parameter group

The following table shows the event category and a list of events when a DB instance is the source type.

Category	Amazon RDS event ID	Description
availability	RDS-EVENT-0006	The DB instance restarted.
availability	RDS-EVENT-0004	DB instance shutdown.
availability	RDS-EVENT-0022	An error has occurred while restarting MySQL or MariaDB.
backup	RDS-EVENT-0001	Backing up DB instance.
backup	RDS-EVENT-0002	Finished DB Instance backup.
configuration change	RDS-EVENT-0009	The DB instance has been added to a security group.
configuration change	RDS-EVENT-0024	The DB instance is being converted to a Multi-AZ DB instance.
configuration change	RDS-EVENT-0030	The DB instance is being converted to a Single-AZ DB instance.
configuration change	RDS-EVENT-0012	Applying modification to database instance class.
configuration change	RDS-EVENT-0018	The current storage settings for this DB instance are being changed.
configuration change	RDS-EVENT-0011	A parameter group for this DB instance has changed.
configuration change	RDS-EVENT-0092	A parameter group for this DB instance has finished updating.

Category	Amazon RDS event ID	Description
configuration change	RDS-EVENT-0028	Automatic backups for this DB instance have been disabled.
configuration change	RDS-EVENT-0032	Automatic backups for this DB instance have been enabled.
configuration change	RDS-EVENT-0033	There are [count] users that match the master user name. Users not tied to a specific host have been reset.
configuration change	RDS-EVENT-0025	The DB instance has been converted to a Multi-AZ DB instance.
configuration change	RDS-EVENT-0029	The DB instance has been converted to a Single-AZ DB instance.
configuration change	RDS-EVENT-0014	The DB instance class for this DB instance has changed.
configuration change	RDS-EVENT-0017	The storage settings for this DB instance have changed.
configuration change	RDS-EVENT-0010	The DB instance has been removed from a security group.
configuration change	RDS-EVENT-0016	The master password for the DB instance has been reset.
configuration change	RDS-EVENT-0067	An attempt to reset the master password for the DB instance has failed.
configuration change	RDS-EVENT-0078	The Enhanced Monitoring configuration has been changed.
creation	RDS-EVENT-0005	DB instance created.
deletion	RDS-EVENT-0003	The DB instance has been deleted.
failover	RDS-EVENT-0034	Amazon RDS is not attempting a requested failover because a failover recently occurred on the DB instance.
failover	RDS-EVENT-0013	A Multi-AZ failover that resulted in the promotion of a standby instance has started.
failover	RDS-EVENT-0015	A Multi-AZ failover that resulted in the promotion of a standby instance is complete. It may take several minutes for the DNS to transfer to the new primary DB instance.
failover	RDS-EVENT-0065	The instance has recovered from a partial failover.
failover	RDS-EVENT-0049	A Multi-AZ failover has completed.
failover	RDS-EVENT-0050	A Multi-AZ activation has started after a successful instance recovery.
failover	RDS-EVENT-0051	A Multi-AZ activation is complete. Your database should be accessible now.

Category	Amazon RDS event ID	Description
failure	RDS-EVENT-0031	The DB instance has failed due to an incompatible configuration or an underlying storage issue. Begin a point-in-time-restore for the DB instance.
failure	RDS-EVENT-0036	The DB instance is in an incompatible network. Some of the specified subnet IDs are invalid or do not exist.
failure	RDS-EVENT-0035	The DB instance has invalid parameters. For example, if the DB instance could not start because a memory-related parameter is set too high for this instance class, the customer action would be to modify the memory parameter and reboot the DB instance.
failure	RDS-EVENT-0058	Error while creating Statspack user account PERFSTAT. Please drop the account before adding the Statspack option.
failure	RDS-EVENT-0079	Enhanced Monitoring cannot be enabled without the enhanced monitoring IAM role. For information on creating the enhanced monitoring IAM role, see To create an IAM role for Amazon RDS enhanced monitoring (p. 472) .
failure	RDS-EVENT-0080	Enhanced Monitoring was disabled due to an error making the configuration change. It is likely that the enhanced monitoring IAM role is configured incorrectly. For information on creating the enhanced monitoring IAM role, see To create an IAM role for Amazon RDS enhanced monitoring (p. 472) .
failure	RDS-EVENT-0081	The IAM role that you use to access your Amazon S3 bucket for SQL Server native backup and restore is configured incorrectly. For more information, see Setting up for native backup and restore (p. 672) .
failure	RDS-EVENT-0188	Amazon RDS was unable to upgrade a MySQL DB instance from version 5.7 to version 8.0 because of incompatibilities related to the data dictionary. The DB instance was rolled back to MySQL version 5.7. For more information, see Rollback after failure to upgrade from MySQL 5.7 to 8.0 (p. 857) .
low storage	RDS-EVENT-0089	The DB instance has consumed more than 90% of its allocated storage. You can monitor the storage space for a DB instance using the Free Storage Space metric. For more information, see Viewing DB instance metrics (p. 548) .
low storage	RDS-EVENT-0007	The allocated storage for the DB instance has been consumed. To resolve this issue, allocate additional storage for the DB instance. For more information, see the RDS FAQ . You can monitor the storage space for a DB instance using the Free Storage Space metric. For more information, see Viewing DB instance metrics (p. 548) .

Category	Amazon RDS event ID	Description
maintenance	RDS-EVENT-0026	Offline maintenance of the DB instance is taking place. The DB instance is currently unavailable.
maintenance	RDS-EVENT-0027	Offline maintenance of the DB instance is complete. The DB instance is now available.
maintenance	RDS-EVENT-0047	Patching of the DB instance has completed.
maintenance	RDS-EVENT-0155	The DB instance has a DB engine minor version upgrade available.
notification	RDS-EVENT-0044	Operator-issued notification. For more information, see the event message.
notification	RDS-EVENT-0048	Patching of the DB instance has been delayed.
notification	RDS-EVENT-0054	The MySQL storage engine you are using is not InnoDB, which is the recommended MySQL storage engine for Amazon RDS. For information about MySQL storage engines, see Supported storage engines for MySQL on Amazon RDS .
notification	RDS-EVENT-0055	<p>The number of tables you have for your DB instance exceeds the recommended best practices for Amazon RDS. Please reduce the number of tables on your DB instance.</p> <p>For information about recommended best practices, see Amazon RDS basic operational guidelines (p. 128).</p>
notification	RDS-EVENT-0056	<p>The number of databases you have for your DB instance exceeds the recommended best practices for Amazon RDS. Please reduce the number of databases on your DB instance.</p> <p>For information about recommended best practices, see Amazon RDS basic operational guidelines (p. 128).</p>
notification	RDS-EVENT-0064	The TDE key has been rotated. For information about recommended best practices, see Amazon RDS basic operational guidelines (p. 128) .
notification	RDS-EVENT-0084	You attempted to convert a DB instance to Multi-AZ, but it contains in-memory file groups that are not supported for Multi-AZ. For more information, see Multi-AZ deployments for Microsoft SQL Server (p. 698) .
notification	RDS-EVENT-0087	The DB instance has been stopped.
notification	RDS-EVENT-0088	The DB instance has been started.
notification	RDS-EVENT-0154	The DB instance is being started due to it exceeding the maximum allowed time being stopped.

Category	Amazon RDS event ID	Description
notification	RDS-EVENT-0157	<p>RDS can't modify the DB instance class because the target instance class can't support the number of databases that exist on the source DB instance. The error message appears as: "The instance has N databases, but after conversion it would only support N".</p> <p>For more information, see Limits for Microsoft SQL Server DB instances (p. 632).</p>
notification	RDS-EVENT-0158	DB instance is in a state that can't be upgraded.
read replica	RDS-EVENT-0045	<p>An error has occurred in the read replication process. For more information, see the event message.</p> <p>In addition, see the troubleshooting section for read replicas for your DB engine.</p> <ul style="list-style-type: none"> • Troubleshooting a MariaDB read replica problem (p. 612) • Troubleshooting a SQL Server read replica problem (p. 697) • Troubleshooting a MySQL read replica problem (p. 908) • Troubleshooting Oracle replicas (p. 1124) • Troubleshooting a PostgreSQL read replica problem (p. 1546)
read replica	RDS-EVENT-0046	The read replica has resumed replication. This message appears when you first create a read replica, or as a monitoring message confirming that replication is functioning properly. If this message follows an RDS-EVENT-0045 notification, then replication has resumed following an error or after replication was stopped.
read replica	RDS-EVENT-0057	Replication on the read replica was terminated.
read replica	RDS-EVENT-0062	Replication on the read replica was manually stopped.
read replica	RDS-EVENT-0063	Replication on the read replica was reset.
recovery	RDS-EVENT-0020	Recovery of the DB instance has started. Recovery time will vary with the amount of data to be recovered.
recovery	RDS-EVENT-0021	Recovery of the DB instance is complete.
recovery	RDS-EVENT-0023	A manual backup has been requested but Amazon RDS is currently in the process of creating a DB snapshot. Submit the request again after Amazon RDS has completed the DB snapshot.
recovery	RDS-EVENT-0052	Recovery of the Multi-AZ instance has started. Recovery time will vary with the amount of data to be recovered.

Category	Amazon RDS event ID	Description
recovery	RDS-EVENT-0053	Recovery of the Multi-AZ instance is complete.
recovery	RDS-EVENT-0066	The SQL Server DB instance is re-establishing its mirror. Performance will be degraded until the mirror is reestablished. A database was found with non-FULL recovery model. The recovery model was changed back to FULL and mirroring recovery was started. (<dbname>: <recovery model found>[,...])"
restoration	RDS-EVENT-0008	The DB instance has been restored from a DB snapshot.
restoration	RDS-EVENT-0019	The DB instance has been restored from a point-in-time backup.

The following table shows the event category and a list of events when a DB parameter group is the source type.

Category	RDS event ID	Description
configuration change	RDS-EVENT-0037	The parameter group was modified.

The following table shows the event category and a list of events when a DB security group is the source type.

Category	RDS event ID	Description
configuration change	RDS-EVENT-0038	The security group has been modified.
failure	RDS-EVENT-0039	The security group owned by [user] does not exist; authorization for the security group has been revoked.

The following table shows the event category and a list of events when a DB snapshot is the source type.

Category	RDS event ID	Description
creation	RDS-EVENT-0040	A manual DB snapshot is being created.
creation	RDS-EVENT-0042	A manual DB snapshot has been created.
creation	RDS-EVENT-0090	An automated DB snapshot is being created.
creation	RDS-EVENT-0091	An automated DB snapshot has been created.
deletion	RDS-EVENT-0041	A DB snapshot has been deleted.
notification	RDS-EVENT-0059	Started copy of snapshot [DB snapshot name] from region [region name].

Category	RDS event ID	Description
notification	RDS-EVENT-0060	Finished copy of snapshot [DB snapshot name] from region [region name] in [time] minutes.
notification	RDS-EVENT-0061	Canceled snapshot copy request of [DB snapshot name] from region %[region name].
notification	RDS-EVENT-0159	DB snapshot export task failed.
notification	RDS-EVENT-0160	DB snapshot export task canceled.
notification	RDS-EVENT-0161	DB snapshot export task completed.
restoration	RDS-EVENT-0043	A DB instance is being restored from a DB snapshot.

Subscribing to Amazon RDS event notification

You can create an Amazon RDS event notification subscription so you can be notified when an event occurs for a given DB instance, DB snapshot, DB security group, or DB parameter group. The simplest way to create a subscription is with the RDS console. If you choose to create event notification subscriptions using the CLI or API, you must create an Amazon Simple Notification Service topic and subscribe to that topic with the Amazon SNS console or Amazon SNS API. You will also need to retain the Amazon Resource Name (ARN) of the topic because it is used when submitting CLI commands or API operations. For information on creating an SNS topic and subscribing to it, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

You can specify the type of source you want to be notified of and the Amazon RDS source that triggers the event. These are defined by the **SourceType** (type of source) and the **SourceIdentifier** (the Amazon RDS source generating the event). If you specify both the **SourceType** and **SourceIdentifier**, such as `SourceType = db-instance` and `SourceIdentifier = myDBInstance1`, you receive all the DB instance events for the specified source. If you specify a **SourceType** but don't specify a **SourceIdentifier**, you receive notice of the events for that source type for all your Amazon RDS sources. If you don't specify either the **SourceType** or the **SourceIdentifier**, you are notified of events generated from all Amazon RDS sources belonging to your customer account.

Note

Event notifications might take up to five minutes to be delivered.

Amazon RDS event notification is only available for unencrypted SNS topics. If you specify an encrypted SNS topic, event notifications aren't sent for the topic.

Console

To subscribe to RDS event notification

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In navigation pane, choose **Event subscriptions**.
3. In the **Event subscriptions** pane, choose **Create event subscription**.
4. In the **Create event subscription** dialog box, do the following:
 - a. For **Name**, enter a name for the event notification subscription.
 - b. For **Send notifications to**, choose an existing Amazon SNS ARN for an Amazon SNS topic, or choose **create topic** to enter the name of a topic and a list of recipients.
 - c. For **Source type**, choose a source type.

- d. Choose **Yes** to enable the subscription. If you want to create the subscription but to not have notifications sent yet, choose **No**.
- e. Depending on the source type you selected, choose the event categories and sources that you want to receive event notifications for.
- f. Choose **Create**.

The Amazon RDS console indicates that the subscription is being created.

Event subscriptions (2)				
		Edit	Delete	Create event subscription
<input type="text"/> Filter event subscriptions		< 1 >	<input type="radio"/>	<input type="radio"/>
Name	Status	Source Type	Enabled	
Configchangerdpgres	active	Instances	Yes	
Test	creating	Instances	Yes	

AWS CLI

To subscribe to RDS event notification, use the AWS CLI [create-event-subscription](#) command. Include the following required parameters:

- `--subscription-name`
- `--sns-topic-arn`

Example

For Linux, macOS, or Unix:

```
aws rds create-event-subscription \
--subscription-name myeventsubscription \
--sns-topic-arn arn:aws:sns:us-east-1:802#####:myawsuser-RDS \
--enabled
```

For Windows:

```
aws rds create-event-subscription ^
--subscription-name myeventsubscription ^
--sns-topic-arn arn:aws:sns:us-east-1:802#####:myawsuser-RDS ^
--enabled
```

API

To subscribe to Amazon RDS event notification, call the Amazon RDS API function [CreateEventSubscription](#). Include the following required parameters:

- `SubscriptionName`
- `SnsTopicArn`

Listing Amazon RDS event notification subscriptions

You can list your current Amazon RDS event notification subscriptions.

Console

To list your current Amazon RDS event notification subscriptions

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Event subscriptions**. The **Event subscriptions** pane shows all your event notification subscriptions.

Event subscriptions (2)		
	Name	Status
<input type="checkbox"/>	Configchangerdpgres	active
<input type="checkbox"/>	Postgresnotification	active

AWS CLI

To list your current Amazon RDS event notification subscriptions, use the AWS CLI [describe-event-subscriptions](#) command.

Example

The following example describes all event subscriptions.

```
aws rds describe-event-subscriptions
```

The following example describes the `myfirsteventsubscription`.

```
aws rds describe-event-subscriptions --subscription-name myfirsteventsubscription
```

API

To list your current Amazon RDS event notification subscriptions, call the Amazon RDS API [DescribeEventSubscriptions](#) action.

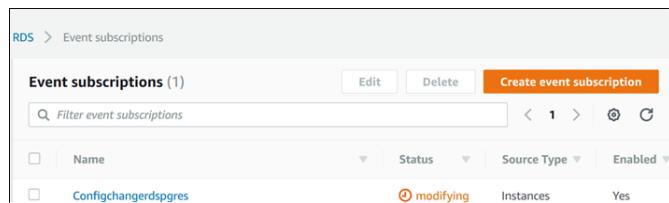
Modifying an Amazon RDS event notification subscription

After you have created a subscription, you can change the subscription name, source identifier, categories, or topic ARN.

Console

To modify an Amazon RDS event notification subscription

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Event subscriptions**.
3. In the **Event subscriptions** pane, choose the subscription that you want to modify and choose **Edit**.
4. Make your changes to the subscription in either the **Target** or **Source** section.
5. Choose **Edit**. The Amazon RDS console indicates that the subscription is being modified.



AWS CLI

To modify an Amazon RDS event notification subscription, use the AWS CLI [modify-event-subscription](#) command. Include the following required parameter:

- `--subscription-name`

Example

The following code enables `myeventsSubscription`.

For Linux, macOS, or Unix:

```
aws rds modify-event-subscription \
--subscription-name myeventsSubscription \
--enabled
```

For Windows:

```
aws rds modify-event-subscription ^
--subscription-name myeventsSubscription ^
--enabled
```

API

To modify an Amazon RDS event, call the Amazon RDS API operation [ModifyEventSubscription](#). Include the following required parameter:

- SubscriptionName

Adding a source identifier to an Amazon RDS event notification subscription

You can add a source identifier (the Amazon RDS source generating the event) to an existing subscription.

Console

You can easily add or remove source identifiers using the Amazon RDS console by selecting or deselecting them when modifying a subscription. For more information, see [Modifying an Amazon RDS event notification subscription \(p. 497\)](#).

AWS CLI

To add a source identifier to an Amazon RDS event notification subscription, use the AWS CLI [add-source-identifier-to-subscription](#) command. Include the following required parameters:

- `--subscription-name`
- `--source-identifier`

Example

The following example adds the source identifier `mysqldb` to the `myrdseventsSubscription` subscription.

For Linux, macOS, or Unix:

```
aws rds add-source-identifier-to-subscription \
  --subscription-name myrdseventsSubscription \
  --source-identifier mysqldb
```

For Windows:

```
aws rds add-source-identifier-to-subscription ^
  --subscription-name myrdseventsSubscription ^
  --source-identifier mysqldb
```

API

To add a source identifier to an Amazon RDS event notification subscription, call the Amazon RDS API [AddSourceIdentifierToSubscription](#). Include the following required parameters:

- `SubscriptionName`
- `SourceIdentifier`

Removing a source identifier from an Amazon RDS event notification subscription

You can remove a source identifier (the Amazon RDS source generating the event) from a subscription if you no longer want to be notified of events for that source.

Console

You can easily add or remove source identifiers using the Amazon RDS console by selecting or deselecting them when modifying a subscription. For more information, see [Modifying an Amazon RDS event notification subscription \(p. 497\)](#).

AWS CLI

To remove a source identifier from an Amazon RDS event notification subscription, use the AWS CLI `remove-source-identifier-from-subscription` command. Include the following required parameters:

- `--subscription-name`
- `--source-identifier`

Example

The following example removes the source identifier `mysqldb` from the `myrdseventsSubscription` subscription.

For Linux, macOS, or Unix:

```
aws rds remove-source-identifier-from-subscription \
--subscription-name myrdseventsSubscription \
--source-identifier mysqldb
```

For Windows:

```
aws rds remove-source-identifier-from-subscription ^
--subscription-name myrdseventsSubscription ^
--source-identifier mysqldb
```

API

To remove a source identifier from an Amazon RDS event notification subscription, use the Amazon RDS API `RemoveSourceIdentifierFromSubscription` command. Include the following required parameters:

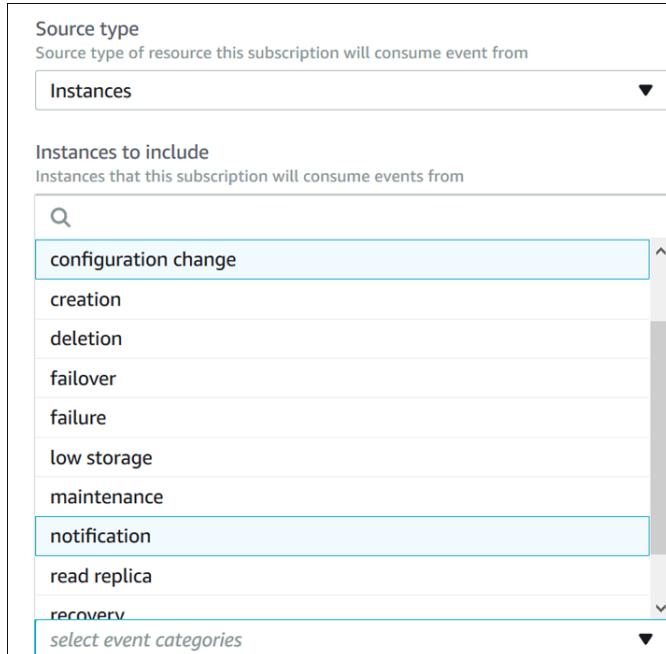
- `SubscriptionName`
- `SourceIdentifier`

List the Amazon RDS event notification categories

All events for a resource type are grouped into categories. To view the list of categories available, use the following procedures.

Console

When you create or modify an event notification subscription, the event categories are displayed in the Amazon RDS console. For more information, see [Modifying an Amazon RDS event notification subscription \(p. 497\)](#).



AWS CLI

To list the Amazon RDS event notification categories, use the AWS CLI `describe-event-categories` command. This command has no required parameters.

Example

```
aws rds describe-event-categories
```

API

To list the Amazon RDS event notification categories, use the Amazon RDS API `DescribeEventCategories` command. This command has no required parameters.

Deleting an Amazon RDS event notification subscription

You can delete a subscription when you no longer need it. All subscribers to the topic will no longer receive event notifications specified by the subscription.

Console

To delete an Amazon RDS event notification subscription

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **DB Event Subscriptions**.
3. In the **My DB Event Subscriptions** pane, choose the subscription that you want to delete.
4. Choose **Delete**.
5. The Amazon RDS console indicates that the subscription is being deleted.

The screenshot shows the 'Event subscriptions' section of the AWS RDS console. It displays two entries:

Name	Status
Configchangerdpgres	active
Postgresnotification	active

AWS CLI

To delete an Amazon RDS event notification subscription, use the AWS CLI `delete-event-subscription` command. Include the following required parameter:

- `--subscription-name`

Example

The following example deletes the subscription `myrdssubscription`.

```
aws rds delete-event-subscription --subscription-name myrdssubscription
```

API

To delete an Amazon RDS event notification subscription, use the RDS API `DeleteEventSubscription` command. Include the following required parameter:

- `SubscriptionName`

Viewing Amazon RDS events

Amazon RDS keeps a record of events that relate to your DB instances, DB snapshots, DB security groups, and DB parameter groups. This information includes the date and time of the event, the source name and source type of the event, and a message associated with the event.

You can retrieve events for your RDS resources through the AWS Management Console, which shows events from the past 24 hours. You can also retrieve events for your RDS resources by using the [describe-events](#) AWS CLI command, or the [DescribeEvents](#) RDS API operation. If you use the AWS CLI or the RDS API to view events, you can retrieve events for up to the past 14 days.

Note

If you need to store events for longer periods of time, you can send Amazon RDS events to CloudWatch Events. For more information, see [Getting CloudWatch Events and Amazon EventBridge events for Amazon RDS \(p. 551\)](#)

Console

To view all Amazon RDS instance events for the past 24 hours

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Events**. The available events appear in a list.
3. Use the **Filter** list to filter the events by type, and use the text box to the right of the **Filter** list to further filter your results. For example, the following screenshot shows a list of events filtered by the characters **stopped**.

The screenshot shows the AWS Management Console interface for viewing RDS events. The top navigation bar has 'RDS' and 'Events'. Below it, a search bar contains the text 'stopped'. A table displays one event row:

Source	Type	Time	Message
orclb	Instances	March 19, 2021, 7:34:09 PM UTC	DB instance stopped

AWS CLI

You can view all Amazon RDS instance events for the past 7 days by calling the [describe-events](#) AWS CLI command and setting the `--duration` parameter to 10080.

```
aws rds describe-events --duration 10080
```

API

You can view all Amazon RDS instance events for the past 14 days by calling the [DescribeEvents](#) RDS API operation and setting the `Duration` parameter to 20160.

Accessing Amazon RDS database log files

You can view, download, and watch database logs using the AWS Management Console, the AWS Command Line Interface (AWS CLI), or the Amazon RDS API. Viewing, downloading, or watching transaction logs isn't supported.

For engine-specific information, see the following:

- [MariaDB database log files \(p. 508\)](#)
- [Microsoft SQL Server database log files \(p. 516\)](#)
- [Accessing MySQL database log files \(p. 519\)](#)
- [Oracle database log files \(p. 527\)](#)
- [PostgreSQL database log files \(p. 534\)](#)

Viewing and listing database log files

You can view database log files for your DB engine by using the AWS Management Console. You can list what log files are available for download or monitoring by using the AWS CLI or Amazon RDS API.

Note

If you can't view the list of log files for an existing Oracle DB instance, reboot the instance to view the list.

Console

To view a database log file

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the name of the DB instance that has the log file that you want to view.
4. Choose the **Logs & events** tab.
5. Scroll down to the **Logs** section.
6. In the **Logs** section, choose the log that you want to view, and then choose **View**.

AWS CLI

To list the available database log files for a DB instance, use the AWS CLI `describe-db-log-files` command.

The following example returns a list of log files for a DB instance named `my-db-instance`.

Example

```
aws rds describe-db-log-files --db-instance-identifier my-db-instance
```

RDS API

To list the available database log files for a DB instance, use the Amazon RDS API `DescribeDBLogFiles` action.

Downloading a database log file

You can use the AWS Management Console, AWS CLI or API to download a database log file.

Console

To download a database log file

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the name of the DB instance that has the log file that you want to view.
4. Choose the **Logs & events** tab.
5. Scroll down to the **Logs** section.
6. In the **Logs** section, choose the button next to the log that you want to download, and then choose **Download**.
7. Open the context (right-click) menu for the link provided, and then choose **Save Link As**. Enter the location where you want the log file to be saved, and then choose **Save**.



AWS CLI

To download a database log file, use the AWS CLI command `download-db-log-file-portion`. By default, this command downloads only the latest portion of a log file. However, you can download an entire file by specifying the parameter `--starting-token 0`.

The following example shows how to download the entire contents of a log file called `log/ERROR.4` and store it in a local file called `errorlog.txt`.

Example

For Linux, macOS, or Unix:

```
aws rds download-db-log-file-portion \
    --db-instance-identifier myexampledb \
    --starting-token 0 --output text \
    --log-file-name log/ERROR.4 > errorlog.txt
```

For Windows:

```
aws rds download-db-log-file-portion ^
    --db-instance-identifier myexampledb ^
    --starting-token 0 --output text ^
    --log-file-name log/ERROR.4 > errorlog.txt
```

RDS API

To download a database log file, use the Amazon RDS API `DownloadDBLogFilePortion` action.

Watching a database log file

You can monitor the contents of a log file by using the AWS Management Console.

Console

To watch a database log file

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the name of the DB instance that has the log file that you want to view.
4. Choose the **Logs & events** tab.
5. In the **Logs** section, choose a log file, and then choose **Watch**.

Publishing database logs to Amazon CloudWatch Logs

In addition to viewing and downloading DB instance logs, you can publish logs to Amazon CloudWatch Logs. With CloudWatch Logs, you can perform real-time analysis of the log data, store the data in highly durable storage, and manage the data with the CloudWatch Logs Agent. AWS retains log data published to CloudWatch Logs for an indefinite time period unless you specify a retention period. For more information, see [Change log data retention in CloudWatch Logs](#).

For engine-specific information, see the following:

- the section called “Publishing MariaDB logs to Amazon CloudWatch Logs” (p. 509)
- the section called “Publishing MySQL logs to Amazon CloudWatch Logs” (p. 521)
- the section called “Publishing Oracle logs to Amazon CloudWatch Logs” (p. 529)
- the section called “Publishing PostgreSQL logs to Amazon CloudWatch Logs” (p. 537)
- the section called “Publishing SQL Server logs to Amazon CloudWatch Logs” (p. 516)

Reading log file contents using REST

Amazon RDS provides a REST endpoint that allows access to DB instance log files. This is useful if you need to write an application to stream Amazon RDS log file contents.

The syntax is:

```
GET /v13/downloadCompleteLogFile/DBInstanceIdentifier/LogFileName HTTP/1.1
Content-type: application/json
host: rds.region.amazonaws.com
```

The following parameters are required:

- *DBInstanceIdentifier*—the name of the DB instance that contains the log file you want to download.
- *LogFileName*—the name of the log file to be downloaded.

The response contains the contents of the requested log file, as a stream.

The following example downloads the log file named *log/ERROR.6* for the DB instance named *sample-sql* in the *us-west-2* region.

```
GET /v13/downloadCompleteLogFile/sample-sql/log/ERROR.6 HTTP/1.1
host: rds.us-west-2.amazonaws.com
X-Amz-Security-Token: AQoDYXdzEIH//////////wEa0AIXLhngC5zp9CyB1R6abwKrXHVR5efnAVN3XvR7IwqKYalFSn6UyJuEFTft9nObglx4QJ+GXV9cpACkETq=
X-Amz-Date: 20140903T233749Z
X-Amz-Algorithm: AWS4-HMAC-SHA256
X-Amz-Credential: AKIADQKE4SARGYLE/20140903/us-west-2/rds/aws4_request
X-Amz-SignedHeaders: host
X-Amz-Content-SHA256: e3b0c44298fc1c229afbf4c8996fb92427ae41e4649b934de495991b7852b855
X-Amz-Expires: 86400
X-Amz-Signature: 353a4f14b3f250142d9afc34f9f9948154d46ce7d4ec091d0cdabbcf8b40c558
```

If you specify a nonexistent DB instance, the response consists of the following error:

- **DBInstanceNotFound**—*DB Instance Identifier* does not refer to an existing DB instance. (HTTP status code: 404)

MariaDB database log files

You can monitor the MariaDB error log, slow query log, and the general log. The MariaDB error log is generated by default; you can generate the slow query and general logs by setting parameters in your DB parameter group. Amazon RDS rotates all of the MariaDB log files; the intervals for each type are given following.

You can monitor the MariaDB logs directly through the Amazon RDS console, Amazon RDS API, Amazon RDS CLI, or AWS SDKs. You can also access MariaDB logs by directing the logs to a database table in the main database and querying that table. You can use the `mysqlbinlog` utility to download a binary log.

For more information about viewing, downloading, and watching file-based database logs, see [Accessing Amazon RDS database log files \(p. 504\)](#).

Accessing MariaDB error logs

The MariaDB error log is written to the `<host-name>.err` file. You can view this file by using the Amazon RDS console or by retrieving the log using the Amazon RDS API, Amazon RDS CLI, or AWS SDKs. The `<host-name>.err` file is flushed every 5 minutes, and its contents are appended to `mysql-error-running.log`. The `mysql-error-running.log` file is then rotated every hour and the hourly files generated during the last 24 hours are retained. Each log file has the hour it was generated (in UTC) appended to its name. The log files also have a timestamp that helps you determine when the log entries were written.

MariaDB writes to the error log only on startup, shutdown, and when it encounters errors. A DB instance can go hours or days without new entries being written to the error log. If you see no recent entries, it's because the server did not encounter an error that resulted in a log entry.

Accessing the MariaDB slow query and general logs

The MariaDB slow query log and the general log can be written to a file or a database table by setting parameters in your DB parameter group. For information about creating and modifying a DB parameter group, see [Working with DB parameter groups \(p. 228\)](#). You must set these parameters before you can view the slow query log or general log in the Amazon RDS console or by using the Amazon RDS API, AWS CLI, or AWS SDKs.

You can control MariaDB logging by using the parameters in this list:

- `slow_query_log`: To create the slow query log, set to 1. The default is 0.
- `general_log`: To create the general log, set to 1. The default is 0.
- `long_query_time`: To prevent fast-running queries from being logged in the slow query log, specify a value for the shortest query run time to be logged, in seconds. The default is 10 seconds; the minimum is 0. If `log_output = FILE`, you can specify a floating point value that goes to microsecond resolution. If `log_output = TABLE`, you must specify an integer value with second resolution. Only queries whose run time exceeds the `long_query_time` value are logged. For example, setting `long_query_time` to 0.1 prevents any query that runs for less than 100 milliseconds from being logged.
- `log_queries_not_using_indexes`: To log all queries that do not use an index to the slow query log, set this parameter to 1. The default is 0. Queries that do not use an index are logged even if their run time is less than the value of the `long_query_time` parameter.
- `log_output option`: You can specify one of the following options for the `log_output` parameter:
 - **TABLE** (default)– Write general queries to the `mysql.general_log` table, and slow queries to the `mysql.slow_log` table.
 - **FILE**– Write both general and slow query logs to the file system. Log files are rotated hourly.

- **NONE**—Disable logging.

When logging is enabled, Amazon RDS rotates table logs or deletes log files at regular intervals. This measure is a precaution to reduce the possibility of a large log file either blocking database use or affecting performance. **FILE** and **TABLE** logging approach rotation and deletion as follows:

- When **FILE** logging is enabled, log files are examined every hour and log files older than 24 hours are deleted. In some cases, the remaining combined log file size after the deletion might exceed the threshold of 2 percent of a DB instance's allocated space. In these cases, the largest log files are deleted until the log file size no longer exceeds the threshold.
- When **TABLE** logging is enabled, in some cases log tables are rotated every 24 hours. This rotation occurs if the space used by the table logs is more than 20 percent of the allocated storage space or the size of all logs combined is greater than 10 GB. If the amount of space used for a DB instance is greater than 90 percent of the DB instance's allocated storage space, then the thresholds for log rotation are reduced. Log tables are then rotated if the space used by the table logs is more than 10 percent of the allocated storage space or the size of all logs combined is greater than 5 GB.

When log tables are rotated, the current log table is copied to a backup log table and the entries in the current log table are removed. If the backup log table already exists, then it is deleted before the current log table is copied to the backup. You can query the backup log table if needed. The backup log table for the `mysql.general_log` table is named `mysql.general_log_backup`. The backup log table for the `mysql.slow_log` table is named `mysql.slow_log_backup`.

You can rotate the `mysql.general_log` table by calling the `mysql.rds_rotate_general_log` procedure. You can rotate the `mysql.slow_log` table by calling the `mysql.rds_rotate_slow_log` procedure.

Table logs are rotated during a database version upgrade.

Amazon RDS records both **TABLE** and **FILE** log rotation in an Amazon RDS event and sends you a notification.

To work with the logs from the Amazon RDS console, Amazon RDS API, Amazon RDS CLI, or AWS SDKs, set the `log_output` parameter to **FILE**. Like the MariaDB error log, these log files are rotated hourly. The log files that were generated during the previous 24 hours are retained.

For more information about the slow query and general logs, go to the following topics in the MariaDB documentation:

- [Slow query log](#)
- [General query log](#)

Publishing MariaDB logs to Amazon CloudWatch Logs

You can configure your MariaDB DB instance to publish log data to a log group in Amazon CloudWatch Logs. With CloudWatch Logs, you can perform real-time analysis of the log data, and use CloudWatch to create alarms and view metrics. You can use CloudWatch Logs to store your log records in highly durable storage.

Amazon RDS publishes each MariaDB database log as a separate database stream in the log group. For example, if you configure the export function to include the slow query log, slow query data is stored in a slow query log stream in the `/aws/rds/instance/my_instance/slowquery` log group.

The error log is enabled by default. The following table summarizes the requirements for the other MariaDB logs.

Log	Requirement
Audit log	The DB instance must use a custom option group with the MARIADB_AUDIT_PLUGIN option.
General log	The DB instance must use a custom parameter group with the parameter setting general_log = 1 to enable the general log.
Slow query log	The DB instance must use a custom parameter group with the parameter setting slow_query_log = 1 to enable the slow query log.
Log output	The DB instance must use a custom parameter group with the parameter setting log_output = FILE to write logs to the file system and publish them to CloudWatch Logs.

Console

To publish MariaDB logs to CloudWatch Logs from the console

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
3. Choose **Modify**.
4. In the **Log exports** section, choose the logs that you want to start publishing to CloudWatch Logs.
5. Choose **Continue**, and then choose **Modify DB Instance** on the summary page.

AWS CLI

You can publish a MariaDB logs with the AWS CLI. You can call the `modify-db-instance` command with the following parameters:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

A change to the `--cloudwatch-logs-export-configuration` option is always applied to the DB instance immediately. Therefore, the `--apply-immediately` and `--no-apply-immediately` options have no effect.

You can also publish MariaDB logs by calling the following AWS CLI commands:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Run one of these AWS CLI commands with the following options:

- `--db-instance-identifier`

- --enable-cloudwatch-logs-exports
- --db-instance-class
- --engine

Other options might be required depending on the AWS CLI command you run.

Example

The following example modifies an existing MariaDB DB instance to publish log files to CloudWatch Logs. The --cloudwatch-logs-export-configuration value is a JSON object. The key for this object is EnableLogTypes, and its value is an array of strings with any combination of audit, error, general, and slowquery.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
  --db-instance-identifier mydbinstance \
  --cloudwatch-logs-export-configuration '{"EnableLogTypes": \
  ["audit", "error", "general", "slowquery"]}'
```

For Windows:

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --cloudwatch-logs-export-configuration '{"EnableLogTypes": \
  ["audit", "error", "general", "slowquery"]}'
```

Example

The following command creates a MariaDB DB instance and publishes log files to CloudWatch Logs. The --enable-cloudwatch-logs-exports value is a JSON array of strings. The strings can be any combination of audit, error, general, and slowquery.

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
  --db-instance-identifier mydbinstance \
  --enable-cloudwatch-logs-exports '[{"audit", "error", "general", "slowquery"}]' \
  --db-instance-class db.m4.large \
  --engine mariadb
```

For Windows:

```
aws rds create-db-instance ^
  --db-instance-identifier mydbinstance ^
  --enable-cloudwatch-logs-exports '[{"audit", "error", "general", "slowquery"}]' ^
  --db-instance-class db.m4.large ^
  --engine mariadb
```

RDS API

You can publish MariaDB logs with the RDS API. You can call the [ModifyDBInstance](#) operation with the following parameters:

- [DBInstanceIdentifier](#)
- [CloudwatchLogsExportConfiguration](#)

Note

A change to the [CloudwatchLogsExportConfiguration](#) parameter is always applied to the DB instance immediately. Therefore, the [ApplyImmediately](#) parameter has no effect.

You can also publish MariaDB logs by calling the following RDS API operations:

- [CreateDBInstance](#)
- [RestoreDBInstanceFromDBSnapshot](#)
- [RestoreDBInstanceFromS3](#)
- [RestoreDBInstanceToPointInTime](#)

Run one of these RDS API operations with the following parameters:

- [DBInstanceIdentifier](#)
- [EnableCloudwatchLogsExports](#)
- [Engine](#)
- [DBInstanceClass](#)

Other parameters might be required depending on the AWS CLI command you run.

Log file size

The MariaDB slow query log, error log, and the general log file sizes are constrained to no more than 2 percent of the allocated storage space for a DB instance. To maintain this threshold, logs are automatically rotated every hour and log files older than 24 hours are removed. If the combined log file size exceeds the threshold after removing old log files, then the largest log files are deleted until the log file size no longer exceeds the threshold.

Managing table-based MariaDB logs

You can direct the general and slow query logs to tables on the DB instance by creating a DB parameter group and setting the `log_output` server parameter to `TABLE`. General queries are then logged to the `mysql.general_log` table, and slow queries are logged to the `mysql.slow_log` table. You can query the tables to access the log information. Enabling this logging increases the amount of data written to the database, which can degrade performance.

Both the general log and the slow query logs are disabled by default. In order to enable logging to tables, you must also set the `general_log` and `slow_query_log` server parameters to 1.

Log tables keep growing until the respective logging activities are turned off by resetting the appropriate parameter to 0. A large amount of data often accumulates over time, which can use up a considerable percentage of your allocated storage space. Amazon RDS does not allow you to truncate the log tables, but you can move their contents. Rotating a table saves its contents to a backup table and then creates a new empty log table. You can manually rotate the log tables with the following command line procedures, where the command prompt is indicated by `PROMPT>`:

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

To completely remove the old data and reclaim the disk space, call the appropriate procedure twice in succession.

Binary logging format

MariaDB on Amazon RDS supports the *row-based*, *statement-based*, and *mixed* binary logging formats. The default binary logging format is *mixed*. For details on the different MariaDB binary log formats, see [Binary log formats](#) in the MariaDB documentation.

If you plan to use replication, the binary logging format is important because it determines the record of data changes that is recorded in the source and sent to the replication targets. For information about the advantages and disadvantages of different binary logging formats for replication, see [Advantages and disadvantages of statement-based and row-based replication](#) in the MySQL documentation.

Important

Setting the binary logging format to row-based can result in very large binary log files. Large binary log files reduce the amount of storage available for a DB instance and can increase the amount of time to perform a restore operation of a DB instance.

Statement-based replication can cause inconsistencies between the source DB instance and a read replica. For more information, see [Unsafe statements for statement-based replication](#) in the MariaDB documentation.

To set the MariaDB binary logging format

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. Choose the parameter group that is used by the DB instance that you want to modify.

You can't modify a default parameter group. If the DB instance is using a default parameter group, create a new parameter group and associate it with the DB instance.

For more information on DB parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

4. For **Parameter group actions**, choose **Edit**.
5. Set the `binlog_format` parameter to the binary logging format of your choice (**ROW**, **STATEMENT**, or **MIXED**).
6. Choose **Save changes** to save the updates to the DB parameter group.

Accessing MariaDB binary logs

You can use the `mysqlbinlog` utility to download binary logs in text format from MariaDB DB instances. The binary log is downloaded to your local computer. For more information about using the `mysqlbinlog` utility, go to [Using mysqlbinlog](#) in the MariaDB documentation.

To run the `mysqlbinlog` utility against an Amazon RDS instance, use the following options:

- Specify the `--read-from-remote-server` option.
- `--host`: Specify the DNS name from the endpoint of the instance.
- `--port`: Specify the port used by the instance.
- `--user`: Specify a MariaDB user that has been granted the replication slave permission.
- `--password`: Specify the password for the user, or omit a password value so the utility prompts you for a password.
- `--result-file`: Specify the local file that receives the output.
- Specify the names of one or more binary log files. To get a list of the available logs, use the SQL command `SHOW BINARY LOGS`.

For more information about mysqlbinlog options, go to [mysqlbinlog options](#) in the MariaDB documentation.

The following is an example:

For Linux, macOS, or Unix:

```
mysqlbinlog \
--read-from-remote-server \
--host=mariadbinstance1.1234abcd.region.rds.amazonaws.com \
--port=3306 \
--user ReplUser \
--password <password> \
--result-file=/tmp/binlog.txt
```

For Windows:

```
mysqlbinlog ^
--read-from-remote-server ^
--host=mariadbinstance1.1234abcd.region.rds.amazonaws.com ^
--port=3306 ^
--user ReplUser ^
--password <password> ^
--result-file=/tmp/binlog.txt
```

Amazon RDS normally purges a binary log as soon as possible, but the binary log must still be available on the instance to be accessed by mysqlbinlog. To specify the number of hours for RDS to retain binary logs, use the `mysql.rds_set_configuration` stored procedure and specify a period with enough time for you to download the logs. After you set the retention period, monitor storage usage for the DB instance to ensure that the retained binary logs do not take up too much storage.

The following example sets the retention period to 1 day:

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

To display the current setting, use the `mysql.rds_show_configuration` stored procedure:

```
call mysql.rds_show_configuration;
```

Binary log annotation

In a MariaDB DB instance, you can use the `Annotate_rows` event to annotate a row event with a copy of the SQL query that caused the row event. This approach provides similar functionality to enabling the `binlog_rows_query_log_events` parameter on a DB instance on MySQL version 5.6 or later.

You can enable binary log annotations globally by creating a custom parameter group and setting the `binlog_annotation_row_events` parameter to 1. You can also enable annotations at the session level, by calling `SET SESSION binlog_annotation_row_events = 1`. Use the `replicate_annotation_row_events` to replicate binary log annotations to the slave instance if binary logging is enabled on it. No special privileges are required to use these settings.

The following is an example of a row-based transaction in MariaDB. The use of row-based logging is triggered by setting the transaction isolation level to `read-committed`.

```
CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE square(x INT PRIMARY KEY, y INT NOT NULL) ENGINE = InnoDB;
```

```
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
BEGIN
INSERT INTO square(x, y) VALUES(5, 5 * 5);
COMMIT;
```

Without annotations, the binary log entries for the transaction look like the following:

```
BEGIN
/*!*/
# at 1163
# at 1209
#150922 7:55:57 server id 1855786460 end_log_pos 1209      Table_map: `test`.`square`
mapped to number 76
#150922 7:55:57 server id 1855786460 end_log_pos 1247      Write_rows: table id 76
flags: STMT_END_F
### INSERT INTO `test`.`square`
### SET
### @1=5
### @2=25
# at 1247
#150922 7:56:01 server id 1855786460 end_log_pos 1274      Xid = 62
COMMIT/*!*/;
```

The following statement enables session-level annotations for this same transaction, and disables them after committing the transaction:

```
CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE square(x INT PRIMARY KEY, y INT NOT NULL) ENGINE = InnoDB;
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET SESSION binlog_annotation_row_events = 1;
BEGIN;
INSERT INTO square(x, y) VALUES(5, 5 * 5);
COMMIT;
SET SESSION binlog_annotation_row_events = 0;
```

With annotations, the binary log entries for the transaction look like the following:

```
BEGIN
/*!*/
# at 423
# at 483
# at 529
#150922 8:04:24 server id 1855786460 end_log_pos 483 Annotate_rows:
#Q> INSERT INTO square(x, y) VALUES(5, 5 * 5)
#150922 8:04:24 server id 1855786460 end_log_pos 529 Table_map: `test`.`square` mapped
to number 76
#150922 8:04:24 server id 1855786460 end_log_pos 567 Write_rows: table id 76 flags:
STMT_END_F
### INSERT INTO `test`.`square`
### SET
### @1=5
### @2=25
# at 567
#150922 8:04:26 server id 1855786460 end_log_pos 594 Xid = 88
COMMIT/*!*/;
```

Microsoft SQL Server database log files

You can access Microsoft SQL Server error logs, agent logs, trace files, and dump files by using the Amazon RDS console, AWS CLI, or RDS API. For more information about viewing, downloading, and watching file-based database logs, see [Accessing Amazon RDS database log files \(p. 504\)](#).

Retention schedule

Log files are rotated each day and whenever your DB instance is restarted. The following is the retention schedule for Microsoft SQL Server logs on Amazon RDS.

Log type	Retention schedule
Error logs	A maximum of 30 error logs are retained. Amazon RDS might delete error logs older than 7 days.
Agent logs	A maximum of 10 agent logs are retained. Amazon RDS might delete agent logs older than 7 days.
Trace files	Trace files are retained according to the trace file retention period of your DB instance. The default trace file retention period is 7 days. To modify the trace file retention period for your DB instance, see Setting the retention period for trace and dump files (p. 824) .
Dump files	Dump files are retained according to the dump file retention period of your DB instance. The default dump file retention period is 7 days. To modify the dump file retention period for your DB instance, see Setting the retention period for trace and dump files (p. 824) .

Viewing the SQL Server error log by using the rds_read_error_log procedure

You can use the Amazon RDS stored procedure `rds_read_error_log` to view error logs and agent logs. For more information, see [Viewing error and agent logs \(p. 824\)](#).

Publishing SQL Server logs to Amazon CloudWatch Logs

With Amazon RDS for SQL Server, you can publish error and agent log events directly to Amazon CloudWatch Logs. Analyze the log data with CloudWatch Logs, then use CloudWatch to create alarms and view metrics.

With CloudWatch Logs, you can do the following:

- Store logs in highly durable storage space with a retention period that you define.
- Search and filter log data.
- Share log data between accounts.
- Export logs to Amazon S3.
- Stream data to Amazon Elasticsearch Service.
- Process log data in real time with Amazon Kinesis Data Streams.

Amazon RDS publishes each SQL Server database log as a separate database stream in the log group. For example, if you publish error logs, error data is stored in an error log stream in the `/aws/rds/instance/my_instance/error` log group.

Note

Publishing SQL Server logs to CloudWatch Logs isn't enabled by default. Publishing trace and dump files isn't supported. Publishing SQL Server logs to CloudWatch Logs is supported in all regions, except for Asia Pacific (Hong Kong).

Console

To publish SQL Server DB logs to CloudWatch Logs from the AWS Management Console

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
3. Choose **Modify**.
4. In the **Log exports** section, choose the logs that you want to start publishing to CloudWatch Logs.
You can choose **Agent log**, **Error log**, or both.
5. Choose **Continue**, and then choose **Modify DB Instance** on the summary page.

AWS CLI

To publish SQL Server logs, you can use the `modify-db-instance` command with the following parameters:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

A change to the `--cloudwatch-logs-export-configuration` option is always applied to the DB instance immediately. Therefore, the `--apply-immediately` and `--no-apply-immediately` options have no effect.

You can also publish SQL Server logs using the following commands:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

Example

The following example creates an SQL Server DB instance with CloudWatch Logs publishing enabled. The `--enable-cloudwatch-logs-exports` value is a JSON array of strings that can include `error`, `agent`, or both.

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
--db-instance-identifier mydbinstance \
--enable-cloudwatch-logs-exports '[{"error", "agent"}]' \
--db-instance-class db.m4.large \
--engine sqlserver-se
```

For Windows:

```
aws rds create-db-instance ^
```

```
--db-instance-identifier mydbinstance ^
--enable-cloudwatch-logs-exports "[\"error\", \"agent\"]" ^
--db-instance-class db.m4.large ^
--engine sqlserver-se
```

Note

When using the Windows command prompt, you must escape double quotes ("") in JSON code by prefixing them with a backslash (\).

Example

The following example modifies an existing SQL Server DB instance to publish log files to CloudWatch Logs. The --cloudwatch-logs-export-configuration value is a JSON object. The key for this object is `EnableLogTypes`, and its value is an array of strings that can include `error`, `agent`, or both.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--cloudwatch-logs-export-configuration '{"EnableLogTypes": ["error", "agent"]}'
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--cloudwatch-logs-export-configuration "{\"EnableLogTypes\": [\"error\", \"agent\"]}"
```

Note

When using the Windows command prompt, you must escape double quotes ("") in JSON code by prefixing them with a backslash (\).

Example

The following example modifies an existing SQL Server DB instance to disable publishing agent log files to CloudWatch Logs. The --cloudwatch-logs-export-configuration value is a JSON object. The key for this object is `DisableLogTypes`, and its value is an array of strings that can include `error`, `agent`, or both.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--cloudwatch-logs-export-configuration '{"DisableLogTypes": ["agent"]}'
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--cloudwatch-logs-export-configuration "{\"DisableLogTypes\": [\"agent\"]}"
```

Note

When using the Windows command prompt, you must escape double quotes ("") in JSON code by prefixing them with a backslash (\).

Accessing MySQL database log files

You can monitor the MySQL logs directly through the Amazon RDS console, Amazon RDS API, AWS CLI, or AWS SDKs. You can also access MySQL logs by directing the logs to a database table in the main database and querying that table. You can use the `mysqlbinlog` utility to download a binary log.

For more information about viewing, downloading, and watching file-based database logs, see [Accessing Amazon RDS database log files \(p. 504\)](#).

Topics

- [Overview of MySQL database logs \(p. 519\)](#)
- [Accessing MySQL error logs \(p. 519\)](#)
- [Accessing the MySQL slow query and general logs \(p. 520\)](#)
- [Accessing the MySQL audit log \(p. 521\)](#)
- [Publishing MySQL logs to Amazon CloudWatch Logs \(p. 521\)](#)
- [Managing table-based MySQL logs \(p. 524\)](#)
- [Setting the binary logging format \(p. 524\)](#)
- [Accessing MySQL binary logs \(p. 525\)](#)

Overview of MySQL database logs

You can monitor the following types of MySQL log files:

- Error log
- Slow query log
- General log

The MySQL error log is generated by default. You can generate the slow query and general logs by setting parameters in your DB parameter group.

Log rotation and retention

The MySQL slow query log, error log, and the general log file sizes are constrained to no more than 2 percent of the allocated storage space for a DB instance. To maintain this threshold, logs are automatically rotated every hour. MySQL removes log files more than two weeks old. If the combined log file size exceeds the threshold after removing old log files, then the oldest log files are deleted until the log file size no longer exceeds the threshold.

Size limits on BLOBs

For MySQL, there is a size limit on BLOBs written to the redo log. To account for this limit, ensure that the `innodb_log_file_size` parameter for your MySQL DB instance is 10 times larger than the largest BLOB data size found in your tables, plus the length of other variable length fields (`VARCHAR`, `VARBINARY`, `TEXT`) in the same tables. For information on how to set parameter values, see [Working with DB parameter groups \(p. 228\)](#). For information on the redo log BLOB size limit, go to [Changes in MySQL 5.6.20](#).

Accessing MySQL error logs

MySQL writes errors in the `mysql-error.log` file. Each log file has the hour it was generated (in UTC) appended to its name. The log files also have a timestamp that helps you determine when the log entries were written.

MySQL writes to the error log only on startup, shutdown, and when it encounters errors. A DB instance can go hours or days without new entries being written to the error log. If you see no recent entries, it's because the server did not encounter an error that would result in a log entry.

MySQL writes `mysql-error.log` to disk every 5 minutes. MySQL appends the contents of the log to `mysql-error-running.log`.

MySQL rotates the `mysql-error-running.log` file every hour. RDS for MySQL retains the logs generated during the last two weeks.

Note

The log retention period is different between Amazon RDS and Aurora.

Accessing the MySQL slow query and general logs

The MySQL slow query log and the general log can be written to a file or a database table by setting parameters in your DB parameter group. For information about creating and modifying a DB parameter group, see [Working with DB parameter groups \(p. 228\)](#). You must set these parameters before you can view the slow query log or general log in the Amazon RDS console or by using the Amazon RDS API, Amazon RDS CLI, or AWS SDKs.

You can control MySQL logging by using the parameters in this list:

- `slow_query_log`: To create the slow query log, set to 1. The default is 0.
- `general_log`: To create the general log, set to 1. The default is 0.
- `long_query_time`: To prevent fast-running queries from being logged in the slow query log, specify a value for the shortest query run time to be logged, in seconds. The default is 10 seconds; the minimum is 0. If `log_output = FILE`, you can specify a floating point value that goes to microsecond resolution. If `log_output = TABLE`, you must specify an integer value with second resolution. Only queries whose run time exceeds the `long_query_time` value are logged. For example, setting `long_query_time` to 0.1 prevents any query that runs for less than 100 milliseconds from being logged.
- `log_queries_not_using_indexes`: To log all queries that do not use an index to the slow query log, set to 1. The default is 0. Queries that do not use an index are logged even if their run time is less than the value of the `long_query_time` parameter.
- `log_output option`: You can specify one of the following options for the `log_output` parameter.
 - **TABLE** (default) – Write general queries to the `mysql.general_log` table, and slow queries to the `mysql.slow_log` table.
 - **FILE** – Write both general and slow query logs to the file system. Log files are rotated hourly.
 - **NONE** – Disable logging.

When logging is enabled, Amazon RDS rotates table logs or deletes log files at regular intervals. This measure is a precaution to reduce the possibility of a large log file either blocking database use or affecting performance. `FILE` and `TABLE` logging approach rotation and deletion as follows:

- When `FILE` logging is enabled, log files are examined every hour and log files more than two weeks old are deleted. In some cases, the remaining combined log file size after the deletion might exceed the threshold of 2 percent of a DB instance's allocated space. In these cases, the largest log files are deleted until the log file size no longer exceeds the threshold.
- When `TABLE` logging is enabled, in some cases log tables are rotated every 24 hours. This rotation occurs if the space used by the table logs is more than 20 percent of the allocated storage space or the size of all logs combined is greater than 10 GB. If the amount of space used for a DB instance is greater than 90 percent of the DB instance's allocated storage space, then the thresholds for log rotation are reduced. Log tables are then rotated if the space used by the table logs is more than 10 percent of

the allocated storage space or the size of all logs combined is greater than 5 GB. You can subscribe to the `low_free_storage` event to be notified when log tables are rotated to free up space. For more information, see [Using Amazon RDS event notification \(p. 487\)](#).

When log tables are rotated, the current log table is copied to a backup log table and the entries in the current log table are removed. If the backup log table already exists, then it is deleted before the current log table is copied to the backup. You can query the backup log table if needed. The backup log table for the `mysql.general_log` table is named `mysql.general_log_backup`. The backup log table for the `mysql.slow_log` table is named `mysql.slow_log_backup`.

You can rotate the `mysql.general_log` table by calling the `mysql.rds_rotate_general_log` procedure. You can rotate the `mysql.slow_log` table by calling the `mysql.rds_rotate_slow_log` procedure.

Table logs are rotated during a database version upgrade.

To work with the logs from the Amazon RDS console, Amazon RDS API, Amazon RDS CLI, or AWS SDKs, set the `log_output` parameter to `FILE`. Like the MySQL error log, these log files are rotated hourly. The log files that were generated during the previous two weeks are retained. Note that the retention period is different between Amazon RDS and Aurora.

For more information about the slow query and general logs, go to the following topics in the MySQL documentation:

- [The slow query log](#)
- [The general query log](#)

Accessing the MySQL audit log

To access the audit log, the DB instance must use a custom option group with the `MARIADB_AUDIT_PLUGIN` option. For more information, see [MariaDB Audit Plugin support \(p. 926\)](#).

Publishing MySQL logs to Amazon CloudWatch Logs

You can configure your MySQL DB instance to publish log data to a log group in Amazon CloudWatch Logs. With CloudWatch Logs, you can perform real-time analysis of the log data, and use CloudWatch to create alarms and view metrics. You can use CloudWatch Logs to store your log records in highly durable storage.

Amazon RDS publishes each MySQL database log as a separate database stream in the log group. For example, if you configure the export function to include the slow query log, slow query data is stored in a slow query log stream in the `/aws/rds/instance/my_instance/slowquery` log group.

The error log is enabled by default. The following table summarizes the requirements for the other MySQL logs.

Log	Requirement
Audit log	The DB instance must use a custom option group with the <code>MARIADB_AUDIT_PLUGIN</code> option.
General log	The DB instance must use a custom parameter group with the parameter setting <code>general_log = 1</code> to enable the general log.

Log	Requirement
Slow query log	The DB instance must use a custom parameter group with the parameter setting <code>slow_query_log = 1</code> to enable the slow query log.
Log output	The DB instance must use a custom parameter group with the parameter setting <code>log_output = FILE</code> to write logs to the file system and publish them to CloudWatch Logs.

Note

Publishing log files to CloudWatch Logs is only supported for MySQL versions 5.6, 5.7, and 8.0.

Console

To publish MySQL logs to CloudWatch Logs using the console

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
3. Choose **Modify**.
4. In the **Log exports** section, choose the logs that you want to start publishing to CloudWatch Logs.
5. Choose **Continue**, and then choose **Modify DB Instance** on the summary page.

AWS CLI

You can publish MySQL logs with the AWS CLI. You can call the `modify-db-instance` command with the following parameters:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

A change to the `--cloudwatch-logs-export-configuration` option is always applied to the DB instance immediately. Therefore, the `--apply-immediately` and `--no-apply-immediately` options have no effect.

You can also publish MySQL logs by calling the following AWS CLI commands:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Run one of these AWS CLI commands with the following options:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`

- `--engine`

Other options might be required depending on the AWS CLI command you run.

Example

The following example modifies an existing MySQL DB instance to publish log files to CloudWatch Logs. The `--cloudwatch-logs-export-configuration` value is a JSON object. The key for this object is `EnableLogTypes`, and its value is an array of strings with any combination of `audit`, `error`, `general`, and `slowquery`.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes": \
    ["audit", "error", "general", "slowquery"]}'
```

For Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes": \
    ["audit", "error", "general", "slowquery"]}'
```

Example

The following example creates a MySQL DB instance and publishes log files to CloudWatch Logs. The `--enable-cloudwatch-logs-exports` value is a JSON array of strings. The strings can be any combination of `audit`, `error`, `general`, and `slowquery`.

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '[{"audit", "error", "general", "slowquery"}]' \
    --db-instance-class db.m4.large \
    --engine MySQL
```

For Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '[{"audit", "error", "general", "slowquery"}]' ^
    --db-instance-class db.m4.large ^
    --engine MySQL
```

RDS API

You can publish MySQL logs with the RDS API. You can call the `ModifyDBInstance` action with the following parameters:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

Note

A change to the `CloudwatchLogsExportConfiguration` parameter is always applied to the DB instance immediately. Therefore, the `ApplyImmediately` parameter has no effect.

You can also publish MySQL logs by calling the following RDS API operations:

- [CreateDBInstance](#)
- [RestoreDBInstanceFromDBSnapshot](#)
- [RestoreDBInstanceFromS3](#)
- [RestoreDBInstanceToPointInTime](#)

Run one of these RDS API operations with the following parameters:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Other parameters might be required depending on the AWS CLI command you run.

Managing table-based MySQL logs

You can direct the general and slow query logs to tables on the DB instance by creating a DB parameter group and setting the `log_output` server parameter to `TABLE`. General queries are then logged to the `mysql.general_log` table, and slow queries are logged to the `mysql.slow_log` table. You can query the tables to access the log information. Enabling this logging increases the amount of data written to the database, which can degrade performance.

Both the general log and the slow query logs are disabled by default. In order to enable logging to tables, you must also set the `general_log` and `slow_query_log` server parameters to 1.

Log tables keep growing until the respective logging activities are turned off by resetting the appropriate parameter to 0. A large amount of data often accumulates over time, which can use up a considerable percentage of your allocated storage space. Amazon RDS does not allow you to truncate the log tables, but you can move their contents. Rotating a table saves its contents to a backup table and then creates a new empty log table. You can manually rotate the log tables with the following command line procedures, where the command prompt is indicated by `PROMPT>`:

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

To completely remove the old data and reclaim the disk space, call the appropriate procedure twice in succession.

Setting the binary logging format

MySQL on Amazon RDS supports the *row-based*, *statement-based*, and *mixed* binary logging formats for MySQL version 5.6 and later. The default binary logging format is mixed. For DB instances running MySQL versions 5.1 and 5.5, only mixed binary logging is supported. For details on the different MySQL binary log formats, see [Binary logging formats](#) in the MySQL documentation.

If you plan to use replication, the binary logging format is important because it determines the record of data changes that is recorded in the source and sent to the replication targets. For information about the advantages and disadvantages of different binary logging formats for replication, see [Advantages and disadvantages of statement-based and row-based replication](#) in the MySQL documentation.

Important

Setting the binary logging format to row-based can result in very large binary log files. Large binary log files reduce the amount of storage available for a DB instance and can increase the amount of time to perform a restore operation of a DB instance.

Statement-based replication can cause inconsistencies between the source DB instance and a read replica. For more information, see [Determination of safe and unsafe statements in binary logging](#) in the MySQL documentation.

To set the MySQL binary logging format

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. Choose the parameter group used by the DB instance you want to modify.

You can't modify a default parameter group. If the DB instance is using a default parameter group, create a new parameter group and associate it with the DB instance.

For more information on parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

4. From **Parameter group actions**, choose **Edit**.
5. Set the `binlog_format` parameter to the binary logging format of your choice (**ROW**, **STATEMENT**, or **MIXED**).
6. Choose **Save changes** to save the updates to the DB parameter group.

Important

Changing a DB parameter group affects all DB instances that use that parameter group. If you want to specify different binary logging formats for different MySQL DB instances in an AWS Region, the DB instances must use different DB parameter groups. These parameter groups identify different logging formats. Assign the appropriate DB parameter group to the each DB instance.

Accessing MySQL binary logs

You can use the `mysqlbinlog` utility to download or stream binary logs from Amazon RDS instances running MySQL 5.6 or later. The binary log is downloaded to your local computer, where you can perform actions such as replaying the log using the `mysql` utility. For more information about using the `mysqlbinlog` utility, go to [Using mysqlbinlog to back up binary log files](#).

To run the `mysqlbinlog` utility against an Amazon RDS instance, use the following options:

- Specify the `--read-from-remote-server` option.
- `--host`: Specify the DNS name from the endpoint of the instance.
- `--port`: Specify the port used by the instance.
- `--user`: Specify a MySQL user that has been granted the replication slave permission.
- `--password`: Specify the password for the user, or omit a password value so that the utility prompts you for a password.
- To have the file downloaded in binary format, specify the `--raw` option.
- `--result-file`: Specify the local file to receive the raw output.
- Specify the names of one or more binary log files. To get a list of the available logs, use the SQL command `SHOW BINARY LOGS`.
- To stream the binary log files, specify the `--stop-never` option.

For more information about `mysqlbinlog` options, go to [mysqlbinlog - utility for processing binary log files](#).

For example, see the following.

For Linux, macOS, or Unix:

```
mysqlbinlog \
--read-from-remote-server \
--host=MySQL56Instance1.cg034hpkmmt.region.rds.amazonaws.com \
--port=3306 \
--user ReplUser \
--password \
--raw \
--result-file=/tmp/ \
binlog.00098
```

For Windows:

```
mysqlbinlog ^
--read-from-remote-server ^
--host=MySQL56Instance1.cg034hpkmmt.region.rds.amazonaws.com ^
--port=3306 ^
--user ReplUser ^
--password ^
--raw ^
--result-file=/tmp/ ^
binlog.00098
```

Amazon RDS normally purges a binary log as soon as possible, but the binary log must still be available on the instance to be accessed by mysqlbinlog. To specify the number of hours for RDS to retain binary logs, use the `mysql.rds_set_configuration` stored procedure and specify a period with enough time for you to download the logs. After you set the retention period, monitor storage usage for the DB instance to ensure that the retained binary logs don't take up too much storage.

Note

The `mysql.rds_set_configuration` stored procedure is only available for MySQL version 5.6 or later.

The following example sets the retention period to 1 day.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

To display the current setting, use the `mysql.rds_show_configuration` stored procedure.

```
call mysql.rds_show_configuration;
```

Oracle database log files

You can access Oracle alert logs, audit files, and trace files by using the Amazon RDS console or API. For more information about viewing, downloading, and watching file-based database logs, see [Accessing Amazon RDS database log files \(p. 504\)](#).

The Oracle audit files provided are the standard Oracle auditing files. Amazon RDS supports the Oracle fine-grained auditing (FGA) feature. However, log access doesn't provide access to FGA events that are stored in the `SYS.FGA_LOG$` table and that are accessible through the `DBA_FGA_AUDIT_TRAIL` view.

The `DescribeDBLogFiles` API operation that lists the Oracle log files that are available for a DB instance ignores the `MaxRecords` parameter and returns up to 1,000 records. The call returns `LastWritten` as a POSIX date in milliseconds.

Retention schedule

The Oracle database engine might rotate log files if they get very large. To retain audit or trace files, download them. If you store the files locally, you reduce your Amazon RDS storage costs and make more space available for your data.

The following table shows the retention schedule for Oracle alert logs, audit files, and trace files on Amazon RDS.

Log type	Retention schedule
Alert logs	The text alert log is rotated daily with 30-day retention managed by Amazon RDS. The XML alert log is retained for at least seven days. You can access this log by using the <code>ALERTLOG</code> view.
Audit files	The default retention period for audit files is seven days. Amazon RDS might delete audit files older than seven days.
Trace files	The default retention period for trace files is seven days. Amazon RDS might delete trace files older than seven days.
Listener logs	The default retention period for the listener logs is seven days. Amazon RDS might delete listener logs older than seven days.

Note

Audit files and trace files share the same retention configuration.

Working with Oracle trace files

Following, you can find descriptions of Amazon RDS procedures to create, refresh, access, and delete trace files.

Listing files

You can use either of two procedures to allow access to any file in the `background_dump_dest` path. The first procedure refreshes a view containing a listing of all files currently in `background_dump_dest`.

```
EXEC rdsadmin.manage_tracefiles.refresh_tracefile_listing;
```

After the view is refreshed, query the following view to access the results.

```
SELECT * FROM rdsadmin.tracefile_listing;
```

An alternative to the previous process is to use `FROM table` to stream nonrelational data in a table-like format to list database directory contents.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('BDUMP'));
```

The following query shows the text of a log file.

```
SELECT text FROM
  TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP','alert_dbname.log.date'));
```

On a read replica, get the name of the BDUMP directory by querying `V$DATABASE.DB_UNIQUE_NAME`. If the unique name is `DATABASE_B`, then the BDUMP directory is `BDUMP_B`. The following example queries the BDUMP name on a replica and then uses this name to query the contents of `alert_DATABASE.log.2020-06-23`.

```
SELECT 'BDUMP' || (SELECT regexp_replace(DB_UNIQUE_NAME,'.*([A-Z])','\1') FROM V
$DATABASE) AS BDUMP_VARIABLE FROM DUAL;

BDUMP_VARIABLE
-----
BDUMP_B

SELECT TEXT FROM
  table(rdsadmin.rds_file_util.read_text_file('BDUMP_B','alert_DATABASE.log.2020-06-23'));
```

Generating trace files and tracing a session

Because there are no restrictions on `ALTER SESSION`, many standard methods to generate trace files in Oracle remain available to an Amazon RDS DB instance. The following procedures are provided for trace files that require greater access.

Oracle method	Amazon RDS method
<code>oradebug hanganalyze 3</code>	<code>EXEC rdsadmin.manage_tracefiles.hanganalyze;</code>
<code>oradebug dump systemstate 266</code>	<code>EXEC rdsadmin.manage_tracefiles.dump_systemstate;</code>

You can use many standard methods to trace individual sessions connected to an Oracle DB instance in Amazon RDS. To enable tracing for a session, you can run subprograms in PL/SQL packages supplied by Oracle, such as `DBMS_SESSION` and `DBMS_MONITOR`. For more information, see [Enabling tracing for a session](#) in the Oracle documentation.

Retrieving trace files

You can retrieve any trace file in `background_dump_dest` using a standard SQL query on an Amazon RDS-managed external table. To use this method, you must execute the procedure to set the location for this table to the specific trace file.

For example, you can use the `rdsadmin.tracefile_listing` view mentioned preceding to list all of the trace files on the system. You can then set the `tracefile_table` view to point to the intended trace file using the following procedure.

```
EXEC
rdsadmin.manage_tracefiles.set_tracefile_table_location('CUST01_ora_3260_SYSTEMSTATE.trc');
```

The following example creates an external table in the current schema with the location set to the file provided. You can retrieve the contents into a local file using a SQL query.

```
SPOOL /tmp/tracefile.txt
SELECT * FROM tracefile_table;
SPOOL OFF;
```

Purging trace files

Trace files can accumulate and consume disk space. Amazon RDS purges trace files by default and log files that are older than seven days. You can view and set the trace file retention period using the `show_configuration` procedure. You should run the command `SET SERVEROUTPUT ON` so that you can view the configuration results.

The following example shows the current trace file retention period, and then sets a new trace file retention period.

```
# Show the current tracefile retention
SQL> EXEC rdsadmin.rdsadmin_util.show_configuration;
NAME:tracefile retention
VALUE:10080
DESCRIPTION:tracefile expiration specifies the duration in minutes before tracefiles in
bdump are automatically deleted.

# Set the tracefile retention to 24 hours:
SQL> EXEC rdsadmin.rdsadmin_util.set_configuration('tracefile retention',1440);

#show the new tracefile retention
SQL> EXEC rdsadmin.rdsadmin_util.show_configuration;
NAME:tracefile retention
VALUE:1440
DESCRIPTION:tracefile expiration specifies the duration in minutes before tracefiles in
bdump are automatically deleted.
```

In addition to the periodic purge process, you can manually remove files from the `background_dump_dest`. The following example shows how to purge all files older than five minutes.

```
EXEC rdsadmin.manage_tracefiles.purge_tracefiles(5);
```

You can also purge all files that match a specific pattern (if you do, don't include the file extension, such as .trc). The following example shows how to purge all files that start with `SCHPOC1_ora_5935`.

```
EXEC rdsadmin.manage_tracefiles.purge_tracefiles('SCHPOC1_ora_5935');
```

Publishing Oracle logs to Amazon CloudWatch Logs

You can configure your Amazon RDS for Oracle DB instance to publish log data to a log group in Amazon CloudWatch Logs. With CloudWatch Logs, you can analyze the log data, and use CloudWatch to create alarms and view metrics. You can use CloudWatch Logs to store your log records in highly durable storage.

Amazon RDS publishes each Oracle database log as a separate database stream in the log group. For example, if you configure the export function to include the audit log, audit data is stored in an audit

log stream in the `/aws/rds/instance/my_instance/audit` log group. RDS for Oracle supports the following logs:

- Alert log
- Trace log
- Audit log
- Listener log
- Oracle Management Agent log

This Oracle Management Agent log consists of the log streams shown in the following table.

Log name	CloudWatch log stream
<code>emctl.log</code>	<code>oemagent-emctl</code>
<code>emdctlj.log</code>	<code>oemagent-emdctlj</code>
<code>gcagent.log</code>	<code>oemagent-gcagent</code>
<code>gcagent_errors.log</code>	<code>oemagent-gcagent-errors</code>
<code>emagent.nohup</code>	<code>oemagent-emagent-nohup</code>
<code>secure.log</code>	<code>oemagent-secure</code>

For more information, see [Locating Management Agent Log and Trace Files](#) in the Oracle documentation.

Console

To publish Oracle DB logs to CloudWatch Logs from the AWS Management Console

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
3. Choose **Modify**.
4. In the **Log exports** section, choose the logs that you want to start publishing to CloudWatch Logs.
5. Choose **Continue**, and then choose **Modify DB Instance** on the summary page.

AWS CLI

To publish Oracle logs, you can use the `modify-db-instance` command with the following parameters:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

A change to the `--cloudwatch-logs-export-configuration` option is always applied to the DB instance immediately. Therefore, the `--apply-immediately` and `--no-apply-immediately` options have no effect.

You can also publish Oracle logs using the following commands:

- `create-db-instance`

- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Example

The following example creates an Oracle DB instance with CloudWatch Logs publishing enabled. The `--cloudwatch-logs-export-configuration` value is a JSON array of strings. The strings can be any combination of alert, audit, listener, and trace.

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration
    '[["trace","audit","alert","listener","oemagent"]]' \
    --db-instance-class db.m5.large \
    --allocated-storage 20 \
    --engine oracle-ee \
    --engine-version 12.1.0.2.v18 \
    --license-model bring-your-own-license \
    --master-username myadmin \
    --master-user-password mypassword
```

For Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration trace alert audit listener oemagent ^
    --db-instance-class db.m5.large ^
    --allocated-storage 20 ^
    --engine oracle-ee ^
    --engine-version 12.1.0.2.v18 ^
    --license-model bring-your-own-license ^
    --master-username myadmin ^
    --master-user-password mypassword
```

Example

The following example modifies an existing Oracle DB instance to publish log files to CloudWatch Logs. The `--cloudwatch-logs-export-configuration` value is a JSON object. The key for this object is `EnableLogTypes`, and its value is an array of strings with any combination of alert, audit, listener, and trace.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":'
    ['trace','alert','audit','listener','oemagent']}'
```

For Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration EnableLogTypes=\"trace\","alert\","audit\",
    "listener\","oemagent\"
```

Example

The following example modifies an existing Oracle DB instance to disable publishing audit and listener log files to CloudWatch Logs. The --cloudwatch-logs-export-configuration value is a JSON object. The key for this object is DisableLogTypes, and its value is an array of strings with any combination of alert, audit, listener, and trace.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit","listener"]}'
```

For Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration DisableLogTypes=\"audit\",\"listener\"
```

RDS API

You can publish Oracle DB logs with the RDS API. You can call the [ModifyDBInstance](#) action with the following parameters:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

Note

A change to the `CloudwatchLogsExportConfiguration` parameter is always applied to the DB instance immediately. Therefore, the `ApplyImmediately` parameter has no effect.

You can also publish Oracle logs by calling the following RDS API operations:

- [CreateDBInstance](#)
- [RestoreDBInstanceFromDBSnapshot](#)
- [RestoreDBInstanceFromS3](#)
- [RestoreDBInstanceToPointInTime](#)

Run one of these RDS API operations with the following parameters:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Other parameters might be required depending on the RDS operation that you run.

Previous methods for accessing alert logs and listener logs

You can view the alert log using the Amazon RDS console. You can also use the following SQL statement to access the alert log.

```
SELECT message_text FROM alertlog;
```

The `listenerlog` view contains entries for Oracle Database version 12.1.0.2 and earlier. To access the listener log for these database versions, use the following query.

```
SELECT message_text FROM listenerlog;
```

For Oracle Database versions 12.2.0.1 and later, access the listener log using Amazon CloudWatch Logs.

Note

Oracle rotates the alert and listener logs when they exceed 10 MB, at which point they are unavailable from Amazon RDS views.

PostgreSQL database log files

Amazon RDS PostgreSQL generates query and error logs. You can use log messages to troubleshoot performance and auditing issues while using the database.

To view, download, and watch file-based database logs, see [Accessing Amazon RDS database log files \(p. 504\)](#).

Topics

- [Overview of PostgreSQL logs \(p. 534\)](#)
- [Setting the log retention period \(p. 535\)](#)
- [Setting the message format \(p. 535\)](#)
- [Enabling query logging \(p. 535\)](#)
- [Publishing PostgreSQL logs to Amazon CloudWatch Logs \(p. 537\)](#)

Overview of PostgreSQL logs

PostgreSQL generates event log files that contain useful information for DBAs.

Log contents

The default logging level captures errors that affect your server. By default, Amazon RDS PostgreSQL logging parameters capture all server errors, including the following:

- Query failures
- Login failures
- Fatal server errors
- Deadlocks

To identify application issues, you can use the preceding error messages. For example, if you converted a legacy application from Oracle to Amazon RDS PostgreSQL, some queries may not convert correctly. These incorrectly formatted queries generate error messages in the logs, which you can use to identify the problematic code.

You can modify PostgreSQL logging parameters to capture additional information, including the following:

- Connections and disconnections
- Checkpoints
- Schema modification queries
- Queries waiting for locks
- Queries consuming temporary disk storage
- Backend autovacuum process consuming resources

The preceding log information can help troubleshoot potential performance and auditing issues. For more information, see [Error reporting and logging](#) in the PostgreSQL documentation. For a useful AWS blog about PostgreSQL logging, see [Working with RDS and Aurora PostgreSQL logs: Part 1](#) and [Working with RDS and Aurora PostgreSQL logs: Part 2](#).

Parameter groups

Each Amazon RDS PostgreSQL instance is associated with a *parameter group* that contains the engine specific configurations. The engine configurations also include several parameters that control

PostgreSQL logging behavior. AWS provides the parameter groups with default configuration settings to use for your instances. However, to change the default settings, you must create a clone of the default parameter group, modify it, and attach it to your instance.

To set logging parameters for a DB instance, set the parameters in a DB parameter group and associate that parameter group with the DB instance. For more information, see [Working with DB parameter groups \(p. 228\)](#).

Setting the log retention period

To set the retention period for system logs, use the `rds.log_retention_period` parameter. You can find `rds.log_retention_period` in the DB parameter group associated with your DB instance. The unit for this parameter is minutes. For example, a setting of 1,440 retains logs for one day. The default value is 4,320 (three days). The maximum value is 10,080 (seven days). Your instance must have enough allocated storage to contain the retained log files.

To retain older logs, publish them to Amazon CloudWatch Logs. For more information, see [Publishing PostgreSQL logs to Amazon CloudWatch Logs \(p. 537\)](#).

Setting the message format

By default, Amazon RDS PostgreSQL generates logs in standard error (stderr) format. In this format, each log message is prefixed with the information specified by the parameter `log_line_prefix`. Amazon RDS only allows the following value for `log_line_prefix`:

```
%t:%r:%u@%d:[%p]:t
```

The preceding value maps to the following code:

```
log-time : remote-host : user-name @ db-name : [ process-id ]
```

For example, the following error message results from querying a column using the wrong name.

```
2019-03-10 03:54:59 UTC:10.0.0.123(52834):postgres@tstldb:[20175]:ERROR: column "wrong" does not exist at character 8
```

To specify the format for output logs, use the parameter `log_destination`. To make the instance generate both standard and CSV output files, set `log_destination` to `csvlog` in your instance parameter group. For a discussion of PostgreSQL logs, see [Working with RDS and Aurora PostgreSQL logs: Part 1](#).

Enabling query logging

To enable query logging for your PostgreSQL DB instance, set two parameters in the DB parameter group associated with your DB instance: `log_statement` and `log_min_duration_statement`.

The `log_statement` parameter controls which SQL statements are logged. The default value is `none`. We recommend that when you debug issues in your DB instance, set this parameter to `all` to log all statements. To log all data definition language (DDL) statements (CREATE, ALTER, DROP, and so on), set this value to `ddl`. To log all DDL and data modification language (DML) statements (INSERT, UPDATE, DELETE, and so on), set the value to `mod`.

Warning

Sensitive information such as passwords can be exposed if you set the `log_statement` parameter to `ddl`, `mod`, or `all`. To avoid this risk, set the `log_statement` to `none`. Also consider the following solutions:

- Encrypt the sensitive information on the client side and use the `ENCRYPTED` and `UNENCRYPTED` options of the `CREATE` and `ALTER` statements.
- Restrict access to the CloudWatch logs.
- Use stronger authentication mechanisms such as IAM.

For auditing, you can use the PostgreSQL pgAudit extension because it redacts the sensitive information for `CREATE` and `ALTER` commands.

The `log_min_duration_statement` parameter sets the limit in milliseconds of a statement to be logged. All SQL statements that run longer than the parameter setting are logged. This parameter is disabled and set to -1 by default. Enabling this parameter can help you find unoptimized queries.

To set up query logging, take the following steps:

1. Set the `log_statement` parameter to `all`. The following example shows the information that is written to the `postgres.log` file.

```
2013-11-05 16:48:56 UTC::@[2952]:LOG: received SIGHUP, reloading configuration files
2013-11-05 16:48:56 UTC::@[2952]:LOG: parameter "log_statement" changed to "all"
```

Additional information is written to the `postgres.log` file when you run a query. The following example shows the type of information written to the file after a query.

```
2013-11-05 16:41:07 UTC::@[2955]:LOG: checkpoint starting: time
2013-11-05 16:41:07 UTC::@[2955]:LOG: checkpoint complete: wrote 1 buffers (0.3%); 
0 transaction log file(s) added, 0 removed, 1 recycled; write=0.000 s, sync=0.003 s
total=0.012 s; sync files=1, longest=0.003 s, average=0.003 s
2013-11-05 16:45:14 UTC:[local]:master@postgres:[8839]:LOG: statement: SELECT d.datname
as "Name",
    pg_catalog.pg_get_userbyid(d.datdba) as "Owner",
    pg_catalog.pg_encoding_to_char(d.encoding) as "Encoding",
    d.datcollate as "Collate",
    d.datctype as "Ctype",
    pg_catalog.array_to_string(d.datacl, E'\n') AS "Access privileges"
FROM pg_catalog.pg_database d
ORDER BY 1;
2013-11-05 16:45:
```

2. Set the `log_min_duration_statement` parameter. The following example shows the information that is written to the `postgres.log` file when the parameter is set to 1.

```
2013-11-05 16:48:56 UTC::@[2952]:LOG: received SIGHUP, reloading configuration files
2013-11-05 16:48:56 UTC::@[2952]:LOG: parameter "log_min_duration_statement" changed to
"1"
```

Additional information is written to the `postgres.log` file when you run a query that exceeds the duration parameter setting. The following example shows the type of information written to the file after a query.

```
2013-11-05 16:51:10 UTC:[local]:master@postgres:[9193]:LOG: statement: SELECT
c2.relname, i.indisprimary, i.indisunique, i.indisclustered, i.indisvalid,
pg_catalog.pg_get_indexdef(i.indexrelid, 0, true),
    pg_catalog.pg_get_constraintdef(con.oid, true), contype, condeferrable, condeferred,
c2.reltablename
FROM pg_catalog.pg_class c, pg_catalog.pg_class c2, pg_catalog.pg_index i
    LEFT JOIN pg_catalog.pg_constraint con ON (conrelid = i.indrelid AND conindid =
i.indexrelid AND contype IN ('p','u','x'))
WHERE c.oid = '1255' AND c.oid = i.indrelid AND i.indexrelid = c2.oid
```

```
ORDER BY i.indisprimary DESC, i.indisunique DESC, c2.relname;
2013-11-05 16:51:10 UTC:[local]:master@postgres:[9193]:LOG: duration: 3.367 ms
2013-11-05 16:51:10 UTC:[local]:master@postgres:[9193]:LOG: statement: SELECT
c.oid::pg_catalog.regclass FROM pg_catalog.pg_class c, pg_catalog.pg_inherits i WHERE
c.oid=i.inhparent AND i.inhrelid = '1255' ORDER BY inhseqno;
2013-11-05 16:51:10 UTC:[local]:master@postgres:[9193]:LOG: duration: 1.002 ms
2013-11-05 16:51:10 UTC:[local]:master@postgres:[9193]:LOG: statement:
SELECT c.oid::pg_catalog.regclass FROM pg_catalog.pg_class c,
pg_catalog.pg_inherits i WHERE c.oid=i.inhrelid AND i.inhparent = '1255' ORDER BY
c.oid::pg_catalog.regclass::pg_catalog.text;
2013-11-05 16:51:18 UTC:[local]:master@postgres:[9193]:LOG: statement: select proname
from pg_proc;
2013-11-05 16:51:18 UTC:[local]:master@postgres:[9193]:LOG: duration: 3.469 ms
```

Publishing PostgreSQL logs to Amazon CloudWatch Logs

To store your PostgreSQL log records in highly durable storage, you can use Amazon CloudWatch Logs. With CloudWatch Logs, you can also perform real-time analysis of log data and use CloudWatch to view metrics and create alarms. For example, if you set `log_statements` to `ddl`, you can set up an alarm to alert whenever a DDL statement is executed.

To work with CloudWatch Logs, configure your RDS for PostgreSQL DB instance to publish log data to a log group.

Note

Publishing log files to CloudWatch Logs is supported only for PostgreSQL versions 9.6.6 and later and 10.4 and later.

You can publish the following log types to CloudWatch Logs for RDS for PostgreSQL:

- Postgresql log
- Upgrade log (not available for Aurora PostgreSQL)

After you complete the configuration, Amazon RDS publishes the log events to log streams within a CloudWatch log group. For example, the PostgreSQL log data is stored within the log group `/aws/rds/instance/my_instance/postgresql`. To view your logs, open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.

Console

To publish PostgreSQL logs to CloudWatch Logs using the console

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the DB instance that you want to modify, and then choose **Modify**.
4. In the **Log exports** section, choose the logs that you want to start publishing to CloudWatch Logs.

The **Log exports** section is available only for PostgreSQL versions that support publishing to CloudWatch Logs.

5. Choose **Continue**, and then choose **Modify DB Instance** on the summary page.

AWS CLI

You can publish PostgreSQL logs with the AWS CLI. You can call the `modify-db-instance` command with the following parameters:

- `--db-instance-identifier`

- `--cloudwatch-logs-export-configuration`

Note

A change to the `--cloudwatch-logs-export-configuration` option is always applied to the DB instance immediately. Therefore, the `--apply-immediately` and `--no-apply-immediately` options have no effect.

You can also publish PostgreSQL logs by calling the following CLI commands:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

Run one of these CLI commands with the following options:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Other options might be required depending on the CLI command you run.

Example Modify an instance to publish logs to CloudWatch Logs

The following example modifies an existing PostgreSQL DB instance to publish log files to CloudWatch Logs. The `--cloudwatch-logs-export-configuration` value is a JSON object. The key for this object is `EnableLogTypes`, and its value is an array of strings with any combination of `postgresql` and `upgrade`.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--cloudwatch-logs-export-configuration '{"EnableLogTypes":["postgresql", "upgrade"]}'
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--cloudwatch-logs-export-configuration '{"EnableLogTypes":["postgresql", "upgrade"]}'
```

Example Create an instance to publish logs to CloudWatch Logs

The following example creates a PostgreSQL DB instance and publishes log files to CloudWatch Logs. The `--enable-cloudwatch-logs-exports` value is a JSON array of strings. The strings can be any combination of `postgresql` and `upgrade`.

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
--db-instance-identifier mydbinstance \
--enable-cloudwatch-logs-exports '[ "postgresql", "upgrade" ]' \
--db-instance-class db.m4.large \
```

```
--engine postgres
```

For Windows:

```
aws rds create-db-instance ^
--db-instance-identifier mydbinstance ^
--enable-cloudwatch-logs-exports '[["postgresql","upgrade"]]' ^
--db-instance-class db.m4.large ^
--engine postgres
```

RDS API

You can publish PostgreSQL logs with the RDS API. You can call the [ModifyDBInstance](#) action with the following parameters:

- [DBInstanceIdentifier](#)
- [CloudwatchLogsExportConfiguration](#)

Note

A change to the [CloudwatchLogsExportConfiguration](#) parameter is always applied to the DB instance immediately. Therefore, the [ApplyImmediately](#) parameter has no effect.

You can also publish PostgreSQL logs by calling the following RDS API operations:

- [CreateDBInstance](#)
- [RestoreDBInstanceFromDBSnapshot](#)
- [RestoreDBInstanceToPointInTime](#)

Run one of these RDS API operations with the following parameters:

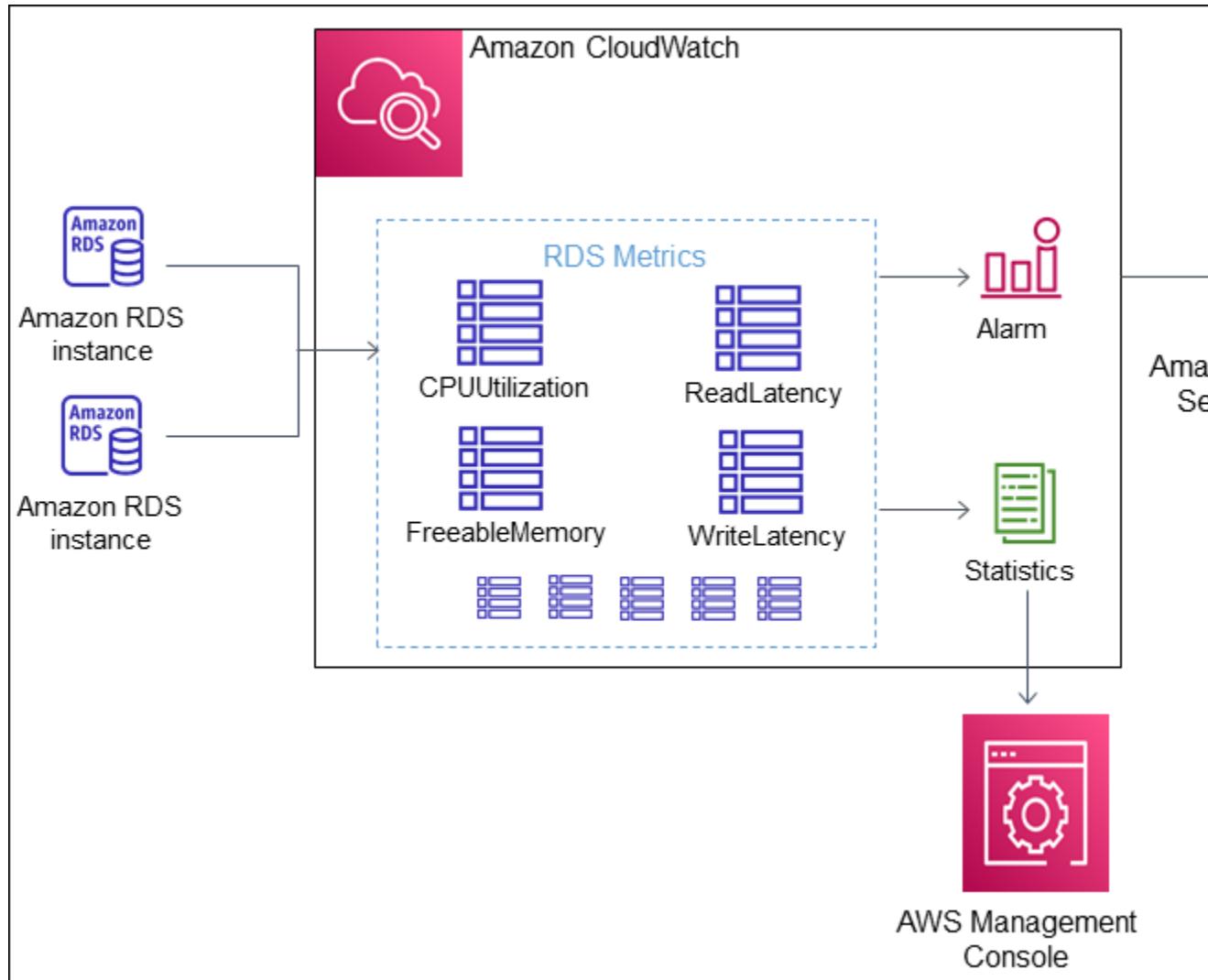
- [DBInstanceIdentifier](#)
- [EnableCloudwatchLogsExports](#)
- [Engine](#)
- [DBInstanceClass](#)

Other parameters might be required depending on the operation that you run.

Monitoring Amazon RDS metrics with Amazon CloudWatch

Amazon CloudWatch is a metrics repository. The repository collects and processes raw data from Amazon RDS into readable, near real-time metrics.

By default, Amazon RDS automatically sends metric data to CloudWatch in 1-minute periods. Data points with a period of 60 seconds (1 minute) are available for 15 days. This means that you can access historical information and see how your web application or service is performing.



For more information about CloudWatch, see [What is Amazon CloudWatch?](#) in the *Amazon CloudWatch User Guide*. For more information about CloudWatch metrics retention, see [Metrics retention](#).

Note

If you are using Amazon RDS Performance Insights, additional metrics are available. For more information, see [Performance Insights metrics published to Amazon CloudWatch \(p. 467\)](#).

Amazon RDS metrics

The AWS/RDS namespace includes the following metrics.

Note

The Amazon RDS console might display metrics in units that are different from the units sent to Amazon CloudWatch. For example, the Amazon RDS console might display a metric in megabytes (MB), while the metric is sent to Amazon CloudWatch in bytes.

Metric	Console name	Description	Units
BinLogDiskUsage	Binary Log Disk Usage (MB)	The amount of disk space occupied by binary logs on the primary. Applies to MySQL read replicas.	Bytes
BurstBalance	Burst Balance (Percent)	The percent of General Purpose SSD (gp2) burst-bucket I/O credits available.	Percent
CPUUtilization	CPU Utilization (Percent)	The percentage of CPU utilization.	Percent
CPUCreditUsage	CPU Credit Usage (Count)	(T2 instances) The number of CPU credits spent by the instance for CPU utilization. One CPU credit equals one vCPU running at 100 percent utilization for one minute or an equivalent combination of vCPUs, utilization, and time. For example, you might have one vCPU running at 50 percent utilization for two minutes or two vCPUs running at 25 percent utilization for two minutes. CPU credit metrics are available at a five-minute frequency only. If you specify a period greater than five minutes, use the <code>Sum</code> statistic instead of the <code>Average</code> statistic.	Credits (vCPU-minutes)
CPUCreditBalance	CPU Credit Balance (Count)	(T2 instances) The number of earned CPU credits that an instance has accrued since it was launched or started. For T2 Standard, the CPUCreditBalance also includes the number of launch credits that have been accrued. Credits are accrued in the credit balance after they are earned, and removed from the credit balance when they are spent. The credit balance has a maximum limit, determined by the instance size. After the limit is reached, any new credits that are earned are discarded. For T2 Standard, launch credits don't count towards the limit. The credits in the CPUCreditBalance are available for the instance to spend to burst beyond its baseline CPU utilization.	Credits (vCPU-minutes)

Metric	Console name	Description	Units
		<p>When an instance is running, credits in the <code>CPUCreditBalance</code> don't expire. When the instance stops, the <code>CPUCreditBalance</code> does not persist, and all accrued credits are lost.</p> <p>CPU credit metrics are available at a five-minute frequency only.</p>	
DatabaseConnections	DB Connections (Count)	<p>The number of database connections in use.</p> <p>The metric value might not include broken database connections that haven't been cleaned up by your database yet. So, the number of database connections recorded by your database might be higher than the metric value.</p>	Count
DiskQueueDepth	Queue Depth (Count)	The number of outstanding I/Os (read/write requests) waiting to access the disk.	Count
EBSByteBalance%	EBS Byte Balance (percent)	<p>The percentage of throughput credits remaining in the burst bucket of your RDS database. This metric is available for basic monitoring only.</p> <p>To find the instance sizes that support this metric, see the instance sizes with an asterisk (*) in the EBS optimized by default table in <i>Amazon EC2 User Guide for Linux Instances</i>. The Sum statistic is not applicable to this metric.</p>	Percent
EBSIOBalance%	EBS IO Balance (percent)	<p>The percentage of I/O credits remaining in the burst bucket of your RDS database. This metric is available for basic monitoring only.</p> <p>To find the instance sizes that support this metric, see the instance sizes with an asterisk (*) in the EBS optimized by default table in <i>Amazon EC2 User Guide for Linux Instances</i>. The Sum statistic is not applicable to this metric.</p> <p>This metric is different from <code>BurstBalance</code>. To learn how to use this metric, see Improving application performance and reducing costs with Amazon EBS-Optimized Instance burst capability.</p>	Percent
FailedSQLServerJobs	Failed SQL Server Agent Jobs Count (Count/Minute)	The number of failed Microsoft SQL Server Agent jobs during the last minute.	Count/Minute

Metric	Console name	Description	Units
FreeableMemory	Freeable Memory (MB)	The amount of available random access memory. For MariaDB, MySQL, Oracle, and PostgreSQL DB instances, this metric reports the value of the <code>MemAvailable</code> field of <code>/proc/meminfo</code> .	Bytes
FreeStorageSpace	Free Storage Space (MB/Second)	The amount of available storage space.	Bytes
MaximumUsedTransactions	Maximum Used Transaction IDs (Count)	The maximum transaction IDs that have been used. Applies to PostgreSQL.	Count
NetworkReceiveThroughput	Network Receive Throughput (MB/Second)	The incoming (receive) network traffic on the DB instance, including both customer database traffic and Amazon RDS traffic used for monitoring and replication.	Bytes/Second
NetworkTransmitThroughput	Network Transmit Throughput (MB/Second)	The outgoing (transmit) network traffic on the DB instance, including both customer database traffic and Amazon RDS traffic used for monitoring and replication.	Bytes/Second
OldestReplicationSlotLag	Oldest Replication Slot Lag (MB)	The lagging size of the replica lagging the most in terms of write-ahead log (WAL) data received. Applies to PostgreSQL.	Bytes
ReadIOPS	Read IOPS (Count/Second)	The average number of disk read I/O operations per second.	Count/Second
ReadLatency	Read Latency (Milliseconds)	The average amount of time taken per disk I/O operation.	Seconds
ReadThroughput	Read Throughput (MB/Second)	The average number of bytes read from disk per second.	Bytes/Second
ReplicaLag	Replica Lag (Milliseconds)	The amount of time a read replica DB instance lags behind the source DB instance. Applies to MySQL, MariaDB, Oracle, PostgreSQL, and SQL Server read replicas.	Seconds
ReplicationSlotDiskUsage	Replica Slot Disk Usage (MB)	The disk space used by replication slot files. Applies to PostgreSQL.	Bytes
SwapUsage	Swap Usage (MB)	The amount of swap space used on the DB instance. This metric is not available for SQL Server.	Bytes
TransactionLogsDiskUsage	Transaction Logs Disk Usage (MB)	The disk space used by transaction logs. Applies to PostgreSQL.	Bytes
TransactionLogsGeneration	TransactionLogs Generation (MB/Second)	The size of transaction logs generated per second. Applies to PostgreSQL.	Bytes/Second

Metric	Console name	Description	Units
WriteIOPS	Write IOPS (Count/Second)	The average number of disk write I/O operations per second.	Count/Second
WriteLatency	Write Latency (Milliseconds)	The average amount of time taken per disk I/O operation.	Seconds
WriteThroughput	Write Throughput (MB/Second)	The average number of bytes written to disk per second.	Bytes/Second

Amazon RDS dimensions

You can filter Amazon RDS metrics data by using any dimension in the following table.

Dimension	Filters the requested data for ...
DBInstanceIdentifier	A specific DB instance.
DatabaseClass	All instances in a database class. For example, you can aggregate metrics for all instances that belong to the database class db.r5.large.
EngineName	The identified engine name only. For example, you can aggregate metrics for all instances that have the engine name mysql.
SourceRegion	The specified Region only. For example, you can aggregate metrics for all DB instances in the us-east-1 Region.

Viewing Amazon RDS metrics and dimensions

When you use Amazon RDS resources, Amazon RDS sends metrics and dimensions to Amazon CloudWatch every minute. You can use the following procedures to view the metrics for Amazon RDS.

To view metrics using the Amazon CloudWatch console

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, change the AWS Region. From the navigation bar, choose the AWS Region where your AWS resources are. For more information, see [Regions and endpoints](#).
3. In the navigation pane, choose **Metrics**. Choose the **RDS** metric namespace.

The screenshot shows the RDS metrics dashboard for the N. Virginia region. At the top, there are tabs for 'All metrics' (selected), 'Graphed metrics', 'Graph options', and 'Source'. Below the tabs, the region is set to 'N. Virginia' and the search bar contains 'Search for any metric, dimension or resource'. The main content area displays three sections: 'DBClusterIdentifier, Role' (188 Metrics), 'Per-Database Metrics' (379 Metrics), and 'Across All Databases' (61 Metrics). Each section has a link to a detailed view.

4. Choose a metric dimension, for example **By Database Class**.
5. To sort the metrics, use the column heading. To graph a metric, select the check box next to the metric. To filter by resource, choose the resource ID, and then choose **Add to search**. To filter by metric, choose the metric name, and then choose **Add to search**.

The screenshot shows the 'By Database Class' search results for the 'db.r5.large' metric. The results table has columns for 'Metric Name' and 'Value'. A context menu is open over the first row, showing options: 'Add to search', 'Search for this only', 'Remove from graph', 'Graph this metric only', 'Graph all search results', and 'What is this?'. The 'db.r5.large' checkbox is checked in the table.

To view metrics using the AWS CLI

- At a command prompt, use the following command.

```
aws cloudwatch list-metrics --namespace AWS/RDS
```

Creating CloudWatch alarms to monitor Amazon RDS

You can create a CloudWatch alarm that sends an Amazon SNS message when the alarm changes state. An alarm watches a single metric over a time period that you specify. The alarm can also perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon SNS topic or Amazon EC2 Auto Scaling policy.

Alarms invoke actions for sustained state changes only. CloudWatch alarms don't invoke actions simply because they are in a particular state. The state must have changed and have been maintained for a specified number of time periods. The following procedures show how to create alarms for Amazon RDS.

To set alarms using the CloudWatch console

- Sign in to the AWS Management Console and open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
- Choose **Alarms** and then choose **Create Alarm**. Doing this launches the Create Alarm Wizard.
- Choose **RDS Metrics** and scroll through the Amazon RDS metrics to find the metric that you want to place an alarm on. To display just Amazon RDS metrics, search for the identifier of your resource. Choose the metric to create an alarm on and then choose **Next**.
- Enter **Name**, **Description**, and **Whenever** values for the metric.
- If you want CloudWatch to send you an email when the alarm state is reached, for **Whenever this alarm**, choose **State is ALARM**. For **Send notification to**, choose an existing SNS topic. If you choose **Create topic**, you can set the name and email addresses for a new email subscription list. This list is saved and appears in the field for future alarms.

Note

If you use **Create topic** to create a new Amazon SNS topic, the email addresses must be verified before they receive notifications. Emails are only sent when the alarm enters an alarm state. If this alarm state change happens before the email addresses are verified, the addresses don't receive a notification.

- Preview the alarm that you're about to create in the **Alarm Preview** area, and then choose **Create Alarm**.

To set an alarm using the AWS CLI

- Call [put-metric-alarm](#). For more information, see [AWS CLI Command Reference](#).

To set an alarm using the CloudWatch API

- Call [PutMetricAlarm](#). For more information, see [Amazon CloudWatch API Reference](#)

Publishing database engine logs to Amazon CloudWatch Logs

You can configure your Amazon RDS database engine to publish log data to a log group in Amazon CloudWatch Logs. With CloudWatch Logs, you can perform real-time analysis of the log data, and use CloudWatch to create alarms and view metrics. You can use CloudWatch Logs to store your log records in highly durable storage, which you can manage with the CloudWatch Logs Agent. For example, you can determine when to rotate log records from a host to the log service, so you can access the raw logs when you need to.

For engine-specific information, see the following topics:

- the section called “[Publishing MariaDB logs to Amazon CloudWatch Logs](#)” (p. 509)
- the section called “[Publishing MySQL logs to Amazon CloudWatch Logs](#)” (p. 521)
- the section called “[Publishing Oracle logs to Amazon CloudWatch Logs](#)” (p. 529)
- the section called “[Publishing PostgreSQL logs to Amazon CloudWatch Logs](#)” (p. 537)
- the section called “[Publishing SQL Server logs to Amazon CloudWatch Logs](#)” (p. 516)

Note

Before you enable log data publishing, make sure that you have a service-linked role in AWS Identity and Access Management (IAM). For more information about service-linked roles, see [Using service-linked roles for Amazon RDS](#) (p. 1714).

Configuring CloudWatch log integration

To publish your database log files to CloudWatch Logs, choose which logs to publish. Make this choice in the **Advanced Settings** section when you create a new DB instance. You can also modify an existing DB instance to begin publishing.

The screenshot shows the 'Log exports' configuration screen. At the top, it says 'Select the log types to publish to Amazon CloudWatch Logs'. Below this is a list of four checked checkboxes: 'Audit log', 'Error log', 'General log', and 'Slow query log'. Underneath the checkboxes is a section titled 'IAM role' with the sub-instruction 'The following service-linked role is used for publishing logs to CloudWatch Logs.' A button labeled 'RDS Service Linked Role' is shown. At the bottom of the screen, there is a note: 'Ensure that General, Slow Query, and Audit Logs are turned on. Error logs are enabled by default.'

After you have enabled publishing, Amazon RDS continuously streams all of the DB instance log records to a log group. For example, you have a log group `/aws/rds/instance/log` type for each type of log that you publish. This log group is in the same AWS Region as the database instance that generates the log.

After you have published log records, you can use CloudWatch Logs to search and filter the records. For more information about searching and filtering logs, see [Searching and filtering log data](#). For a tutorial explaining how to monitor RDS logs, see [Build proactive database monitoring for Amazon RDS with Amazon CloudWatch Logs, AWS Lambda, and Amazon SNS](#).

Viewing DB instance metrics

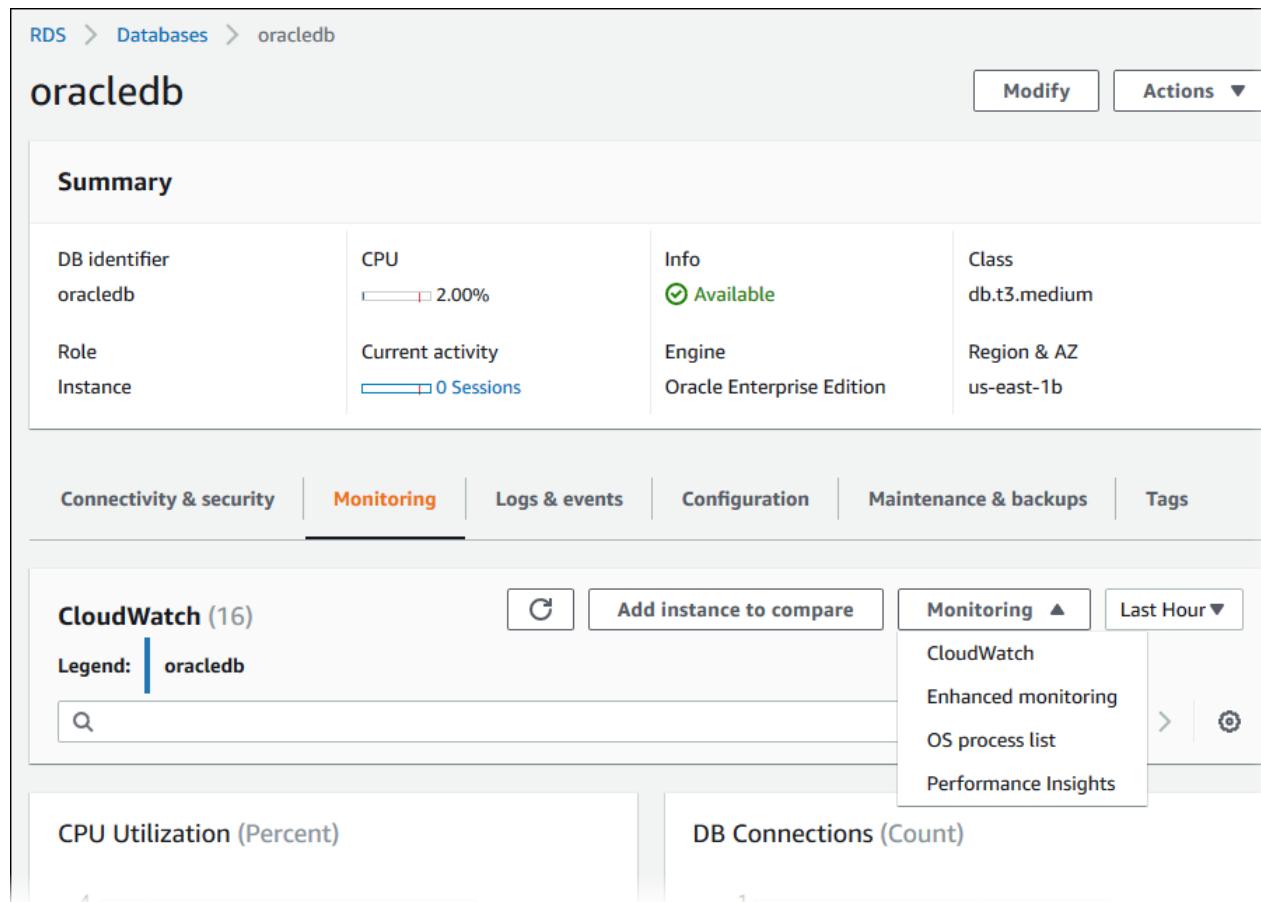
Amazon RDS provides metrics so that you can monitor the health of your DB instances. You can monitor both DB instance metrics and operating system (OS) metrics.

Following, you can find details about how to view metrics for your DB instance using the RDS console and CloudWatch. For information on monitoring metrics for your DB instance's operating system in real time using CloudWatch Logs, see [Using Enhanced Monitoring \(p. 471\)](#).

Viewing metrics by using the console

To view DB and OS metrics for a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the name of the DB instance that you need information about to show its details.
4. Choose the **Monitoring** tab.
5. For **Monitoring**, choose the option for how you want to view your metrics from these:
 - **CloudWatch** – Shows a summary of DB instance metrics available from Amazon CloudWatch. Each metric includes a graph showing the metric monitored over a specific time span.
 - **Enhanced monitoring** – Shows a summary of OS metrics available for a DB instance with Enhanced Monitoring enabled. Each metric includes a graph showing the metric monitored over a specific time span.
 - **OS Process list** – Shows details for each process running in the selected instance.
 - **Performance Insights** – Opens the Amazon RDS Performance Insights console for your DB instance.



Tip

To choose the time range of the metrics represented by the graphs, you can use the time range list.

To bring up a more detailed view, you can choose any graph. You can also apply metric-specific filters to the data.

Viewing DB instance metrics with the CLI or API

Amazon RDS integrates with CloudWatch metrics to provide a variety of DB instance metrics. You can view CloudWatch metrics using the RDS console, AWS CLI, or API.

For a complete list of Amazon RDS metrics, go to [Amazon RDS dimensions and metrics](#) in the *Amazon CloudWatch User Guide*.

Viewing DB metrics by using the CloudWatch CLI

Note

The following CLI example requires the CloudWatch command line tools. For more information on CloudWatch and to download the developer tools, see [Amazon CloudWatch](#) on the AWS website. The `StartTime` and `EndTime` values supplied in this example are for illustration only. Substitute appropriate start and end time values for your DB instance.

To view usage and performance statistics for a DB instance

- Use the CloudWatch command `mon-get-stats` with the following parameters.

```
PROMPT>mon-get-stats FreeStorageSpace --dimensions="DBInstanceIdentifier=mydbinstance"
--statistics= Average
--namespace="AWS/RDS" --start-time 2009-10-16T00:00:00 --end-time 2009-10-16T00:02:00
```

[Viewing DB metrics by using the CloudWatch API](#)

The `StartTime` and `EndTime` values supplied in this example are for illustration only. Substitute appropriate start and end time values for your DB instance.

To view usage and performance statistics for a DB instance

- Call the CloudWatch API `GetMetricStatistics` with the following parameters:
 - `Statistics.member.1 = Average`
 - `Namespace = AWS/RDS`
 - `StartTime = 2009-10-16T00:00:00`
 - `EndTime = 2009-10-16T00:02:00`
 - `Period = 60`
 - `MeasureName = FreeStorageSpace`

Getting CloudWatch Events and Amazon EventBridge events for Amazon RDS

Using Amazon CloudWatch Events and Amazon EventBridge, you can automate AWS services and respond to system events such as application availability issues or resource changes.

Topics

- [Overview of events for Amazon RDS \(p. 551\)](#)
- [Creating rules to send Amazon RDS events to CloudWatch Events \(p. 553\)](#)
- [Tutorial: log the state of an Amazon RDS instance using EventBridge \(p. 554\)](#)

Overview of events for Amazon RDS

An *event* indicates a change in an environment. This can be an AWS environment, an SaaS partner service or application, or one of your own custom applications or services. For example, Amazon RDS generates an event when the state of an instance changes from pending to running. Amazon RDS deliver events to CloudWatch Events and EventBridge in near real time.

Note

Amazon RDS emits events on a best effort basis. We recommend that you avoid writing programs that depends on the order or existence of notification events, as they might be out of sequence or missing.

You can write simple rules to indicate which events interest you and what automated actions to take when an event matches a rule. You can set a variety of targets, such as an AWS Lambda function or an Amazon SNS topic, which receive events in JSON format. For example, you can configure Amazon RDS to send events to CloudWatch Events or Amazon EventBridge whenever a DB instance is created or deleted. For more information, see the [Amazon CloudWatch Events User Guide](#) and the [Amazon EventBridge User Guide](#).

Topics

- [DB instance events \(p. 551\)](#)
- [DB parameter group events \(p. 552\)](#)
- [DB security group events \(p. 552\)](#)
- [DB snapshot events \(p. 553\)](#)

DB instance events

The following is an example of a DB instance event.

```
{  
  "version": "0",  
  "id": "68f6e973-1a0c-d37b-f2f2-94a7f62ffd4e",  
  "detail-type": "RDS DB Instance Event",  
  "source": "aws.rds",  
  "account": "123456789012",  
  "time": "2018-09-27T22:36:43Z",  
  "region": "us-east-1",  
  "resources": [  
    "arn:aws:rds:us-east-1:123456789012:db:my-db-instance"  
  ],  
  "detail": {
```

```
    "EventCategories": [
        "failover"
    ],
    "SourceType": "DB_INSTANCE",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
    "Date": "2018-09-27T22:36:43.292Z",
    "Message": "A Multi-AZ failover has completed.",
    "SourceIdentifier": "rds:my-db-instance",
    "EventID": "RDS-EVENT-0049"
}
}
```

DB parameter group events

The following is an example of a DB parameter group event.

```
{
    "version": "0",
    "id": "844e2571-85d4-695f-b930-0153b71dc42",
    "detail-type": "RDS DB Parameter Group Event",
    "source": "aws.rds",
    "account": "123456789012",
    "time": "2018-10-06T12:26:13Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:rds:us-east-1:123456789012:pg:my-db-param-group"
    ],
    "detail": {
        "EventCategories": [
            "configuration change"
        ],
        "SourceType": "DB_PARAM",
        "SourceArn": "arn:aws:rds:us-east-1:123456789012:pg:my-db-param-group",
        "Date": "2018-10-06T12:26:13.882Z",
        "Message": "Updated parameter time_zone to UTC with apply method immediate",
        "SourceIdentifier": "rds:my-db-param-group",
        "EventID": "RDS-EVENT-0037"
    }
}
```

DB security group events

The following is an example of a DB security group event.

```
{
    "version": "0",
    "id": "844e2571-85d4-695f-b930-0153b71dc42",
    "detail-type": "RDS DB Security Group Event",
    "source": "aws.rds",
    "account": "123456789012",
    "time": "2018-10-06T12:26:13Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:rds:us-east-1:123456789012:secgrp:my-security-group"
    ],
    "detail": {
        "EventCategories": [
            "configuration change"
        ],
        "SourceType": "SECURITY_GROUP",
        "SourceArn": "arn:aws:rds:us-east-1:123456789012:secgrp:my-security-group",
        "Date": "2018-10-06T12:26:13.882Z",
        "Message": "Updated security group rule for my-security-group with port 3306"
    }
}
```

```
        "Message": "Applied change to security group",
        "SourceIdentifier": "rds:my-security-group",
        "EventID": "RDS-EVENT-0038"
    }
}
```

DB snapshot events

The following is an example of a DB snapshot event.

```
{
    "version": "0",
    "id": "844e2571-85d4-695f-b930-0153b71dc42",
    "detail-type": "RDS DB Snapshot Event",
    "source": "aws.rds",
    "account": "123456789012",
    "time": "2018-10-06T12:26:13Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:rds:us-east-1:123456789012:snapshot:rds:my-db-snapshot"
    ],
    "detail": {
        "EventCategories": [
            "deletion"
        ],
        "SourceType": "SNAPSHOT",
        "SourceArn": "arn:aws:rds:us-east-1:123456789012:snapshot:rds:my-db-snapshot",
        "Date": "2018-10-06T12:26:13.882Z",
        "Message": "Deleted manual snapshot",
        "SourceIdentifier": "rds:my-db-snapshot",
        "EventID": "RDS-EVENT-0041"
    }
}
```

Creating rules to send Amazon RDS events to CloudWatch Events

You can create CloudWatch Events rules to send Amazon RDS events to CloudWatch Events.

Use the following steps to create a CloudWatch Events rule that triggers on an event emitted by an AWS service.

To create a rule that triggers on an event:

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. Under **Events** in the navigation pane, choose **Rules**.
3. Choose **Create rule**.
4. For **Event Source**, do the following:
 - a. Choose **Event Pattern**.
 - b. For **Service Name**, choose **Relational Database Service (RDS)**.
 - c. For **Event Type**, choose the type of Amazon RDS resource that triggers the event. For example, if a DB instance triggers the event, choose **RDS DB Instance Event**.
5. For **Targets**, choose **Add Target** and choose the AWS service that is to act when an event of the selected type is detected.
6. In the other fields in this section, enter information specific to this target type, if any is needed.

7. For many target types, CloudWatch Events needs permissions to send events to the target. In these cases, CloudWatch Events can create the IAM role needed for your event to run:
 - To create an IAM role automatically, choose **Create a new role for this specific resource**.
 - To use an IAM role that you created before, choose **Use existing role**.
8. Optionally, repeat steps 5-7 to add another target for this rule.
9. Choose **Configure details**. For **Rule definition**, type a name and description for the rule.

The rule name must be unique within this Region.
10. Choose **Create rule**.

For more information, see [Creating a CloudWatch Events Rule That Triggers on an Event](#) in the *Amazon CloudWatch User Guide*.

Tutorial: log the state of an Amazon RDS instance using EventBridge

You can create an AWS Lambda function that logs the changes in state for an Amazon RDS instance. You can choose to create a rule that runs the function whenever there is a state transition or a transition to one or more states that are of interest.

In this tutorial, you log any state change of an existing RDS DB instance. The tutorial assumes that you have a small running test instance that you can shut down temporarily.

Important

Don't perform this tutorial on a running production instance.

Step 1: Create an AWS Lambda Function

Create a Lambda function to log the state change events. You specify this function when you create your rule.

To create a Lambda function

1. Open the AWS Lambda console at <https://console.aws.amazon.com/lambda/>.
2. If you're new to Lambda, you see a welcome page. Choose **Get Started Now**. Otherwise, choose **Create function**.
3. Choose **Author from scratch**.
4. On the **Create function** page, do the following:
 - a. Enter a name and description for the Lambda function. For example, name the function **RDSInstanceStateChange**.
 - b. In **Runtime**, select **Node.js 14x**.
 - c. In **Execution role**, choose **Create a new role with basic Lambda permissions**. For **Existing role**, select your basic execution role. Otherwise, create a basic execution role.
 - d. Choose **Create function**.
5. On the **RDSInstanceStateChange** page, do the following:
 - a. In **Code source**, select **index.js**.
 - b. Right-click **index.js**, and choose **Open**.
 - c. In the **index.js** pane, delete the existing code.
 - d. Enter the following code:

```
console.log('Loading function');

exports.handler = async (event, context) => {
    console.log('Received event:', JSON.stringify(event));
};
```

- e. Choose **Deploy**.

Step 2: Create a Rule

Create a rule to run your Lambda function whenever you launch an Amazon RDS instance.

To create the EventBridge rule

1. Open the Amazon EventBridge console at <https://console.aws.amazon.com/events/>.
2. In the navigation pane, choose **Rules**.
3. Choose **Create rule**.
4. Enter a name and description for the rule. For example, enter **RDSInstanceStateChangeRule**.
5. For **Define pattern**, do the following:
 - a. Choose **Event pattern**.
 - b. Choose **Pre-defined pattern by service**.
 - c. For **Service provider**, choose **AWS**.
 - d. For **Service Name**, choose **Relational Database Service (RDS)**.
 - e. For **Event type**, choose **RDS DB Instance Event**.
6. For **Select event bus**, choose **AWS default event bus**. When an AWS service in your account emits an event, it always goes to your account's default event bus.
7. For **Target**, choose **Lambda function**.
8. For **Function**, select the Lambda function that you created.
9. Choose **Create**.

Step 3: Test the Rule

To test your rule, shut down an RDS DB instance. After waiting a few minutes for the instance to shut down, verify that your Lambda function was invoked.

To test your rule by stopping a DB instance

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Stop an RDS DB instance.
3. Open the Amazon EventBridge console at <https://console.aws.amazon.com/events/>.
4. In the navigation pane, choose **Rules**, choose the name of the rule that you created.
5. In **Rule details**, choose **Metrics for the rule**.

You are redirected to the Amazon CloudWatch console.
6. In **All metrics**, choose the name of the rule that you created.

The graph should indicate that the rule was invoked.
7. In the navigation pane, choose **Log groups**.
8. Choose the name of the log group for your Lambda function (**/aws/lambda/*function-name***).

9. Choose the name of the log stream to view the data provided by the function for the instance that you launched. You should see a received event similar to the following:

```
{  
    "version": "0",  
    "id": "12a345b6-78c9-01d2-34e5-123f4ghi5j6k",  
    "detail-type": "RDS DB Instance Event",  
    "source": "aws.rds",  
    "account": "111111111111",  
    "time": "2021-03-19T19:34:09Z",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:rds:us-east-1:111111111111:db:testdb"  
    ],  
    "detail": {  
        "EventCategories": [  
            "notification"  
        ],  
        "SourceType": "DB_INSTANCE",  
        "SourceArn": "arn:aws:rds:us-east-1:111111111111:db:testdb",  
        "Date": "2021-03-19T19:34:09.293Z",  
        "Message": "DB instance stopped",  
        "SourceIdentifier": "testdb",  
        "EventID": "RDS-EVENT-0087"  
    }  
}
```

10. (Optional) When you're finished, you can open the Amazon RDS console and start the instance that you stopped.

Working with AWS CloudTrail and Amazon RDS

AWS CloudTrail is an AWS service that helps you audit your AWS account. CloudTrail is enabled on your AWS account when you create it.

For complete information about CloudTrail, see the [AWS CloudTrail User Guide](#).

CloudTrail integration with Amazon RDS

All Amazon RDS actions are logged by CloudTrail. CloudTrail provides a record of actions taken by a user, role, or an AWS service in Amazon RDS.

CloudTrail events

CloudTrail captures API calls for Amazon RDS as events. An *event* represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. Events include calls from the Amazon RDS console and from code calls to the Amazon RDS APIs.

Amazon RDS activity is recorded in a CloudTrail event in **Event history**. You can use the CloudTrail console to view the last 90 days of recorded API activity and events in an AWS Region. For more information, see [Viewing events with CloudTrail event history](#).

CloudTrail trails

For an ongoing record of events in your AWS account, including events for Amazon RDS, create a *trail*. A trail is a configuration that enables delivery of events to a specified Amazon S3 bucket. CloudTrail typically delivers log files within 15 minutes of account activity.

Note

If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**.

You can create two types of trails for an AWS account: a trail that applies to all regions, or a trail that applies to one region. By default, when you create a trail in the console, the trail applies to all regions.

Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- Overview for creating a trail
 - CloudTrail supported services and integrations
 - Configuring Amazon SNS notifications for CloudTrail
 - Receiving CloudTrail log files from multiple Regions and Receiving CloudTrail log files from multiple accounts

Amazon RDS log file entries

CloudTrail log files contain one or more log entries. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `CreateDBInstance` action.

```
{  
    "eventVersion": "1.04",  
    "userIdentity": {  
        "type": "TAMUser"  
    }  
}
```

```

"principalId": "AKIAIOSFODNN7EXAMPLE",
"arn": "arn:aws:iam::123456789012:user/johndoe",
"accountId": "123456789012",
"accessKeyId": "AKIAI44QH8DHBEXAMPLE",
"userName": "johndoe"
},
"eventTime": "2018-07-30T22:14:06Z",
"eventSource": "rds.amazonaws.com",
"eventName": "CreateDBInstance",
"awsRegion": "us-east-1",
"sourceIPAddress": "192.0.2.0",
"userAgent": "aws-cli/1.15.42 Python/3.6.1 Darwin/17.7.0 botocore/1.10.42",
"requestParameters": {
    "enableCloudwatchLogsExports": [
        "audit",
        "error",
        "general",
        "slowquery"
    ],
    "dBInstanceIdentifier": "test-instance",
    "engine": "mysql",
    "masterUsername": "myawsuser",
    "allocatedStorage": 20,
    "dBInstanceClass": "db.m1.small",
    "masterUserPassword": "*****"
},
"responseElements": {
    "dBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance",
    "storageEncrypted": false,
    "preferredBackupWindow": "10:27-10:57",
    "preferredMaintenanceWindow": "sat:05:47-sat:06:17",
    "backupRetentionPeriod": 1,
    "allocatedStorage": 20,
    "storageType": "standard",
    "engineVersion": "5.6.39",
    "dBInstancePort": 0,
    "optionGroupMemberships": [
        {
            "status": "in-sync",
            "optionGroupName": "default:mysql-5-6"
        }
    ],
    "dBParameterGroups": [
        {
            "dBParameterGroupName": "default.mysql5.6",
            "parameterApplyStatus": "in-sync"
        }
    ],
    "monitoringInterval": 0,
    "dBInstanceClass": "db.m1.small",
    "readReplicaDBInstanceIdentifiers": [],
    "dBSubnetGroup": {
        "dBSubnetGroupName": "default",
        "dBSubnetGroupDescription": "default",
        "subnets": [
            {
                "subnetAvailabilityZone": {"name": "us-east-1b"},
                "subnetIdentifier": "subnet-cbfff283",
                "subnetStatus": "Active"
            },
            {
                "subnetAvailabilityZone": {"name": "us-east-1e"},
                "subnetIdentifier": "subnet-d7c825e8",
                "subnetStatus": "Active"
            },
            {

```

```

        "subnetAvailabilityZone": {"name": "us-east-1f"},  

        "subnetIdentifier": "subnet-6746046b",  

        "subnetStatus": "Active"  

    },  

    {  

        "subnetAvailabilityZone": {"name": "us-east-1c"},  

        "subnetIdentifier": "subnet-bac383e0",  

        "subnetStatus": "Active"  

    },  

    {  

        "subnetAvailabilityZone": {"name": "us-east-1d"},  

        "subnetIdentifier": "subnet-42599426",  

        "subnetStatus": "Active"  

    },  

    {  

        "subnetAvailabilityZone": {"name": "us-east-1a"},  

        "subnetIdentifier": "subnet-da327bf6",  

        "subnetStatus": "Active"  

    }  

],  

"vpcId": "vpc-136a4c6a",  

"subnetGroupStatus": "Complete"  

},  

"masterUsername": "myawsuser",  

"multiAZ": false,  

"autoMinorVersionUpgrade": true,  

"engine": "mysql",  

"caCertificateIdentifier": "rds-ca-2015",  

"dbiResourceId": "db-ETDZIIXHEWY5N7GXVC4SH7H5IA",  

"dBSecurityGroups": [],  

"pendingModifiedValues": {  

    "masterUserPassword": "*****",  

    "pendingCloudwatchLogsExports": {  

        "logTypesToEnable": [  

            "audit",  

            "error",  

            "general",  

            "slowquery"
        ]
    }
},  

"dBInstanceStatus": "creating",  

"publiclyAccessible": true,  

"domainMemberships": [],  

"copyTagsToSnapshot": false,  

"dBInstanceIdentifier": "test-instance",  

"licenseModel": "general-public-license",  

"iAMDatabaseAuthenticationEnabled": false,  

"performanceInsightsEnabled": false,  

"vpcSecurityGroups": [  

    {
        "status": "active",
        "vpcSecurityGroupId": "sg-f839b688"
    }
]
},  

"requestID": "daf2e3f5-96a3-4df7-a026-863f96db793e",  

"eventID": "797163d3-5726-441d-80a7-6eeb7464acd4",  

"eventType": "AwsApiCall",  

"recipientAccountId": "123456789012"
}

```

As shown in the `userIdentity` element in the preceding example, every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information about the `userIdentity`, see the [CloudTrail userIdentity element](#). For more information about `CreateDBInstance` and other Amazon RDS actions, see the [Amazon RDS API Reference](#).

Working with Amazon RDS on AWS Outposts

Amazon RDS on AWS Outposts extends Amazon RDS for MySQL and PostgreSQL databases to AWS Outposts environments. AWS Outposts uses the same hardware as in public AWS Regions to bring AWS services, infrastructure, and operation models on-premises. With RDS on Outposts, you can provision managed DB instances close to the business applications that must run on-premises. For more information about AWS Outposts, see [AWS Outposts](#).

You use the same AWS Management Console, AWS CLI, and RDS API to provision and manage on-premises RDS on Outposts DB instances as you do for RDS DB instances running in the AWS Cloud. RDS on Outposts automates tasks, such as database provisioning, operating system and database patching, backup, and long-term archival in Amazon S3.

RDS on Outposts supports automated backups of DB instances. Network connectivity between your Outpost and your AWS Region is required to back up and restore DB instances. All DB snapshots and transaction logs from an Outpost are stored in your AWS Region. From your AWS Region, you can restore a DB instance from a DB snapshot to a different Outpost. For more information, see [Working with backups \(p. 328\)](#).

RDS on Outposts supports automated maintenance and upgrades of DB instances. For more information, see [Maintaining a DB instance \(p. 264\)](#).

RDS on Outposts uses encryption at rest for DB instances and DB snapshots using your AWS Key Management Service (AWS KMS) key. For more information about encryption at rest, see [Encrypting Amazon RDS resources \(p. 1630\)](#).

By default, EC2 instances in Outposts subnets can use the Amazon Route 53 DNS Service to resolve domain names to IP addresses. You might encounter longer DNS resolution times with Route 53, depending on the path latency between your Outpost and the AWS Region. In such cases, you can use the DNS servers installed locally in your on-premises environment. For more information, see [DNS](#) in the [AWS Outposts User Guide](#).

When network connectivity to the AWS Region isn't available, your DB instance continues to run locally. You can continue to access DB instances using DNS name resolution by configuring a local DNS server as a secondary server. However, you can't create new DB instances or take new actions on existing DB instances. Automatic backups don't occur when there is no connectivity. If there is a DB instance failure, the DB instance isn't automatically replaced until connectivity is restored. We recommend restoring network connectivity as soon as possible.

Topics

- [Prerequisites for Amazon RDS on AWS Outposts \(p. 561\)](#)
- [Amazon RDS on AWS Outposts support for Amazon RDS features \(p. 562\)](#)
- [Supported DB instance classes for Amazon RDS on AWS Outposts \(p. 564\)](#)
- [Customer-owned IP addresses for RDS on Outposts \(p. 565\)](#)
- [Creating DB instances for Amazon RDS on AWS Outposts \(p. 567\)](#)

Prerequisites for Amazon RDS on AWS Outposts

The following are prerequisites for using Amazon RDS on AWS Outposts:

- Install AWS Outposts in your on-premises data center. For more information about AWS Outposts, see [AWS Outposts](#).
- Make sure that you have at least one subnet available for RDS on Outposts. You can use the same subnet for other workloads.
- Make sure that you have a reliable network connection between your Outpost and an AWS Region.

Amazon RDS on AWS Outposts support for Amazon RDS features

Feature	Supported	Notes	More information
DB instance provisioning	Yes	You can only create DB instances for RDS for MySQL and RDS for PostgreSQL DB engines. The following versions are supported: <ul style="list-style-type: none"> MySQL versions 8.0.17, 8.0.19, 8.0.20, and 8.0.21 PostgreSQL versions 12.2, 12.3, and 12.4 	Creating an Amazon RDS DB instance (p. 141)
Modifying the master user password	Yes	—	Modifying an Amazon RDS DB instance (p. 250)
Renaming a DB instance	Yes	—	Modifying an Amazon RDS DB instance (p. 250)
Rebooting a DB instance	Yes	—	Rebooting a DB instance (p. 276)
Stopping a DB instance	Yes	—	Stopping an Amazon RDS DB instance temporarily (p. 246)
Starting a DB instance	Yes	—	Starting an Amazon RDS DB instance that was previously stopped (p. 249)
Multi-AZ deployments	No	—	High availability (Multi-AZ) for Amazon RDS (p. 53)
DB parameter groups	Yes	—	Working with DB parameter groups (p. 228)
Read replicas	No	—	Working with read replicas (p. 278)
Encryption at rest	Yes	RDS on Outposts doesn't support unencrypted DB instances.	Encrypting Amazon RDS resources (p. 1630)
AWS Identity and Access Management	No	—	IAM database authentication for MySQL and PostgreSQL (p. 1660)

Feature	Supported	Notes	More information
(IAM) database authentication		—	
Associating an IAM role with a DB instance	No	—	add-role-to-db-instance CLI command and AddRoleToDBInstance RDS API operation
Kerberos authentication	No	—	Kerberos authentication (p. 1629)
Tagging Amazon RDS resources	Yes	—	Tagging Amazon RDS resources (p. 299)
Option groups	Yes	—	Working with option groups (p. 212)
Modifying the maintenance window	Yes	—	Maintaining a DB instance (p. 264)
Automatic minor version upgrade	Yes	—	Automatically upgrading the minor engine version (p. 273)
Modifying the backup window	Yes	—	Working with backups (p. 328) and Modifying an Amazon RDS DB instance (p. 250)
DB instance scaling	Yes	To scale a DB instance, modify its on-premises DB instance class. Storage scaling isn't supported.	Modifying an Amazon RDS DB instance (p. 250)
Manual and automatic DB instance snapshots	Yes	Manual and automatic DB instance snapshots are stored in your AWS Region.	Creating a DB snapshot (p. 346)
Restoring from a DB snapshot	Yes	—	Restoring from a DB snapshot (p. 349)
Restoring a DB instance from Amazon S3	No	—	Restoring a backup into a MySQL DB instance (p. 871)
Exporting snapshot data to Amazon S3	Yes	—	Exporting DB snapshot data to Amazon S3 (p. 373)
Point-in-time recovery	Yes	—	Restoring a DB instance to a specified time (p. 389)
Enhanced monitoring	No	—	Using Enhanced Monitoring (p. 471)

Feature	Supported	Notes	More information
Amazon CloudWatch monitoring	Yes	You can view the same set of metrics that are available for your databases in the AWS Region.	Monitoring Amazon RDS metrics with Amazon CloudWatch (p. 540)
Publishing database engine logs to CloudWatch Logs	No	—	Publishing database logs to Amazon CloudWatch Logs (p. 506)
Event notification	Yes	—	Using Amazon RDS event notification (p. 487)
Amazon RDS Performance Insights	No	—	Using Performance Insights on Amazon RDS (p. 412)
Viewing or downloading database logs	No	RDS on Outposts doesn't support viewing database logs using the console or describing database logs using the CLI or RDS API. RDS on Outposts doesn't support downloading database logs using the console or downloading database logs using the CLI or RDS API.	Accessing Amazon RDS database log files (p. 504)
Amazon RDS Proxy	No	—	Managing connections with Amazon RDS Proxy (p. 167)
Stored procedures for Amazon RDS for MySQL	Yes	—	MySQL on Amazon RDS SQL reference (p. 952)
Replication with external databases for Amazon RDS for MySQL	No	—	Replication with a MySQL or MariaDB instance running external to Amazon RDS (p. 914)

Note

RDS on Outposts doesn't support use cases that require all data to remain in your data center.
RDS on Outposts stores database backups and logs in your AWS Region.

Supported DB instance classes for Amazon RDS on AWS Outposts

Amazon RDS on AWS Outposts supports the following DB instance classes:

- General Purpose DB instance classes
 - db.m5.24xlarge

- db.m5.12xlarge
- db.m5.4xlarge
- db.m5.2xlarge
- db.m5.xlarge
- db.m5.large
- Memory Optimized DB instance classes
 - db.r5.24xlarge
 - db.r5.12xlarge
 - db.r5.4xlarge
 - db.r5.2xlarge
 - db.r5.xlarge
 - db.r5.large

Only General Purpose SSD storage is supported for RDS on Outposts DB instances. For more information about DB instance classes, see [DB instance classes \(p. 7\)](#).

Amazon RDS manages maintenance and recovery for your DB instances and requires active capacity on the Outpost to do so. We recommend that you configure N+1 EC2 instances for each DB instance class in your production environments. RDS on Outposts can use the extra capacity of these EC2 instances for maintenance and repair operations. For example, if your production environments have 3 db.m5.large and 5 db.r5.xlarge DB instance classes, then we recommend that they have at least 4 m5.large EC2 instances and 6 r5.xlarge EC2 instances. For more information, see [Resilience in AWS Outposts](#) in the [AWS Outposts User Guide](#).

Customer-owned IP addresses for RDS on Outposts

AWS Outposts uses information that you provide about your on-premises network to create an address pool, known as a *customer-owned IP address pool* (CoIP pool). *Customer-owned IP addresses* (CoIPs) provide local or external connectivity to resources in your Outpost subnets through your on-premises network. For more information about CoIPs, see [Customer-owned IP addresses](#) in the [AWS Outposts User Guide](#).

Each RDS on Outposts DB instance has a private IP address for traffic inside its virtual private cloud (VPC). This private IP address isn't publicly accessible. You can use the **Public** option to designate whether the DB instance also has a public IP address in addition to the private IP address. Using the public IP address for connections routes them through the internet and can result in high latencies in some cases.

Instead of using these private and public IP addresses, RDS on Outposts supports enabling a CoIP for DB instances through their subnets. When you enable a CoIP for an RDS on Outposts DB instance, you connect to the DB instance with the DB instance endpoint. RDS on Outposts automatically uses the CoIP for all connections from both inside and outside of the VPC.

CoIPs can provide the following benefits for RDS on Outposts DB instances:

- Lower connection latency
- Enhanced security

You can enable or disable a CoIP for an RDS on Outposts DB instance using the AWS Management Console, the AWS CLI, or the RDS API:

- With the AWS Management Console, use the **Customer-owned IP address (CoIP)** setting in **Access type** to enable a CoIP. Use one of the other settings to disable it.

▼ Additional configuration

Access type [Info](#)

Private

RDS will not assign a public IP address to the database. Amazon EC2 instances and devices inside the VPC can connect to your database. EC2 instances and devices outside your VPC can't connect unless they use AWS Site-to-Site VPN or AWS Direct Connect.

Customer-owned IP address (CoIP)

Devices on your on-premises network can connect to your database through a CoIP.

Public

Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices can connect to the database.

Database port

TCP/IP port that the database will use for application connections.

3306

- With the AWS CLI, use the `--enable-customer-owned-ip` | `--no-enable-customer-owned-ip` option.
- With the RDS API, use the `EnableCustomerOwnedIp` parameter.

You can enable or disable a CoIP when you perform any of the following actions:

- Create a DB instance

For more information, see [Creating DB instances for Amazon RDS on AWS Outposts \(p. 567\)](#).

- Modify a DB instance

For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

- Restore a DB instance from a snapshot

For more information, see [Restoring from a DB snapshot \(p. 349\)](#).

- Restore a DB instance to a specified time

For more information, see [Restoring a DB instance to a specified time \(p. 389\)](#).

Note

If you enable a CoIP for a DB instance, but Amazon RDS is unable to allocate a CoIP for the DB instance, the DB instance status is changed to **incompatible-network**. For more information about the DB instance status, see [DB instance status \(p. 404\)](#).

The following limitations apply to CoIP support for RDS on Outposts DB instances:

- When a CoIP is enabled for a DB instance, make sure that public accessibility is disabled for the DB instance.
- You can't assign a CoIP from a CoIP pool to a DB instance. When you enable a CoIP for a DB instance, Amazon RDS automatically assigns a CoIP from a CoIP pool to the DB instance.

- You must use the AWS account that owns the Outpost resources (owner) or share the following resources with other AWS accounts (consumers) in the same organization.
 - The Outpost
 - The local gateway (LGW) route table for the DB instance's VPC
 - The CoIP pool or pools for the LGW route table

For more information, see [Working with shared AWS Outposts resources](#) in the *AWS Outposts User Guide*.

Creating DB instances for Amazon RDS on AWS Outposts

Creating an Amazon RDS on AWS Outposts DB instance is similar to creating an Amazon RDS DB instance in the AWS Cloud. However, you must specify a DB subnet group that is associated with your Outpost.

An Amazon VPC can span all of the Availability Zones in an AWS Region. You can extend any VPC in the AWS Region to your Outpost by adding an Outpost subnet. To add an Outpost subnet to a VPC, specify the Amazon Resource Name (ARN) of the Outpost when you create the subnet.

Before you create an RDS on Outposts DB instance, you can create a DB subnet group that includes one subnet that is associated with your Outpost. When you create an RDS on Outposts DB instance, specify this DB subnet group. You can also choose to create a new DB subnet group when you create your DB instance.

For information about configuring AWS Outposts, see the [AWS Outposts User Guide](#).

Console

To create an RDS on Outposts DB instance using the console

1. Create a DB subnet group with one subnet that is associated with your Outpost.

To create a new DB subnet group for the Outpost when you create your DB instance, skip this step.

Note

To create a DB subnet group for the AWS Cloud, you specify at least two subnets. However, for an Outpost DB subnet group, you can specify only one subnet.

- a. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
- b. In the upper-right corner of the Amazon RDS console, choose the AWS Region where you want to create the DB subnet group.
- c. Choose **Subnet groups**, and then choose **Create DB Subnet Group**.

The **Create DB subnet group** page appears.

The screenshot shows the 'Create DB Subnet Group' wizard. At the top, a breadcrumb navigation shows 'RDS > Subnet groups > Create DB subnet group'. The main title is 'Create DB Subnet Group'. A descriptive text below the title says: 'To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.' The first section, 'Subnet group details', contains fields for 'Name' (with a note: 'You won't be able to modify the name after your subnet group has been created.') and 'Description'. The 'VPC' section allows selecting a VPC identifier from a dropdown menu labeled 'Choose a VPC'. The second section, 'Add subnets', contains fields for 'Availability Zones' (with a note: 'Choose the Availability Zones that include the subnets you want to add.') and 'Subnets' (with a note: 'Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.').

- d. Set the following values for your new DB subnet group:
 - **Name** – The name of the DB subnet group
 - **Description** – A description for the DB subnet group
 - **VPC** – The VPC for which you're creating the DB subnet group
 - e. For **Availability Zones**, choose the Availability Zone for your Outpost.
 - f. For **Subnets**, choose the subnet for use by RDS on Outposts.

Your DB subnet group must have only one subnet.
 - g. Choose **Create** to create the DB subnet group.
2. Create the DB instance, and choose the Outpost for your DB instance.
 - a. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
 - b. In the upper-right corner of the Amazon RDS console, choose the AWS Region where you want to create the DB instance.

- c. In the navigation pane, choose **Databases**.
- d. Choose **Create database**.

The AWS Management Console detects available Outposts that you have configured and presents the **On-premises** option in the **Database location** section.

Create database

Database location
Choose a location to meet your use case. [Info](#)

Database location options

Amazon Cloud
Use Amazon's cloud to store and provision a database instance with RDS.

On-premises
Create a DB instance on-premises using an AWS Outpost or VMware datacenter.

Choose an on-premises creation method [Info](#)

On-premises database options

RDS on Outposts
Use an AWS outpost to create a DB instance on-premises.

RDS on VMware
Use a custom AZ to create a DB instance with an on-premises VMware datacenter.

Outposts Connectivity

Virtual private cloud (VPC) [Info](#)
VPC that defines the virtual networking environment for this DB instance.

vpc-0069a9625caeda794

Only VPCs with an Outpost subnet are listed.

VPC security group

Choose VPC security groups

Subnet group [Info](#)
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

Choose existing
Choose existing subnet group

Create new
Create new subnet group

Add subnets to DB subnet group.

subnet-0dbe9e254c8463e33 (Outpost: op-09d7a5baee9402a08)

Note

If you haven't configured any Outposts, either the **Database location** section doesn't appear or the **RDS on Outposts** option isn't available in the **Choose an on-premises creation method** section.

- e. Choose the following settings:
 - **Database location – On-premises**
 - **On-premises creation method – RDS on Outposts**
 - **Outpost** – The Outpost that uses the virtual private cloud (VPC) that has the DB subnet group for your DB instance. Your VPC here must be based on the Amazon VPC service.
 - **Virtual Private Cloud (VPC)** – The VPC that contains the DB subnet group for your DB instance.
 - **VPC security group** – The Amazon VPC security group for your DB instance.

- **Subnet group** – The DB subnet group for your DB instance.

You can choose an existing DB subnet group that is associated with the Outpost. If you didn't create a DB subnet group, you can create a new DB subnet group for the Outpost. Only one subnet is allowed in this DB subnet group.

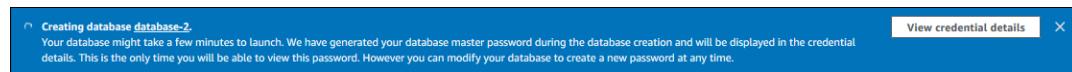
- For the remaining sections, specify your DB instance settings.

For information about each setting when creating a DB instance, see [Settings for DB instances \(p. 145\)](#).

- Choose **Create database**.

If you chose to use an automatically generated password, the **View credential details** button appears on the **Databases** page.

To view the master user name and password for the DB instance, choose **View credential details**.



To connect to the DB instance as the master user, use the user name and password that appear.

Important

You can't view the master user password again. If you don't record it, you might have to change it. To change the master user password after the DB instance is available, modify the DB instance. For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

- For **Databases**, choose the name of the new DB instance.

On the RDS console, the details for the new DB instance appear. The DB instance has a status of **Creating** until the DB instance is created and ready for use. When the state changes to **Available**, you can connect to the DB instance. Depending on the DB instance class and storage allocated, it can take several minutes for the new DB instance to be available.

The screenshot shows the RDS Databases page. The top navigation bar shows "RDS > Databases > database-1". On the right, there are "Modify" and "Actions" buttons. The main area shows a table with the following data:

Summary			
DB identifier database-1	CPU -	Info Creating	Class db.m5.xlarge
Role Instance	Current activity 0 Sessions	Engine MySQL Community	Region & AZ -

Below the table, there are tabs for "Connectivity & security" (which is highlighted in red), "Monitoring", "Logs & events", "Configuration", and "Maintenance & backups".

After the DB instance is available, you can manage it the same way that you manage RDS DB instances in the cloud.

AWS CLI

To create a new DB instance in an Outpost with the AWS CLI, first create a DB subnet group for use by RDS on Outposts by calling the [create-db-subnet-group](#) command. For `--subnet-ids`, specify the subnet group in the Outpost for use by RDS on Outposts.

For Linux, macOS, or Unix:

```
aws rds create-db-subnet-group \
--db-subnet-group-name myoutpostdbsubnetgr \
--db-subnet-group-description "DB subnet group for RDS on Outposts" \
--subnet-ids subnet-abc123
```

For Windows:

```
aws rds create-db-subnet-group ^
--db-subnet-group-name myoutpostdbsubnetgr ^
--db-subnet-group-description "DB subnet group for RDS on Outposts" ^
--subnet-ids subnet-abc123
```

Next, call the [create-db-instance](#) command with the parameters below. Specify an Availability Zone for the Outpost, an Amazon VPC security group associated with the Outpost, and the DB subnet group you created for the Outpost. You can include the following options:

- `--db-instance-identifier`
- `--db-instance-class`
- `--engine`
- `--availability-zone`
- `--vpc-security-group-ids`
- `--db-subnet-group-name`
- `--allocated-storage`
- `--master-user-name`
- `--master-user-password`
- `--backup-retention-period`
- `--storage-encrypted`
- `--kms-key-id`

Example

The following example creates a MySQL DB instance named `myoutpostdbinstance`.

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
--db-instance-identifier myoutpostdbinstance \
--engine-version 8.0.17 \
--db-instance-class db.m5.large \
--engine mysql \
--availability-zone us-east-1d \
```

```
--vpc-security-group-ids outpost-sg \
--db-subnet-group-name myoutpostdbsubnetgr \
--allocated-storage 100 \
--master-username masterawsuser \
--master-user-password masteruserpassword \
--backup-retention-period 3 \
--storage-encrypted \
--kms-key-id mykey
```

For Windows:

```
aws rds create-db-instance ^
--db-instance-identifier myoutpostdbinstance ^
--engine-version 8.0.17 ^
--db-instance-class db.m5.large ^
--engine mysql ^
--availability-zone us-east-1d ^
--vpc-security-group-ids outpost-sg ^
--db-subnet-group-name myoutpostdbsubnetgr ^
--allocated-storage 100 ^
--master-username masterawsuser ^
--master-user-password masteruserpassword ^
--backup-retention-period 3 ^
--storage-encrypted ^
--kms-key-id mykey
```

To create a PostgreSQL DB instance, specify `postgres` for the `--engine` option.

For information about each setting when creating a DB instance, see [Settings for DB instances \(p. 145\)](#).

RDS API

To create a new DB instance in an Outpost with the RDS API, first create a DB subnet group for use by RDS on Outposts by calling the [CreateDBSubnetGroup](#) operation. For `SubnetIds`, specify the subnet group in the Outpost for use by RDS on Outposts.

Next, call the [CreateDBInstance](#) operation with the parameters below. Specify an Availability Zone for the Outpost, an Amazon VPC security group associated with the Outpost, and the DB subnet group you created for the Outpost.

- `AllocatedStorage`
- `AvailabilityZone`
- `BackupRetentionPeriod`
- `DBInstanceState`
- `DBInstanceIdentifier`
- `VpcSecurityGroupIds`
- `DBSubnetGroupName`
- `Engine`
- `EngineVersion`
- `MasterUsername`
- `MasterUserPassword`
- `StorageEncrypted`
- `KmsKeyID`

For information about each setting when creating a DB instance, see [Settings for DB instances \(p. 145\)](#).

MariaDB on Amazon RDS

Amazon RDS supports DB instances running several versions of MariaDB. You can use the following major versions:

- MariaDB 10.5
- MariaDB 10.4
- MariaDB 10.3
- MariaDB 10.2
- MariaDB 10.1
- MariaDB 10.0

For more information about minor version support, see [MariaDB on Amazon RDS versions \(p. 576\)](#).

You first use the Amazon RDS management tools or interfaces to create a MariaDB DB instance. You can then use the Amazon RDS tools to perform management actions for the DB instance, such as reconfiguring or resizing the DB instance, authorizing connections to the DB instance, creating and restoring from backups or snapshots, creating Multi-AZ secondaries, creating read replicas, and monitoring the performance of the DB instance. You use standard MariaDB utilities and applications to store and access the data in the DB instance.

MariaDB is available in all of the AWS Regions. For more information about AWS Regions, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

You can use Amazon RDS for MariaDB databases to build HIPAA-compliant applications. You can store healthcare-related information, including protected health information (PHI), under a Business Associate Agreement (BAA) with AWS. For more information, see [HIPAA compliance](#). AWS Services in Scope have been fully assessed by a third-party auditor and result in a certification, attestation of compliance, or Authority to Operate (ATO). For more information, see [AWS services in scope by compliance program](#).

Before creating your first DB instance, you should complete the steps in the setting up section of this guide. For more information, see [Setting up for Amazon RDS \(p. 67\)](#).

Common management tasks for MariaDB on Amazon RDS

The following are the common management tasks you perform with an Amazon RDS DB instance running MariaDB, with links to relevant documentation for each task.

Task area	Relevant documentation
Instance Classes, Storage, and PIOPS If you are creating a DB instance for production purposes, you should understand how instance classes, storage types, and Provisioned IOPS work in Amazon RDS.	DB instance classes (p. 7) Amazon RDS storage types (p. 40)
Multi-AZ Deployments Provide high availability with synchronous standby replication in a different Availability Zone, automatic failover, fault tolerance for DB instances using Multi-AZ deployments, and read replicas.	High availability (Multi-AZ) for Amazon RDS (p. 53)

Task area	Relevant documentation
Amazon Virtual Private Cloud (VPC) If your AWS account has a default VPC, then your DB instance is automatically created inside the default VPC. If your account does not have a default VPC, and you want the DB instance in a VPC, you must create the VPC and subnet groups before you create the DB instance.	Determining whether you are using the EC2-VPC or EC2-Classic platform (p. 1718) Working with a DB instance in a VPC (p. 1727)
Security Groups By default, DB instances are created with a firewall that prevents access to them. You therefore must create a security group with the correct IP addresses and network configuration to access the DB instance. The security group you create depends on what Amazon EC2 platform your DB instance is on, and whether you access your DB instance from an Amazon EC2 instance. In general, if your DB instance is on the <i>EC2-Classic</i> platform, you will need to create a DB security group; if your DB instance is on the <i>EC2-VPC</i> platform, you will need to create a VPC security group.	Determining whether you are using the EC2-VPC or EC2-Classic platform (p. 1718) Controlling access with security groups (p. 1699)
Parameter Groups If your DB instance is going to require specific database parameters, you should create a parameter group before you create the DB instance.	Working with DB parameter groups (p. 228)
Importing and Exporting Data Establish procedures for importing or exporting data.	Importing data into a MariaDB DB instance (p. 616)
Replication You can offload read traffic from your source MariaDB DB instance by creating read replicas.	Working with read replicas (p. 278)
Connecting to Your DB Instance Connect to your DB instance using a standard SQL client application.	Connecting to a DB instance running the MariaDB database engine (p. 588)
Backup and Restore When you create your DB instance, you can configure it to take automated backups. You can also back up and restore your databases manually by using full backup files (.bak files).	Working with backups (p. 328)
Monitoring Monitor your MariaDB DB instance by using Amazon CloudWatch RDS metrics, events, and Enhanced Monitoring. View log files for your MariaDB DB instance.	Viewing DB instance metrics (p. 548) Viewing Amazon RDS events (p. 503)
Log Files You can access the log files for your MariaDB DB instance.	Accessing Amazon RDS database log files (p. 504) MariaDB database log files (p. 508)

There are also advanced administrative tasks for working with DB instances running MariaDB. For more information, see the following documentation:

- [Parameters for MariaDB \(p. 620\)](#)
- [MariaDB on Amazon RDS SQL reference \(p. 625\)](#)

MariaDB on Amazon RDS versions

For MariaDB, version numbers are organized as version X.Y.Z. In Amazon RDS terminology, X.Y denotes the major version, and Z is the minor version number. For Amazon RDS implementations, a version change is considered major if the major version number changes, for example going from version 10.4 to 10.5. A version change is considered minor if only the minor version number changes, for example going from version 10.3.20 to 10.3.23.

Amazon RDS currently supports the following versions of MariaDB:

Major version	Minor version
MariaDB 10.5	<ul style="list-style-type: none">• 10.5.8 (supported in all AWS Regions)
MariaDB 10.4	<ul style="list-style-type: none">• 10.4.13 (supported in all AWS Regions)
MariaDB 10.3	<ul style="list-style-type: none">• 10.3.23 (supported in all AWS Regions)• 10.3.20 (supported in all AWS Regions)• 10.3.13 (supported in all AWS Regions)• 10.3.8 (supported in all AWS Regions)
MariaDB 10.2	<ul style="list-style-type: none">• 10.2.32 (supported in all AWS Regions)• 10.2.21 (supported in all AWS Regions)• 10.2.15 (supported in all AWS Regions)• 10.2.12 (supported in all AWS Regions)• 10.2.11 (supported in all AWS Regions)
MariaDB 10.1	<ul style="list-style-type: none">• 10.1.34 (supported in all AWS Regions)• 10.1.31 (supported in all AWS Regions)• 10.1.26 (supported in all AWS Regions)• 10.1.23 (supported in all AWS Regions)• 10.1.19 (supported in all AWS Regions)• 10.1.14 (supported in all AWS Regions except US East (Ohio))
MariaDB 10.0	<ul style="list-style-type: none">• 10.0.35 (supported in all AWS Regions)• 10.0.34 (supported in all AWS Regions)• 10.0.32 (supported in all AWS Regions)• 10.0.31 (supported in all AWS Regions)• 10.0.28 (supported in all AWS Regions)• 10.0.24 (supported in all AWS Regions)• 10.0.17 (supported in all AWS Regions except US East (Ohio), Canada (Central), and Europe (London))

You can specify any currently supported MariaDB version when creating a new DB instance. You can specify the major version (such as MariaDB 10.5), and any supported minor version for the specified

major version. If no version is specified, Amazon RDS defaults to a supported version, typically the most recent version. If a major version is specified but a minor version is not, Amazon RDS defaults to a recent release of the major version you have specified. To see a list of supported versions, as well as defaults for newly created DB instances, use the [describe-db-engine-versions](#) AWS CLI command.

The default MariaDB version might vary by AWS Region. To create a DB instance with a specific minor version, specify the minor version during DB instance creation. You can determine the default minor version for an AWS Region using the following AWS CLI command:

```
aws rds describe-db-engine-versions --default-only --engine mariadb --engine-version major-engine-version --region region --query "*[].{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

Replace *major-engine-version* with the major engine version, and replace *region* with the AWS Region. For example, the following AWS CLI command returns the default MariaDB minor engine version for the 10.3 major version and the US West (Oregon) AWS Region (us-west-2):

```
aws rds describe-db-engine-versions --default-only --engine mariadb --engine-version 10.3 --region us-west-2 --query '*[].{Engine:Engine,EngineVersion:EngineVersion}' --output text
```

Deprecation of MariaDB versions 10.0 and 10.1

On May 18, 2021, Amazon RDS plans to deprecate support for MariaDB 10.0 and 10.1 using the following schedule, which includes upgrade recommendations. For more information, see [Upgrading the MariaDB DB engine \(p. 598\)](#).

Action or recommendation	Dates
We recommend that you upgrade MariaDB 10.0 and 10.1 DB instances manually to the version of your choice.	Now–May 18, 2021
We recommend that you upgrade MariaDB 10.0 and 10.1 snapshots manually to the version of your choice.	Now–May 18, 2021
You can no longer create new MariaDB 10.0 and 10.1 DB instances.	December 3, 2020
Amazon RDS starts automatic upgrades of your MariaDB 10.0 and 10.1 DB instances to version 10.2.	March 22, 2021
Amazon RDS starts automatic upgrades to version 10.2 for any MariaDB 10.0 and 10.1 DB instances restored from snapshots.	March 22, 2021
Amazon RDS automatically upgrades any remaining MariaDB 10.0 and 10.1 DB instances to version 10.2 whether or not they are in a maintenance window.	May 18, 2021

For more information, see [Announcement: Extending end-of-life Process for Amazon RDS for MariaDB 10.0 and 10.1](#).

For information about the Amazon RDS deprecation policy for MariaDB, see [Amazon RDS FAQs](#).

MariaDB feature support on Amazon RDS

In the following sections, find MariaDB feature support on Amazon RDS for MariaDB major versions:

Topics

- [MariaDB 10.5 support on Amazon RDS \(p. 578\)](#)
- [MariaDB 10.4 support on Amazon RDS \(p. 579\)](#)
- [MariaDB 10.3 support on Amazon RDS \(p. 579\)](#)
- [MariaDB 10.2 support on Amazon RDS \(p. 579\)](#)
- [MariaDB 10.1 support on Amazon RDS \(p. 580\)](#)
- [MariaDB 10.0 support on Amazon RDS \(p. 580\)](#)

For information about supported minor versions of Amazon RDS for MariaDB, see [MariaDB on Amazon RDS versions \(p. 576\)](#).

MariaDB 10.5 support on Amazon RDS

Amazon RDS supports the following new features for your DB instances running MariaDB version 10.5 or later:

- **InnoDB enhancements** – MariaDB version 10.5 includes InnoDB enhancements. For more information, see [InnoDB: Performance Improvements etc..](#)
- **Performance schema updates** – MariaDB version 10.5 includes performance schema updates. For more information, see [Performance Schema Updates to Match MySQL 5.7 Instrumentation and Tables](#).
- **One file in the InnoDB redo log** – In versions of MariaDB before version 10.5, the value of the `innodb_log_files_in_group` parameter was set to 2. In MariaDB version 10.5, the value of this parameter is set to 1.

If you are upgrading from a prior version to MariaDB version 10.5, and you don't modify the parameters, the `innodb_log_file_size` parameter value is unchanged. However, it applies to one log file instead of two. The result is that your upgraded MariaDB version 10.5 DB instance uses half of the redo log size that it was using before the upgrade. This change can have a noticeable performance impact. To address this issue, you can double the value of the `innodb_log_file_size` parameter. For information about modifying parameters, see [Modifying parameters in a DB parameter group \(p. 232\)](#).

- **SHOW SLAVE STATUS command not supported** – In versions of MariaDB before version 10.5, the `SHOW SLAVE STATUS` command required the `REPLICATION SLAVE` privilege. In MariaDB version 10.5, this command requires the `REPLICATION SLAVE ADMIN` privilege. This new privilege isn't granted to the RDS master user.

Instead of using the `SHOW SLAVE STATUS` command, run the new `mysql.rds_replica_status` stored procedure to return similar information. For more information, see [mysql.rds_replica_status \(p. 625\)](#).

- **SHOW RELAYLOG EVENTS command not supported** – In versions of MariaDB before version 10.5, the `SHOW RELAYLOG EVENTS` command required the `REPLICATION SLAVE` privilege. In MariaDB version 10.5, this command requires the `REPLICATION SLAVE ADMIN` privilege. This new privilege isn't granted to the RDS master user.
- **New default values for parameters** – The following parameters have new default values for MariaDB version 10.5 DB instances:
 - The default value of the `max_connections` parameter has changed to `LEAST({DBInstanceClassMemory/25165760}, 12000)`. For information about the `LEAST` parameter function, see [DB parameter functions \(p. 242\)](#).

- The default value of the `innodb_adaptive_hash_index` parameter has changed to `OFF` (0).
- The default value of the `innodb_checksum_algorithm` parameter has changed to `full_crc32`.
- The default value of the `innodb_log_file_size` parameter has changed to 2 GB.

For a list of all MariaDB 10.5 features and their documentation, see [Changes and improvements in MariaDB 10.5](#) and [Release notes - MariaDB 10.5 series](#) on the MariaDB website.

For a list of unsupported features, see [Features not supported \(p. 580\)](#).

MariaDB 10.4 support on Amazon RDS

Amazon RDS supports the following new features for your DB instances running MariaDB version 10.4 or later:

- **User account security enhancements** – Password expiration and `account locking` improvements
- **Optimizer enhancements** – Optimizer trace feature
- **InnoDB enhancements** – Instant `DROP COLUMN` support and instant VARCHAR extension for `ROW_FORMAT=DYNAMIC` and `ROW_FORMAT=COMPACT`
- **New parameters** – Including `tcp_nodelay`, `tls_version`, and `gtid_cleanup_batch_size`

For a list of all MariaDB 10.4 features and their documentation, see [Changes and improvements in MariaDB 10.4](#) and [Release notes - MariaDB 10.4 series](#) on the MariaDB website.

For a list of unsupported features, see [Features not supported \(p. 580\)](#).

MariaDB 10.3 support on Amazon RDS

Amazon RDS supports the following new features for your DB instances running MariaDB version 10.3 or later:

- **Oracle compatibility** – PL/SQL compatibility parser, sequences, `INTERSECT` and `EXCEPT` to complement `UNION`, new `TYPE OF` and `ROW TYPE OF` declarations, and invisible columns
- **Temporal data processing** – System versioned tables for querying of past and present states of the database
- **Flexibility** – User-defined aggregates, storage-independent column compression, and proxy protocol support to relay the client IP address to the server
- **Manageability** – Instant `ADD COLUMN` operations and fast-fail data definition language (DDL) operations

For a list of all MariaDB 10.3 features and their documentation, see [Changes & improvements in MariaDB 10.3](#) and [Release notes - MariaDB 10.3 series](#) on the MariaDB website.

For a list of unsupported features, see [Features not supported \(p. 580\)](#).

MariaDB 10.2 support on Amazon RDS

Amazon RDS supports the following new features for your DB instances running MariaDB version 10.2 or later:

- `ALTER USER`
- Common Table Expressions

- Compressing Events to Reduce Size of the Binary Log
- CREATE USER — new options for limiting resource usage and TLS/SSL
- EXECUTE IMMEDIATE
- Flashback
- InnoDB — now the default storage engine instead of XtraDB
- InnoDB — set the buffer pool size dynamically
- JSON Functions
- Window Functions
- WITH

For a list of all MariaDB 10.2 features and their documentation, see [Changes & improvements in MariaDB 10.2](#) and [Release notes - MariaDB 10.2 series](#) on the MariaDB website.

For a list of unsupported features, see [Features not supported \(p. 580\)](#).

MariaDB 10.1 support on Amazon RDS

Amazon RDS supports the following new features for your DB instances running MariaDB version 10.1 or later:

- Optimistic in-order parallel replication
- Page Compression
- XtraDB data scrubbing and defragmentation

For a list of all MariaDB 10.1 features and their documentation, see [Changes & improvements in MariaDB 10.1](#) and [Release notes - MariaDB 10.1 series](#) on the MariaDB website.

For a list of unsupported features, see [Features not supported \(p. 580\)](#).

MariaDB 10.0 support on Amazon RDS

For a list of all MariaDB 10.0 features and their documentation, see [Changes & improvements in MariaDB 10.0](#) and [Release notes - MariaDB 10.0 series](#) on the MariaDB website.

For a list of unsupported features, see [Features not supported \(p. 580\)](#).

Features not supported

The following MariaDB features are not supported on Amazon RDS:

- ColumnStore storage engine
- S3 storage engine
- Authentication plugin – GSSAPI
- Authentication plugin – Unix Socket
- AWS Key Management encryption plugin
- Delayed replication

- Native MariaDB encryption at rest for XtraDB, InnoDB, and Aria.

You can enable encryption at rest for a MariaDB DB instance by following the instructions in [Encrypting Amazon RDS resources \(p. 1630\)](#).

- HandlerSocket
- JSON table type
- MariaDB ColumnStore
- MariaDB Galera Cluster
- Multisource replication
- MyRocks storage engine
- Password validation plugin, `simple_password_check`, and `cracklib_password_check`
- Spider storage engine
- Sphinx storage engine
- TokuDB storage engine
- Storage engine-specific object attributes, as described in [Engine-defined new Table/Field/Index attributes](#) in the MariaDB documentation
- Table and tablespace encryption

To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances, and it restricts access to certain system procedures and tables that require advanced privileges. Amazon RDS supports access to databases on a DB instance using any standard SQL client application. Amazon RDS doesn't allow direct host access to a DB instance by using Telnet, Secure Shell (SSH), or Windows Remote Desktop Connection.

Supported storage engines for MariaDB on Amazon RDS

While MariaDB supports multiple storage engines with varying capabilities, not all of them are optimized for recovery and data durability. InnoDB (for version 10.2 and higher) and XtraDB (for version 10.0 and 10.1) are the recommended and supported storage engines for MariaDB DB instances on Amazon RDS. Amazon RDS features such as Point-In-Time Restore and snapshot restore require a recoverable storage engine and are supported only for the recommended storage engine for the MariaDB version. Amazon RDS also supports Aria, although using Aria might have a negative impact on recovery in the event of an instance failure. However, if you need to use spatial indexes to handle geographic data on MariaDB 10.1 or 10.0, you should use Aria because spatial indexes are not supported by XtraDB. On MariaDB 10.2 and higher, the InnoDB storage engine supports spatial indexes.

Other storage engines are not currently supported by Amazon RDS for MariaDB.

MariaDB file size limits in Amazon RDS

For MariaDB DB instances, the maximum provisioned storage limit constrains the size of a table to a maximum size of 16 TB when using InnoDB file-per-table tablespaces. This limit also constrains the system tablespace to a maximum size of 16 TB. InnoDB file-per-table tablespaces (with tables each in their own tablespace) are set by default for MariaDB DB instances. For more information, see [Amazon RDS DB instance storage \(p. 40\)](#).

There are advantages and disadvantages to using InnoDB file-per-table tablespaces, depending on your application. To determine the best approach for your application, see [File-per-table tablespaces](#) in the MySQL documentation.

We don't recommend allowing tables to grow to the maximum file size. In general, a better practice is to partition data into smaller tables, which can improve performance and recovery times.

One option that you can use for breaking a large table up into smaller tables is partitioning. *Partitioning* distributes portions of your large table into separate files based on rules that you specify. For example, if you store transactions by date, you can create partitioning rules that distribute older transactions into separate files using partitioning. Then periodically, you can archive the historical transaction data that doesn't need to be readily available to your application. For more information, see [Partitioning](#) in the MySQL documentation.

To determine the file size of a table

Use the following SQL command to determine if any of your tables are too large and are candidates for partitioning. To update table statistics, issue an `ANALYZE TABLE` command on each table. For more information, see [ANALYZE TABLE statement](#) in the MySQL documentation.

```
SELECT TABLE_SCHEMA, TABLE_NAME,
       round(((DATA_LENGTH + INDEX_LENGTH) / 1024 / 1024), 2) As "Approximate size (MB)",
       DATA_FREE
  FROM information_schema.TABLES
 WHERE TABLE_SCHEMA NOT IN ('mysql', 'information_schema', 'performance_schema');
```

To enable InnoDB file-per-table tablespaces

- To enable InnoDB file-per-table tablespaces, set the `innodb_file_per_table` parameter to 1 in the parameter group for the DB instance.

To disable InnoDB file-per-table tablespaces

- To disable InnoDB file-per-table tablespaces, set the `innodb_file_per_table` parameter to 0 in the parameter group for the DB instance.

For information on updating a parameter group, see [Working with DB parameter groups \(p. 228\)](#).

When you have enabled or disabled InnoDB file-per-table tablespaces, you can issue an `ALTER TABLE` command. You can use this command to move a table from the global tablespace to its own tablespace. Or you can move a table from its own tablespace to the global tablespace. Following is an example.

```
ALTER TABLE table_name ENGINE=InnoDB, ALGORITHM=COPY;
```

MariaDB security on Amazon RDS

Security for MariaDB DB instances is managed at three levels:

- AWS Identity and Access Management controls who can perform Amazon RDS management actions on DB instances. When you connect to AWS using IAM credentials, your IAM account must have IAM policies that grant the permissions required to perform Amazon RDS management operations. For more information, see [Identity and access management in Amazon RDS \(p. 1644\)](#).
- When you create a DB instance, you use either a VPC security group or a DB security group to control which devices and Amazon EC2 instances can open connections to the endpoint and port of the DB

instance. These connections can be made using Secure Socket Layer (SSL). In addition, firewall rules at your company can control whether devices running at your company can open connections to the DB instance.

- Once a connection has been opened to a MariaDB DB instance, authentication of the login and permissions are applied the same way as in a stand-alone instance of MariaDB. Commands such as `CREATE USER`, `RENAME USER`, `GRANT`, `REVOKE`, and `SET PASSWORD` work just as they do in stand-alone databases, as does directly modifying database schema tables.

When you create an Amazon RDS DB instance, the master user has the following default privileges:

- `alter`
- `alter routine`
- `create`
- `create routine`
- `create temporary tables`
- `create user`
- `create view`
- `delete`
- `drop`
- `event`
- `execute`
- `grant option`
- `index`
- `insert`
- `lock tables`
- `process`
- `references`
- `reload`

This privilege is limited on MariaDB DB instances. It doesn't grant access to the `FLUSH LOGS` or `FLUSH TABLES WITH READ LOCK` operations.

- `replication client`
- `replication slave`
- `select`
- `show databases`
- `show view`
- `trigger`
- `update`

For more information about these privileges, see [User account management](#) in the MariaDB documentation.

Note

Although you can delete the master user on a DB instance, we don't recommend doing so. To recreate the master user, use the `ModifyDBInstance` API or the `modify-db-instance` AWS CLI and specify a new master user password with the appropriate parameter. If the master user does not exist in the instance, the master user is created with the specified password.

To provide management services for each DB instance, the `rdsadmin` user is created when the DB instance is created. Attempting to drop, rename, change the password for, or change privileges for the `rdsadmin` account results in an error.

To allow management of the DB instance, the standard `kill` and `kill_query` commands have been restricted. The Amazon RDS commands `mysql.rds_kill`, `mysql.rds_kill_query`, and `mysql.rds_kill_query_id` are provided for use in MariaDB and also MySQL so that you can end user sessions or queries on DB instances.

Using SSL with a MariaDB DB instance

Amazon RDS supports Secure Sockets Layer (SSL) connections with DB instances running the MariaDB database engine.

Amazon RDS creates an SSL certificate and installs the certificate on the DB instance when Amazon RDS provisions the instance. These certificates are signed by a certificate authority. The SSL certificate includes the DB instance endpoint as the Common Name (CN) for the SSL certificate to guard against spoofing attacks.

For information about downloading certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

MariaDB uses yaSSL for secure connections in the following versions:

- MariaDB version 10.1.26 and earlier 10.1 versions
- MariaDB version 10.0.32 and earlier 10.0 versions

MariaDB uses OpenSSL for secure connections in the following versions:

- MariaDB 10.5 versions
- MariaDB 10.4 versions
- MariaDB 10.3 versions
- MariaDB 10.2 versions
- MariaDB version 10.1.31 and later 10.1 versions
- MariaDB version 10.0.34 and later 10.0 versions

Amazon RDS for MariaDB supports Transport Layer Security (TLS) versions 1.0, 1.1, and 1.2. The following table shows the TLS support for MySQL versions.

MariaDB version	TLS 1.0	TLS 1.1	TLS 1.2
MariaDB 10.5	Supported	Supported	Supported
MariaDB 10.4	Supported	Supported	Supported
MariaDB 10.3	Supported	Supported	Supported
MariaDB 10.2	Supported	Supported	Supported
MariaDB 10.1	Supported	Supported for 10.1.31 and later 10.1 versions	Supported for 10.1.31 and later 10.1 versions

MariaDB version	TLS 1.0	TLS 1.1	TLS 1.2
MariaDB 10.0	Supported	Supported for 10.0.34 and later 10.0 versions	Supported for 10.0.34 and later 10.0 versions

To encrypt connections using the default mysql client, launch the mysql client using the --ssl-ca parameter to reference the public key, as shown in the examples following.

The following example shows how to launch the client using the --ssl-ca parameter for MariaDB 10.2 and later.

```
mysql -h myinstance.c9akciq32.rds-us-east-1.amazonaws.com  
--ssl-ca=[full path]rds-combined-ca-bundle.pem --ssl-mode=REQUIRED
```

The following example shows how to launch the client using the --ssl-ca parameter for MariaDB 10.1 and earlier.

```
mysql -h myinstance.c9akciq32.rds-us-east-1.amazonaws.com  
--ssl-ca=[full path]rds-combined-ca-bundle.pem --ssl-verify-server-cert
```

You can require SSL connections for specific users accounts. For example, you can use one of the following statements, depending on your MariaDB version, to require SSL connections on the user account `encrypted_user`.

For MariaDB 10.2 and later, use the following statement.

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

For MariaDB 10.1 and earlier, use the following statement.

```
GRANT USAGE ON *.* TO 'encrypted_user'@'%' REQUIRE SSL;
```

For more information on SSL connections with MariaDB, see [SSL overview](#) in the MariaDB documentation.

Cache warming

InnoDB (version 10.2 and later) and XtraDB (versions 10.0 and 10.1) cache warming can provide performance gains for your MariaDB DB instance by saving the current state of the buffer pool when the DB instance is shut down, and then reloading the buffer pool from the saved information when the DB instance starts up. This approach bypasses the need for the buffer pool to "warm up" from normal database use and instead preloads the buffer pool with the pages for known common queries. For more information on cache warming, see [Dumping and restoring the buffer pool](#) in the MariaDB documentation.

Cache warming is enabled by default on MariaDB 10.2 and higher DB instances. To enable it, set the `innodb_buffer_pool_dump_at_shutdown` and `innodb_buffer_pool_load_at_startup`

parameters to 1 in the parameter group for your DB instance. Changing these parameter values in a parameter group affects all MariaDB DB instances that use that parameter group. To enable cache warming for specific MariaDB DB instances, you might need to create a new parameter group for those DB instances. For information on parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

Cache warming primarily provides a performance benefit for DB instances that use standard storage. If you use PIOPS storage, you don't commonly see a significant performance benefit.

Important

If your MariaDB DB instance doesn't shut down normally, such as during a failover, then the buffer pool state isn't saved to disk. In this case, MariaDB loads whatever buffer pool file is available when the DB instance is restarted. No harm is done, but the restored buffer pool might not reflect the most recent state of the buffer pool before the restart. To ensure that you have a recent state of the buffer pool available to warm the cache on startup, we recommend that you periodically dump the buffer pool "on demand." You can dump or load the buffer pool on demand.

You can create an event to dump the buffer pool automatically and at a regular interval. For example, the following statement creates an event named `periodic_buffer_pool_dump` that dumps the buffer pool every hour.

```
CREATE EVENT periodic_buffer_pool_dump
  ON SCHEDULE EVERY 1 HOUR
  DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

For more information, see [Events](#) in the MariaDB documentation.

Dumping and loading the buffer pool on demand

You can save and load the cache on demand using the following stored procedures:

- To dump the current state of the buffer pool to disk, call the [mysql.rds_innodb_buffer_pool_dump_now \(p. 971\)](#) stored procedure.
- To load the saved state of the buffer pool from disk, call the [mysql.rds_innodb_buffer_pool_load_now \(p. 971\)](#) stored procedure.
- To cancel a load operation in progress, call the [mysql.rds_innodb_buffer_pool_load_abort \(p. 972\)](#) stored procedure.

Database parameters for MariaDB

By default, a MariaDB DB instance uses a DB parameter group that is specific to a MariaDB database. This parameter group contains some but not all of the parameters contained in the Amazon RDS DB parameter groups for the MySQL database engine. It also contains a number of new, MariaDB-specific parameters. For more information on the parameters available for the RDS for MariaDB DB engine, see [Parameters for MariaDB \(p. 620\)](#).

Common DBA tasks for MariaDB

Ending sessions or queries, skipping replication errors, working with InnoDB (version 10.2 and later) and XtraDB (versions 10.0 and 10.1) tablespaces to improve crash recovery times, and managing the global status history are common DBA tasks you might perform in a MariaDB DB instance. You can handle these tasks just as in a MySQL DB instance, as described in [Common DBA tasks for MySQL DB](#)

[instances \(p. 933\)](#). The crash recovery instructions there refer to the MySQL InnoDB engine, but they are applicable to a MariaDB instance running InnoDB or XtraDB as well.

Local time zone for MariaDB DB instances

By default, the time zone for a MariaDB DB instance is Universal Time Coordinated (UTC). You can set the time zone for your DB instance to the local time zone for your application instead.

To set the local time zone for a DB instance, set the `time_zone` parameter in the parameter group for your DB instance to one of the supported values listed later in this section. When you set the `time_zone` parameter for a parameter group, all DB instances and read replicas that are using that parameter group change to use the new local time zone. For information on setting parameters in a parameter group, see [Working with DB parameter groups \(p. 228\)](#).

After you set the local time zone, all new connections to the database reflect the change. If you have any open connections to your database when you change the local time zone, you won't see the local time zone update until after you close the connection and open a new connection.

You can set a different local time zone for a DB instance and one or more of its read replicas. To do this, use a different parameter group for the DB instance and the replica or replicas and set the `time_zone` parameter in each parameter group to a different local time zone.

If you are replicating across AWS Regions, then the source DB instance and the read replica use different parameter groups (parameter groups are unique to an AWS Region). To use the same local time zone for each instance, you must set the `time_zone` parameter in the instance's and read replica's parameter groups.

When you restore a DB instance from a DB snapshot, the local time zone is set to UTC. You can update the time zone to your local time zone after the restore is complete. If you restore a DB instance to a point in time, then the local time zone for the restored DB instance is the time zone setting from the parameter group of the restored DB instance.

You can set your local time zone to one of the following values.

Africa/Cairo	Asia/Bangkok	Australia/Darwin
Africa/Casablanca	Asia/Beirut	Australia/Hobart
Africa/Harare	Asia/Calcutta	Australia/Perth
Africa/Monrovia	Asia/Damascus	Australia/Sydney
Africa/Nairobi	Asia/Dhaka	Brazil/East
Africa/Tripoli	Asia/Irkutsk	Canada/Newfoundland
Africa/Windhoek	Asia/Jerusalem	Canada/Saskatchewan
America/Araguaina	Asia/Kabul	Europe/Amsterdam
America/Asuncion	Asia/Karachi	Europe/Athens
America/Bogota	Asia/Kathmandu	Europe/Dublin
America/Caracas	Asia/Krasnoyarsk	Europe/Helsinki
America/Chihuahua	Asia/Magadan	Europe/Istanbul

America/Cuiaba	Asia/Muscat	Europe/Kaliningrad
America/Denver	Asia/Novosibirsk	Europe/Moscow
America/Fortaleza	Asia/Riyadh	Europe/Paris
America/Guatemala	Asia/Seoul	Europe/Prague
America/Halifax	Asia/Shanghai	Europe/Sarajevo
America/Manaus	Asia/Singapore	Pacific/Auckland
America/Matamoros	Asia/Taipei	Pacific/Fiji
America/Monterrey	Asia/Tehran	Pacific/Guam
America/Montevideo	Asia/Tokyo	Pacific/Honolulu
America/Phoenix	Asia/Ulaanbaatar	Pacific/Samoa
America/Santiago	Asia/Vladivostok	US/Alaska
America/Tijuana	Asia/Yakutsk	US/Central
Asia/Amman	Asia/Yerevan	US/Eastern
Asia/Ashgabat	Atlantic/Azores	US/East-Indiana
Asia/Baghdad	Australia/Adelaide	US/Pacific
Asia/Baku	Australia/Brisbane	UTC

Connecting to a DB instance running the MariaDB database engine

After Amazon RDS provisions your DB instance, you can use any standard MariaDB client application or utility to connect to the instance. In the connection string, you specify the Domain Name System (DNS) address from the DB instance endpoint as the host parameter. You also specify the port number from the DB instance endpoint as the port parameter.

You can connect to an Amazon RDS for MariaDB DB instance by using tools like the mysql command line utility. For more information on using the mysql utility, see [mysql command-line client](#) in the MariaDB documentation. One GUI-based application that you can use to connect is Heidi. For more information, see the [Download heidi](#) page.

To connect to a DB instance from outside of a virtual private cloud (VPC) based on Amazon VPC, the DB instance must be publicly accessible. Also, access must be granted using the inbound rules of the DB instance's security group, and other requirements must be met. For more information, see [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).

You can use SSL encryption on connections to a MariaDB DB instance. For information, see [Using SSL with a MariaDB DB instance \(p. 584\)](#).

Topics

- [Finding the connection information for a MariaDB DB instance \(p. 589\)](#)
- [Connecting from the mysql utility \(p. 591\)](#)

- [Connecting with SSL \(p. 592\)](#)
- [Troubleshooting connections to your MariaDB DB instance \(p. 592\)](#)

Finding the connection information for a MariaDB DB instance

The connection information for a DB instance includes its endpoint, port, and a valid database user, such as the master user. For example, suppose that an endpoint value is `mydb.123456789012.us-east-1.rds.amazonaws.com`. In this case, the port value is 3306, and the database user is `admin`. Given this information, you specify the following values in a connection string:

- For host or host name or DNS name, specify `mydb.123456789012.us-east-1.rds.amazonaws.com`.
- For port, specify 3306.
- For user, specify `admin`.

To connect to a DB instance, use any client for a DB engine. For example, you might use the `mysql` utility to connect to a MariaDB or MySQL DB instance. You might use Microsoft SQL Server Management Studio to connect to a SQL Server DB instance. You might use Oracle SQL Developer to connect to an Oracle DB instance, or the `psql` command line utility to connect to a PostgreSQL DB instance.

To find the connection information for a DB instance, you can use the AWS Management Console, the AWS Command Line Interface (AWS CLI) [describe-db-instances](#) command, or the Amazon RDS API [DescribeDBInstances](#) operation to list its details.

Console

To find the connection information for a DB instance in the AWS Management Console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases** to display a list of your DB instances.
3. Choose the name of the MariaDB DB instance to display its details.
4. On the **Connectivity & security** tab, copy the endpoint. Also, note the port number. You need both the endpoint and the port number to connect to the DB instance.

RDS > Databases > mydb

mydb

Summary

DB identifier	mydb	CPU	2.33%
Role	Instance	Current activity	0 Connections

Connectivity & security | Monitoring | Logs & events | Configuration

Connectivity & security

Endpoint & port	Network
Endpoint	Available
mydb. [REDACTED].us-east-1.rds.amazonaws.com	us-eas
Port	VPC
3306	vpc-65
Subnet	default
Security groups	[REDACTED]

5. If you need to find the master user name, choose the **Configuration** tab and view the **Master username** value.

AWS CLI

To find the connection information for a MariaDB DB instance by using the AWS CLI, call the [describe-db-instances](#) command. In the call, query for the DB instance ID, endpoint, port, and master user name.

For Linux, macOS, or Unix:

```
aws rds describe-db-instances \
--filters "Name=engine,Values=mariadb" \
--query "[].{DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername}"
```

For Windows:

```
aws rds describe-db-instances ^
--filters "Name=engine,Values=mariadb" ^
--query "[].{DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername}"
```

Your output should be similar to the following.

```
[  
  [  
    "mydb1",  
    "mydb1.123456789012.us-east-1.rds.amazonaws.com",  
    3306,  
    "admin"  
  ],  
  [  
    "mydb2",  
    "mydb2.123456789012.us-east-1.rds.amazonaws.com",  
    3306,  
    "admin"  
  ]  
]
```

RDS API

To find the connection information for a DB instance by using the Amazon RDS API, call the [DescribeDBInstances](#) operation. In the output, find the values for the endpoint address, endpoint port, and master user name.

Connecting from the mysql utility

To connect to a DB instance using the mysql utility, enter the following command at a command prompt on a client computer. Doing this connects you to a database on a MariaDB DB instance. Substitute the DNS name (endpoint) for your DB instance for `<endpoint>` and the master user name that you used for `<quartermaster>`. Provide the master password that you used when prompted for a password.

```
mysql -h <endpoint> -P 3306 -u <mymasteruser> -p
```

After you enter the password for the user, you see output similar to the following.

```
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 272  
Server version: 5.5.5-10.0.17-MariaDB-log MariaDB Server  
  
Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
|mysql >
```

Connecting with SSL

Amazon RDS creates an SSL certificate for your DB instance when the instance is created. If you enable SSL certificate verification, then the SSL certificate includes the DB instance endpoint as the Common Name (CN) for the SSL certificate to guard against spoofing attacks. To connect to your DB instance using SSL, follow these steps:

To connect to a DB instance with SSL using the mysql utility

1. Download a root certificate that works for all AWS Regions.

For information about downloading certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

2. Enter the following command at a command prompt to connect to a DB instance with SSL using the mysql utility. For the -h parameter, substitute the DNS name for your DB instance. For the --ssl-ca parameter, substitute the SSL certificate file name as appropriate.

```
mysql -h mariadb-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=rds-ca-2015-root.pem -p
```

3. Include the --ssl-verify-server-cert parameter so that the SSL connection verifies the DB instance endpoint against the endpoint in the SSL certificate. For example:

```
mysql -h mariadb-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=rds-ca-2015-root.pem --ssl-verify-server-cert -p
```

4. Enter the master user password when prompted.

You should see output similar to the following.

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 272
Server version: 5.5.5-10.0.17-MariaDB-log MariaDB Server

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql >
```

Troubleshooting connections to your MariaDB DB instance

Two common causes of connection failures to a new DB instance are the following:

- The DB instance was created using a security group that doesn't authorize connections from the device or Amazon EC2 instance where the MariaDB application or utility is running. If the DB instance was created in an Amazon VPC, it must have a VPC security group that authorizes the connections. For more information, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).

You can add or edit an inbound rule in the security group. For **Source**, choose **My IP**. This allows access to the DB instance from the IP address detected in your browser.

If the DB instance was created outside of a VPC, it must have a DB security group that authorizes the connections.

- The DB instance was created using the default port of 3306, and your company has firewall rules blocking connections to that port from devices in your company network. To fix this failure, recreate the instance with a different port.

For more information on connection issues, see [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).

Updating applications to connect to MariaDB DB instances using new SSL/TLS certificates

As of September 19, 2019, Amazon RDS has published new Certificate Authority (CA) certificates for connecting to your RDS DB instances using Secure Socket Layer or Transport Layer Security (SSL/TLS). Following, you can find information about updating your applications to use the new certificates.

This topic can help you to determine whether your applications require certificate verification to connect to your DB instances.

Note

Some applications are configured to connect to MariaDB only if they can successfully verify the certificate on the server. For such applications, you must update your client application trust stores to include the new CA certificates.

You can specify the following SSL modes: `disabled`, `preferred`, and `required`. When you use the `preferred` SSL mode and the CA certificate doesn't exist or isn't up to date, the following behavior applies:

- For MariaDB version 10.2 and higher, and newer minor versions of 10.0 and 10.1, the connection falls back to not using SSL and still connects successfully.

Because these later versions use the OpenSSL protocol, an expired server certificate doesn't prevent successful connections unless the `required` SSL mode is specified.

- For older MariaDB 10.0 and 10.1 minor versions, an error is returned.

Because these older versions use the yaSSL protocol, certificate verification is strictly enforced and the connection is unsuccessful.

The following MariaDB minor versions use the yaSSL protocol:

- 10.0.17, 10.0.24, 10.0.28, 10.0.31, 10.0.32
- 10.1.14, 10.1.19, 10.1.23, 10.1.26

After you update your CA certificates in the client application trust stores, you can rotate the certificates on your DB instances. We strongly recommend testing these procedures in a development or staging environment before implementing them in your production environments.

For more information about certificate rotation, see [Rotating your SSL/TLS certificate \(p. 1636\)](#). For more information about downloading certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#). For information about using SSL/TLS with MariaDB DB instances, see [Using SSL with a MariaDB DB instance \(p. 584\)](#).

Topics

- [Determining whether a client requires certificate verification in order to connect \(p. 594\)](#)
- [Updating your application trust store \(p. 595\)](#)
- [Example Java code for establishing SSL connections \(p. 597\)](#)

Determining whether a client requires certificate verification in order to connect

You can check whether JDBC clients and MySQL clients require certificate verification to connect.

JDBC

The following example with MySQL Connector/J 8.0 shows one way to check an application's JDBC connection properties to determine whether successful connections require a valid certificate. For more information on all of the JDBC connection options for MySQL, see [Configuration properties](#) in the MySQL documentation.

When using the MySQL Connector/J 8.0, an SSL connection requires verification against the server CA certificate if your connection properties have `sslMode` set to `VERIFY_CA` or `VERIFY_IDENTITY`, as in the following example.

```
Properties properties = new Properties();
properties.setProperty("sslMode", "VERIFY_IDENTITY");
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
```

Note

If you use either the MySQL Java Connector v5.1.38 or later, or the MySQL Java Connector v8.0.9 or later to connect to your databases, even if you haven't explicitly configured your applications to use SSL/TLS when connecting to your databases, these client drivers default to using SSL/TLS. In addition, when using SSL/TLS, they perform partial certificate verification and fail to connect if the database server certificate is expired.

MySQL

The following examples with the MySQL Client show two ways to check a script's MySQL connection to determine whether successful connections require a valid certificate. For more information on all of the connection options with the MySQL Client, see [Client-side configuration for encrypted connections](#) in the MySQL documentation.

When using the MySQL 5.7 or MySQL 8.0 Client, an SSL connection requires verification against the server CA certificate if for the `--ssl-mode` option you specify `VERIFY_CA` or `VERIFY_IDENTITY`, as in the following example.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem --
ssl-mode=VERIFY_CA
```

When using the MySQL 5.6 Client, an SSL connection requires verification against the server CA certificate if you specify the `--ssl-verify-server-cert` option, as in the following example.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem --
ssl-verify-server-cert
```

Updating your application trust store

For information about updating the trust store for MySQL applications, see [Using TLS/SSL with MariaDB Connector/J](#) in the MariaDB documentation.

Note

When you update the trust store, you can retain older certificates in addition to adding the new certificates.

Updating your application trust store for JDBC

You can update the trust store for applications that use JDBC for SSL/TLS connections.

To update the trust store for JDBC applications

1. Download the 2019 root certificate that works for all AWS Regions and put the file in the trust store directory.

For information about downloading the root certificate, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

2. Convert the certificate to .der format using the following command.

```
openssl x509 -outform der -in rds-ca-2019-root.pem -out rds-ca-2019-root.der
```

Replace the file name with the one that you downloaded.

3. Import the certificate into the key store using the following command.

```
keytool -import -alias rds-root -keystore clientkeystore -file rds-ca-2019-root.der
```

4. Confirm that the key store was updated successfully.

```
keytool -list -v -keystore clientkeystore.jks
```

Enter the key store password when you are prompted for it.

Your output should contain the following.

```
rds-root,date, trustedCertEntry,  
Certificate fingerprint (SHA1):  
D4:0D:DB:29:E3:75:0D:FF:A6:71:C3:14:0B:BF:5F:47:8D:1C:80:96  
# This fingerprint should match the output from the below command  
openssl x509 -fingerprint -in rds-ca-2019-root.pem -noout
```

If you are using the MariaDB Connector/J JDBC driver in an application, set the following properties in the application.

```
System.setProperty("javax.net.ssl.trustStore", certs);  
System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

When you start the application, set the following properties.

```
java -Djavax.net.ssl.trustStore=/path_to_truststore/MyTruststore.jks -  
Djavax.net.ssl.trustStorePassword=my_truststore_password com.companyName.MyApplication
```

Example Java code for establishing SSL connections

The following code example shows how to set up the SSL connection using JDBC.

```
private static final String DB_USER = "admin";  
  
private static final String DB_USER = "user name";  
private static final String DB_PASSWORD = "password";  
// This key store has only the prod root ca.  
private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";  
private static final String KEY_STORE_PASS = "keystore-password";  
  
public static void main(String[] args) throws Exception {  
    Class.forName("org.mariadb.jdbc.Driver");  
  
    System.setProperty("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);  
    System.setProperty("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);  
  
    Properties properties = new Properties();  
    properties.put("user", DB_USER);  
    properties.put("password", DB_PASSWORD);  
  
    Connection connection = DriverManager.getConnection("jdbc:mysql://ssl-mariadb-  
public.cn162e2e7kwh.us-east-1.rds.amazonaws.com:3306?useSSL=true", properties);  
    Statement stmt=connection.createStatement();  
  
    ResultSet rs=stmt.executeQuery("SELECT 1 from dual");  
  
    return;  
}
```

Important

After you have determined that your database connections use SSL/TLS and have updated your application trust store, you can update your database to use the rds-ca-2019 certificates. For instructions, see step 3 in [Updating your CA certificate by modifying your DB instance \(p. 1636\)](#).

Upgrading the MariaDB DB engine

When Amazon RDS supports a new version of a database engine, you can upgrade your DB instances to the new version. There are two kinds of upgrades for MariaDB DB instances: major version upgrades and minor version upgrades.

Major version upgrades can contain database changes that are not backward-compatible with existing applications. As a result, you must manually perform major version upgrades of your DB instances. You can initiate a major version upgrade by modifying your DB instance. However, before you perform a major version upgrade, we recommend that you follow the instructions in [Major version upgrades for MariaDB \(p. 599\)](#).

In contrast, *minor version upgrades* include only changes that are backward-compatible with existing applications. You can initiate a minor version upgrade manually by modifying your DB instance. Or you can enable the **Auto minor version upgrade** option when creating or modifying a DB instance. Doing so means that your DB instance is automatically upgraded after Amazon RDS tests and approves the new version. For information about performing an upgrade, see [Upgrading a DB instance engine version \(p. 271\)](#).

If your MariaDB DB instance is using read replicas, you must upgrade all of the read replicas before upgrading the source instance. If your DB instance is in a Multi-AZ deployment, both the writer and standby replicas are upgraded. Your DB instance might not be available until the upgrade is complete.

For more information about MariaDB supported versions and version management, see [MariaDB on Amazon RDS versions \(p. 576\)](#).

Topics

- [Overview of upgrading \(p. 598\)](#)
- [Major version upgrades for MariaDB \(p. 599\)](#)
- [Upgrading a MariaDB DB instance \(p. 600\)](#)
- [Automatic minor version upgrades for MariaDB \(p. 600\)](#)

Overview of upgrading

When you use the AWS Management Console to upgrade a DB instance, it shows the valid upgrade targets for the DB instance. You can also use the following AWS CLI command to identify the valid upgrade targets for a DB instance:

For Linux, macOS, or Unix:

```
aws rds describe-db-engine-versions \
--engine mariadb \
--engine-version version-number \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

For Windows:

```
aws rds describe-db-engine-versions ^
--engine mariadb ^
--engine-version version-number ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

For example, to identify the valid upgrade targets for a MariaDB version 10.1.19 DB instance, run the following AWS CLI command:

For Linux, macOS, or Unix:

```
aws rds describe-db-engine-versions \
--engine mariadb \
--engine-version 10.1.19 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

For Windows:

```
aws rds describe-db-engine-versions ^
--engine mariadb ^
--engine-version 10.1.19 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Amazon RDS takes two DB snapshots during the upgrade process. The first DB snapshot is of the DB instance before any upgrade changes have been made. If the upgrade doesn't work for your databases, you can restore this snapshot to create a DB instance running the old version. The second DB snapshot is taken when the upgrade completes.

Note

Amazon RDS only takes DB snapshots if you have set the backup retention period for your DB instance to a number greater than 0. To change your backup retention period, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

After the upgrade is complete, you can't revert to the previous version of the database engine. If you want to return to the previous version, restore the first DB snapshot taken to create a new DB instance.

You control when to upgrade your DB instance to a new version supported by Amazon RDS. This level of control helps you maintain compatibility with specific database versions and test new versions with your application before deploying in production. When you are ready, you can perform version upgrades at the times that best fit your schedule.

If your DB instance is using read replication, you must upgrade all of the Read Replicas before upgrading the source instance.

If your DB instance is in a Multi-AZ deployment, both the primary and standby DB instances are upgraded. The primary and standby DB instances are upgraded at the same time and you will experience an outage until the upgrade is complete. The time for the outage varies based on your database engine, engine version, and the size of your DB instance.

Major version upgrades for MariaDB

Major version upgrades can contain database changes that are not backward-compatible with existing applications. As a result, Amazon RDS doesn't apply major version upgrades automatically. You must manually modify your DB instance. We recommend that you thoroughly test any upgrade before applying it to your production instances.

Amazon RDS supports the following in-place upgrades for major versions of the MariaDB database engine:

- MariaDB 10.0 to MariaDB 10.1
- MariaDB 10.1 to MariaDB 10.2

- MariaDB 10.2 to MariaDB 10.3
- MariaDB 10.3 to MariaDB 10.4
- MariaDB 10.4 to MariaDB 10.5

To perform a major version upgrade for a MariaDB version 10.0 DB instance on Amazon RDS to MariaDB version 10.1 or later, first upgrade to each major version: 10.0 to 10.1, then 10.1 to 10.2, then 10.2 to 10.3, 10.3 to 10.4, and then 10.4 to 10.5.

If you are using a custom parameter group, and you perform a major version upgrade, you must specify either a default parameter group for the new DB engine version or create your own custom parameter group for the new DB engine version. Associating the new parameter group with the DB instance requires a customer-initiated database reboot after the upgrade completes. The instance's parameter group status will show pending-reboot if the instance needs to be rebooted to apply the parameter group changes. An instance's parameter group status can be viewed in the AWS console or by using a "describe" call such as `describe-db-instances`.

Upgrading a MariaDB DB instance

For information about manually or automatically upgrading a MariaDB DB instance, see [Upgrading a DB instance engine version \(p. 271\)](#).

Automatic minor version upgrades for MariaDB

If you specify the following settings when creating or modifying a DB instance, you can have your DB instance automatically upgraded.

- The **Auto minor version upgrade** setting is enabled.
- The **Backup retention period** setting is greater than 0.

For more information about these settings, see [Settings for DB instances \(p. 251\)](#).

For some RDS for MariaDB major versions in some AWS Regions, one minor version is designated by RDS as the automatic upgrade version. After a minor version has been tested and approved by Amazon RDS, the minor version upgrade occurs automatically during your maintenance window. RDS doesn't automatically set newer released minor versions as the automatic upgrade version. Before RDS designates a newer automatic upgrade version, several criteria are considered, such as the following:

- Known security issues
- Bugs in the MySQL community version
- Overall fleet stability since the minor version was released

You can use the following AWS CLI command and script to determine the current automatic minor upgrade target version for a specified MariaDB minor version in a specific AWS Region.

```
aws rds describe-db-engine-versions --output=table --engine mariadb --engine-version minor-version --region region
```

For example, the following AWS CLI command determines the automatic minor upgrade target for MariaDB minor version 10.2.11 in the US East (Ohio) AWS Region (us-east-2).

```
aws rds describe-db-engine-versions --output=table --engine mariadb --engine-version 10.2.11 --region us-east-2
```

Your output is similar to the following.

DescribeDBEngineVersions								
<hr/>								
<hr/>								
<hr/>								
DBEngineVersions								
<hr/>								
<hr/>								
<hr/>								
DBEngineDescription	MariaDb Community Edition							
DBEngineVersionDescription	MariaDB 10.2.11							
DBParameterGroupFamily	mariadb10.2							
Engine	mariadb							
EngineVersion	10.2.11							
Status	available							
SupportsGlobalDatabases	False							
SupportsExportsToCloudwatchLogs	True							
SupportsParallelQuery	False							
SupportsReadReplica	True							
<hr/>								
<hr/>								
<hr/>								
ExportableLogTypes								
<hr/>								
<hr/>								
<hr/>								
audit								
error								
general								
slowquery								
<hr/>								
<hr/>								
<hr/>								
ValidUpgradeTarget								
<hr/>								
<hr/>								
<hr/>								
AutoUpgrade	Description	Engine	EngineVersion	IsMajorVersionUpgrade				
False	MariaDB 10.2.12	mariadb	10.2.12	False				
False	MariaDB 10.2.15	mariadb	10.2.15	False				
True	MariaDB 10.2.21	mariadb	10.2.21	False				
False	MariaDB 10.3.8	mariadb	10.3.8	True				
False	MariaDB 10.3.20	mariadb	10.3.20	True				

False	MariaDB 10.3.23 mariadb 10.3.23	True
	-----+-----+-----+-----+	
+		

In this example, the `AutoUpgrade` value is `True` for MariaDB version 10.2.21. So, the automatic minor upgrade target is MariaDB version 10.2.21, which is highlighted in the output.

A MariaDB DB instance is automatically upgraded during your maintenance window if the following criteria are met:

- The **Auto minor version upgrade** setting is enabled.
- The **Backup retention period** setting is greater than 0.
- The DB instance is running a minor DB engine version that is less than the current automatic upgrade minor version.

For more information, see [Automatically upgrading the minor engine version \(p. 273\)](#).

Migrating data from a MySQL DB snapshot to a MariaDB DB instance

You can migrate an RDS for MySQL DB snapshot to a new DB instance running MariaDB 10.1 using the AWS CLI or Amazon RDS API. You must create the DB snapshot from an Amazon RDS DB instance running MySQL 5.6. To learn how to create an RDS for MySQL DB snapshot, see [Creating a DB snapshot \(p. 346\)](#).

After you migrate from MySQL to MariaDB, the MariaDB DB instance will be associated with the default DB parameter group and option group. After you restore the DB snapshot, you can associate a custom DB parameter group for the new DB instance. However, a MariaDB parameter group has a different set of configurable system variables. For information about the differences between MySQL and MariaDB system variables, see [System variable differences between MariaDB 10.1 and MySQL 5.6](#). To learn about DB parameter groups, see [Working with DB parameter groups \(p. 228\)](#). To learn about option groups, see [Working with option groups \(p. 212\)](#).

Incompatibilities between MariaDB and MySQL

Incompatibilities between MySQL and MariaDB include the following:

- You can't migrate a DB snapshot created with MySQL 5.5 to MariaDB 10.1.
- You can't migrate a DB snapshot created with MySQL 5.7 to MariaDB.
- You can't migrate a DB snapshot created with MySQL 8.0 to MariaDB.
- You can't migrate an encrypted snapshot.
- If the source MySQL database uses a SHA256 password hash, you need to reset user passwords that are SHA256 hashed before you can connect to the MariaDB database. The following code shows how to reset a password that is SHA256 hashed:

```
SET old_passwords = 0;
UPDATE mysql.user SET plugin = 'mysql_native_password',
Password = PASSWORD('new_password')
WHERE (User, Host) = ('master_user_name', %);
FLUSH PRIVILEGES;
```

- If your RDS master user account uses the SHA-256 password hash, you must reset the password using the AWS Management Console, the [modify-db-instance](#) AWS CLI command, or the [ModifyDBInstance](#) RDS API operation. For information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
- MariaDB doesn't support the Memcached plugin; however, the data used by the Memcached plugin is stored as InnoDB tables. After you migrate a MySQL DB snapshot, you can access the data used by the Memcached plugin using SQL. For more information about the innodb_memcache database, see [InnoDB memcached plugin internals](#).

Performing the migration

You can migrate an RDS for MySQL DB snapshot to a new MariaDB DB instance using the AWS CLI or the RDS API.

Note

You can't use the AWS Management Console to migrate an RDS for MySQL DB snapshot to a new MariaDB DB instance.

AWS CLI

To migrate data from a MySQL DB snapshot to a MariaDB DB instance, use the AWS CLI [restore-db-instance-from-db-snapshot](#) command with the following parameters:

- `--db-instance-identifier` – Name of the DB instance to create from the DB snapshot.
- `--db-snapshot-identifier` – The identifier for the DB snapshot to restore from.
- `--engine` – The database engine to use for the new instance.

Example

For Linux, macOS, or Unix:

```
aws rds restore-db-instance-from-db-snapshot \
  --db-instance-identifier newmariadbinstance \
  --db-snapshot-identifier mysqlsnapshot \
  --engine mariadb
```

For Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
  --db-instance-identifier newmariadbinstance ^
  --db-snapshot-identifier mysqlsnapshot ^
  --engine mariadb
```

API

To migrate data from a MySQL DB snapshot to a MariaDB DB instance, call the Amazon RDS API operation [RestoreDBInstanceFromDBSnapshot](#).

Working with MariaDB replication in Amazon RDS

You usually use read replicas to configure replication between Amazon RDS DB instances. For general information about read replicas, see [Working with read replicas \(p. 278\)](#). For specific information about working with read replicas on Amazon RDS for MariaDB, see [Working with MariaDB read replicas \(p. 605\)](#).

You can also configure replication based on binary log coordinates for a MariaDB DB instance. For MariaDB instances, you can also configure replication based on global transaction IDs (GTIDs), which provides better crash safety. For more information, see [Configuring GTID-based replication into a MariaDB DB instance \(p. 613\)](#).

The following are other replication options available with Amazon RDS for MariaDB:

- You can set up replication between an Amazon RDS for MariaDB DB instance and a MySQL or MariaDB instance that is external to Amazon RDS. For information about configuring replication with an external source, see [Replication with a MySQL or MariaDB instance running external to Amazon RDS \(p. 914\)](#).
- You can configure replication to import databases from a MySQL or MariaDB instance that is external to Amazon RDS, or to export databases to such instances. For more information, see [Importing data to an Amazon RDS MySQL or MariaDB DB instance with reduced downtime \(p. 881\)](#) and [Exporting data from a MySQL DB instance by using replication \(p. 921\)](#).

For any of these replication options, you can use either row-based replication, statement-based, or mixed replication. Row-based replication only replicates the changed rows that result from a SQL statement. Statement-based replication replicates the entire SQL statement. Mixed replication uses statement-based replication when possible, but switches to row-based replication when SQL statements that are unsafe for statement-based replication are run. In most cases, mixed replication is recommended. The binary log format of the DB instance determines whether replication is row-based, statement-based, or mixed. For information about setting the binary log format, see [Binary logging format \(p. 513\)](#).

Topics

- [Working with MariaDB read replicas \(p. 605\)](#)
- [Configuring GTID-based replication into a MariaDB DB instance \(p. 613\)](#)

Working with MariaDB read replicas

This section contains specific information about working with read replicas on Amazon RDS for MariaDB. For general information about read replicas and instructions for using them, see [Working with read replicas \(p. 278\)](#).

Topics

- [Read replica configuration with MariaDB \(p. 606\)](#)
- [Configuring replication filters with MariaDB \(p. 606\)](#)
- [Read replica updates with MariaDB \(p. 611\)](#)
- [Multi-AZ read replica deployments with MariaDB \(p. 611\)](#)
- [Monitoring MariaDB read replicas \(p. 611\)](#)
- [Starting and stopping replication with MariaDB read replicas \(p. 612\)](#)
- [Troubleshooting a MariaDB read replica problem \(p. 612\)](#)

Read replica configuration with MariaDB

Before a MariaDB DB instance can serve as a replication source, you must enable automatic backups on the source DB instance by setting the backup retention period to a value other than 0. This requirement also applies to a read replica that is the source DB instance for another read replica.

You can create up to five read replicas from one DB instance. For replication to operate effectively, each read replica should have as the same amount of compute and storage resources as the source DB instance. If you scale the source DB instance, also scale the read replicas.

If a read replica is running any version of MariaDB, you can specify it as the source DB instance for another read replica. For example, you can create ReadReplica1 from MyDBInstance, and then create ReadReplica2 from ReadReplica1. Updates made to MyDBInstance are replicated to ReadReplica1 and then replicated from ReadReplica1 to ReadReplica2. You can't have more than four instances involved in a replication chain. For example, you can create ReadReplica1 from MySourceDBInstance, and then create ReadReplica2 from ReadReplica1, and then create ReadReplica3 from ReadReplica2, but you can't create a ReadReplica4 from ReadReplica3.

If you promote a MariaDB read replica that is in turn replicating to other read replicas, those read replicas remain active. Consider an example where MyDBInstance1 replicates to MyDBInstance2, and MyDBInstance2 replicates to MyDBInstance3. If you promote MyDBInstance2, replication from MyDBInstance1 to MyDBInstance2 no longer occurs, but MyDBInstance2 still replicates to MyDBInstance3.

To enable automatic backups on a read replica for Amazon RDS for MariaDB, first create the read replica, then modify the read replica to enable automatic backups.

You can run multiple concurrent read replica create or delete actions that reference the same source DB instance, as long as you stay within the limit of five read replicas for the source instance.

Configuring replication filters with MariaDB

You can use replication filters to specify which databases and tables are replicated with a read replica. Replication filters can include databases and tables in replication or exclude them from replication.

The following are some use cases for replication filters:

- To reduce the size of a read replica. With replication filtering, you can exclude the databases and tables that aren't needed on the read replica.
- To exclude databases and tables from read replicas for security reasons.
- To replicate different databases and tables for specific use cases at different read replicas. For example, you might use specific read replicas for analytics or sharding.
- For a DB instance that has read replicas in different AWS Regions, to replicate different databases or tables in different AWS Regions.

Topics

- [Replication filtering parameters for Amazon RDS for MariaDB \(p. 606\)](#)
- [Replication filtering limitations for Amazon RDS for MariaDB \(p. 607\)](#)
- [Replication filtering examples for Amazon RDS for MariaDB \(p. 608\)](#)
- [Viewing the replication filters for a read replica \(p. 611\)](#)

Replication filtering parameters for Amazon RDS for MariaDB

To configure replication filters, set the following replication filtering parameters on the read replica:

- `replicate-do-db` – Replicate changes to the specified databases. When you set this parameter for a read replica, only the databases specified in the parameter are replicated.
- `replicate-ignore-db` – Don't replicate changes to the specified databases. When the `replicate-do-db` parameter is set for a read replica, this parameter isn't evaluated.
- `replicate-do-table` – Replicate changes to the specified tables. When you set this parameter for a read replica, only the tables specified in the parameter are replicated. Also, when the `replicate-do-db` or `replicate-ignore-db` parameter is set, the database that includes the specified tables must be included in replication with the read replica.
- `replicate-ignore-table` – Don't replicate changes to the specified tables. When the `replicate-do-table` parameter is set for a read replica, this parameter isn't evaluated.
- `replicate-wild-do-table` – Replicate tables based on the specified database and table name patterns. The % and _ wildcard characters are supported. When the `replicate-do-db` or `replicate-ignore-db` parameter is set, make sure to include the database that includes the specified tables in replication with the read replica.
- `replicate-wild-ignore-table` – Don't replicate tables based on the specified database and table name patterns. The % and _ wildcard characters are supported. When the `replicate-do-table` or `replicate-wild-do-table` parameter is set for a read replica, this parameter isn't evaluated.

The parameters are evaluated in the order that they are listed. For more information about how these parameters work, see [the MariaDB documentation](#).

By default, each of these parameters has an empty value. On each read replica, you can use these parameters to set, change, and delete replication filters. When you set one of these parameters, separate each filter from others with a comma.

You can use the % and _ wildcard characters in the `replicate-wild-do-table` and `replicate-wild-ignore-table` parameters. The % wildcard matches any number of characters, and the _ wildcard matches only one character.

The binary logging format of the source DB instance is important for replication because it determines the record of data changes. The setting of the `binlog_format` parameter determines whether the replication is row-based or statement-based. For more information, see [Binary logging format \(p. 513\)](#).

Note

All data definition language (DDL) statements are replicated as statements, regardless of the `binlog_format` setting on the source DB instance.

Replication filtering limitations for Amazon RDS for MariaDB

The following limitations apply to replication filtering for Amazon RDS for MariaDB:

- Each replication filtering parameter has a 2,000-character limit.
- Commas aren't supported in replication filters.
- The MariaDB `binlog_do_db` and `binlog_ignore_db` options for binary log filtering aren't supported.
- Replication filtering doesn't support XA transactions.

For more information, see [Restrictions on XA Transactions](#) in the MySQL documentation.

- Replication filtering is supported for Amazon RDS for MariaDB version 10.3.13 and higher 10.3 versions, all 10.4 versions, and all 10.5 versions.
- Replication filtering isn't supported for Amazon RDS for MariaDB version 10.0, 10.1, and 10.2.

Replication filtering examples for Amazon RDS for MariaDB

To configure replication filtering for a read replica, modify the replication filtering parameters in the parameter group associated with the read replica.

Note

You can't modify a default parameter group. If the read replica is using a default parameter group, create a new parameter group and associate it with the read replica. For more information on DB parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

You can set parameters in a parameter group using the AWS Management Console, AWS CLI, or RDS API. For information about setting parameters, see [Modifying parameters in a DB parameter group \(p. 232\)](#). When you set parameters in a parameter group, all of the DB instances associated with the parameter group use the parameter settings. If you set the replication filtering parameters in a parameter group, make sure that the parameter group is associated only with read replicas. Leave the replication filtering parameters empty for source DB instances.

The following examples set the parameters using the AWS CLI. These examples set `ApplyMethod` to `immediate` so that the parameter changes occur immediately after the CLI command completes. If you want a pending change to be applied after the read replica is rebooted, set `ApplyMethod` to `pending-reboot`.

The following examples set replication filters:

- [Including databases in replication](#)
- [Including tables in replication](#)
- [Including tables in replication with wildcard characters](#)
- [Escaping wildcard characters in names](#)
- [Excluding databases from replication](#)
- [Excluding tables from replication](#)
- [Excluding tables from replication using wildcard characters](#)

Example Including databases in replication

The following example includes the `mydb1` and `mydb2` databases in replication. When you set `replicate-do-db` for a read replica, only the databases specified in the parameter are replicated.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
"ApplyMethod": "immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
"ApplyMethod": "immediate"}]"
```

Example Including tables in replication

The following example includes the `table1` and `table2` tables in database `mydb1` in replication.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-do-table", "ParameterValue": "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-do-table", "ParameterValue": "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example Including tables in replication using wildcard characters

The following example includes tables with names that begin with `orders` and `returns` in database `mydb` in replication.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "mydb.orders%,mydb_returns%", "ApplyMethod":"immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "mydb.orders%,mydb_returns%", "ApplyMethod":"immediate"}]"
```

Example Escaping wildcard characters in names

The following example shows you how to use the escape character `\` to escape a wildcard character that is part of a name.

Assume that you have several table names in database `mydb1` that start with `my_table`, and you want to include these tables in replication. The table names include an underscore, which is also a wildcard character, so the example escapes the underscore in the table names.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my\_table%", "ApplyMethod":"immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my\_table%", "ApplyMethod":"immediate"}]"
```

Example Excluding databases from replication

The following example excludes the `mydb1` and `mydb2` databases from replication.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue": "mydb1,mydb2",
"ApplyMethod": "immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue": "mydb1,mydb2",
"ApplyMethod": "immediate"}]"
```

Example Excluding tables from replication

The following example excludes tables `table1` and `table2` in database `mydb1` from replication.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue": "mydb1.table1,mydb1.table2",
"ApplyMethod": "immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue": "mydb1.table1,mydb1.table2",
"ApplyMethod": "immediate"}]"
```

Example Excluding tables from replication using wildcard characters

The following example excludes tables with names that begin with `orders` and `returns` in database `mydb` from replication.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue": "mydb.orders%,mydb>Returns%",
"ApplyMethod": "immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
```

```
--parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue": "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Viewing the replication filters for a read replica

You can view the replication filters for a read replica in the following ways:

- Check the settings of the replication filtering parameters in the parameter group associated with the read replica.

For instructions, see [Viewing parameter values for a DB parameter group \(p. 239\)](#).

- In a MariaDB client, connect to the read replica and run the `SHOW SLAVE STATUS` statement.

In the output, the following fields show the replication filters for the read replica:

- `Replicate_Do_DB`
- `Replicate_Ignore_DB`
- `Replicate_Do_Table`
- `Replicate_Ignore_Table`
- `Replicate_Wild_Do_Table`
- `Replicate_Wild_Ignore_Table`

For more information about these fields, see [Checking Replication Status](#) in the MySQL documentation.

Read replica updates with MariaDB

Read replicas are designed to support read queries, but you might need occasional updates. For example, you might need to add an index to speed the specific types of queries accessing the replica. You can enable updates by setting the `read_only` parameter to `0` in the DB parameter group for the read replica.

Multi-AZ read replica deployments with MariaDB

You can create a read replica from either single-AZ or Multi-AZ DB instance deployments. You use Multi-AZ deployments to improve the durability and availability of critical data, but you can't use the Multi-AZ secondary to serve read-only queries. Instead, you can create read replicas from high-traffic Multi-AZ DB instances to offload read-only queries. If the source instance of a Multi-AZ deployment fails over to the secondary, any associated read replicas automatically switch to use the secondary (now primary) as their replication source. For more information, see [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).

You can create a read replica as a Multi-AZ DB instance. Amazon RDS creates a standby of your replica in another Availability Zone for failover support for the replica. Creating your read replica as a Multi-AZ DB instance is independent of whether the source database is a Multi-AZ DB instance.

Monitoring MariaDB read replicas

For MariaDB read replicas, you can monitor replication lag in Amazon CloudWatch by viewing the Amazon RDS `ReplicaLag` metric. The `ReplicaLag` metric reports the value of the `Seconds_Behind_Master` field of the `SHOW SLAVE STATUS` command.

Common causes for replication lag for MariaDB are the following:

- A network outage.
- Writing to tables with indexes on a read replica. If the `read_only` parameter is not set to `0` on the read replica, it can break replication.

- Using a nontransactional storage engine such as MyISAM. Replication is only supported for the InnoDB storage engine on MariaDB 10.2 and later and the XtraDB storage engine on MariaDB 10.1 and earlier.

When the `ReplicaLag` metric reaches 0, the replica has caught up to the source DB instance. If the `ReplicaLag` metric returns -1, then replication is currently not active. `ReplicaLag = -1` is equivalent to `Seconds_Behind_Master = NULL`.

Starting and stopping replication with MariaDB read replicas

You can stop and restart the replication process on an Amazon RDS DB instance by calling the system stored procedures [mysql.rds_stop_replication \(p. 967\)](#) and [mysql.rds_start_replication \(p. 964\)](#). You can do this when replicating between two Amazon RDS instances for long-running operations such as creating large indexes. You also need to stop and start replication when importing or exporting databases. For more information, see [Importing data to an Amazon RDS MySQL or MariaDB DB instance with reduced downtime \(p. 881\)](#) and [Exporting data from a MySQL DB instance by using replication \(p. 921\)](#).

If replication is stopped for more than 30 consecutive days, either manually or due to a replication error, Amazon RDS ends replication between the source DB instance and all read replicas. It does so to prevent increased storage requirements on the source DB instance and long failover times. The read replica DB instance is still available. However, replication can't be resumed because the binary logs required by the read replica are deleted from the source DB instance after replication is ended. You can create a new read replica for the source DB instance to reestablish replication.

Troubleshooting a MariaDB read replica problem

The replication technologies for MariaDB are asynchronous. Because they are asynchronous, occasional `BinLogDiskUsage` increases on the source DB instance and `ReplicaLag` on the read replica are to be expected. For example, a high volume of write operations to the source DB instance can occur in parallel. In contrast, write operations to the read replica are serialized using a single I/O thread, which can lead to a lag between the source instance and read replica. For more information about read-only replicas in the MariaDB documentation, go to [Replication overview](#).

You can do several things to reduce the lag between updates to a source DB instance and the subsequent updates to the read replica, such as the following:

- Sizing a read replica to have a storage size and DB instance class comparable to the source DB instance.
- Ensuring that parameter settings in the DB parameter groups used by the source DB instance and the read replica are compatible. For more information and an example, see the discussion of the `max_allowed_packet` parameter later in this section.

Amazon RDS monitors the replication status of your read replicas and updates the `Replication State` field of the read replica instance to `Error` if replication stops for any reason. An example might be if DML queries run on your read replica conflict with the updates made on the source DB instance.

You can review the details of the associated error thrown by the MariaDB engine by viewing the `Replication_Error` field. Events that indicate the status of the read replica are also generated, including [RDS-EVENT-0045 \(p. 492\)](#), [RDS-EVENT-0046 \(p. 492\)](#), and [RDS-EVENT-0047 \(p. 491\)](#). For more information about events and subscribing to events, see [Using Amazon RDS event notification \(p. 487\)](#). If a MariaDB error message is returned, review the error in the [MariaDB error message documentation](#).

One common issue that can cause replication errors is when the value for the `max_allowed_packet` parameter for a read replica is less than the `max_allowed_packet` parameter for the source DB instance. The `max_allowed_packet` parameter is a custom parameter that you can set in a DB

parameter group that is used to specify the maximum size of DML code that can be run on the database. In some cases, the `max_allowed_packet` parameter value in the DB parameter group associated with a source DB instance is smaller than the `max_allowed_packet` parameter value in the DB parameter group associated with the source's read replica. In these cases, the replication process can throw an error (Packet bigger than '`max_allowed_packet`' bytes) and stop replication. You can fix the error by having the source and read replica use DB parameter groups with the same `max_allowed_packet` parameter values.

Other common situations that can cause replication errors include the following:

- Writing to tables on a read replica. If you are creating indexes on a read replica, you need to have the `read_only` parameter set to **0** to create the indexes. If you are writing to tables on the read replica, it might break replication.
- Using a non-transactional storage engine such as MyISAM. read replicas require a transactional storage engine. Replication is only supported for the InnoDB storage engine on MariaDB 10.2 and later and the XtraDB storage engine on MariaDB 10.1 and earlier.
- Using unsafe nondeterministic queries such as `SYSDATE()`. For more information, see [Determination of safe and unsafe statements in binary logging](#).

If you decide that you can safely skip an error, you can follow the steps described in the section [Skipping the current replication error \(p. 933\)](#). Otherwise, you can delete the read replica and create an instance using the same DB instance identifier so that the endpoint remains the same as that of your old read replica. If a replication error is fixed, the `Replication State` changes to *replicating*.

For MariaDB DB instances, in some cases read replicas can't be switched to the secondary if some binlog events aren't flushed during the failure. In these cases, you must manually delete and recreate the read replicas. You can reduce the chance of this happening by setting the following parameter values: `sync_binlog=1` and `innodb_flush_log_at_trx_commit=1`. These settings might reduce performance, so test their impact before implementing the changes in a production environment.

Configuring GTID-based replication into a MariaDB DB instance

You can set up replication based on global transaction identifiers (GTIDs) from an external MariaDB instance of version 10.0.24 or greater into a MariaDB DB instance. Follow these guidelines when you set up an external source instance and a replica on Amazon RDS:

- Monitor failover events for the Amazon RDS for MariaDB DB instance that is your replica. If a failover occurs, then the DB instance that is your replica might be recreated on a new host with a different network address. For information on how to monitor failover events, see [Using Amazon RDS event notification \(p. 487\)](#).
- Maintain the binlogs on your source instance until you have verified that they have been applied to the replica. This maintenance ensures that you can restore your source instance in the event of a failure.
- Turn on automated backups on your MariaDB DB instance on Amazon RDS. Turning on automated backups ensures that you can restore your replica to a particular point in time if you need to resynchronize your source instance and replica. For information on backups and Point-In-Time Restore, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#).

Note

The permissions required to start replication on a MariaDB DB instance are restricted and not available to your Amazon RDS master user. Because of this, you must use the Amazon RDS `mysql.rds_set_external_master_gtid` (p. 626) and `mysql.rds_start_replication` (p. 964) commands to set up replication between your live database and your Amazon RDS for MariaDB database.

To start replication between an external source instance and a MariaDB DB instance on Amazon RDS, use the following procedure.

To start replication

1. Make the source MariaDB instance read-only:

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

2. Get the current GTID of the external MariaDB instance. You can do this by using mysql or the query editor of your choice to run `SELECT @@gtid_current_pos;`.

The GTID is formatted as <domain-id>-<server-id>-<sequence-id>. A typical GTID looks something like **0-1234510749-1728**. For more information about GTIDs and their component parts, see [Global transaction ID](#) in the MariaDB documentation.

3. Copy the database from the external MariaDB instance to the MariaDB DB instance using mysqldump. For very large databases, you might want to use the procedure in [Importing data to an Amazon RDS MySQL or MariaDB DB instance with reduced downtime \(p. 881\)](#).

Note

Make sure there is not a space between the `-p` option and the entered password.

For Linux, macOS, or Unix:

```
mysqldump \
--databases <database_name> \
--single-transaction \
--compress \
--order-by-primary \
-u <local_user> \
-p<local_password> | mysql \
--host=hostname \
--port=3306 \
-u <RDS_user_name> \
-p <RDS_password>
```

For Windows:

```
mysqldump ^
--databases <database_name> ^
--single-transaction ^
--compress ^
--order-by-primary \
-u <local_user> \
-p<local_password> | mysql ^
--host=hostname ^
--port=3306 ^
-u <RDS_user_name> ^
-p <RDS_password>
```

Use the `--host`, `--user` (`-u`), `--port` and `-p` options in the mysql command to specify the host name, user name, port, and password to connect to your MariaDB DB instance. The host name is the DNS name from the MariaDB DB instance endpoint, for example `myinstance.123456789012.us-east-1.rds.amazonaws.com`. You can find the endpoint value in the instance details in the [Amazon RDS Management Console](#).

4. Make the source MariaDB instance writeable again.

```
mysql> SET GLOBAL read_only = OFF;
```

```
mysql> UNLOCK TABLES;
```

5. In the Amazon RDS Management Console, add the IP address of the server that hosts the external MariaDB database to the VPC security group for the MariaDB DB instance. For more information on modifying a VPC security group, go to [Security groups for your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

The IP address can change when the following conditions are met:

- You are using a public IP address for communication between the external source instance and the DB instance.
- The external source instance was stopped and restarted.

If these conditions are met, verify the IP address before adding it.

You might also need to configure your local network to permit connections from the IP address of your MariaDB DB instance, so that it can communicate with your external MariaDB instance. To find the IP address of the MariaDB DB instance, use the host command.

```
host <RDS_MariaDB_DB_host_name>
```

The host name is the DNS name from the MariaDB DB instance endpoint.

6. Using the client of your choice, connect to the external MariaDB instance and create a MariaDB user to be used for replication. This account is used solely for replication and must be restricted to your domain to improve security. The following is an example.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY '<password>;
```

7. For the external MariaDB instance, grant REPLICATION CLIENT and REPLICATION SLAVE privileges to your replication user. For example, to grant the REPLICATION CLIENT and REPLICATION SLAVE privileges on all databases for the 'repl_user' user for your domain, issue the following command.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.*  
TO 'repl_user'@'mydomain.com'  
IDENTIFIED BY '<password>;
```

8. Make the MariaDB DB instance the replica. Connect to the MariaDB DB instance as the master user and identify the external MariaDB database as the replication source instance by using the [mysql.rds_set_external_master_gtid \(p. 626\)](#) command. Use the GTID that you determined in Step 2. The following is an example.

```
CALL mysql.rds_set_external_master_gtid ('mymasterserver.mydomain.com', 3306,  
'repl_user', '<password>', '<GTID>', 0);
```

9. On the MariaDB DB instance, issue the [mysql.rds_start_replication \(p. 964\)](#) command to start replication.

```
CALL mysql.rds_start_replication;
```

Importing data into a MariaDB DB instance

Following, you can find information about methods to import your MariaDB data to an Amazon RDS DB instance running MariaDB.

To do an initial data import into a MariaDB DB instance, you can use the procedures documented in [Restoring a backup into a MySQL DB instance \(p. 871\)](#), as follows:

- To move data from a MySQL DB instance, a MariaDB or MySQL instance in Amazon Elastic Compute Cloud (Amazon EC2) in the same VPC as your MariaDB DB instance, or a small on-premises instance of MariaDB or MySQL, you can use the procedure documented in [Importing data from a MySQL or MariaDB DB to a MySQL or MariaDB DB instance \(p. 879\)](#).
- To move data from a large or production on-premises instance of MariaDB or MySQL, you can use the procedure documented in [Importing data to an Amazon RDS MySQL or MariaDB DB instance with reduced downtime \(p. 881\)](#).
- To move data from an instance of MariaDB or MySQL that is in EC2 in a different VPC than your MariaDB DB instance, or to move data from any data source that can output delimited text files, you can use the procedure documented in [Importing data from any source to a MySQL or MariaDB DB instance \(p. 894\)](#).

You can also use AWS Database Migration Service (AWS DMS) to import data into an Amazon RDS DB instance. AWS DMS can migrate databases without downtime and, for many database engines, continue ongoing replication until you are ready to switch over to the target database. You can migrate to MariaDB from either the same database engine or a different database engine using AWS DMS. If you are migrating from a different database engine, you can use the AWS Schema Conversion Tool to migrate schema objects that are not migrated by AWS DMS. For more information about AWS DMS, see [What is AWS Database Migration Service](#).

You can configure replication into a MariaDB DB instance using MariaDB global transaction identifiers (GTIDs) when the external instance is MariaDB version 10.0.24 or greater, or using binary log coordinates for MySQL instances or MariaDB instances on earlier versions than 10.0.24. Note that MariaDB GTIDs are implemented differently than MySQL GTIDs, which are not supported by Amazon RDS.

To configure replication into a MariaDB DB instance, you can use the following procedures:

- To configure replication into a MariaDB DB instance from an external MySQL instance or an external MariaDB instance running a version prior to 10.0.24, you can use the procedure documented in [Replication with a MySQL or MariaDB instance running external to Amazon RDS \(p. 914\)](#).
- To configure replication into a MariaDB DB instance from an external MariaDB instance running version 10.0.24 or greater, you can use the procedure documented in [Configuring GTID-based replication into a MariaDB DB instance \(p. 613\)](#).

Note

The mysql system database contains authentication and authorization information required to log into your DB instance and access your data. Dropping, altering, renaming, or truncating tables, data, or other contents of the mysql database in your DB instance can result in errors and might render the DB instance and your data inaccessible. If this occurs, the DB instance can be restored from a snapshot using the AWS CLI `restore-db-instance-from-db-snapshot` or recovered using `restore-db-instance-to-point-in-time` commands.

Options for MariaDB database engine

This appendix describes options, or additional features, that are available for Amazon RDS instances running the MariaDB DB engine. To enable these options, you add them to a custom option group, and

then associate the option group with your DB instance. For more information about working with option groups, see [Working with option groups \(p. 212\)](#).

Amazon RDS supports the following options for MariaDB:

Option ID	Engine versions
MARIADB_AUDIT_PLUGIN	MariaDB 10.0.24 and later

MariaDB Audit Plugin support

Amazon RDS supports using the MariaDB Audit Plugin on MariaDB database instances. The MariaDB Audit Plugin records database activity such as users logging on to the database, queries run against the database, and more. The record of database activity is stored in a log file.

Audit Plugin option settings

Amazon RDS supports the following settings for the MariaDB Audit Plugin option.

Option setting	Valid values	Default value	Description
			<ul style="list-style-type: none"> • <code>QUERY_DCL</code>: Similar to the <code>QUERY</code> event, but returns only data control language (DCL) queries (<code>GRANT</code>, <code>REVOKE</code>, and so on).
<code>SERVER_AUDIT_INCL_USERS</code>	Multiple users comma-separated values	None	Include only activity from the specified users. By default, activity is recorded for all users. If a user is specified in both <code>SERVER_AUDIT_EXCL_USERS</code> and <code>SERVER_AUDIT_INCL_USERS</code> , then activity is recorded for the user.
<code>SERVER_AUDIT_EXCL_USERS</code>	Multiple users comma-separated values	None	Exclude activity from the specified users. By default, activity is recorded for all users. If a user is specified in both <code>SERVER_AUDIT_EXCL_USERS</code> and <code>SERVER_AUDIT_INCL_USERS</code> , then activity is recorded for the user.
			<p>The <code>rdsadmin</code> user queries the database every second to check the health of the database. Depending on your other settings, this activity can possibly cause the size of your log file to grow very large, very quickly. If you don't need to record this activity, add the <code>rdsadmin</code> user to the <code>SERVER_AUDIT_EXCL_USERS</code> list.</p> <p>Note CONNECT activity is always recorded for all users, even if the user is specified for this option setting.</p>
<code>SERVER_AUDIT_LOGGING</code>	ON	ON	Logging is active. The only valid value is ON. Amazon RDS does not support deactivating logging. If you want to deactivate logging, remove the MariaDB Audit Plugin. For more information, see Removing the MariaDB Audit Plugin (p. 619) .
<code>SERVER_AUDIT_QUERY_LENGTH</code>	0 to 147483647	1024	The limit on the length of the query string in a record.

Adding the MariaDB Audit Plugin

The general process for adding the MariaDB Audit Plugin to a DB instance is the following:

1. Create a new option group, or copy or modify an existing option group.
2. Add the option to the option group.
3. Associate the option group with the DB instance.

After you add the MariaDB Audit Plugin, you don't need to restart your DB instance. As soon as the option group is active, auditing begins immediately.

To add the MariaDB Audit Plugin

1. Determine the option group you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group. Choose `mariadb` for **Engine**, and choose **10.0** or later for **Major engine version**. For more information, see [Creating an option group \(p. 214\)](#).

2. Add the **MARIADB_AUDIT_PLUGIN** option to the option group, and configure the option settings. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#). For more information about each setting, see [Audit Plugin option settings \(p. 617\)](#).
3. Apply the option group to a new or existing DB instance.
 - For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
 - For an existing DB instance, you apply the option group by modifying the DB instance and attaching the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Viewing and downloading the MariaDB Audit Plugin log

After you enable the MariaDB Audit Plugin, you access the results in the log files the same way you access any other text-based log files. The audit log files are located at `/rdsdbdata/log/audit/`. For information about viewing the log file in the console, see [Viewing and listing database log files \(p. 504\)](#). For information about downloading the log file, see [Downloading a database log file \(p. 504\)](#).

Modifying MariaDB Audit Plugin settings

After you enable the MariaDB Audit Plugin, you can modify settings for the plugin. For more information about how to modify option settings, see [Modifying an option setting \(p. 221\)](#). For more information about each setting, see [Audit Plugin option settings \(p. 617\)](#).

Removing the MariaDB Audit Plugin

Amazon RDS doesn't support turning off logging in the MariaDB Audit Plugin. However, you can remove the plugin from a DB instance. When you remove the MariaDB Audit Plugin, the DB instance is restarted automatically to stop auditing.

To remove the MariaDB Audit Plugin from a DB instance, do one of the following:

- Remove the MariaDB Audit Plugin option from the option group it belongs to. This change affects all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).
- Modify the DB instance and specify a different option group that doesn't include the plugin. This change affects a single DB instance. You can specify the default (empty) option group, or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Parameters for MariaDB

By default, a MariaDB DB instance uses a DB parameter group that is specific to a MariaDB database. This parameter group contains some but not all of the parameters contained in the Amazon RDS DB parameter groups for the MySQL database engine. It also contains a number of new, MariaDB-specific parameters. For information about working with parameter groups and setting parameters, see [Working with DB parameter groups \(p. 228\)](#).

The following MySQL parameters are not available in MariaDB-specific DB parameter groups:

- bind_address
- binlog_error_action
- binlog_gtid_simple_recovery
- binlog_max_flush_queue_time
- binlog_order_commits
- binlog_row_image
- binlog_rows_query_log_events
- binlogging_impossible_mode
- block_encryption_mode
- core_file
- default_tmp_storage_engine
- div_precision_increment
- end_markers_in_json
- enforce_gtid_consistency
- eq_range_index_dive_limit
- explicit_defaults_for_timestamp
- gtid_executed
- gtid-mode
- gtid_next
- gtid_owned
- gtid_purged
- log_bin_basename
- log_bin_index
- log_bin_use_v1_row_events
- log_slow_admin_statements
- log_slow_slave_statements
- log_throttle_queries_not_using_indexes
- master-info-repository
- optimizer_trace
- optimizer_trace_features
- optimizer_trace_limit
- optimizer_trace_max_mem_size
- optimizer_trace_offset
- relay_log_info_repository
- rpl_stop_slave_timeout
- slave_parallel_workers
- slave_pending_jobs_size_max

- slave_rows_search_algorithms
- storage_engine
- table_open_cache_instances
- timed_mutexes
- transaction_allow_batching
- validate-password
- validate_password_dictionary_file
- validate_password_length
- validate_password_mixed_case_count
- validate_password_number_count
- validate_password_policy
- validate_password_special_char_count

For more information on MySQL parameters, go to the [MySQL documentation](#).

The MariaDB-specific DB parameter groups also contain the following parameters that are applicable to MariaDB only. Acceptable ranges for the modifiable parameters are the same as specified in the MariaDB documentation except where noted. RDS for MariaDB parameters are set to the default values of the storage engine you have selected.

- aria_block_size
- aria_checkpoint_interval
- aria_checkpoint_log_activity
- aria_force_start_after_recovery_failures
- aria_group_commit
- aria_group_commit_interval
- aria_log_dir_path
- aria_log_file_size
- aria_log_purge_type
- aria_max_sort_file_size
- aria_page_checksum
- aria_pagecache_age_threshold
- aria_pagecache_division_limit
- aria_recover

RDS for MariaDB supports the values of NORMAL, OFF, and QUICK, but not FORCE or BACKUP.

- aria_repair_threads
- aria_sort_buffer_size
- aria_stats_method
- aria_sync_log_dir
- binlog_annotation_row_events
- binlog_commit_wait_count
- binlog_commit_wait_usec
- binlog_row_image (MariaDB version 10.1 and later)
- deadlock_search_depth_long
- deadlock_search_depth_short
- deadlock_timeout_long
- deadlock_timeout_short

- `explicit_defaults_for_timestamp` (MariaDB version 10.1 and later)
- `extra_max_connections`
- `extra_port`
- `feedback`
- `feedback_send_retry_wait`
- `feedback_send_timeout`
- `feedback_url`
- `feedback_user_info`
- `gtid_domain_id`
- `gtid_strict_mode`
- `histogram_size`
- `histogram_type`
- `innodb_adaptive_hash_index_partitions`
- `innodb_background_scrub_data_check_interval` (MariaDB version 10.1 and later)
- `innodb_background_scrub_data_compressed` (MariaDB version 10.1 and later)
- `innodb_background_scrub_data_interval` (MariaDB version 10.1 and later)
- `innodb_background_scrub_data_uncompressed` (MariaDB version 10.1 and later)
- `innodb_buf_dump_status_frequency` (MariaDB version 10.1 and later)
- `innodb_buffer_pool_populate`
- `innodb_cleaner_lsn_age_factor`
- `innodb_compression_algorithm` (MariaDB version 10.1 and later)
- `innodb_corrupt_table_action`
- `innodb_defragment` (MariaDB version 10.1 and later)
- `innodb_defragment_fill_factor` (MariaDB version 10.1 and later)
- `innodb_defragment_fill_factor_n_recs` (MariaDB version 10.1 and later)
- `innodb_defragment_frequency` (MariaDB version 10.1 and later)
- `innodb_defragment_n_pages` (MariaDB version 10.1 and later)
- `innodb_defragment_stats_accuracy` (MariaDB version 10.1 and later)
- `innodb_empty_free_List_algorithm`
- `innodb_fake_changes`
- `innodb_fatal_semaphore_wait_threshold` (MariaDB version 10.1 and later)
- `innodb_foreground_preflush`
- `innodb_idle_flush_pct` (MariaDB version 10.1 and later)
- `innodb_immediate_scrub_data_uncompressed` (MariaDB version 10.1 and later)
- `innodb_instrument_semaphores` (MariaDB version 10.1 and later)
- `innodb_locking_fake_changes`
- `innodb_log_arch_dir`
- `innodb_log_arch_expire_sec`
- `innodb_log_archive`
- `innodb_log_block_size`
- `innodb_log_checksum_algorithm`
- `innodb_max_bitmap_file_size`
- `innodb_max_changed_pages`
- `innodb_prefix_index_cluster_optimization` (MariaDB version 10.1 and later)
- `innodb_sched_priority_cleaner`
- `innodb_scrub_log` (MariaDB version 10.1 and later)

- [innodb_scrub_log_speed](#) (MariaDB version 10.1 and later)
- [innodb_show_locks_held](#)
- [innodb_show_verbose_locks](#)
- [innodb_simulate_comp_failures](#)
- [innodb_stats_modified_counter](#)
- [innodb_stats_traditional](#)
- [innodb_use_atomic_writes](#)
- [innodb_use_fallocate](#)
- [innodb_use_global_flush_log_at_trx_commit](#)
- [innodb_use_stacktrace](#)
- [innodb_use_trim](#) (MariaDB version 10.1 and later)
- [join_buffer_space_limit](#)
- [join_cache_level](#)
- [key_cache_file_hash_size](#)
- [key_cache_segments](#)
- [max_digest_length](#) (MariaDB version 10.1 and later)
- [max_statement_time](#) (MariaDB version 10.1 and later)
- [mysql56_temporal_format](#) (MariaDB version 10.1 and later)
- [progress_report_time](#)
- [query_cache_strip_comments](#)
- [replicate_annotation_row_events](#)
- [replicate_do_db](#)
- [replicate_do_table](#)
- [replicate_events_marked_for_skip](#)
- [replicate_ignore_db](#)
- [replicate_ignore_table](#)
- [replicate_wild_ignore_table](#)
- [slave_domain_parallel_threads](#)
- [slave_parallel_max_queued](#)
- [slave_parallel_mode](#) (MariaDB version 10.1 and later)
- [slave_parallel_threads](#)
- [slave_run_triggers_for_rbr](#) (MariaDB version 10.1 and later)
- [sql_error_log_filename](#)
- [sql_error_log_rate](#)
- [sql_error_log_rotate](#)
- [sql_error_log_rotations](#)
- [sql_error_log_size_limit](#)
- [thread_handling](#)
- [thread_pool_idle_timeout](#)
- [thread_pool_max_threads](#)
- [thread_pool_min_threads](#)
- [thread_pool_oversubscribe](#)
- [thread_pool_size](#)
- [thread_pool_stall_limit](#)
- [transaction_write_set_extraction](#)
- [use_stat_tables](#)

- **userstat**

For more information on MariaDB parameters, go to the [MariaDB documentation](#).

MariaDB on Amazon RDS SQL reference

This appendix describes system stored procedures that are available for Amazon RDS instances running the MariaDB DB engine.

You can use the system stored procedures that are available for MySQL DB instances and MariaDB DB instances. These stored procedures are documented at [MySQL on Amazon RDS SQL reference \(p. 952\)](#). MariaDB DB instances support all of the stored procedures, except for `mysql.rds_set_source_delay`.

Additionally, the following system stored procedures are supported only for Amazon RDS DB instances running MariaDB:

- [mysql.rds_replica_status \(p. 625\)](#)
- [mysql.rds_set_external_master_gtid \(p. 626\)](#)
- [mysql.rds_kill_query_id \(p. 628\)](#)

[mysql.rds_replica_status](#)

Shows the replication status of a MariaDB read replica.

Call this procedure on the read replica to show status information on essential parameters of the replica threads.

Syntax

```
CALL mysql.rds_replica_status;
```

Usage notes

This procedure is only supported for MariaDB DB instances running MariaDB version 10.5 and higher.

This procedure is the equivalent of the `SHOW SLAVE STATUS` command. This command isn't supported for MariaDB version 10.5 and higher DB instances.

In prior versions of MariaDB, this command required the `REPLICATION SLAVE` privilege. In MariaDB version 10.5, it requires the `REPLICATION SLAVE ADMIN` privilege. To protect the RDS management of MariaDB 10.5 DB instances, this new privilege isn't granted to the RDS master user.

Examples

The following example shows the status of a MariaDB read replica:

```
call mysql.rds_replica_status;
```

The response is similar to the following:

```
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: XX.XX.XX.XXX
Master_User: rdsrepladmin
Master_Port: 3306
```

```

        Connect_Retry: 60
        Master_Log_File: mysql-bin-changelog.003988
        Read_Master_Log_Pos: 405
            Relay_Log_File: relaylog.011024
            Relay_Log_Pos: 657
        Relay_Master_Log_File: mysql-bin-changelog.003988
            Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
                Replicate_Do_DB:
                Replicate_Ignore_DB:
                Replicate_Do_Table:
                Replicate_Ignore_Table:
        mysql.rds_sysinfo,mysql.rds_history,mysql.rds_replication_status
            Replicate_Wild_Do_Table:
            Replicate_Wild_Ignore_Table:
                Last_Error:
                Skip_Counter: 0
            Exec_Master_Log_Pos: 405
                Relay_Log_Space: 1016
                Until_Condition: None
                Until_Log_File:
                Until_Log_Pos: 0
            Master_SSL_Allowed: No
            Master_SSL_CA_File:
            Master_SSL_CA_Path:
                Master_SSL_Cert:
                Master_SSL_Cipher:
                Master_SSL_Key:
            Seconds_Behind_Master: 0
        Master_SSL_Verify_Server_Cert: No
            Last_IO_Error:
            Last_SQL_Error:
        Replicate_Ignore_Servers:
            Master_Server_Id: 807509301
            Master_SSL_Crl:
            Master_SSL_Crlpath:
                Using_Gtid: Slave_Pos
                Gtid_IO_Pos: 0-807509301-3980
        Replicate_Do_Domain_Ids:
        Replicate_Ignore_Domain_Ids:
            Parallel_Mode: optimistic
            SQL_Delay: 0
            SQL_Remaining_Delay: NULL
        Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
        Slave_DDL_Groups: 15
    Slave_Non_Transactional_Groups: 0
        Slave_Transactional_Groups: 3658
1 row in set (0.000 sec)

Query OK, 0 rows affected (0.000 sec)

```

[mysql.rds_set_external_master_gtid](#)

Configures GTID-based replication from a MariaDB instance running external to Amazon RDS to a MariaDB DB instance. This stored procedure is supported only where the external MariaDB instance is version 10.0.24 or greater. When setting up replication where one or both instances do not support MariaDB global transaction identifiers (GTIDs), use [mysql.rds_set_external_master \(p. 954\)](#).

Using GTIDs for replication provides crash-safety features not offered by binary log replication, so we recommend it in cases where the replicating instances support it.

Syntax

```
CALL mysql.rds_set_external_master_gtid(
    host_name
    , host_port
    , replication_user_name
    , replication_user_password
    , gtid
    , ssl_encryption
);
```

Parameters

host_name

String. The host name or IP address of the MariaDB instance running external to Amazon RDS that will become the source instance.

host_port

Integer. The port used by the MariaDB instance running external to Amazon RDS to be configured as the source instance. If your network configuration includes SSH port replication that converts the port number, specify the port number that is exposed by SSH.

replication_user_name

String. The ID of a user with REPLICATION SLAVE permissions in the MariaDB DB instance to be configured as the read replica.

replication_user_password

String. The password of the user ID specified in *replication_user_name*.

gtid

String. The global transaction ID on the source instance that replication should start from.

You can use @@gtid_current_pos to get the current GTID if the source instance has been locked while you are configuring replication, so the binary log doesn't change between the points when you get the GTID and when replication starts.

Otherwise, if you are using mysqldump version 10.0.13 or greater to populate the replica instance prior to starting replication, you can get the GTID position in the output by using the --master-data or --dump-slave options. If you are not using mysqldump version 10.0.13 or greater, you can run the SHOW MASTER STATUS or use those same mysqldump options to get the binary log file name and position, then convert them to a GTID by running BINLOG_GTID_POS on the external MariaDB instance:

```
SELECT BINLOG_GTID_POS('<binary log file name>', <binary log file position>);
```

For more information about the MariaDB implementation of GTIDs, go to [Global transaction ID](#) in the MariaDB documentation.

ssl_encryption

A value that specifies whether Secure Socket Layer (SSL) encryption is used on the replication connection. 1 specifies to use SSL encryption, 0 specifies to not use encryption. The default is 0.

Note

The MASTER_SSL_VERIFY_SERVER_CERT option isn't supported. This option is set to 0, which means that the connection is encrypted, but the certificates aren't verified.

Usage notes

The `mysql.rds_set_external_master_gtid` procedure must be run by the master user. It must be run on the MariaDB DB instance that you are configuring as the replica of a MariaDB instance running external to Amazon RDS. Before running `mysql.rds_set_external_master_gtid`, you must have configured the instance of MariaDB running external to Amazon RDS as a source instance. For more information, see [Importing data into a MariaDB DB instance \(p. 616\)](#).

Warning

Do not use `mysql.rds_set_external_master_gtid` to manage replication between two Amazon RDS DB instances. Use it only when replicating with a MariaDB instance running external to RDS. For information about managing replication between Amazon RDS DB instances, see [Working with read replicas \(p. 278\)](#).

After calling `mysql.rds_set_external_master_gtid` to configure an Amazon RDS DB instance as a read replica, you can call [mysql.rds_start_replication \(p. 964\)](#) on the replica to start the replication process. You can call [mysql.rds_reset_external_master \(p. 961\)](#) to remove the read replica configuration.

When `mysql.rds_set_external_master_gtid` is called, Amazon RDS records the time, user, and an action of "set master" in the `mysql.rds_history` and `mysql.rds_replication_status` tables.

Examples

When run on a MariaDB DB instance, the following example configures it as the replica of an instance of MariaDB running external to Amazon RDS.

```
call mysql.rds_set_external_master_gtid
('Sourcedb.some.com',3306,'ReplicationUser','SomePassWOrd','0-123-456',0);
```

mysql.rds_kill_query_id

Ends a query running against the MariaDB server.

Syntax

```
CALL mysql.rds_kill_query_id(queryID);
```

Parameters

queryID

Integer. The identity of the query to be ended.

Usage notes

To stop a query running against the MariaDB server, use the `mysql.rds_kill_query_id` procedure and pass in the ID of that query. To obtain the query ID, query the MariaDB [Information schema PROCESSLIST table](#), as shown following:

```
SELECT USER, HOST, COMMAND, TIME, STATE, INFO, QUERY_ID FROM
INFORMATION_SCHEMA.PROCESSLIST WHERE USER = '<user name>';
```

The connection to the MariaDB server is retained.

Examples

The following example ends a query with a query ID of 230040:

```
call mysql.rds_kill_query_id(230040);
```

Microsoft SQL Server on Amazon RDS

Amazon RDS supports DB instances running several versions and editions of Microsoft SQL Server. Following, you can find the most recent supported version of each major version. For the full list of supported versions, editions, and RDS engine versions, see [Microsoft SQL Server versions on Amazon RDS \(p. 637\)](#).

- SQL Server 2019 CU8 15.00.4073.23, released per [KB4577194](#) on October 1, 2020.
- SQL Server 2017 CU22 14.00.3356.20, released per [KB4577467](#) on September 10, 2020.
- SQL Server 2016 SP2 CU15 13.00.5850.14, released per [KB4577775](#) on September 28, 2020.
- SQL Server 2014 SP3 CU4 12.00.6329.1, released per [KB4500181](#) on July 29, 2019.
- SQL Server 2012 SP4 GDR 11.0.7493.4, released per [KB4532098](#) on February 11, 2020.
- SQL Server 2008: It's no longer possible to provision new instances in any Region. Amazon RDS is actively migrating existing instances off this version.

For information about licensing for SQL Server, see [Licensing Microsoft SQL Server on Amazon RDS \(p. 655\)](#). For information about SQL Server builds, see this Microsoft support article about [the latest SQL Server builds](#).

With Amazon RDS, you can create DB instances and DB snapshots, point-in-time restores, and automated or manual backups. DB instances running SQL Server can be used inside a VPC. You can also use Secure Sockets Layer (SSL) to connect to a DB instance running SQL Server, and you can use transparent data encryption (TDE) to encrypt data at rest. Amazon RDS currently supports Multi-AZ deployments for SQL Server using SQL Server Database Mirroring (DBM) or Always On Availability Groups (AGs) as a high-availability, failover solution.

To deliver a managed service experience, Amazon RDS does not provide shell access to DB instances, and it restricts access to certain system procedures and tables that require advanced privileges. Amazon RDS supports access to databases on a DB instance using any standard SQL client application such as Microsoft SQL Server Management Studio. Amazon RDS does not allow direct host access to a DB instance via Telnet, Secure Shell (SSH), or Windows Remote Desktop Connection. When you create a DB instance, the master user is assigned to the `db_owner` role for all user databases on that instance, and has all database-level permissions except for those that are used for backups. Amazon RDS manages backups for you.

Before creating your first DB instance, you should complete the steps in the setting up section of this guide. For more information, see [Setting up for Amazon RDS \(p. 67\)](#).

Common management tasks for Microsoft SQL Server on Amazon RDS

The following are the common management tasks you perform with an Amazon RDS for SQL Server DB instance, with links to relevant documentation for each task.

Task area	Relevant documentation
Instance classes, storage, and PIOPS If you are creating a DB instance for production purposes, you should understand how instance classes, storage types, and Provisioned IOPS work in Amazon RDS.	DB instance class support for Microsoft SQL Server (p. 634) Amazon RDS storage types (p. 40)
Multi-AZ deployments A production DB instance should use Multi-AZ deployments. Multi-AZ deployments provide increased availability, data durability, and fault tolerance for DB instances. Multi-AZ deployments for SQL Server are implemented using SQL Server's native DBM or AGs technology.	High availability (Multi-AZ) for Amazon RDS (p. 53) Multi-AZ deployments using Microsoft SQL Server Database Mirroring or Always On availability groups (p. 643)
Amazon Virtual Private Cloud (VPC) If your AWS account has a default VPC, then your DB instance is automatically created inside the default VPC. If your account does not have a default VPC, and you want the DB instance in a VPC, you must create the VPC and subnet groups before you create the DB instance.	Determining whether you are using the EC2-VPC or EC2-Classic platform (p. 1718) Working with a DB instance in a VPC (p. 1727)
Security groups By default, DB instances are created with a firewall that prevents access to them. You therefore must create a security group with the correct IP addresses and network configuration to access the DB instance. The security group you create depends on what Amazon EC2 platform your DB instance is on, and whether you will access your DB instance from an Amazon EC2 instance. In general, if your DB instance is on the <i>EC2-Classic</i> platform, you will need to create a DB security group; if your DB instance is on the <i>EC2-VPC</i> platform, you will need to create a VPC security group.	Determining whether you are using the EC2-VPC or EC2-Classic platform (p. 1718) Controlling access with security groups (p. 1699)
Parameter groups If your DB instance is going to require specific database parameters, you should create a parameter group before you create the DB instance.	Working with DB parameter groups (p. 228)
Option groups If your DB instance is going to require specific database options, you should create an option group before you create the DB instance.	Options for the Microsoft SQL Server database engine (p. 749)
Connecting to your DB instance After creating a security group and associating it to a DB instance, you can connect to the DB instance using any standard SQL client application such as Microsoft SQL Server Management Studio.	Connecting to a DB instance running the Microsoft SQL Server database engine (p. 656)
Backup and restore	Working with backups (p. 328) Importing and exporting SQL Server databases (p. 671)

Task area	Relevant documentation
When you create your DB instance, you can configure it to take automated backups. You can also back up and restore your databases manually by using full backup files (.bak files).	
Monitoring You can monitor your SQL Server DB instance by using CloudWatch Amazon RDS metrics, events, and enhanced monitoring.	Viewing DB instance metrics (p. 548) Viewing Amazon RDS events (p. 503)
Log files You can access the log files for your SQL Server DB instance.	Accessing Amazon RDS database log files (p. 504) Microsoft SQL Server database log files (p. 516)

There are also advanced administrative tasks for working with SQL Server DB instances. For more information, see the following documentation:

- [Common DBA tasks for Microsoft SQL Server \(p. 809\)](#).
- [Using Windows Authentication with a SQL Server DB instance \(p. 711\)](#)
- [Accessing the tempdb database \(p. 810\)](#)

Limits for Microsoft SQL Server DB instances

The Amazon RDS implementation of Microsoft SQL Server on a DB instance has some limitations that you should be aware of:

- The maximum number of databases supported on a DB instance depends on the instance class type and the availability mode—Single-AZ, Multi-AZ Database Mirroring (DBM), or Multi-AZ Availability Groups (AGs). The Microsoft SQL Server system databases don't count toward this limit.

The following table shows the maximum number of supported databases for each instance class type and availability mode. Use this table to help you decide if you can move from one instance class type to another, or from one availability mode to another. If your source DB instance has more databases than the target instance class type or availability mode can support, modifying the DB instance fails. You can see the status of your request in the **Events** pane.

Instance class type	Single-AZ	Multi-AZ with DBM	Multi-AZ with Always On AGs
db.*.micro to db.*.medium	30	N/A	N/A
db.*.large	30	30	30
db.*.xlarge to db.*.16xlarge	100	50	75
db.*.24xlarge	100	50	100

* Represents the different instance class types.

For example, let's say that your DB instance runs on a db.*.16xlarge with Single-AZ and that it has 76 databases. You modify the DB instance to upgrade to using Multi-AZ Always On AGs. This upgrade fails, because your DB instance contains more databases than your target configuration can support. If you upgrade your instance class type to db.*.24xlarge instead, the modification succeeds.

If the upgrade fails, you see events and messages similar to the following:

- Unable to modify database instance class. The instance has 76 databases, but after conversion it would only support 75.
- Unable to convert the DB instance to Multi-AZ: The instance has 76 databases, but after conversion it would only support 75.

If the point-in-time restore or snapshot restore fails, you see events and messages similar to the following:

- Database instance put into incompatible-restore. The instance has 76 databases, but after conversion it would only support 75.
- Some ports are reserved for Amazon RDS, and you can't use them when you create a DB instance.
- Client connections from IP addresses within the range 169.254.0.0/16 are not permitted. This is the Automatic Private IP Addressing Range (APIPA), which is used for local-link addressing.
- SQL Server Standard Edition uses only a subset of the available processors if the DB instance has more processors than the software limits (24 cores, 4 sockets, and 128GB RAM). Examples of this are the db.m5.24xlarge and db.r5.24xlarge instance classes.

For more information, see the table of scale limits under [Editions and supported features of SQL Server 2019 \(15.x\)](#) in the Microsoft documentation.

- Amazon RDS for SQL Server doesn't support importing data into the msdb database.
- You can't rename databases on a DB instance in a SQL Server Multi-AZ deployment.
- Make sure that you use these guidelines when setting the following DB parameters on RDS for SQL Server:
 - `max server memory (mb) >= 256 MB`
 - `max worker threads >= (number of logical CPUs * 7)`
 - For the upper limit on `max worker threads`, see [Configure the max worker threads server configuration option](#) in the Microsoft documentation.

For more information on setting DB parameters, see [Working with DB parameter groups \(p. 228\)](#).

- The maximum storage size for SQL Server DB instances is the following:
 - General Purpose (SSD) storage – 16 TiB for all editions
 - Provisioned IOPS storage – 16 TiB for all editions
 - Magnetic storage – 1 TiB for all editions

If you have a scenario that requires a larger amount of storage, you can use sharding across multiple DB instances to get around the limit. This approach requires data-dependent routing logic in applications that connect to the sharded system. You can use an existing sharding framework, or you can write custom code to enable sharding. If you use an existing framework, the framework can't install any components on the same server as the DB instance.

- The minimum storage size for SQL Server DB instances is the following:
 - General Purpose (SSD) storage – 20 GiB for Enterprise, Standard, Web, and Express Editions
 - Provisioned IOPS storage – 20 GiB for Enterprise, Standard, Web, and Express Editions
 - Magnetic storage – 20 GiB for Enterprise, Standard, Web, and Express Editions
- Amazon RDS doesn't support running these services on the same server as your RDS DB instance:
 - Data Quality Services
 - Master Data Services

To use these features, we recommend that you install SQL Server on an Amazon EC2 instance, or use an on-premises SQL Server instance. In these cases, the EC2 or SQL Server instance acts as the Master Data Services server for your SQL Server DB instance on Amazon RDS. You can install SQL Server on an Amazon EC2 instance with Amazon EBS storage, pursuant to Microsoft licensing policies.

- Because of limitations in Microsoft SQL Server, restoring to a point in time before successfully running `DROP DATABASE` might not reflect the state of that database at that point in time. For example, the dropped database is typically restored to its state up to 5 minutes before the `DROP DATABASE` command was issued. This type of restore means that you can't restore the transactions made during those few minutes on your dropped database. To work around this, you can reissue the `DROP DATABASE` command after the restore operation is completed. Dropping a database removes the transaction logs for that database.
 - For SQL Server, you create your databases after you create your DB instance. Database names follow the usual SQL Server naming rules with the following differences:
 - Database names can't start with `rdsadmin`.
 - They can't start or end with a space or a tab.
 - They can't contain any of the characters that create a new line.
 - They can't contain a single quote (`'`).

DB instance class support for Microsoft SQL Server

The computation and memory capacity of a DB instance is determined by its DB instance class. The DB instance class you need depends on your processing power and memory requirements. For more information, see [DB instance classes \(p. 7\)](#).

The following list of DB instance classes supported for Microsoft SQL Server is provided here for your convenience. For the most current list, see the RDS console: <https://console.aws.amazon.com/rds/>.

SQL Server edition	2019 support range	2017 and 2016 support range	2014 and 2012 support range
Enterprise Edition	db.t3.xlarge-db.t3.2xlarge db.r5.xlarge-db.r5.24xlarge db.r5b.xlarge-db.r5b.24xlarge db.r5d.xlarge-db.r5d.24xlarge db.m5.xlarge-db.m5.24xlarge db.m5d.xlarge-db.m5d.24xlarge db.x1.16xlarge-db.x1.32xlarge db.xle.xlarge-db.xle.32xlarge db.z1d.xlarge-db.z1d.12xlarge	db.t3.xlarge-db.t3.2xlarge db.r3.xlarge-db.r3.8xlarge db.r4.16xlarge-db.r4.8xlarge db.r5.24xlarge-db.r5.24xlarge db.r5b.24xlarge-db.r5b.24xlarge db.r5d.24xlarge-db.r5d.24xlarge db.m4.16xlarge-db.m4.10xlarge db.m5.24xlarge-db.m5.24xlarge db.x1.32xlarge-db.x1.16xlarge db.xle.32xlarge-db.xle.16xlarge db.z1d.12xlarge-db.z1d.xlarge	db.t3.xlarge-db.t3.2xlarge db.r3.xlarge-db.r3.8xlarge db.r4.16xlarge-db.r4.8xlarge db.r5.24xlarge-db.r5.24xlarge db.r5b.24xlarge-db.r5b.24xlarge db.r5d.24xlarge-db.r5d.24xlarge db.m4.16xlarge-db.m4.10xlarge db.m5.24xlarge-db.m5.24xlarge db.x1.32xlarge-db.x1.16xlarge db.xle.32xlarge-db.xle.16xlarge db.z1d.12xlarge-db.z1d.xlarge

Microsoft SQL Server security

The Microsoft SQL Server database engine uses role-based security. The master user name you use when you create a DB instance is a SQL Server Authentication login that is a member of the `processadmin`, `public`, and `setupadmin` fixed server roles.

Any user who creates a database is assigned to the db_owner role for that database and has all database-level permissions except for those that are used for backups. Amazon RDS manages backups for you.

The following server-level roles are not currently available in Amazon RDS:

- bulkadmin
- dbcreator
- diskadmin
- securityadmin
- serveradmin
- sysadmin

The following server-level permissions are not available on SQL Server DB instances:

- ALTER ANY CREDENTIAL
- ALTER ANY EVENT NOTIFICATION
- ALTER ANY EVENT SESSION
- ALTER RESOURCES
- ALTER SETTINGS (you can use the DB parameter group API operations to modify parameters; for more information, see [Working with DB parameter groups \(p. 228\)](#))
- AUTHENTICATE SERVER
- CONTROL_SERVER
- CREATE DDL EVENT NOTIFICATION
- CREATE ENDPOINT
- CREATE TRACE EVENT NOTIFICATION
- EXTERNAL ACCESS ASSEMBLY
- SHUTDOWN (You can use the RDS reboot option instead)
- UNSAFE ASSEMBLY
- ALTER ANY AVAILABILITY GROUP (SQL Server 2012 only)
- CREATE ANY AVAILABILITY GROUP (SQL Server 2012 only)

Compliance program support for Microsoft SQL Server DB instances

AWS Services in scope have been fully assessed by a third-party auditor and result in a certification, attestation of compliance, or Authority to Operate (ATO). For more information, see [AWS services in scope by compliance program](#).

HIPAA support for Microsoft SQL Server DB instances

You can use Amazon RDS for Microsoft SQL Server databases to build HIPAA-compliant applications. You can store healthcare-related information, including protected health information (PHI), under a Business Associate Agreement (BAA) with AWS. For more information, see [HIPAA compliance](#).

Amazon RDS for SQL Server supports HIPAA for the following versions and editions:

- SQL Server 2019 Enterprise, Standard, and Web Editions

- SQL Server 2017 Enterprise, Standard, and Web Editions
- SQL Server 2016 Enterprise, Standard, and Web Editions
- SQL Server 2014 Enterprise, Standard, and Web Editions
- SQL Server 2012 Enterprise, Standard, and Web Editions

To enable HIPAA support on your DB instance, set up the following three components.

Component	Details
Auditing	To set up auditing, set the parameter <code>rds.sqlserver_audit</code> to the value <code>fedramp_hipaa</code> . If your DB instance is not already using a custom DB parameter group, you must create a custom parameter group and attach it to your DB instance before you can modify the <code>rds.sqlserver_audit</code> parameter. For more information, see Working with DB parameter groups (p. 228) .
Transport encryption	To set up transport encryption, force all connections to your DB instance to use Secure Sockets Layer (SSL). For more information, see Forcing connections to your DB instance to use SSL (p. 704) .
Encryption at rest	To set up encryption at rest, you have two options: <ol style="list-style-type: none"> 1. If you're running SQL Server 2012–2019 Enterprise Edition or 2019 Standard Edition, you can use Transparent Data Encryption (TDE) to achieve encryption at rest. For more information, see Support for Transparent Data Encryption in SQL Server (p. 754). 2. You can set up encryption at rest by using AWS Key Management Service (AWS KMS) encryption keys. For more information, see Encrypting Amazon RDS resources (p. 1630).

SSL support for Microsoft SQL Server DB instances

You can use SSL to encrypt connections between your applications and your Amazon RDS DB instances running Microsoft SQL Server. You can also force all connections to your DB instance to use SSL. If you force connections to use SSL, it happens transparently to the client, and the client doesn't have to do any work to use SSL.

SSL is supported in all AWS Regions and for all supported SQL Server editions. For more information, see [Using SSL with a Microsoft SQL Server DB instance \(p. 704\)](#).

Microsoft SQL Server versions on Amazon RDS

You can specify any currently supported Microsoft SQL Server version when creating a new DB instance. You can specify the Microsoft SQL Server major version (such as Microsoft SQL Server 14.00), and any supported minor version for the specified major version. If no version is specified, Amazon RDS defaults to a supported version, typically the most recent version. If a major version is specified but a minor version is not, Amazon RDS defaults to a recent release of the major version you have specified.

The following table shows the supported versions for all editions and all AWS Regions, except where noted. You can also use the `describe-db-engine-versions` AWS CLI command to see a list of supported versions, as well as defaults for newly created DB instances.

SQL Server versions supported in RDS

Major version	Minor version	RDS API EngineVersion and CLI engine-version
SQL Server 2019	15.00.4073.23 (CU8)	15.00.4073.23.v1
	15.00.4043.16 (CU5)	15.00.4043.16.v1
SQL Server 2017	14.00.3356.20 (CU22)	14.00.3356.20.v1
	14.00.3294.2 (CU20)	14.00.3294.2.v1
	14.00.3281.6 (CU19)	14.00.3281.6.v1
	14.00.3223.3 (CU16)	14.00.3223.3.v1
	14.00.3192.2 (CU15 GDR)	14.00.3192.2.v1
	14.00.3049.1 (CU13 hotfix update)	14.00.3049.1.v1
	14.00.3035.2 (CU9 GDR)	14.00.3035.2.v1
	14.00.3015.40 (CU3)	14.00.3015.40.v1
	14.00.1000.169 (RTM)	14.00.1000.169.v1
SQL Server 2016	13.00.5850.14 (SP2 CU15)	13.00.5850.14.v1
	13.00.5820.21 (SP2 CU13)	13.00.5820.21.v1
	13.00.5598.27 (SP2 CU11)	13.00.5598.27.v1
	13.00.5426.0 (SP2 CU8)	13.00.5426.0.v1
	13.00.5366.0 (SP2)	13.00.5366.0.v1
	13.00.5292.0 (CU6)	13.00.5292.0.v1
	13.00.5216.0 (CU3)	13.00.5216.0.v1
	13.00.4522.0 (SP1 CU10 security update)	13.00.4522.0.v1
	13.00.4466.4 (SP1 CU7)	13.00.4466.4.v1
	13.00.4451.0 (SP1 CU5)	13.00.4451.0.v1
	13.00.4422.0 (SP1 CU2)	13.00.4422.0.v1
	13.00.2164.0 (RTM CU2)	13.00.2164.0.v1
SQL Server 2014	12.00.6329.1 (SP3 CU4)	12.00.6329.1.v1
	12.00.6293.0 (SP3 CU3)	12.00.6293.0.v1
	12.00.5571.0 (SP2 CU10)	12.00.5571.0.v1
	12.00.5546.0 (SP2 CU5)	12.00.5546.0.v1
	12.00.5000.0 (SP2)	12.00.5000.0.v1

Major version	Minor version	RDS API EngineVersion and CLI engine-version
SQL Server 2012	11.00.7493.4 (SP4 GDR)	11.00.7493.4.v1
	11.00.7462.6 (SP4 GDR)	11.00.7462.6.v1
	11.00.6594.0 (SP3 CU8)	11.00.6594.0.v1
	11.00.6020.0 (SP3)	11.00.6020.0.v1
	11.00.5058.0 (SP2), except US East (Ohio), Canada (Central), and Europe (London)	11.00.5058.0.v1

Version management in Amazon RDS

Amazon RDS includes flexible version management that enables you to control when and how your DB instance is patched or upgraded. This enables you to do the following for your DB engine:

- Maintain compatibility with database engine patch versions.
- Test new patch versions to verify that they work with your application before you deploy them in production.
- Plan and perform version upgrades to meet your service level agreements and timing requirements.

Microsoft SQL Server engine patching in Amazon RDS

Amazon RDS periodically aggregates official Microsoft SQL Server database patches into a DB instance engine version that's specific to Amazon RDS. For more information about the Microsoft SQL Server patches in each engine version, see [Version and feature support on Amazon RDS](#).

Currently, you manually perform all engine upgrades on your DB instance. For more information, see [Upgrading the Microsoft SQL Server DB engine \(p. 666\)](#).

Deprecation schedule for major engine versions of Microsoft SQL Server on Amazon RDS

The table following displays the planned schedule of deprecations for major engine versions of Microsoft SQL Server.

Date	Information
July 12, 2019	<p>The Amazon RDS team deprecated support for Microsoft SQL Server 2008 R2 in June 2019. Customers who are migrating to SQL Server 2012 (latest minor version available).</p> <p>To avoid an automatic upgrade from Microsoft SQL Server 2008 R2, you can upgrade at any time. For more information, see Upgrading a DB instance engine version (p. 271).</p>
April 25, 2019	Before the end of April 2019, you will no longer be able to create new Amazon RDS for Microsoft SQL Server 2008R2.

Microsoft SQL Server features on Amazon RDS

The supported SQL Server versions on Amazon RDS include the following features.

Microsoft SQL Server 2019 features

SQL Server 2019 includes many new features, such as the following:

- Accelerated database recovery (ADR) – Reduces crash recovery time after a restart or a long-running transaction rollback.
- Intelligent Query Processing (IQP):
 - Row mode memory grant feedback – Corrects excessive grants automatically, that would otherwise result in wasted memory and reduced concurrency.
 - Batch mode on rowstore – Enables batch mode execution for analytic workloads without requiring columnstore indexes.
 - Table variable deferred compilation – Improves plan quality and overall performance for queries that reference table variables.
- Intelligent performance:
 - `OPTIMIZE_FOR_SEQUENTIAL_KEY` index option – Improves throughput for high-concurrency inserts into indexes.
 - Improved indirect checkpoint scalability – Helps databases with heavy DML workloads.
 - Concurrent Page Free Space (PFS) updates – Enables handling as a shared latch rather than an exclusive latch.
- Monitoring improvements:
 - `WAIT_ON_SYNC_STATISTICS_REFRESH` wait type – Shows accumulated instance-level time spent on synchronous statistics refresh operations.
 - Database-scoped configurations – Include `LIGHTWEIGHT_QUERY_PROFILING` and `LAST_QUERY_PLAN_STATS`.
 - Dynamic management functions (DMFs) – Include `sys.dm_exec_query_plan_stats` and `sys.dm_db_page_info`.
- Verbose truncation warnings – The data truncation error message defaults to include table and column names and the truncated value.
- Resumable online index creation – In SQL Server 2017, only resumable online index rebuild is supported.

For the full list of SQL Server 2019 features, see [What's new in SQL Server 2019 \(15.x\)](#) in the Microsoft documentation.

For a list of unsupported features, see [Features not supported and features with limited support \(p. 643\)](#).

Microsoft SQL Server 2017 features

SQL Server 2017 includes many new features, such as the following:

- Adaptive query processing
- Automatic plan correction
- GraphDB
- Resumable index rebuilds

For the full list of SQL Server 2017 features, see [What's new in SQL Server 2017](#) in the Microsoft documentation.

For a list of unsupported features, see [Features not supported and features with limited support \(p. 643\)](#).

Microsoft SQL Server 2016 features

Amazon RDS supports the following features of SQL Server 2016:

- Always Encrypted
- JSON Support
- Operational Analytics
- Query Store
- Temporal Tables

For the full list of SQL Server 2016 features, see [What's new in SQL Server 2016](#) in the Microsoft documentation.

Microsoft SQL Server 2014 features

In addition to supported features of SQL Server 2012, Amazon RDS supports the new query optimizer available in SQL Server 2014, and also the delayed durability feature.

For a list of unsupported features, see [Features not supported and features with limited support \(p. 643\)](#).

SQL Server 2014 supports all the parameters from SQL Server 2012 and uses the same default values. SQL Server 2014 includes one new parameter, backup checksum default. For more information, see [How to enable the CHECKSUM option if backup utilities do not expose the option](#) in the Microsoft documentation.

Microsoft SQL Server 2012 features

In addition to supported features of SQL Server 2008 R2, Amazon RDS supports the following SQL Server 2012 features:

- Columnstore indexes (Enterprise Edition)
- Online Index Create, Rebuild and Drop for XML, varchar(max), nvarchar(max), and varbinary(max) data types (Enterprise Edition)
- Flexible Server Roles
- Service Broker is supported, Service Broker endpoints are not supported
- Partially Contained Databases
- Sequences
- Transparent Data Encryption (Enterprise Edition only)
- THROW statement
- New and enhanced spatial types
- UTF-16 Support
- ALTER ANY SERVER ROLE server-level permission

For more information about SQL Server 2012, see [Features supported by the editions of SQL Server 2012](#) in the Microsoft documentation.

For a list of unsupported features, see [Features not supported and features with limited support \(p. 643\)](#).

Some SQL Server parameters have changed in SQL Server 2012.

- The following parameters have been removed from SQL Server 2012: `awe_enabled`, `precompute_rank`, and `sql_mail_xps`. These parameters were not modifiable in SQL Server DB Instances and their removal should have no impact on your SQL Server use.
- A new `contained_database_authentication` parameter in SQL Server 2012 supports partially contained databases. When you enable this parameter and then create a partially contained database, an authorized user's user name and password is stored within the partially contained database instead of in the primary database. For more information about partially contained databases, see [Contained databases](#) in the Microsoft documentation.

Microsoft SQL Server 2008 R2 deprecated on Amazon RDS

We are upgrading all existing instances that are still using SQL Server 2008 R2 to the latest minor version of SQL Server 2012. For more information, see [Version management in Amazon RDS \(p. 639\)](#).

For more information about SQL Server 2008 R2, see [Features supported by the editions of SQL Server 2008 R2](#) in the Microsoft documentation.

Change data capture support for Microsoft SQL Server DB instances

Amazon RDS supports change data capture (CDC) for your DB instances running Microsoft SQL Server. CDC captures changes that are made to the data in your tables, and stores metadata about each change that you can access later. For more information, see [Change data capture](#) in the Microsoft documentation.

Amazon RDS supports CDC for the following SQL Server editions and versions:

- Microsoft SQL Server Enterprise Edition (All versions)
- Microsoft SQL Server Standard Edition:
 - 2019
 - 2017
 - 2016 version 13.00.4422.0 SP1 CU2 and later

To use CDC with your Amazon RDS DB instances, first enable or disable CDC at the database level by using RDS-provided stored procedures. After that, any user that has the `db_owner` role for that database can use the native Microsoft stored procedures to control CDC on that database. For more information, see [Using change data capture \(p. 820\)](#).

You can use CDC and AWS Database Migration Service to enable ongoing replication from SQL Server DB instances.

Features not supported and features with limited support

The following Microsoft SQL Server features are not supported on Amazon RDS:

- Backing up to Microsoft Azure Blob Storage
- Buffer pool extension
- Custom password policies
- Data Quality Services
- Database Log Shipping
- Extended stored procedures, including xp_cmdshell
- FILESTREAM support
- File tables
- Machine Learning and R Services (requires OS access to install it)
- Maintenance Plans
- Performance Data Collector
- Policy-Based Management
- PolyBase
- Replication
- Resource Governor
- Server-level triggers
- Service Broker endpoints
- Stretch database
- T-SQL endpoints (all operations using CREATE ENDPOINT are unavailable)
- WCF Data Services

The following Microsoft SQL Server features have limited support on Amazon RDS:

- Distributed Queries / Linked Servers. For more information, see: [Implementing linked servers with Amazon RDS for Microsoft SQL Server](#).

Multi-AZ deployments using Microsoft SQL Server Database Mirroring or Always On availability groups

Amazon RDS supports Multi-AZ deployments for DB instances running Microsoft SQL Server by using SQL Server Database Mirroring (DBM) or Always On Availability Groups (AGs). Multi-AZ deployments provide increased availability, data durability, and fault tolerance for DB instances. In the event of planned database maintenance or unplanned service disruption, Amazon RDS automatically fails over to the up-to-date secondary replica so database operations can resume quickly without manual intervention. The primary and secondary instances use the same endpoint, whose physical network address transitions to the passive secondary replica as part of the failover process. You don't have to reconfigure your application when a failover occurs.

Amazon RDS manages failover by actively monitoring your Multi-AZ deployment and initiating a failover when a problem with your primary occurs. Failover doesn't occur unless the standby and primary are

fully in sync. Amazon RDS actively maintains your Multi-AZ deployment by automatically repairing unhealthy DB instances and re-establishing synchronous replication. You don't have to manage anything. Amazon RDS handles the primary, the witness, and the standby instance for you. When you set up SQL Server Multi-AZ, RDS configures passive secondary instances for all of the databases on the instance.

For more information, see [Multi-AZ deployments for Microsoft SQL Server \(p. 698\)](#).

Using Transparent Data Encryption to encrypt data at rest

Amazon RDS supports Microsoft SQL Server Transparent Data Encryption (TDE), which transparently encrypts stored data. Amazon RDS uses option groups to enable and configure these features. For more information about the TDE option, see [Support for Transparent Data Encryption in SQL Server \(p. 754\)](#).

Functions and stored procedures for Amazon RDS for Microsoft SQL Server

The following table lists Amazon RDS functions and stored procedures that help automate SQL Server tasks.

SQL Server functions and stored procedures

Task type	Procedure or function	Where it's used
Administrative tasks	<code>rds_drop_database</code>	Dropping a Microsoft SQL Server database (p. 818)
	<code>rds_failover</code>	Determining the last failover time (p. 817)
	<code>rds_modify_db</code>	Renaming a Microsoft SQL Server database in a Multi-AZ deployment (p. 818)
	<code>rds_read_error_log</code>	Viewing error and agent logs (p. 824)
	<code>rds_set_configuration</code>	Setting DB instance configurations: <ul style="list-style-type: none">• Change data capture for Multi-AZ instances (p. 821)• Setting the retention period for trace and dump files (p. 824)• Compressing backup files (p. 685)
	<code>rds_set_database</code>	Transitioning a Microsoft SQL Server database from OFFLINE to ONLINE (p. 820)
	<code>rds_show_configuration</code>	Show values set using <code>rds_set_configuration</code> : <ul style="list-style-type: none">• Change data capture for Multi-AZ instances (p. 821)• Setting the retention period for trace and dump files (p. 824)
	<code>rds_shrink_tempdb</code>	Shrinking the tempdb database (p. 810)
Change data capture (CDC)	<code>rds_cdc_disable</code>	Disabling CDC (p. 820)

Task type	Procedure or function	Where it's used
	<code>rds_cdc_enable_db</code>	Enabling CDC (p. 820)
Database Mail	<code>rds_fn_sysmail_viewitem</code>	Viewing messages, logs, and attachments (p. 742)
	<code>rds_fn_sysmail_viewitemlog</code>	Viewing messages, logs, and attachments (p. 742)
	<code>rds_fn_sysmail_viewitemattachment</code>	Viewing item attachments, and attachments (p. 742)
	<code>rds_sysmail_start</code>	Starting and stopping the mail queue: <ul style="list-style-type: none"> Starting the mail queue (p. 743) Stopping the mail queue (p. 743)
	<code>rds_sysmail_delete</code>	Deleting messages (p. 742)
Native backup and restore	<code>rds_backup_database</code>	Backing up a database (p. 675)
	<code>rds_cancel_task</code>	Canceling a task (p. 682)
	<code>rds_finish_restore</code>	Finishing a database restore (p. 681)
	<code>rds_restore_database</code>	Restoring a database (p. 678)
	<code>rds_restore_log</code>	Restoring a log (p. 680)
Amazon S3 file transfer	<code>rds_delete_file</code>	Deleting files on the RDS DB instance (p. 729)
	<code>rds_download_file</code>	Downloading files from an Amazon S3 bucket to a SQL Server DB instance (p. 727)
	<code>rds_gather_file</code>	Gathering files on the RDS DB instance (p. 728)
	<code>rds_upload_to_s3</code>	Uploading files from a SQL Server DB instance to an Amazon S3 bucket (p. 728)
Microsoft Distributed Transaction Coordinator (MSDTC)	<code>rds_msdtc_transaction_tracing</code>	Using transaction tracing (p. 804)
SQL Server Audit	<code>rds_fn_get_audit_file</code>	Viewing audit logs (p. 760)

Task type	Procedure or function	Where it's used
Microsoft Business Intelligence (MSBI)	rds_msbi_task	SQL Server Analysis Services (SSAS): <ul style="list-style-type: none"> Deploying SSAS projects on Amazon RDS (p. 767) Adding a domain user as a database administrator (p. 770) Backing up an SSAS database (p. 770) Restoring an SSAS database (p. 771) SQL Server Integration Services (SSIS): <ul style="list-style-type: none"> Administrative permissions on SSISDB (p. 779) Deploying an SSIS project (p. 781) SQL Server Reporting Services (SSRS): <ul style="list-style-type: none"> Granting access to domain users (p. 792) Revoking system-level permissions (p. 793)
	rds_fn_task_Status	Status of MSBI tasks: <ul style="list-style-type: none"> SSAS: Monitoring the status of a deployment task (p. 768) SSIS: Monitoring the status of a deployment task (p. 781) SSRS: Monitoring the status of a task (p. 793)
SSIS	rds_drop_ssis	Dropping the SSISDB database (p. 786)
	rds_sqlagent	Creating an SSIS proxy (p. 783)
SSRS	rds_drop_ssrs	Deleting databases (p. 796)

Local time zone for Microsoft SQL Server DB instances

The time zone of an Amazon RDS DB instance running Microsoft SQL Server is set by default. The current default is Coordinated Universal Time (UTC). You can set the time zone of your DB instance to a local time zone instead, to match the time zone of your applications.

You set the time zone when you first create your DB instance. You can create your DB instance by using the [AWS Management Console](#), the Amazon RDS API [CreateDBInstance](#) action, or the AWS CLI [create-db-instance](#) command.

If your DB instance is part of a Multi-AZ deployment (using SQL Server DBM or AGs), then when you fail over, your time zone remains the local time zone that you set. For more information, see [Multi-AZ deployments using Microsoft SQL Server Database Mirroring or Always On availability groups \(p. 643\)](#).

When you request a point-in-time restore, you specify the time to restore to. The time is shown in your local time zone. For more information, see [Restoring a DB instance to a specified time \(p. 389\)](#).

The following are limitations to setting the local time zone on your DB instance:

- You can't modify the time zone of an existing SQL Server DB instance.
- You can't restore a snapshot from a DB instance in one time zone to a DB instance in a different time zone.
- We strongly recommend that you don't restore a backup file from one time zone to a different time zone. If you restore a backup file from one time zone to a different time zone, you must audit your queries and applications for the effects of the time zone change. For more information, see [Importing and exporting SQL Server databases \(p. 671\)](#).

Supported time zones

You can set your local time zone to one of the values listed in the following table.

Time zones supported for Amazon RDS on SQL Server

Time zone	Standard time offset	Description	Notes
Afghanistan Standard Time	(UTC+04:30)	Kabul	This time zone doesn't observe daylight saving time.
Alaskan Standard Time	(UTC–09:00)	Alaska	
Aleutian Standard Time	(UTC–10:00)	Aleutian Islands	
Altai Standard Time	(UTC+07:00)	Barnaul, Gorno-Altaysk	
Arab Standard Time	(UTC+03:00)	Kuwait, Riyadh	This time zone doesn't observe daylight saving time.
Arabian Standard Time	(UTC+04:00)	Abu Dhabi, Muscat	
Arabic Standard Time	(UTC+03:00)	Baghdad	This time zone doesn't observe daylight saving time.
Argentina Standard Time	(UTC–03:00)	City of Buenos Aires	This time zone doesn't observe daylight saving time.
Astrakhan Standard Time	(UTC+04:00)	Astrakhan, Ulyanovsk	
Atlantic Standard Time	(UTC–04:00)	Atlantic Time (Canada)	
AUS Central Standard Time	(UTC+09:30)	Darwin	This time zone doesn't observe daylight saving time.
Aus Central W. Standard Time	(UTC+08:45)	Eucla	
AUS Eastern Standard Time	(UTC+10:00)	Canberra, Melbourne, Sydney	
Azerbaijan Standard Time	(UTC+04:00)	Baku	

Time zone	Standard time offset	Description	Notes
Azores Standard Time	(UTC-01:00)	Azores	
Bahia Standard Time	(UTC-03:00)	Salvador	
Bangladesh Standard Time	(UTC+06:00)	Dhaka	This time zone doesn't observe daylight saving time.
Belarus Standard Time	(UTC+03:00)	Minsk	This time zone doesn't observe daylight saving time.
Bougainville Standard Time	(UTC+11:00)	Bougainville Island	
Canada Central Standard Time	(UTC-06:00)	Saskatchewan	This time zone doesn't observe daylight saving time.
Cape Verde Standard Time	(UTC-01:00)	Cabo Verde Is.	This time zone doesn't observe daylight saving time.
Caucasus Standard Time	(UTC+04:00)	Yerevan	
Cen. Australia Standard Time	(UTC+09:30)	Adelaide	
Central America Standard Time	(UTC-06:00)	Central America	This time zone doesn't observe daylight saving time.
Central Asia Standard Time	(UTC+06:00)	Astana	This time zone doesn't observe daylight saving time.
Central Brazilian Standard Time	(UTC-04:00)	Cuiaba	
Central Europe Standard Time	(UTC+01:00)	Belgrade, Bratislava, Budapest, Ljubljana, Prague	
Central European Standard Time	(UTC+01:00)	Sarajevo, Skopje, Warsaw, Zagreb	
Central Pacific Standard Time	(UTC+11:00)	Solomon Islands, New Caledonia	This time zone doesn't observe daylight saving time.
Central Standard Time	(UTC-06:00)	Central Time (US and Canada)	
Central Standard Time (Mexico)	(UTC-06:00)	Guadalajara, Mexico City, Monterrey	
Chatham Islands Standard Time	(UTC+12:45)	Chatham Islands	

Time zone	Standard time offset	Description	Notes
China Standard Time	(UTC+08:00)	Beijing, Chongqing, Hong Kong, Urumqi	This time zone doesn't observe daylight saving time.
Cuba Standard Time	(UTC-05:00)	Havana	
Dateline Standard Time	(UTC-12:00)	International Date Line West	This time zone doesn't observe daylight saving time.
E. Africa Standard Time	(UTC+03:00)	Nairobi	This time zone doesn't observe daylight saving time.
E. Australia Standard Time	(UTC+10:00)	Brisbane	This time zone doesn't observe daylight saving time.
E. Europe Standard Time	(UTC+02:00)	Chisinau	
E. South America Standard Time	(UTC-03:00)	Brasilia	
Easter Island Standard Time	(UTC-06:00)	Easter Island	
Eastern Standard Time	(UTC-05:00)	Eastern Time (US and Canada)	
Eastern Standard Time (Mexico)	(UTC-05:00)	Chetumal	
Egypt Standard Time	(UTC+02:00)	Cairo	
Ekaterinburg Standard Time	(UTC+05:00)	Ekaterinburg	
Fiji Standard Time	(UTC+12:00)	Fiji	
FLE Standard Time	(UTC+02:00)	Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius	
Georgian Standard Time	(UTC+04:00)	Tbilisi	This time zone doesn't observe daylight saving time.
GMT Standard Time	(UTC)	Dublin, Edinburgh, Lisbon, London	This time zone isn't the same as Greenwich Mean Time. This time zone does observe daylight saving time.
Greenland Standard Time	(UTC-03:00)	Greenland	
Greenwich Standard Time	(UTC)	Monrovia, Reykjavik	This time zone doesn't observe daylight saving time.
GTB Standard Time	(UTC+02:00)	Athens, Bucharest	
Haiti Standard Time	(UTC-05:00)	Haiti	

Time zone	Standard time offset	Description	Notes
Hawaiian Standard Time	(UTC-10:00)	Hawaii	
India Standard Time	(UTC+05:30)	Chennai, Kolkata, Mumbai, New Delhi	This time zone doesn't observe daylight saving time.
Iran Standard Time	(UTC+03:30)	Tehran	
Israel Standard Time	(UTC+02:00)	Jerusalem	
Jordan Standard Time	(UTC+02:00)	Amman	
Kaliningrad Standard Time	(UTC+02:00)	Kaliningrad	
Kamchatka Standard Time	(UTC+12:00)	Petropavlovsk-Kamchatsky – Old	
Korea Standard Time	(UTC+09:00)	Seoul	This time zone doesn't observe daylight saving time.
Libya Standard Time	(UTC+02:00)	Tripoli	
Line Islands Standard Time	(UTC+14:00)	Kiritimati Island	
Lord Howe Standard Time	(UTC+10:30)	Lord Howe Island	
Magadan Standard Time	(UTC+11:00)	Magadan	This time zone doesn't observe daylight saving time.
Magallanes Standard Time	(UTC-03:00)	Punta Arenas	
Marquesas Standard Time	(UTC-09:30)	Marquesas Islands	
Mauritius Standard Time	(UTC+04:00)	Port Louis	This time zone doesn't observe daylight saving time.
Middle East Standard Time	(UTC+02:00)	Beirut	
Montevideo Standard Time	(UTC-03:00)	Montevideo	
Morocco Standard Time	(UTC+01:00)	Casablanca	
Mountain Standard Time	(UTC-07:00)	Mountain Time (US and Canada)	
Mountain Standard Time (Mexico)	(UTC-07:00)	Chihuahua, La Paz, Mazatlan	
Myanmar Standard Time	(UTC+06:30)	Yangon (Rangoon)	This time zone doesn't observe daylight saving time.
N. Central Asia Standard Time	(UTC+07:00)	Novosibirsk	
Namibia Standard Time	(UTC+02:00)	Windhoek	

Time zone	Standard time offset	Description	Notes
Nepal Standard Time	(UTC+05:45)	Kathmandu	This time zone doesn't observe daylight saving time.
New Zealand Standard Time	(UTC+12:00)	Auckland, Wellington	
Newfoundland Standard Time	(UTC−03:30)	Newfoundland	
Norfolk Standard Time	(UTC+11:00)	Norfolk Island	
North Asia East Standard Time	(UTC+08:00)	Irkutsk	
North Asia Standard Time	(UTC+07:00)	Krasnoyarsk	
North Korea Standard Time	(UTC+09:00)	Pyongyang	
Omsk Standard Time	(UTC+06:00)	Omsk	
Pacific SA Standard Time	(UTC−03:00)	Santiago	
Pacific Standard Time	(UTC−08:00)	Pacific Time (US and Canada)	
Pacific Standard Time (Mexico)	(UTC−08:00)	Baja California	
Pakistan Standard Time	(UTC+05:00)	Islamabad, Karachi	This time zone doesn't observe daylight saving time.
Paraguay Standard Time	(UTC−04:00)	Asuncion	
Romance Standard Time	(UTC+01:00)	Brussels, Copenhagen, Madrid, Paris	
Russia Time Zone 10	(UTC+11:00)	Chokurdakh	
Russia Time Zone 11	(UTC+12:00)	Anadyr, Petropavlovsk-Kamchatsky	
Russia Time Zone 3	(UTC+04:00)	Izhevsk, Samara	
Russian Standard Time	(UTC+03:00)	Moscow, St. Petersburg, Volgograd	This time zone doesn't observe daylight saving time.
SA Eastern Standard Time	(UTC−03:00)	Cayenne, Fortaleza	This time zone doesn't observe daylight saving time.
SA Pacific Standard Time	(UTC−05:00)	Bogota, Lima, Quito, Rio Branco	This time zone doesn't observe daylight saving time.
SA Western Standard Time	(UTC−04:00)	Georgetown, La Paz, Manaus, San Juan	This time zone doesn't observe daylight saving time.

Time zone	Standard time offset	Description	Notes
Saint Pierre Standard Time	(UTC-03:00)	Saint Pierre and Miquelon	
Sakhalin Standard Time	(UTC+11:00)	Sakhalin	
Samoa Standard Time	(UTC+13:00)	Samoa	
Sao Tome Standard Time	(UTC+01:00)	Sao Tome	
Saratov Standard Time	(UTC+04:00)	Saratov	
SE Asia Standard Time	(UTC+07:00)	Bangkok, Hanoi, Jakarta	This time zone doesn't observe daylight saving time.
Singapore Standard Time	(UTC+08:00)	Kuala Lumpur, Singapore	This time zone doesn't observe daylight saving time.
South Africa Standard Time	(UTC+02:00)	Harare, Pretoria	This time zone doesn't observe daylight saving time.
Sri Lanka Standard Time	(UTC+05:30)	Sri Jayawardenepura	This time zone doesn't observe daylight saving time.
Sudan Standard Time	(UTC+02:00)	Khartoum	
Syria Standard Time	(UTC+02:00)	Damascus	
Taipei Standard Time	(UTC+08:00)	Taipei	This time zone doesn't observe daylight saving time.
Tasmania Standard Time	(UTC+10:00)	Hobart	
Tocantins Standard Time	(UTC-03:00)	Araguaina	
Tokyo Standard Time	(UTC+09:00)	Osaka, Sapporo, Tokyo	This time zone doesn't observe daylight saving time.
Tomsk Standard Time	(UTC+07:00)	Tomsk	
Tonga Standard Time	(UTC+13:00)	Nuku'alofa	This time zone doesn't observe daylight saving time.
Transbaikal Standard Time	(UTC+09:00)	Chita	
Turkey Standard Time	(UTC+03:00)	Istanbul	
Turks And Caicos Standard Time	(UTC-05:00)	Turks and Caicos	
Ulaanbaatar Standard Time	(UTC+08:00)	Ulaanbaatar	This time zone doesn't observe daylight saving time.

Time zone	Standard time offset	Description	Notes
US Eastern Standard Time	(UTC-05:00)	Indiana (East)	
US Mountain Standard Time	(UTC-07:00)	Arizona	This time zone doesn't observe daylight saving time.
UTC	UTC	Coordinated Universal Time	This time zone doesn't observe daylight saving time.
UTC-02	(UTC-02:00)	Coordinated Universal Time-02	This time zone doesn't observe daylight saving time.
UTC-08	(UTC-08:00)	Coordinated Universal Time-08	
UTC-09	(UTC-09:00)	Coordinated Universal Time-09	
UTC-11	(UTC-11:00)	Coordinated Universal Time-11	This time zone doesn't observe daylight saving time.
UTC+12	(UTC+12:00)	Coordinated Universal Time+12	This time zone doesn't observe daylight saving time.
UTC+13	(UTC+13:00)	Coordinated Universal Time+13	
Venezuela Standard Time	(UTC-04:00)	Caracas	This time zone doesn't observe daylight saving time.
Vladivostok Standard Time	(UTC+10:00)	Vladivostok	
Volgograd Standard Time	(UTC+04:00)	Volgograd	
W. Australia Standard Time	(UTC+08:00)	Perth	This time zone doesn't observe daylight saving time.
W. Central Africa Standard Time	(UTC+01:00)	West Central Africa	This time zone doesn't observe daylight saving time.
W. Europe Standard Time	(UTC+01:00)	Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna	
W. Mongolia Standard Time	(UTC+07:00)	Hovd	
West Asia Standard Time	(UTC+05:00)	Ashgabat, Tashkent	This time zone doesn't observe daylight saving time.

Time zone	Standard time offset	Description	Notes
West Bank Standard Time	(UTC+02:00)	Gaza, Hebron	
West Pacific Standard Time	(UTC+10:00)	Guam, Port Moresby	This time zone doesn't observe daylight saving time.
Yakutsk Standard Time	(UTC+09:00)	Yakutsk	

Licensing Microsoft SQL Server on Amazon RDS

When you set up an Amazon RDS DB instance for Microsoft SQL Server, the software license is included.

This means that you don't need to purchase SQL Server licenses separately. AWS holds the license for the SQL Server database software. Amazon RDS pricing includes the software license, underlying hardware resources, and Amazon RDS management capabilities.

Amazon RDS supports the following Microsoft SQL Server editions:

- Enterprise
- Standard
- Web
- Express

Note

Licensing for SQL Server Web Edition supports only public and internet-accessible webpages, websites, web applications, and web services. This level of support is required for compliance with Microsoft's usage rights. For more information, see [AWS service terms](#).

Amazon RDS supports Multi-AZ deployments for DB instances running Microsoft SQL Server by using SQL Server Database Mirroring (DBM) or Always On Availability Groups (AGs). There are no additional licensing requirements for Multi-AZ deployments. For more information, see [Multi-AZ deployments for Microsoft SQL Server \(p. 698\)](#).

Restoring license-terminated DB instances

Amazon RDS takes snapshots of license-terminated DB instances. If your instance is terminated for licensing issues, you can restore it from the snapshot to a new DB instance. New DB instances have a license included.

For more information, see [Restoring license-terminated DB instances \(p. 819\)](#).

Development and test

Because of licensing requirements, we can't offer SQL Server Developer Edition on Amazon RDS. You can use Express Edition for many development, testing, and other nonproduction needs. However, if you need the full feature capabilities of an enterprise-level installation of SQL Server for development, you can download and install SQL Server Developer Edition (and other MSDN products) on Amazon EC2. Dedicated infrastructure isn't required for Developer Edition. By using your own host, you also gain access to other programmability features that are not accessible on Amazon RDS. For more information on the difference between SQL Server editions, see [Editions and supported features of SQL Server 2017](#) in the Microsoft documentation.

Connecting to a DB instance running the Microsoft SQL Server database engine

After Amazon RDS provisions your DB instance, you can use any standard SQL client application to connect to the DB instance. In this topic, you connect to your DB instance by using either Microsoft SQL Server Management Studio (SSMS) or SQL Workbench/J.

For an example that walks you through the process of creating and connecting to a sample DB instance, see [Creating a Microsoft SQL Server DB instance and connecting to it \(p. 80\)](#).

Connecting to your DB instance with Microsoft SQL Server Management Studio

In this procedure, you connect to your sample DB instance by using Microsoft SQL Server Management Studio (SSMS). To download a standalone version of this utility, see [Download SQL Server Management Studio \(SSMS\)](#) in the Microsoft documentation.

To connect to a DB instance using SSMS

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region of your DB instance.
3. Find the Domain Name System (DNS) name and port number for your DB instance:
 - a. Open the RDS console and choose **Databases** to display a list of your DB instances.
 - b. Choose the SQL Server DB instance name to display its details.
 - c. On the **Connectivity & security** tab, copy the endpoint. Also, note the port number. You need both the endpoint and the port number to connect to the DB instance.

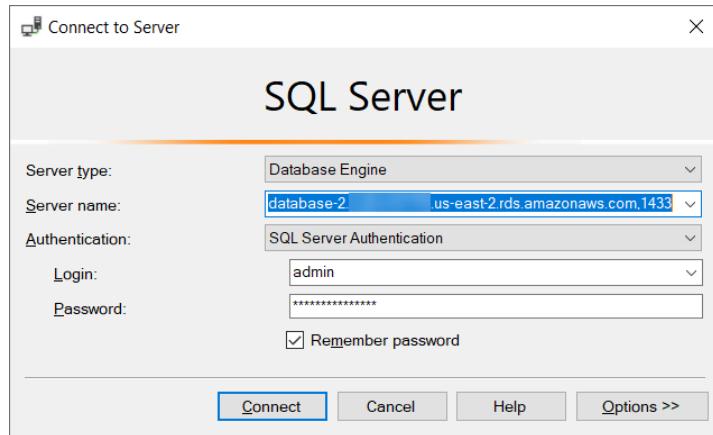
The screenshot shows the Amazon RDS console interface for a DB instance named 'database-2'. The 'Summary' tab is visible at the top, showing basic details like DB identifier and role. Below it, the 'Connectivity & security' tab is selected, showing the endpoint (database-2.REDACTED.us-east-2.rds.amazonaws.com) and port (1433).

Summary	
DB identifier	database-2
Role	Instance

Connectivity & security	
Endpoint	database-2. REDACTED .us-east-2.rds.amazonaws.com
Port	1433

4. Start SQL Server Management Studio.

The **Connect to Server** dialog box appears.



5. Provide the information for your DB instance:

- a. For **Server type**, choose **Database Engine**.
- b. For **Server name**, enter the DNS name and port number of your DB instance, separated by a comma.

Important

Change the colon between the DNS name and port number to a comma.

For example, your server name should look like the following.

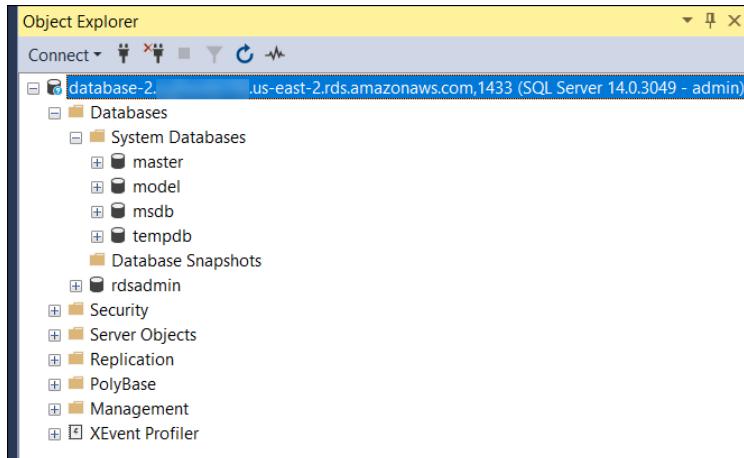
database-2.cg034itsfake.us-east-1.rds.amazonaws.com,1433

- c. For **Authentication**, choose **SQL Server Authentication**.
- d. For **Login**, enter the master user name for your DB instance.
- e. For **Password**, enter the password for your DB instance.

6. Choose **Connect**.

After a few moments, SSMS connects to your DB instance. If you can't connect to your DB instance, see [Security group considerations \(p. 660\)](#) and [Troubleshooting connections to your SQL Server DB instance \(p. 661\)](#).

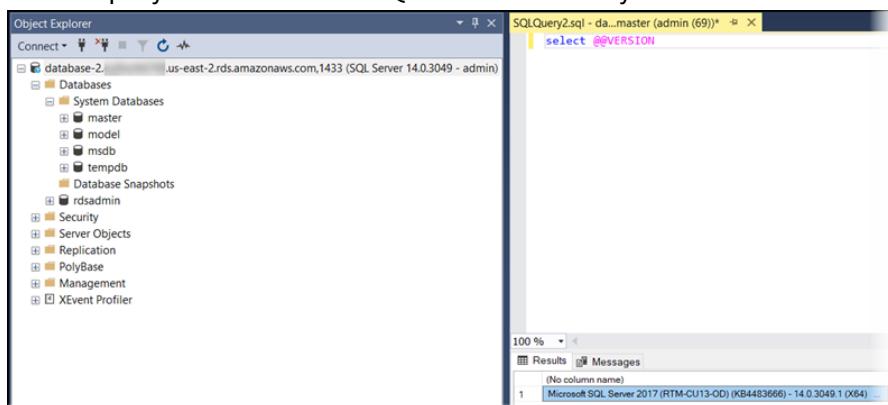
7. Your SQL Server DB instance comes with SQL Server's standard built-in system databases (`master`, `model`, `msdb`, and `tempdb`). To explore the system databases, do the following:
 - a. In SSMS, on the **View** menu, choose **Object Explorer**.
 - b. Expand your DB instance, expand **Databases**, and then expand **System Databases**.



8. Your SQL Server DB instance also comes with a database named `rdsadmin`. Amazon RDS uses this database to store the objects that it uses to manage your database. The `rdsadmin` database also includes stored procedures that you can run to perform advanced tasks. For more information, see [Common DBA tasks for Microsoft SQL Server \(p. 809\)](#).
9. You can now start creating your own databases and running queries against your DB instance and databases as usual. To run a test query against your DB instance, do the following:
 - a. In SSMS, on the **File** menu point to **New** and then choose **Query with Current Connection**.
 - b. Enter the following SQL query.

```
select @@VERSION
```

- c. Run the query. SSMS returns the SQL Server version of your Amazon RDS DB instance.



Connecting to your DB instance with SQL Workbench/J

This example shows how to connect to a DB instance running the Microsoft SQL Server database engine by using the SQL Workbench/J database tool. To download SQL Workbench/J, see [SQL Workbench/J](#).

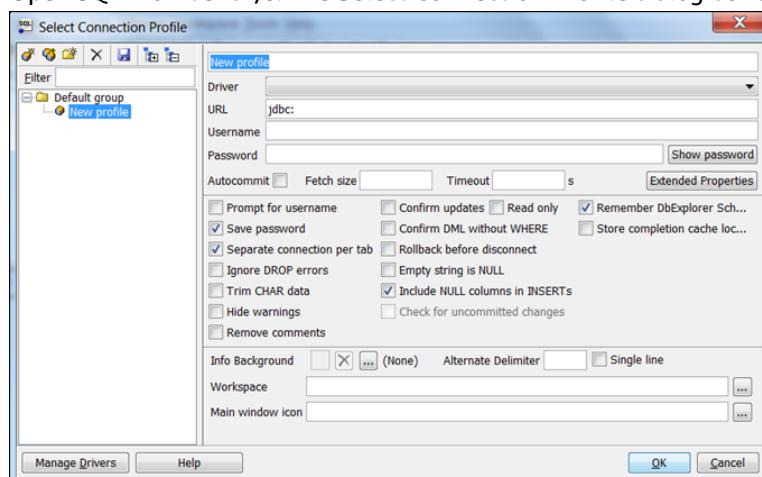
SQL Workbench/J uses JDBC to connect to your DB instance. You also need the JDBC driver for SQL Server. To download this driver, see [Microsoft JDBC drivers 4.1 \(preview\) and 4.0 for SQL Server](#).

To connect to a DB instance using SQL Workbench/J

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region of your DB instance.
3. Find the DNS name and port number for your DB instance:
 - a. Open the RDS console, then choose **Databases** to display a list of your DB instances.
 - b. Choose the name of your SQL Server DB instance to display its details.

The screenshot shows the 'Summary' tab of the RDS console for a database instance named 'database-2'. It displays basic information like DB identifier, CPU usage, and role. Below it, the 'Connectivity & security' tab is selected, showing the endpoint and port details: Endpoint is 'database-2.[REDACTED].us-east-2.rds.amazonaws.com' and Port is '1433'.

- c. On the **Connectivity** tab, copy the endpoint. Also, note the port used by the DB instance.
4. Open SQL Workbench/J. The **Select Connection Profile** dialog box appears, as shown following.

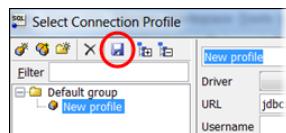


5. In the first box at the top of the dialog box, enter a name for the profile.
6. For **Driver**, choose **SQL JDBC 4.0**.

7. For **URL**, enter `jdbc:sqlserver://`, then enter the endpoint of your DB instance. For example, the URL value might be the following.

```
jdbc:sqlserver://sqlsvr-pdz.abcd12340.us-west-2.rds.amazonaws.com:1433
```

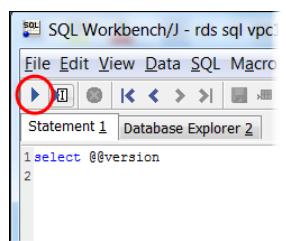
8. For **Username**, enter the master user name for the DB instance.
9. For **Password**, enter the password for the master user.
10. Choose the save icon in the dialog toolbar, as shown following.



11. Choose **OK**. After a few moments, SQL Workbench/J connects to your DB instance. If you can't connect to your DB instance, see [Security group considerations \(p. 660\)](#) and [Troubleshooting connections to your SQL Server DB instance \(p. 661\)](#).
12. In the query pane, enter the following SQL query.

```
select @@VERSION
```

13. Choose the **Execute** icon in the toolbar, as shown following.



The query returns the version information for your DB instance, similar to the following.

```
Microsoft SQL Server 2012 - 11.0.2100.60 (X64)
```

Security group considerations

To connect to your DB instance, your DB instance must be associated with a security group. This security group contains the IP addresses and network configuration that you use to access the DB instance. You might have associated your DB instance with an appropriate security group when you created your DB instance. If you assigned a default, no-configured security group when you created your DB instance, your DB instance firewall prevents connections.

In some cases, you might need to create a new security group to enable access. If so, the type of security group to create depends on what Amazon EC2 platform your DB instance is on. To determine your platform, see [Determining whether you are using the EC2-VPC or EC2-Classic platform \(p. 1718\)](#). In general, if your DB instance is on the EC2-Classic platform, you create a DB security group. If your DB instance is on the VPC platform, you create a VPC security group.

For instructions on creating a new security group, see [Controlling access with security groups \(p. 1699\)](#). For a topic that walks you through the process of setting up rules for your VPC security group, see [Tutorial: Create an Amazon VPC for use with a DB instance \(p. 1737\)](#).

After you have created the new security group, modify your DB instance to associate it with the security group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

You can enhance security by using SSL to encrypt connections to your DB instance. For more information, see [Using SSL with a Microsoft SQL Server DB instance \(p. 704\)](#).

Troubleshooting connections to your SQL Server DB instance

The following table shows error messages that you might encounter when you attempt to connect to your SQL Server DB instance. For more information on connection issues, see [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).

Issue	Troubleshooting suggestions
Could not open a connection to SQL Server – Microsoft SQL Server, Error: 53	<p>Make sure that you specified the server name correctly. For Server name, enter the DNS name and port number of your sample DB instance, separated by a comma.</p> <p>Important If you have a colon between the DNS name and port number, change the colon to a comma. Your server name should look like the following example.</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;">sample-instance.cg034itsfake.us-east-1.rds.amazonaws.com,1433</div>
No connection could be made because the target machine actively refused it – Microsoft SQL Server, Error: 10061	You were able to reach the DB instance but the connection was refused. This issue is usually caused by specifying the user name or password incorrectly. Verify the user name and password, then retry.
A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was not found or was not accessible... The wait operation timed out – Microsoft SQL Server, Error: 258	<p>The access rules enforced by your local firewall and the IP addresses authorized to access your DB instance might not match. The problem is most likely the inbound rules in your security group.</p> <p>Your database instance must be publicly accessible. To connect to it from outside of the VPC, the instance must have a public IP address assigned.</p>

Updating applications to connect to Microsoft SQL Server DB instances using new SSL/TLS certificates

As of September 19, 2019, Amazon RDS has published new Certificate Authority (CA) certificates for connecting to your RDS DB instances using Secure Socket Layer or Transport Layer Security (SSL/TLS). Following, you can find information about updating your applications to use the new certificates.

This topic can help you to determine whether any client applications use SSL/TLS to connect to your DB instances. If they do, you can further check whether those applications require certificate verification to connect.

Note

Some applications are configured to connect to SQL Server DB instances only if they can successfully verify the certificate on the server.

For such applications, you must update your client application trust stores to include the new CA certificates.

After you update your CA certificates in the client application trust stores, you can rotate the certificates on your DB instances. We strongly recommend testing these procedures in a development or staging environment before implementing them in your production environments.

For more information about certificate rotation, see [Rotating your SSL/TLS certificate \(p. 1636\)](#). For more information about downloading certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#). For information about using SSL/TLS with Microsoft SQL Server DB instances, see [Using SSL with a Microsoft SQL Server DB instance \(p. 704\)](#).

Topics

- [Determining whether any applications are connecting to your Microsoft SQL Server DB instance using SSL \(p. 662\)](#)
- [Determining whether a client requires certificate verification in order to connect \(p. 663\)](#)
- [Updating your application trust store \(p. 664\)](#)

Determining whether any applications are connecting to your Microsoft SQL Server DB instance using SSL

Check the DB instance configuration for the value of the `rds.force_ssl` parameter. By default, the `rds.force_ssl` parameter is set to 0 (off). If the `rds.force_ssl` parameter is set to 1 (on), clients are required to use SSL/TLS for connections. For more information about parameter groups, see [Working with DB parameter groups \(p. 228\)](#). You can also find the setting for this parameter in the `sys.dm_server_registry` DMV.

Run the following query to get the current encryption option for all the open connections to a DB instance. The column `ENCRYPT_OPTION` returns `TRUE` if the connection is encrypted.

```
select SESSION_ID,
       ENCRYPT_OPTION,
       NET_TRANSPORT,
       AUTH_SCHEME
  from SYS.DM_EXEC_CONNECTIONS
```

This query shows only the current connections. It doesn't show whether applications that have connected and disconnected in the past have used SSL.

Determining whether a client requires certificate verification in order to connect

You can check whether different types of clients require certificate verification to connect.

Note

If you use connectors other than the ones listed, see the specific connector's documentation for information about how it enforces encrypted connections. For more information, see [Connection modules for Microsoft SQL databases](#) in the Microsoft SQL Server documentation.

SQL Server Management Studio

Check whether encryption is enforced for SQL Server Management Studio connections:

1. Launch SQL Server Management Studio.
2. For **Connect to server**, enter the server information, login user name, and password.
3. Choose **Options**.
4. Check if **Encrypt connection** is selected in the connect page.

For more information about SQL Server Management Studio, see [Use SQL Server Management Studio](#).

Sqlcmd

The following example with the `sqlcmd` client shows how to check a script's SQL Server connection to determine whether successful connections require a valid certificate. For more information, see [Connecting with sqlcmd](#) in the Microsoft SQL Server documentation.

When using `sqlcmd`, an SSL connection requires verification against the server certificate if you use the `-N` command argument to encrypt connections, as in the following example.

```
$ sqlcmd -N -S dbinstance.rds.amazonaws.com -d ExampleDB
```

Note

If `sqlcmd` is invoked with the `-C` option, it trusts the server certificate, even if that doesn't match the client-side trust store.

ADO.NET

In the following example, the application connects using SSL, and the server certificate must be verified.

```
using SQLC = Microsoft.Data.SqlClient;  
  
...  
  
static public void Main()  
{  
    using (var connection = new SQLC.SqlConnection(  
        "Server=tcp:dbinstance.rds.amazonaws.com;" +  
        "Database=ExampleDB;User ID=LOGIN_NAME;" +  
        "Password=YOUR_PASSWORD;" +  
        "Encrypt=True;TrustServerCertificate=False;"  
    ))
```

```
{  
    connection.Open();  
    ...  
}
```

Java

In the following example, the application connects using SSL, and the server certificate must be verified.

```
String connectionUrl =  
    "jdbc:sqlserver://dbinstance.rds.amazonaws.com;" +  
    "databaseName=ExampleDB;integratedSecurity=true;" +  
    "encrypt=true;trustServerCertificate=false";
```

To enable SSL encryption for clients that connect using JDBC, you might need to add the Amazon RDS certificate to the Java CA certificate store. For instructions, see [Configuring the client for encryption](#) in the Microsoft SQL Server documentation. You can also provide the trusted CA certificate file name directly by appending `trustStore=path-to-certificate-trust-store-file` to the connection string.

Note

If you use `TrustServerCertificate=true` (or its equivalent) in the connection string, the connection process skips the trust chain validation. In this case, the application connects even if the certificate can't be verified. Using `TrustServerCertificate=false` enforces certificate validation and is a best practice.

Updating your application trust store

You can update the trust store for applications that use Microsoft SQL Server. For instructions, see [Encrypting specific connections \(p. 705\)](#). Also, see [Configuring the client for encryption](#) in the Microsoft SQL Server documentation.

If you are using an operating system other than Microsoft Windows, see the software distribution documentation for SSL/TLS implementation for information about adding a new root CA certificate. For example, OpenSSL and GnuTLS are popular options. Use the implementation method to add trust to the RDS root CA certificate. Microsoft provides instructions for configuring certificates on some systems.

Note

When you update the trust store, you can retain older certificates in addition to adding the new certificates.

Updating your application trust store for JDBC

You can update the trust store for applications that use JDBC for SSL/TLS connections.

To update the trust store for JDBC applications

1. Download the 2019 root certificate that works for all AWS Regions and put the file in the trust store directory.

For information about downloading the root certificate, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).
2. Convert the certificate to .der format using the following command.

```
openssl x509 -outform der -in rds-ca-2019-root.pem -out rds-ca-2019-root.der
```

Replace the file name with the one that you downloaded.

3. Import the certificate into the key store using the following command.

```
keytool -import -alias rds-root -keystore clientkeystore -file rds-ca-2019-root.der
```

4. Confirm that the key store was updated successfully.

```
keytool -list -v -keystore clientkeystore.jks
```

Enter the key store password when you are prompted for it.

Your output should contain the following.

```
rds-root,date, trustedCertEntry,  
Certificate fingerprint (SHA1):  
D4:0D:DB:29:E3:75:0D:FF:A6:71:C3:14:0B:BF:5F:47:8D:1C:80:96  
# This fingerprint should match the output from the below command  
openssl x509 -fingerprint -in rds-ca-2019-root.pem -noout
```

Important

After you have determined that your database connections use SSL/TLS and have updated your application trust store, you can update your database to use the rds-ca-2019 certificates. For instructions, see step 3 in [Updating your CA certificate by modifying your DB instance \(p. 1636\)](#).

Upgrading the Microsoft SQL Server DB engine

When Amazon RDS supports a new version of a database engine, you can upgrade your DB instances to the new version. There are two kinds of upgrades for SQL Server DB instances: major version upgrades and minor version upgrades.

Major version upgrades can contain database changes that are not backward-compatible with existing applications. As a result, you must manually perform major version upgrades of your DB instances. You can initiate a major version upgrade by modifying your DB instance. However, before you perform a major version upgrade, we recommend that you test the upgrade by following the steps described in [Testing an upgrade \(p. 669\)](#).

In contrast, *minor version upgrades* include only changes that are backward-compatible with existing applications. You can initiate a minor version upgrade manually by modifying your DB instance.

Amazon RDS on SQL Server doesn't support automatic minor version upgrades. You can confirm this by using the `describe-db-engine-versions` AWS CLI command. For example:

```
aws rds describe-db-engine-versions --engine sqlserver-se --engine-version 14.00.3049.1.v1
```

In this example, the CLI command returns the following response that indicates that the upgrade will *not* be automatic, even if **Auto minor version upgrade** had been enabled:

```
...
"ValidUpgradeTarget": [
    {
        "Engine": "sqlserver-se",
        "EngineVersion": "14.00.3192.2.v1",
        "Description": "SQL Server 2017 14.00.3192.2.v1",
        "AutoUpgrade": false,
        "IsMajorVersionUpgrade": false
    }
]
...
```

For more information about performing upgrades, see [Upgrading a SQL Server DB instance \(p. 669\)](#). For information about what SQL Server versions are available on Amazon RDS, see [Microsoft SQL Server on Amazon RDS \(p. 630\)](#).

Topics

- [Overview of upgrading \(p. 666\)](#)
- [Major version upgrades \(p. 667\)](#)
- [Multi-AZ and in-memory optimization considerations \(p. 668\)](#)
- [Option and parameter group considerations \(p. 668\)](#)
- [Testing an upgrade \(p. 669\)](#)
- [Upgrading a SQL Server DB instance \(p. 669\)](#)
- [Upgrading deprecated DB instances before support ends \(p. 670\)](#)

Overview of upgrading

Amazon RDS takes two DB snapshots during the upgrade process. The first DB snapshot is of the DB instance before any upgrade changes have been made. If the upgrade doesn't work for your databases,

you can restore this snapshot to create a DB instance running the old version. The second DB snapshot is taken after the upgrade completes.

Note

Amazon RDS only takes DB snapshots if you have set the backup retention period for your DB instance to a number greater than 0. To change your backup retention period, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

After an upgrade is complete, you can't revert to the previous version of the database engine. If you want to return to the previous version, restore the DB snapshot that was taken before the upgrade to create a new DB instance.

During a minor or major version upgrade of SQL Server, the **Free Storage Space** and **Disk Queue Depth** metrics will display -1. After the upgrade is complete, both metrics will return to normal.

Major version upgrades

Amazon RDS currently supports the following major version upgrades to a Microsoft SQL Server DB instance.

You can upgrade your existing DB instance to SQL Server 2017 or 2019 from any version except SQL Server 2008. To upgrade from SQL Server 2008, first upgrade to one of the other versions.

Current version	Supported upgrade versions
SQL Server 2017	SQL Server 2019
SQL Server 2016	SQL Server 2019 SQL Server 2017
SQL Server 2014	SQL Server 2019 SQL Server 2017 SQL Server 2016
SQL Server 2012	SQL Server 2019 SQL Server 2017 SQL Server 2016 SQL Server 2014
SQL Server 2008 R2 (Deprecated)	SQL Server 2016 SQL Server 2014 SQL Server 2012

You can use an AWS CLI query, such as the following example, to find the available upgrades for a particular database engine version.

Example

For Linux, macOS, or Unix:

```
aws rds describe-db-engine-versions \
--engine sqlserver-se \
```

```
--engine-version 14.00.3049.1.v1 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" \
--output table
```

For Windows:

```
aws rds describe-db-engine-versions ^
--engine sqlserver-se ^
--engine-version 14.00.3049.1.v1 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" ^
--output table
```

The output shows that you can upgrade version 14.00.3049.1 to the latest SQL Server 2017 or 2019 version.

```
-----
|DescribeDBEngineVersions|
+-----+
|   EngineVersion      |
+-----+
| 14.00.3294.2.v1     |
| 15.00.4043.16.v1    |
+-----+
```

Database compatibility level

You can use Microsoft SQL Server database compatibility levels to adjust some database behaviors to mimic previous versions of SQL Server. For more information, see [Compatibility level](#) in the Microsoft documentation.

When you upgrade your DB instance, all existing databases remain at their original compatibility level. For example, if you upgrade from SQL Server 2012 to SQL Server 2014, all existing databases have a compatibility level of 110. Any new database created after the upgrade have compatibility level 120.

You can change the compatibility level of a database by using the ALTER DATABASE command. For example, to change a database named `customeracct` to be compatible with SQL Server 2014, issue the following command:

```
ALTER DATABASE customeracct SET COMPATIBILITY_LEVEL = 120
```

Multi-AZ and in-memory optimization considerations

Amazon RDS supports Multi-AZ deployments for DB instances running Microsoft SQL Server by using SQL Server Database Mirroring (DBM) or Always On Availability Groups (AGs). For more information, see [Multi-AZ deployments for Microsoft SQL Server \(p. 698\)](#).

If your DB instance is in a Multi-AZ deployment, both the primary and standby instances are upgraded. Amazon RDS does rolling upgrades. You have an outage only for the duration of a failover.

SQL Server 2014 through 2019 Enterprise Edition support in-memory optimization.

Option and parameter group considerations

Option group considerations

If your DB instance uses a custom option group, in some cases Amazon RDS can't automatically assign your DB instance a new option group. For example, when you upgrade to a new major version. In that

case, you must specify a new option group when you upgrade. We recommend that you create a new option group, and add the same options to it as your existing custom option group.

For more information, see [Creating an option group \(p. 214\)](#) or [Copying an option group \(p. 215\)](#).

Parameter group considerations

If your DB instance uses a custom parameter group, in some cases Amazon RDS can't automatically assign your DB instance a new parameter group. For example, when you upgrade to a new major version. In that case, you must specify a new parameter group when you upgrade. We recommend that you create a new parameter group, and configure the parameters as in your existing custom parameter group.

For more information, see [Creating a DB parameter group \(p. 229\)](#) or [Copying a DB parameter group \(p. 236\)](#).

Testing an upgrade

Before you perform a major version upgrade on your DB instance, you should thoroughly test your database, and all applications that access the database, for compatibility with the new version. We recommend that you use the following procedure.

To test a major version upgrade

1. Review [Upgrade SQL Server](#) in the Microsoft documentation for the new version of the database engine to see if there are compatibility issues that might affect your database or applications.
2. If your DB instance uses a custom option group, create a new option group compatible with the new version you are upgrading to. For more information, see [Option group considerations \(p. 668\)](#).
3. If your DB instance uses a custom parameter group, create a new parameter group compatible with the new version you are upgrading to. For more information, see [Parameter group considerations \(p. 669\)](#).
4. Create a DB snapshot of the DB instance to be upgraded. For more information, see [Creating a DB snapshot \(p. 346\)](#).
5. Restore the DB snapshot to create a new test DB instance. For more information, see [Restoring from a DB snapshot \(p. 349\)](#).
6. Modify this new test DB instance to upgrade it to the new version, by using one of the following methods:
 - [Console \(p. 271\)](#)
 - [AWS CLI \(p. 272\)](#)
 - [RDS API \(p. 272\)](#)
7. Evaluate the storage used by the upgraded instance to determine if the upgrade requires additional storage.
8. Run as many of your quality assurance tests against the upgraded DB instance as needed to ensure that your database and application work correctly with the new version. Implement any new tests needed to evaluate the impact of any compatibility issues you identified in step 1. Test all stored procedures and functions. Direct test versions of your applications to the upgraded DB instance.
9. If all tests pass, then perform the upgrade on your production DB instance. We recommend that you do not allow write operations to the DB instance until you confirm that everything is working correctly.

Upgrading a SQL Server DB instance

For information about manually or automatically upgrading a SQL Server DB instance, see the following:

- [Upgrading a DB instance engine version \(p. 271\)](#)
- [Best practices for upgrading SQL Server 2008 R2 to SQL Server 2016 on Amazon RDS for SQL Server](#)

Important

If you have any snapshots that are encrypted using AWS KMS, we recommend that you initiate an upgrade before support ends.

Upgrading deprecated DB instances before support ends

After a major version is deprecated, you can't install it on new DB instances. RDS will try to automatically upgrade all existing DB instances.

If you need to restore a deprecated DB instance, you can do point-in-time recovery (PITR) or restore a snapshot. Doing this gives you temporary access a DB instance that uses the version that is being deprecated. However, after a major version is fully deprecated, these DB instances will also be automatically upgraded to a supported version.

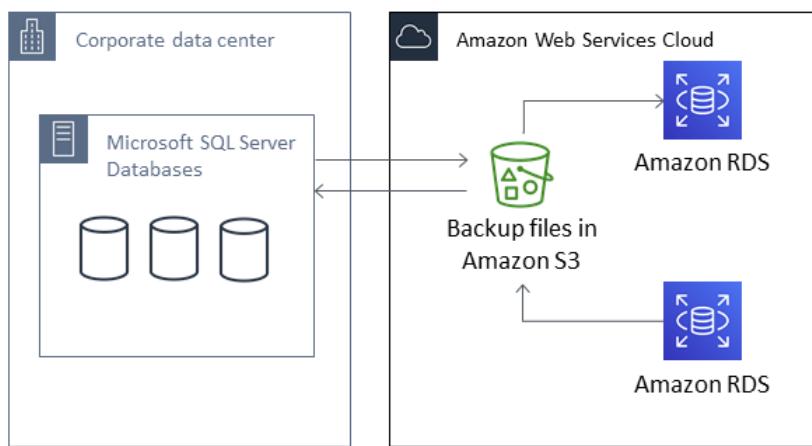
Importing and exporting SQL Server databases

Amazon RDS supports native backup and restore for Microsoft SQL Server databases using full backup files (.bak files). When you use RDS, you access files stored in Amazon S3 rather than using the local file system on the database server.

For example, you can create a full backup from your local server, store it on S3, and then restore it onto an existing Amazon RDS DB instance. You can also make backups from RDS, store them on S3, and then restore them wherever you want.

Native backup and restore is available in all AWS Regions for Single-AZ and Multi-AZ DB instances, including Multi-AZ DB instances with read replicas. Native backup and restore is available for all editions of Microsoft SQL Server supported on Amazon RDS.

The following diagram shows the supported scenarios.



Using native .bak files to back up and restore databases is usually the fastest way to back up and restore databases. There are many additional advantages to using native backup and restore. For example, you can do the following:

- Migrate databases to or from Amazon RDS.
- Move databases between RDS for SQL Server DB instances.
- Migrate data, schemas, stored procedures, triggers, and other database code inside .bak files.
- Backup and restore single databases, instead of entire DB instances.
- Create copies of databases for development, testing, training, and demonstrations.
- Store and transfer backup files with Amazon S3, for an added layer of protection for disaster recovery.

Limitations and recommendations

The following are some limitations to using native backup and restore:

- You can't back up to, or restore from, an Amazon S3 bucket in a different AWS Region from your Amazon RDS DB instance.
- We strongly recommend that you don't restore backups from one time zone to a different time zone. If you restore backups from one time zone to a different time zone, you must audit your queries and applications for the effects of the time zone change.
- Amazon S3 has a size limit of 5 TB per file. For native backups of larger databases, you can use multifile backup.

- The maximum database size that can be backed up to S3 depends on the available memory, CPU, I/O, and network resources on the DB instance. The larger the database, the more memory the backup agent consumes. Our testing shows that you can make a compressed backup of a 16-TB database on our newest-generation instance types from 2xlarge instance sizes and larger, given sufficient system resources.
- You can't back up to or restore from more than 10 backup files at the same time.
- A differential backup is based on the last full backup. For differential backups to work, you can't take a snapshot between the last full backup and the differential backup. If you want a differential backup, but a manual or automated snapshot exists, then do another full backup before proceeding with the differential backup.
- Differential and log restores aren't supported for databases with files that have their file_guid (unique identifier) set to NULL.
- You can run up to two backup or restore tasks at the same time.
- You can't perform native log backups from SQL Server on Amazon RDS.
- RDS supports native restores of databases up to 16 TB. Native restores of databases on SQL Server Express Edition are limited to 10 GB.
- You can't do a native backup during the maintenance window, or any time Amazon RDS is in the process of taking a snapshot of the database. If a native backup task overlaps with the RDS daily backup window, the native backup task is canceled.
- On Multi-AZ DB instances, you can only natively restore databases that are backed up in the full recovery model.
- Restoring from differential backups on Multi-AZ instances isn't supported.
- Calling the RDS procedures for native backup and restore within a transaction isn't supported.
- Use a symmetric AWS KMS customer master key (CMK) to encrypt your backups. Amazon RDS doesn't support asymmetric CMKs. For more information, see [Using symmetric and asymmetric keys](#) in the *AWS Key Management Service Developer Guide*.
- Native backup files are encrypted with the specified AWS KMS CMK using the "Encryption-Only" crypto mode. When you are restoring encrypted backup files, be aware that they were encrypted with the "Encryption-Only" crypto mode.
- You can't restore a database that contains a FILESTREAM file group.

If your database can be offline while the backup file is created, copied, and restored, we recommend that you use native backup and restore to migrate it to RDS. If your on-premises database can't be offline, we recommend that you use the AWS Database Migration Service to migrate your database to Amazon RDS. For more information, see [What is AWS Database Migration Service?](#)

Native backup and restore isn't intended to replace the data recovery capabilities of the cross-region snapshot copy feature. We recommend that you use snapshot copy to copy your database snapshot to another AWS Region for cross-region disaster recovery in Amazon RDS. For more information, see [Copying a snapshot \(p. 352\)](#).

Setting up for native backup and restore

To set up for native backup and restore, you need three components:

1. An Amazon S3 bucket to store your backup files.

You must have an S3 bucket to use for your backup files and then upload backups you want to migrate to RDS. If you already have an Amazon S3 bucket, you can use that. If you don't, you can [create a bucket](#). Alternatively, you can choose to have a new bucket created for you when you add the `SQLSERVER_BACKUP_RESTORE` option by using the AWS Management Console.

For information on using S3, see the *Amazon Simple Storage Service Getting Started Guide* for a simple introduction. For more depth, see the *Amazon Simple Storage Service Console User Guide*.

2. An AWS Identity and Access Management (IAM) role to access the bucket.

If you already have an IAM role, you can use that. You can choose to have a new IAM role created for you when you add the `SQLSERVER_BACKUP_RESTORE` option by using the AWS Management Console. Alternatively, you can create a new one manually.

If you want to create a new IAM role manually, take the approach discussed in the next section. Do the same if you want to attach trust relationships and permissions policies to an existing IAM role.

3. The SOL SERVER BACKUP RESTORE option added to an option group on your DB instance.

To enable native backup and restore on your DB instance, you add the `SQlSERVER_BACKUP_RESTORE` option to an option group on your DB instance. For more information and instructions, see [Support for native backup and restore in SQL Server \(p. 751\)](#).

Manually creating an IAM role for native backup and restore

If you want to manually create a new IAM role to use with native backup and restore, you can do so. In this case, you create a role to delegate permissions from the Amazon RDS service to your Amazon S3 bucket. When you create an IAM role, you attach a trust relationship and a permissions policy. The trust relationship allows RDS to assume this role. The permissions policy defines the actions this role can do. For more information about creating the role, see [Creating a role to delegate permissions to an AWS service](#).

For the native backup and restore feature, use trust relationships and permissions policies similar to the examples in this section. In the following example, we use the service principal name `rds.amazonaws.com` as an alias for all service accounts. In the other examples, we specify an Amazon Resource Name (ARN) to identify another account, user, or role that we're granting access to in the trust policy.

Example Trust relationship for native backup and restore

```
{  
    "Version": "2012-10-17",  
    "Statement":  
    [ {  
        "Effect": "Allow",  
        "Principal": {"Service": "rds.amazonaws.com"},  
        "Action": "sts:AssumeRole"  
    }]  
}
```

The following example uses an ARN to specify a resource. For more information on using ARNs, see [Amazon resource names \(ARNs\)](#).

Example Permissions policy for native backup and restore without encryption support

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Effect": "Allow",
      "Action":
        [
          "s3>ListBucket",
          "s3:GetObject"
        ]
    }
  ]
}
```

```

        "s3:GetBucketLocation"
    ],
    "Resource": "arn:aws:s3:::bucket_name"
},
{
"Effect": "Allow",
"Action":
[
    "s3:GetObject",
    "s3:PutObject",
    "s3>ListMultipartUploadParts",
    "s3:AbortMultipartUpload"
],
"Resource": "arn:aws:s3:::bucket_name/*"
}
]
}

```

Example Permissions policy for native backup and restore with encryption support

If you want to encrypt your backup files, include an encryption key in your permissions policy. For more information about encryption keys, see [Getting started](#) in the *AWS Key Management Service Developer Guide*.

Note

You must use a symmetric AWS KMS CMK to encrypt your backups. Amazon RDS doesn't support asymmetric CMKs. For more information, see [Using symmetric and asymmetric keys](#) in the *AWS Key Management Service Developer Guide*.

The IAM role must also be a key user and key administrator for the AWS KMS CMK, that is, it must be specified in the key policy. For more information, see [Creating symmetric CMKs](#) in the *AWS Key Management Service Developer Guide*.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Action":
            [
                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:Encrypt",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:region:account-id:key/key-id"
        },
        {
            "Effect": "Allow",
            "Action":
            [
                "s3>ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::bucket_name"
        },
        {
            "Effect": "Allow",
            "Action":
            [
                "s3:GetObject",
                "s3:PutObject",
                "s3>ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
            "Resource": "arn:aws:s3:::bucket_name/*"
        }
    ]
}
```

```
        "s3:AbortMultipartUpload"
    ],
    "Resource": "arn:aws:s3:::bucket_name/*"
}
]
```

Using native backup and restore

After you have enabled and configured native backup and restore, you can start using it. First, you connect to your Microsoft SQL Server database, and then you call an Amazon RDS stored procedure to do the work. For instructions on connecting to your database, see [Connecting to a DB instance running the Microsoft SQL Server database engine \(p. 656\)](#).

Some of the stored procedures require that you provide an Amazon Resource Name (ARN) to your Amazon S3 bucket and file. The format for your ARN is `arn:aws:s3:::bucket_name/file_name.extension`. Amazon S3 doesn't require an account number or AWS Region in ARNs.

If you also provide an optional AWS KMS customer master key (CMK), the format for the ARN of the key is `arn:aws:kms:region:account-id:key/key-id`. For more information, see [Amazon resource names \(ARNs\) and AWS service namespaces](#). You must use a symmetric AWS KMS CMK to encrypt your backups. Amazon RDS doesn't support asymmetric CMKs. For more information, see [Using symmetric and asymmetric keys](#) in the [AWS Key Management Service Developer Guide](#).

Note

Whether or not you use a KMS CMK, the native backup and restore tasks enable server-side Advanced Encryption Standard (AES) 256-bit encryption by default for files uploaded to S3.

For instructions on how to call each stored procedure, see the following topics:

- [Backing up a database \(p. 675\)](#)
- [Restoring a database \(p. 678\)](#)
- [Restoring a log \(p. 680\)](#)
- [Finishing a database restore \(p. 681\)](#)
- [Working with partially restored databases \(p. 682\)](#)
- [Canceling a task \(p. 682\)](#)
- [Tracking the status of tasks \(p. 682\)](#)

Backing up a database

To back up your database, use the `rds_backup_database` stored procedure.

Note

You can't back up a database during the maintenance window, or while Amazon RDS is taking a snapshot.

Usage

```
exec msdb.dbo.rds_backup_database
@source_db_name='database_name',
@s3_arn_to_backup_to='arn:aws:s3:::bucket_name/file_name.extension',
[@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
[@overwrite_s3_backup_file=0/1],
[@type='DIFFERENTIAL/FULL'],
[@number_of_files=n];
```

The following parameters are required:

- `@source_db_name` – The name of the database to back up.
- `@s3_arn_to_backup_to` – The ARN indicating the Amazon S3 bucket to use for the backup, plus the name of the backup file.

The file can have any extension, but `.bak` is usually used.

The following parameters are optional:

- `@kms_master_key_arn` – The ARN for the symmetric AWS KMS CMK to use to encrypt the item.
- You can't use the default encryption key. If you use the default key, the database won't be backed up.
- If you don't specify a AWS KMS key identifier, the backup file won't be encrypted. For more information, see [Encrypting Amazon RDS resources](#).
- When you specify a AWS KMS CMK, client-side encryption is used.
- Amazon RDS doesn't support asymmetric CMKs. For more information, see [Using symmetric and asymmetric keys in the AWS Key Management Service Developer Guide](#).
- `@overwrite_s3_backup_file` – A value that indicates whether to overwrite an existing backup file.
 - 0 – Doesn't overwrite an existing file. This value is the default.

Setting `@overwrite_s3_backup_file` to 0 returns an error if the file already exists.

- 1 – Overwrites an existing file that has the specified name, even if it isn't a backup file.
- `@type` – The type of backup.
 - `DIFFERENTIAL` – Makes a differential backup.
 - `FULL` – Makes a full backup. This value is the default.

A differential backup is based on the last full backup. For differential backups to work, you can't take a snapshot between the last full backup and the differential backup. If you want a differential backup, but a snapshot exists, then do another full backup before proceeding with the differential backup.

You can look for the last full backup or snapshot using the following example SQL query:

```
select top 1
    database_name
    , backup_start_date
    , backup_finish_date
from msdb.dbo.backupset
where database_name = 'mydatabase'
and type = 'D'
order by backup_start_date desc;
```

- `@number_of_files` – The number of files into which the backup will be divided (chunked). The maximum number is 10.
 - Multifile backup is supported for both full and differential backups.
 - If you enter a value of 1 or omit the parameter, a single backup file is created.

Provide the prefix that the files have in common, then suffix that with an asterisk (*). The asterisk can be anywhere in the `file_name` part of the S3 ARN. The asterisk is replaced by a series of alphanumeric strings in the generated files, starting with 1-of-`number_of_files`.

For example, if the file names in the S3 ARN are `backup*.bak` and you set `@number_of_files=4`, the backup files generated are `backup1-of-4.bak`, `backup2-of-4.bak`, `backup3-of-4.bak`, and `backup4-of-4.bak`.

- If any of the file names already exists, and `@overwrite_s3_backup_file` is 0, an error is returned.

- Multifile backups can only have one asterisk in the *file_name* part of the S3 ARN.
- Single-file backups can have any number of asterisks in the *file_name* part of the S3 ARN.
Asterisks aren't removed from the generated file name.

Examples

Example of differential backup

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup1.bak',
@overwrite_s3_backup_file=1,
@type='DIFFERENTIAL';
```

Example of full backup with encryption

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup1.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE',
@overwrite_s3_backup_file=1,
@type='FULL';
```

Example of multifile backup

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@number_of_files=4;
```

Example of multifile differential backup

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@type='DIFFERENTIAL',
@number_of_files=4;
```

Example of multifile backup with encryption

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE',
@number_of_files=4;
```

Example of multifile backup with S3 overwrite

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@overwrite_s3_backup_file=1,
@number_of_files=4;
```

Example of single-file backup with the @number_of_files parameter

This example generates a backup file named `backup*.bak`.

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@number_of_files=1;
```

Restoring a database

To restore your database, call the `rds_restore_database` stored procedure. Amazon RDS creates an initial snapshot of the database after the restore task is complete and the database is open.

Usage

```
exec msdb.dbo.rds_restore_database
@restore_db_name='database_name',
@s3_arn_to_restore_from='arn:aws:s3:::bucket_name/file_name.extension',
@with_norecovery=0|1,
[@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
[@type='DIFFERENTIAL|FULL'];
```

The following parameters are required:

- `@restore_db_name` – The name of the database to restore.
- `@s3_arn_to_restore_from` – The ARN indicating the Amazon S3 prefix and names of the backup files used to restore the database.
 - For a single-file backup, provide the entire file name.
 - For a multifile backup, provide the prefix that the files have in common, then suffix that with an asterisk (*).
 - If `@s3_arn_to_restore_from` is empty, the following error message is returned: S3 ARN prefix cannot be empty.

The following parameter is required for differential restores, but optional for full restores:

- `@with_norecovery` – The recovery clause to use for the restore operation.
 - Set it to 0 to restore with RECOVERY. In this case, the database is online after the restore.
 - Set it to 1 to restore with NORECOVERY. In this case, the database remains in the RESTORING state after restore task completion. With this approach, you can do later differential restores.
 - For DIFFERENTIAL restores, specify 0 or 1.
 - For FULL restores, this value defaults to 0.

The following parameters are optional:

- `@kms_master_key_arn` – If you encrypted the backup file, the AWS KMS customer master key (CMK) to use to decrypt the file.

When you specify a AWS KMS CMK, client-side encryption is used.
- `@type` – The type of restore. Valid types are DIFFERENTIAL and FULL. The default value is FULL.

Note

For differential restores, either the database must be in the RESTORING state or a task must already exist that restores with NORECOVERY.

You can't restore later differential backups while the database is online.
You can't submit a restore task for a database that already has a pending restore task with RECOVERY.
Full restores with NORECOVERY and differential restores aren't supported on Multi-AZ instances.
Restoring a database on a Multi-AZ instance with read replicas is similar to restoring a database on a Multi-AZ instance. You don't have to take any additional actions to restore a database on a replica.

Examples

Example of single-file restore

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak';
```

Example of multifile restore

To avoid errors when restoring multiple files, make sure that all the backup files have the same prefix, and that no other files use that prefix.

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup*';
```

Example of full database restore with RECOVERY

The following three examples perform the same task, full restore with RECOVERY.

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak';
```

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
[@type='DIFFERENTIAL | FULL'];
```

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='FULL',
@with_norecovery=0;
```

Example of full database restore with encryption

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Example of full database restore with NORECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
```

```
@type='FULL',
@with_norecovery=1;
```

Example of differential restore with NORECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='DIFFERENTIAL',
@with_norecovery=1;
```

Example of differential restore with RECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='DIFFERENTIAL',
@with_norecovery=0;
```

Restoring a log

To restore your log, call the `rds_restore_log` stored procedure.

Usage

```
exec msdb.dbo.rds_restore_log
@restore_db_name='database_name',
@s3_arn_to_restore_from='arn:aws:s3:::bucket_name/log_file_name.extension',
[@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
[@with_norecovery=0|1],
[@stopat='datetime'];
```

The following parameters are required:

- `@restore_db_name` – The name of the database whose log to restore.
- `@s3_arn_to_restore_from` – The ARN indicating the Amazon S3 prefix and name of the log file used to restore the log. The file can have any extension, but `.trn` is usually used.

If `@s3_arn_to_restore_from` is empty, the following error message is returned: S3 ARN prefix cannot be empty.

The following parameters are optional:

- `@kms_master_key_arn` – If you encrypted the log, the AWS KMS customer master key (CMK) to use to decrypt the log.
- `@with_norecovery` – The recovery clause to use for the restore operation. This value defaults to 1.
 - Set it to 0 to restore with RECOVERY. In this case, the database is online after the restore. You can't restore further log backups while the database is online.
 - Set it to 1 to restore with NORECOVERY. In this case, the database remains in the RESTORING state after restore task completion. With this approach, you can do later log restores.
- `@stopat` – A value that specifies that the database is restored to its state at the date and time specified (in datetime format). Only transaction log records written before the specified date and time are applied to the database.

If this parameter isn't specified (it is NULL), the complete log is restored.

Note

For log restores, either the database must be in a state of restoring or a task must already exist that restores with NORECOVERY.

You can't restore log backups while the database is online.

You can't submit a log restore task on a database that already has a pending restore task with RECOVERY.

Log restores aren't supported on Multi-AZ instances.

Examples

Example of log restore

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn';
```

Example of log restore with encryption

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Example of log restore with NORECOVERY

The following two examples perform the same task, log restore with NORECOVERY.

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=1;
```

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn';
```

Example of log restore with RECOVERY

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=0;
```

Example of log restore with STOPAT clause

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=0,
@stopat='2019-12-01 03:57:09';
```

Finishing a database restore

If the last restore task on the database was performed using @with_norecovery=1, the database is now in the RESTORING state. Open this database for normal operation by using the rds_finish_restore stored procedure.

Usage

```
exec msdb.dbo.rds_finish_restore @db_name='database_name';
```

Note

To use this approach, the database must be in the RESTORING state without any pending restore tasks.

The `rds_finish_restore` procedure isn't supported on Multi-AZ instances.

To finish restoring the database, use the master login. Or use the user login that most recently restored the database or log with NORECOVERY.

Working with partially restored databases

Dropping a partially restored database

To drop a partially restored database (left in the RESTORING state), use the `rds_drop_database` stored procedure.

```
exec msdb.dbo.rds_drop_database @db_name='database_name';
```

Note

You can't submit a DROP database request for a database that already has a pending restore or finish restore task.

To drop the database, use the master login. Or use the user login that most recently restored the database or log with NORECOVERY.

Snapshot restore and point-in-time recovery behavior for partially restored databases

Partially restored databases in the source instance (left in the RESTORING state) are dropped from the target instance during snapshot restore and point-in-time recovery.

Cancelling a task

To cancel a backup or restore task, call the `rds_cancel_task` stored procedure.

Note

You can't cancel a FINISH_RESTORE task.

Usage

```
exec msdb.dbo.rds_cancel_task @task_id=ID_number;
```

The following parameter is required:

- `@task_id` – The ID of the task to cancel. You can get the task ID by calling `rds_task_status`.

Tracking the status of tasks

To track the status of your backup and restore tasks, call the `rds_task_status` stored procedure. If you don't provide any parameters, the stored procedure returns the status of all tasks. The status for tasks is updated approximately every two minutes. Task history is retained for 36 days.

Usage

```
exec msdb.dbo.rds_task_status
```

```
[@db_name='database_name'],
[@task_id=ID_number];
```

The following parameters are optional:

- @db_name – The name of the database to show the task status for.
- @task_id – The ID of the task to show the task status for.

Examples

Example of listing the status for a specific task

```
exec msdb.dbo.rds_task_status @task_id=5;
```

Example of listing the status for a specific database and task

```
exec msdb.dbo.rds_task_status
@db_name='my_database',
@task_id=5;
```

Example of listing all tasks and their statuses on a specific database

```
exec msdb.dbo.rds_task_status @db_name='my_database';
```

Example of listing all tasks and their statuses on the current instance

```
exec msdb.dbo.rds_task_status;
```

Response

The `rds_task_status` stored procedure returns the following columns.

Column	Description
<code>task_id</code>	The ID of the task.
<code>task_type</code>	Task type depending on the input parameters, as follows: <ul style="list-style-type: none">• For backup tasks:<ul style="list-style-type: none">• BACKUP_DB – Full database backup• BACKUP_DB_DIFFERENTIAL – Differential database backup• For restore tasks:<ul style="list-style-type: none">• RESTORE_DB – Full database restore with RECOVERY• RESTORE_DB_NORECOVERY – Full database restore with NORECOVERY• RESTORE_DB_DIFFERENTIAL – Differential database restore with RECOVERY• RESTORE_DB_DIFFERENTIAL_NORECOVERY – Differential database restore with NORECOVERY• RESTORE_DB_LOG – Log restore with RECOVERY• RESTORE_DB_LOG_NORECOVERY – Log restore with NORECOVERY• For tasks that finish a restore:<ul style="list-style-type: none">• FINISH_RESTORE – Finish restore and open database

Column	Description
	<p>Amazon RDS creates an initial snapshot of the database after it is open on completion of the following restore tasks:</p> <ul style="list-style-type: none"> • RESTORE_DB • RESTORE_DB_DIFFERENTIAL • RESTORE_DB_LOG • FINISH_RESTORE
database_name	The name of the database that the task is associated with.
% complete	The progress of the task as a percent value.
duration (mins)	The amount of time spent on the task, in minutes.
lifecycle	<p>The status of the task. The possible statuses are the following:</p> <ul style="list-style-type: none"> • CREATED – As soon as you call <code>rds_backup_database</code> or <code>rds_restore_database</code>, a task is created and the status is set to <code>CREATED</code>. • IN_PROGRESS – After a backup or restore task starts, the status is set to <code>IN_PROGRESS</code>. It can take up to 5 minutes for the status to change from <code>CREATED</code> to <code>IN_PROGRESS</code>. • SUCCESS – After a backup or restore task completes, the status is set to <code>SUCCESS</code>. • ERROR – If a backup or restore task fails, the status is set to <code>ERROR</code>. For more information about the error, see the <code>task_info</code> column. • CANCEL_REQUESTED – As soon as you call <code>rds_cancel_task</code>, the status of the task is set to <code>CANCEL_REQUESTED</code>. • CANCELLED – After a task is successfully canceled, the status of the task is set to <code>CANCELLED</code>.
task_info	<p>Additional information about the task.</p> <p>If an error occurs while backing up or restoring a database, this column contains information about the error. For a list of possible errors, and mitigation strategies, see Troubleshooting (p. 685).</p>
last_updated	The date and time that the task status was last updated. The status is updated after every 5 percent of progress.
created_at	The date and time that the task was created.
S3_object_arn	The ARN indicating the Amazon S3 prefix and the name of the file that is being backed up or restored.
overwrite_s3_backup_file	The value of the <code>@overwrite_s3_backup_file</code> parameter specified when calling a backup task. For more information, see Backing up a database (p. 675) .
KMS_master_key_arn	The ARN for the AWS KMS CMK used for encryption (for backup) and decryption (for restore).
filepath	Not applicable to native backup and restore tasks.
overwrite_file	Not applicable to native backup and restore tasks.

Compressing backup files

To save space in your Amazon S3 bucket, you can compress your backup files. For more information about compressing backup files, see [Backup compression](#) in the Microsoft documentation.

Compressing your backup files is supported for the following database editions:

- Microsoft SQL Server Enterprise Edition
- Microsoft SQL Server Standard Edition

To turn on compression for your backup files, run the following code:

```
exec rdsadmin..rds_set_configuration 'S3 backup compression', 'true';
```

To turn off compression for your backup files, run the following code:

```
exec rdsadmin..rds_set_configuration 'S3 backup compression', 'false';
```

Troubleshooting

The following are issues you might encounter when you use native backup and restore.

Issue	Troubleshooting suggestions
Access Denied	<p>The backup or restore process can't access the backup file. This is usually caused by issues like the following:</p> <ul style="list-style-type: none">• Referencing the incorrect bucket. Referencing the bucket using an incorrect format. Referencing a file name without using the ARN.• Incorrect permissions on the bucket file. For example, if it is created by a different account that is trying to access it now, add the correct permissions.• An IAM policy that is incorrect or incomplete. Your IAM role must include all the necessary elements, including, for example, the correct version. These are highlighted in Importing and exporting SQL Server databases (p. 671).
BACKUP DATABASE WITH COMPRESSION isn't supported on <edition_name> Edition	<p>Compressing your backup files is only supported for Microsoft SQL Server Enterprise Edition and Standard Edition.</p> <p>For more information, see Compressing backup files (p. 685).</p>
Key <ARN> does not exist	<p>You attempted to restore an encrypted backup, but didn't provide a valid encryption key. Check your encryption key and retry.</p> <p>For more information, see Restoring a database (p. 678).</p>
Please reissue task with correct type and overwrite property	<p>If you attempt to back up your database and provide the name of a file that already exists, but set the overwrite property to false, the save operation fails. To fix this error, either provide the name of a file that doesn't already exist, or set the overwrite property to true.</p> <p>For more information, see Backing up a database (p. 675).</p>

Issue	Troubleshooting suggestions
	<p>It's also possible that you intended to restore your database, but called the <code>rds_backup_database</code> stored procedure accidentally. In that case, call the <code>rds_restore_database</code> stored procedure instead.</p> <p>For more information, see Restoring a database (p. 678).</p> <p>If you intended to restore your database and called the <code>rds_restore_database</code> stored procedure, make sure that you provided the name of a valid backup file.</p> <p>For more information, see Using native backup and restore (p. 675).</p>
Please specify a bucket that is in the same region as RDS instance	<p>You can't back up to, or restore from, an Amazon S3 bucket in a different AWS Region from your Amazon RDS DB instance. You can use Amazon S3 replication to copy the backup file to the correct AWS Region.</p> <p>For more information, see Cross-Region replication in the Amazon S3 documentation.</p>
The specified bucket does not exist	<p>Verify that you have provided the correct ARN for your bucket and file, in the correct format.</p> <p>For more information, see Using native backup and restore (p. 675).</p>
User <ARN> is not authorized to perform <kms action> on resource <ARN>	<p>You requested an encrypted operation, but didn't provide correct AWS KMS permissions. Verify that you have the correct permissions, or add them.</p> <p>For more information, see Setting up for native backup and restore (p. 672).</p>
The Restore task is unable to restore from more than 10 backup file(s). Please reduce the number of files matched and try again.	<p>Reduce the number of files that you're trying to restore from. You can make each individual file larger if necessary.</p>

Importing and exporting SQL Server data using other methods

Following, you can find information about using snapshots to import your Microsoft SQL Server data to Amazon RDS. You can also find information about using snapshots to export your data from an RDS DB instance running SQL Server.

If your scenario supports it, it's easier to move data in and out of Amazon RDS by using the native backup and restore functionality. For more information, see [Importing and exporting SQL Server databases \(p. 671\)](#).

Note

Amazon RDS for Microsoft SQL Server does not support importing data into the `msdb` database.

Importing data into SQL Server on Amazon RDS by using a snapshot

To import data into a SQL Server DB instance by using a snapshot

1. Create a DB instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
2. Stop applications from accessing the destination DB instance.

If you prevent access to your DB instance while you are importing data, data transfer is faster. Additionally, you don't need to worry about conflicts while data is being loaded if other applications cannot write to the DB instance at the same time. If something goes wrong and you have to roll back to an earlier database snapshot, the only changes that you lose are the imported data. You can import this data again after you resolve the issue.

For information about controlling access to your DB instance, see [Working with DB security groups \(EC2-Classic platform\) \(p. 1704\)](#).

3. Create a snapshot of the target database.

If the target database is already populated with data, we recommend that you take a snapshot of the database before you import the data. If something goes wrong with the data import or you want to discard the changes, you can restore the database to its previous state by using the snapshot. For information about database snapshots, see [Creating a DB snapshot \(p. 346\)](#).

Note

When you take a database snapshot, I/O operations to the database are suspended for a moment (milliseconds) while the backup is in progress.

4. Disable automated backups on the target database.

Disabling automated backups on the target DB instance improves performance while you are importing your data because Amazon RDS doesn't log transactions when automatic backups are disabled. However, there are some things to consider. Automated backups are required to perform a point-in-time recovery. Thus, you can't restore the database to a specific point in time while you are importing data. Additionally, any automated backups that were created on the DB instance are erased unless you choose to retain them.

Choosing to retain the automated backups can help protect you against accidental deletion of data. Amazon RDS also saves the database instance properties along with each automated backup to make it easy to recover. Using this option lets you can restore a deleted database instance to a specified point in time within the backup retention period even after deleting it. Automated backups are automatically deleted at the end of the specified backup window, just as they are for an active database instance.

You can also use previous snapshots to recover the database, and any snapshots that you have taken remain available. For information about automated backups, see [Working with backups \(p. 328\)](#).

5. Disable foreign key constraints, if applicable.

If you need to disable foreign key constraints, you can do so with the following script.

```
--Disable foreign keys on all tables
DECLARE @table_name SYSNAME;
DECLARE @cmd NVARCHAR(MAX);
DECLARE table_cursor CURSOR FOR SELECT name FROM sys.tables;

OPEN table_cursor;
FETCH NEXT FROM table_cursor INTO @table_name;

WHILE @@FETCH_STATUS = 0 BEGIN
    SELECT @cmd = 'ALTER TABLE '+QUOTENAME(@table_name)+' NOCHECK CONSTRAINT ALL';
    EXEC (@cmd);
    FETCH NEXT FROM table_cursor INTO @table_name;
END

CLOSE table_cursor;
DEALLOCATE table_cursor;

GO
```

6. Drop indexes, if applicable.
7. Disable triggers, if applicable.

If you need to disable triggers, you can do so with the following script.

```
--Disable triggers on all tables
DECLARE @enable BIT = 0;
DECLARE @trigger SYSNAME;
DECLARE @table SYSNAME;
DECLARE @cmd NVARCHAR(MAX);
DECLARE trigger_cursor CURSOR FOR SELECT trigger_object.name trigger_name,
    table_object.name table_name
FROM sysobjects trigger_object
JOIN sysobjects table_object ON trigger_object.parent_obj = table_object.id
WHERE trigger_object.type = 'TR';

OPEN trigger_cursor;
FETCH NEXT FROM trigger_cursor INTO @trigger, @table;

WHILE @@FETCH_STATUS = 0 BEGIN
    IF @enable = 1
        SET @cmd = 'ENABLE ';
    ELSE
        SET @cmd = 'DISABLE ';

    SET @cmd = @cmd + ' TRIGGER dbo.'+QUOTENAME(@trigger)+'' ON
dbo.'+QUOTENAME(@table)+'' ;
    EXEC (@cmd);
    FETCH NEXT FROM trigger_cursor INTO @trigger, @table;
END

CLOSE trigger_cursor;
DEALLOCATE trigger_cursor;

GO
```

8. Query the source SQL Server instance for any logins that you want to import to the destination DB instance.

SQL Server stores logins and passwords in the `master` database. Because Amazon RDS doesn't grant access to the `master` database, you cannot directly import logins and passwords into your destination DB instance. Instead, you must query the `master` database on the source SQL Server instance to generate a data definition language (DDL) file. This file should include all logins and passwords that you want to add to the destination DB instance. This file also should include role memberships and permissions that you want to transfer.

For information about querying the `master` database, see [How to transfer the logins and the passwords between instances of SQL Server 2005 and SQL Server 2008](#) in the Microsoft Knowledge Base.

The output of the script is another script that you can run on the destination DB instance. The script in the Knowledge Base article has the following code:

```
p.type IN
```

Every place `p.type` appears, use the following code instead:

```
p.type = 'S'
```

9. Import the data using the method in [Import the data \(p. 690\)](#).
10. Grant applications access to the target DB instance.

When your data import is complete, you can grant access to the DB instance to those applications that you blocked during the import. For information about controlling access to your DB instance, see [Working with DB security groups \(EC2-Classic platform\) \(p. 1704\)](#).

11. Enable automated backups on the target DB instance.

For information about automated backups, see [Working with backups \(p. 328\)](#).

12. Enable foreign key constraints.

If you disabled foreign key constraints earlier, you can now enable them with the following script.

```
--Enable foreign keys on all tables
DECLARE @table_name SYSNAME;
DECLARE @cmd NVARCHAR(MAX);
DECLARE table_cursor CURSOR FOR SELECT name FROM sys.tables;

OPEN table_cursor;
FETCH NEXT FROM table_cursor INTO @table_name;

WHILE @@FETCH_STATUS = 0 BEGIN
    SELECT @cmd = 'ALTER TABLE '+QUOTENAME(@table_name)+' CHECK CONSTRAINT ALL';
    EXEC (@cmd);
    FETCH NEXT FROM table_cursor INTO @table_name;
END

CLOSE table_cursor;
DEALLOCATE table_cursor;
```

13. Enable indexes, if applicable.
14. Enable triggers, if applicable.

If you disabled triggers earlier, you can now enable them with the following script.

```
--Enable triggers on all tables
DECLARE @enable BIT = 1;
DECLARE @trigger SYSNAME;
DECLARE @table SYSNAME;
DECLARE @cmd NVARCHAR(MAX);
DECLARE trigger_cursor CURSOR FOR SELECT trigger_object.name trigger_name,
    table_object.name table_name
FROM sysobjects trigger_object
JOIN sysobjects table_object ON trigger_object.parent_obj = table_object.id
WHERE trigger_object.type = 'TR';

OPEN trigger_cursor;
FETCH NEXT FROM trigger_cursor INTO @trigger, @table;

WHILE @@FETCH_STATUS = 0 BEGIN
    IF @enable = 1
        SET @cmd = 'ENABLE ';
    ELSE
        SET @cmd = 'DISABLE ';

    SET @cmd = @cmd + ' TRIGGER dbo.'+QUOTENAME(@trigger)+' ON
dbo.'+QUOTENAME(@table)+';
    EXEC (@cmd);
    FETCH NEXT FROM trigger_cursor INTO @trigger, @table;
END

CLOSE trigger_cursor;
DEALLOCATE trigger_cursor;
```

Import the data

Microsoft SQL Server Management Studio is a graphical SQL Server client that is included in all Microsoft SQL Server editions except the Express Edition. SQL Server Management Studio Express is available from Microsoft as a free download. To find this download, see [the Microsoft website](#).

Note

SQL Server Management Studio is available only as a Windows-based application.

SQL Server Management Studio includes the following tools, which are useful in importing data to a SQL Server DB instance:

- Generate and Publish Scripts Wizard
- Import and Export Wizard
- Bulk copy

Generate and Publish Scripts Wizard

The Generate and Publish Scripts Wizard creates a script that contains the schema of a database, the data itself, or both. You can generate a script for a database in your local SQL Server deployment. You can then run the script to transfer the information that it contains to an Amazon RDS DB instance.

Note

For databases of 1 GiB or larger, it's more efficient to script only the database schema. You then use the Import and Export Wizard or the bulk copy feature of SQL Server to transfer the data.

For detailed information about the Generate and Publish Scripts Wizard, see the [Microsoft SQL Server documentation](#).

In the wizard, pay particular attention to the advanced options on the **Set Scripting Options** page to ensure that everything you want your script to include is selected. For example, by default, database triggers are not included in the script.

When the script is generated and saved, you can use SQL Server Management Studio to connect to your DB instance and then run the script.

Import and Export Wizard

The Import and Export Wizard creates a special Integration Services package, which you can use to copy data from your local SQL Server database to the destination DB instance. The wizard can filter which tables and even which tuples within a table are copied to the destination DB instance.

Note

The Import and Export Wizard works well for large datasets, but it might not be the fastest way to remotely export data from your local deployment. For an even faster way, consider the SQL Server bulk copy feature.

For detailed information about the Import and Export Wizard, see the [Microsoft SQL Server documentation](#).

In the wizard, on the **Choose a Destination** page, do the following:

- For **Server Name**, type the name of the endpoint for your DB instance.
- For the server authentication mode, choose **Use SQL Server Authentication**.
- For **User name** and **Password**, type the credentials for the master user that you created for the DB instance.

Bulk copy

The SQL Server bulk copy feature is an efficient means of copying data from a source database to your DB instance. Bulk copy writes the data that you specify to a data file, such as an ASCII file. You can then run bulk copy again to write the contents of the file to the destination DB instance.

This section uses the **bcp** utility, which is included with all editions of SQL Server. For detailed information about bulk import and export operations, see [the Microsoft SQL Server documentation](#).

Note

Before you use bulk copy, you must first import your database schema to the destination DB instance. The Generate and Publish Scripts Wizard, described earlier in this topic, is an excellent tool for this purpose.

The following command connects to the local SQL Server instance. It generates a tab-delimited file of a specified table in the C:\ root directory of your existing SQL Server deployment. The table is specified by its fully qualified name, and the text file has the same name as the table that is being copied.

```
bcp dbname.schema_name.table_name out C:\table_name.txt -n -S localhost -U username -P password -b 10000
```

The preceding code includes the following options:

- **-n** specifies that the bulk copy uses the native data types of the data to be copied.
- **-S** specifies the SQL Server instance that the *bcp* utility connects to.
- **-U** specifies the user name of the account to log in to the SQL Server instance.
- **-P** specifies the password for the user specified by **-U**.

- **-b** specifies the number of rows per batch of imported data.

Note

There might be other parameters that are important to your import situation. For example, you might need the **-E** parameter that pertains to identity values. For more information; see the full description of the command line syntax for the **bcp** utility in the [Microsoft SQL Server documentation](#).

For example, suppose that a database named **store** that uses the default schema, **dbo**, contains a table named **customers**. The user account **admin**, with the password **insecure**, copies 10,000 rows of the **customers** table to a file named **customers.txt**.

```
bcp store.dbo.customers out C:\customers.txt -n -S localhost -U admin -P insecure -b 10000
```

After you generate the data file, you can upload the data to your DB instance by using a similar command. Beforehand, create the database and schema on the target DB instance. Then use the **in** argument to specify an input file instead of **out** to specify an output file. Instead of using **localhost** to specify the local SQL Server instance, specify the endpoint of your DB instance. If you use a port other than 1433, specify that too. The user name and password are the master user and password for your DB instance. The syntax is as follows.

```
bcp dbname.schema_name.table_name in C:\table_name.txt -n -S endpoint,port -  
U master_user_name -P master_user_password -b 10000
```

To continue the previous example, suppose that the master user name is **admin**, and the password is **insecure**. The endpoint for the DB instance is **rds.ckz2kqd4qsn1.us-east-1.rds.amazonaws.com**, and you use port 4080. The command is as follows.

```
bcp store.dbo.customers in C:\customers.txt -n -S rds.ckz2kqd4qsn1.us-east-1.rds.amazonaws.com,4080 -U admin -P insecure -b 10000
```

Exporting data from SQL Server on Amazon RDS

You can choose one of the following options to export data from an RDS for SQL Server DB instance:

- **Native database backup using a full backup file (.bak)** – Using .bak files to backup databases is heavily optimized, and is usually the fastest way to export data. For more information, see [Importing and exporting SQL Server databases \(p. 671\)](#).
- **SQL Server Import and Export Wizard** – For more information, see [SQL Server Import and Export Wizard \(p. 692\)](#).
- **SQL Server Generate and Publish Scripts Wizard and bcp utility** – For more information, see [SQL Server Generate and Publish Scripts Wizard and bcp utility \(p. 694\)](#).

SQL Server Import and Export Wizard

You can use the SQL Server Import and Export Wizard to copy one or more tables, views, or queries from your RDS for SQL Server DB instance to another data store. This choice is best if the target data store is not SQL Server. For more information, see [SQL Server Import and Export Wizard](#) in the SQL Server documentation.

The SQL Server Import and Export Wizard is available as part of Microsoft SQL Server Management Studio. This graphical SQL Server client is included in all Microsoft SQL Server editions except the Express Edition. SQL Server Management Studio is available only as a Windows-based application.

SQL Server Management Studio Express is available from Microsoft as a free download. To find this download, see [the Microsoft website](#).

To use the SQL Server Import and Export Wizard to export data

1. In SQL Server Management Studio, connect to your RDS for SQL Server DB instance. For details on how to do this, see [Connecting to a DB instance running the Microsoft SQL Server database engine \(p. 656\)](#).
2. In **Object Explorer**, expand **Databases**, open the context (right-click) menu for the source database, choose **Tasks**, and then choose **Export Data**. The wizard appears.
3. On the **Choose a Data Source** page, do the following:
 - a. For **Data source**, choose **SQL Server Native Client 11.0**.
 - b. Verify that the **Server name** box shows the endpoint of your RDS for SQL Server DB instance.
 - c. Select **Use SQL Server Authentication**. For **User name** and **Password**, type the master user name and password of your DB instance.
 - d. Verify that the **Database** box shows the database from which you want to export data.
 - e. Choose **Next**.
4. On the **Choose a Destination** page, do the following:
 - a. For **Destination**, choose **SQL Server Native Client 11.0**.

Note

Other target data sources are available. These include .NET Framework data providers, OLE DB providers, SQL Server Native Client providers, ADO.NET providers, Microsoft Office Excel, Microsoft Office Access, and the Flat File source. If you choose to target one of these data sources, skip the remainder of step 4. For details on the connection information to provide next, see [Choose a destination](#) in the SQL Server documentation.

- b. For **Server name**, type the server name of the target SQL Server DB instance.
- c. Choose the appropriate authentication type. Type a user name and password if necessary.
- d. For **Database**, choose the name of the target database, or choose **New** to create a new database to contain the exported data.

If you choose **New**, see [Create database](#) in the SQL Server documentation for details on the database information to provide.

- e. Choose **Next**.
5. On the **Table Copy or Query** page, choose **Copy data from one or more tables or views** or **Write a query to specify the data to transfer**. Choose **Next**.
6. If you chose **Write a query to specify the data to transfer**, you see the **Provide a Source Query** page. Type or paste in a SQL query, and then choose **Parse** to verify it. Once the query validates, choose **Next**.
7. On the **Select Source Tables and Views** page, do the following:
 - a. Select the tables and views that you want to export, or verify that the query you provided is selected.
 - b. Choose **Edit Mappings** and specify database and column mapping information. For more information, see [Column mappings](#) in the SQL Server documentation.
 - c. (Optional) To see a preview of data to be exported, select the table, view, or query, and then choose **Preview**.
 - d. Choose **Next**.
8. On the **Run Package** page, verify that **Run immediately** is selected. Choose **Next**.
9. On the **Complete the Wizard** page, verify that the data export details are as you expect. Choose **Finish**.

10. On the **The execution was successful** page, choose **Close**.

SQL Server Generate and Publish Scripts Wizard and bcp utility

You can use the SQL Server Generate and Publish Scripts Wizard to create scripts for an entire database or just selected objects. You can run these scripts on a target SQL Server DB instance to recreate the scripted objects. You can then use the bcp utility to bulk export the data for the selected objects to the target DB instance. This choice is best if you want to move a whole database (including objects other than tables) or large quantities of data between two SQL Server DB instances. For a full description of the bcp command-line syntax, see [bcp utility](#) in the Microsoft SQL Server documentation.

The SQL Server Generate and Publish Scripts Wizard is available as part of Microsoft SQL Server Management Studio. This graphical SQL Server client is included in all Microsoft SQL Server editions except the Express Edition. SQL Server Management Studio is available only as a Windows-based application. SQL Server Management Studio Express is available from Microsoft as a [free download](#).

To use the SQL Server Generate and Publish Scripts Wizard and the bcp utility to export data

1. In SQL Server Management Studio, connect to your RDS for SQL Server DB instance. For details on how to do this, see [Connecting to a DB instance running the Microsoft SQL Server database engine \(p. 656\)](#).
2. In **Object Explorer**, expand the **Databases** node and select the database you want to script.
3. Follow the instructions in [Generate and publish scripts Wizard](#) in the SQL Server documentation to create a script file.
4. In SQL Server Management Studio, connect to your target SQL Server DB instance.
5. With the target SQL Server DB instance selected in **Object Explorer**, choose **Open** on the **File** menu, choose **File**, and then open the script file.
6. If you have scripted the entire database, review the CREATE DATABASE statement in the script. Make sure that the database is being created in the location and with the parameters that you want. For more information, see [CREATE DATABASE](#) in the SQL Server documentation.
7. If you are creating database users in the script, check to see if server logins exist on the target DB instance for those users. If not, create logins for those users; the scripted commands to create the database users fail otherwise. For more information, see [Create a login](#) in the SQL Server documentation.
8. Choose **!Execute** on the SQL Editor menu to run the script file and create the database objects. When the script finishes, verify that all database objects exist as expected.
9. Use the bcp utility to export data from the RDS for SQL Server DB instance into files. Open a command prompt and type the following command.

```
bcp database_name.schema_name.table_name out data_file -n -S aws_rds_sql_endpoint -U
username -P password
```

The preceding code includes the following options:

- *table_name* is the name of one of the tables that you've recreated in the target database and now want to populate with data.
- *data_file* is the full path and name of the data file to be created.
- *-n* specifies that the bulk copy uses the native data types of the data to be copied.
- *-S* specifies the SQL Server DB instance to export from.
- *-U* specifies the user name to use when connecting to the SQL Server DB instance.
- *-P* specifies the password for the user specified by *-U*.

The following shows an example command.

```
bcp world.dbo.city out C:\Users\JohnDoe\city.dat -n -S sql-jdoe.1234abcd.us-west-2.rds.amazonaws.com,1433 -U JohnDoe -P ClearTextPassword
```

Repeat this step until you have data files for all of the tables you want to export.

10. Prepare your target DB instance for bulk import of data by following the instructions at [Basic guidelines for bulk importing data](#) in the SQL Server documentation.
11. Decide on a bulk import method to use after considering performance and other concerns discussed in [About bulk import and bulk export operations](#) in the SQL Server documentation.
12. Bulk import the data from the data files that you created using the bcp utility. To do so, follow the instructions at either [Import and export bulk data by using the bcp utility](#) or [Import bulk data by using BULK INSERT or OPENROWSET\(BULK...\)](#) in the SQL Server documentation, depending on what you decided in step 11.

Working with read replicas for Microsoft SQL Server in Amazon RDS

You usually use read replicas to configure replication between Amazon RDS DB instances. For general information about read replicas, see [Working with read replicas \(p. 278\)](#).

In this section, you can find specific information about working with read replicas on Amazon RDS for SQL Server.

Topics

- [Configuring read replicas for SQL Server \(p. 696\)](#)
- [Read replica limitations with SQL Server \(p. 696\)](#)
- [Troubleshooting a SQL Server read replica problem \(p. 697\)](#)

Configuring read replicas for SQL Server

Before a DB instance can serve as a source instance for replication, you must enable automatic backups on the source DB instance. To do so, you set the backup retention period to a value other than 0. The source DB instance must be a Multi-AZ deployment with Always On Availability Groups (AGs). Setting this type of deployment also enforces that automatic backups are enabled.

Creating a SQL Server read replica doesn't require an outage for the primary DB instance. Amazon RDS sets the necessary parameters and permissions for the source DB instance and the read replica without any service interruption. A snapshot is taken of the source DB instance, and this snapshot becomes the read replica. No outage occurs when you delete a read replica.

You can create up to five read replicas from one source DB instance. For replication to operate effectively, each read replica should have the same amount of compute and storage resources as the source DB instance. If you scale the source DB instance, also scale the read replicas.

The SQL Server DB engine version of the source DB instance and all of its read replicas must be the same. Amazon RDS upgrades the primary immediately after upgrading the read replicas, regardless of the maintenance window. For more information about upgrading the DB engine version, see [Upgrading the Microsoft SQL Server DB engine \(p. 666\)](#).

For a read replica to receive and apply changes from the source, it should have sufficient compute and storage resources. If a read replica reaches compute, network, or storage resource capacity, the read replica stops receiving or applying changes from its source. You can modify the storage and CPU resources of a read replica independently from its source and other read replicas.

Read replica limitations with SQL Server

The following limitations apply to SQL Server read replicas on Amazon RDS:

- Read replicas are only available on the SQL Server Enterprise Edition (EE) engine.
- Read replicas are available for SQL Server versions 2016–2019.
- The source DB instance to be replicated must be a Multi-AZ deployment with Always On AGs.
- Read replicas are only available for DB instances on the EC2-VPC platform.
- Read replicas are only available for DB instances running on DB instance classes with four or more vCPUs.
- The following aren't supported on Amazon RDS for SQL Server:
 - Creating a read replica in a different AWS Region (a cross-Region read replica)

- Backup retention of read replicas
- Point-in-time recovery from read replicas
- Manual snapshots of read replicas
- Multi-AZ read replicas
- Creating read replicas of read replicas
- Synchronization of user logins to read replicas
- Amazon RDS for SQL Server doesn't intervene to mitigate high replica lag between a source DB instance and its read replicas. Make sure that the source DB instance and its read replicas are sized properly, in terms of computing power and storage, to suit their operational load.
- A SQL Server read replica belongs to the same option group as the source DB instance. Modifications to the source option group or source option group membership propagate to read replicas. These changes are applied to the read replicas immediately after they are applied to the source DB instance, regardless of the read replica's maintenance window.

For more information about option groups, see [Working with option groups \(p. 212\)](#).

Troubleshooting a SQL Server read replica problem

You can monitor replication lag in Amazon CloudWatch by viewing the Amazon RDS `ReplicaLag` metric. For information about replication lag time, see [Monitoring read replication \(p. 288\)](#).

If replication lag is too long, you can use the following query to get information about the lag.

```
SELECT AR.replica_server_name
      , DB_NAME(ARS.database_id) 'database_name'
      , AR.availability_mode_desc
      , ARS.synchronization_health_desc
      , ARS.last_hardened_lsn
      , ARS.last_redone_lsn
      , ARS.secondary_lag_seconds
  FROM sys.dm_hadr_database_replica_states ARS
 INNER JOIN sys.availability_replicas AR ON ARS.replica_id = AR.replica_id
--WHERE DB_NAME(ARS.database_id) = 'database_name'
 ORDER BY AR.replica_server_name;
```

Multi-AZ deployments for Microsoft SQL Server

Multi-AZ deployments provide increased availability, data durability, and fault tolerance for DB instances. In the event of planned database maintenance or unplanned service disruption, Amazon RDS automatically fails over to the up-to-date secondary DB instance. This functionality lets database operations resume quickly without manual intervention. The primary and standby instances use the same endpoint, whose physical network address transitions to the secondary replica as part of the failover process. You don't have to reconfigure your application when a failover occurs.

Amazon RDS supports Multi-AZ deployments for Microsoft SQL Server by using either SQL Server Database Mirroring (DBM) or Always On Availability Groups (AGs). Amazon RDS monitors and maintains the health of your Multi-AZ deployment. If problems occur, RDS automatically repairs unhealthy DB instances, reestablishes synchronization, and initiates failovers. Failover only occurs if the standby and primary are fully in sync. You don't have to manage anything.

When you set up SQL Server Multi-AZ, RDS automatically configures all databases on the instance to use DBM or AGs. Amazon RDS handles the primary, the witness, and the secondary DB instance for you. Because configuration is automatic, RDS selects DBM or Always On AGs based on the version of SQL Server that you deploy.

Amazon RDS supports Multi-AZ with Always On AGs for the following SQL Server versions and editions:

- SQL Server 2019: Standard and Enterprise Editions
- SQL Server 2017: Enterprise Edition 14.00.3049.1 or later
- SQL Server 2016: Enterprise Edition 13.00.5216.0 or later

Amazon RDS supports Multi-AZ with DBM for the following SQL Server versions and editions, except for the versions noted previously:

- SQL Server 2017: Standard and Enterprise Editions
- SQL Server 2016: Standard and Enterprise Editions
- SQL Server 2014: Standard and Enterprise Editions
- SQL Server 2012: Standard and Enterprise Editions

Amazon RDS supports Multi-AZ for SQL Server in all AWS Regions, with the following exceptions:

- Asia Pacific (Osaka): Neither DBM nor Always On AGs are supported here.
- Asia Pacific (Sydney): Supported for [DB instances in VPCs](#).
- Asia Pacific (Tokyo): Supported for [DB instances in VPCs](#).
- South America (São Paulo): Supported on all [DB instance classes](#) except m1 and m2.

You can use the following SQL query to determine whether your SQL Server DB instance is Single-AZ, Multi-AZ with DBM, or Multi-AZ with Always On AGs:

```
SELECT CASE WHEN dm.mirroring_state_desc IS NOT NULL THEN 'Multi-AZ (Mirroring)'
            WHEN dhdrs.group_database_id IS NOT NULL THEN 'Multi-AZ (AlwaysOn)'
            ELSE 'Single-AZ'
        END 'high_availability'
FROM sys.databases sd
LEFT JOIN sys.database_mirroring dm ON sd.database_id = dm.database_id
LEFT JOIN sys.dm_hadr_database_replica_states dhdrs ON sd.database_id = dhdrs.database_id
AND dhdrs.is_local = 1
WHERE DB_NAME(sd.database_id) = 'rdsadmin';
```

The output resembles the following:

```
high_availability  
Multi-AZ (Mirroring)
```

Adding Multi-AZ to a Microsoft SQL Server DB instance

When you create a new SQL Server DB instance using the AWS Management Console, you can add Multi-AZ with Database Mirroring (DBM) or Always On AGs. You do so by choosing **Yes (Mirroring / Always On)** from **Multi-AZ deployment**. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

When you modify an existing SQL Server DB instance using the AWS Management Console, you can add Multi-AZ with DBM or AGs by choosing **Yes (Mirroring / Always On)** from **Multi-AZ deployment** on the **Modify DB instance** page. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Note

If your DB instance is running Database Mirroring (DBM)—not Always On Availability Groups (AGs)—you might need to disable in-memory optimization before you add Multi-AZ. Disable in-memory optimization with DBM before you add Multi-AZ if your DB instance runs SQL Server 2014, 2016, or 2017 Enterprise Edition and has in-memory optimization enabled.

If your DB instance is running AGs, it doesn't require this step.

Microsoft SQL Server Multi-AZ deployment notes and recommendations

The following are some restrictions when working with Multi-AZ deployments for Microsoft SQL Server DB instances:

- Cross-Region Multi-AZ isn't supported.
- You can't configure the secondary DB instance to accept database read activity.
- Multi-AZ with Always On Availability Groups (AGs) supports in-memory optimization.
- Multi-AZ with Always On Availability Groups (AGs) doesn't support Kerberos authentication for the availability group listener. This is because the listener has no Service Principal Name (SPN).
- You can't rename a database on a SQL Server DB instance that is in a SQL Server Multi-AZ deployment. If you need to rename a database on such an instance, first turn off Multi-AZ for the DB instance, then rename the database. Finally, turn Multi-AZ back on for the DB instance.
- You can only restore Multi-AZ DB instances that are backed up using the full recovery model.

The following are some notes about working with Multi-AZ deployments for Microsoft SQL Server DB instances:

- Amazon RDS exposes the Always On AGs **availability group listener endpoint**. The endpoint is visible in the console, and is returned by the `DescribeDBInstances` API as an entry in the endpoints field.
- Amazon RDS supports **availability group multisubnet failovers**.
- To use SQL Server Multi-AZ with a SQL Server DB instance in a VPC, first create a DB subnet group that has subnets in at least two distinct Availability Zones. Then assign the DB subnet group to the primary replica of the SQL Server DB instance.
- When a DB instance is modified to be a Multi-AZ deployment, during the modification it has a status of **modifying**. Amazon RDS creates the standby, and makes a backup of the primary DB instance. After the process is complete, the status of the primary DB instance becomes **available**.

- Multi-AZ deployments maintain all databases on the same node. If a database on the primary host fails over, all your SQL Server databases fail over as one atomic unit to your standby host. Amazon RDS provisions a new healthy host, and replaces the unhealthy host.
- Multi-AZ with DBM or AGs supports a single standby replica.
- Users, logins, and permissions are automatically replicated for you on the secondary. You don't need to recreate them. User-defined server roles (a SQL Server 2012 feature) are only replicated in Multi-AZ instances for AGs instances.
- If you have SQL Server Agent jobs, recreate them on the secondary. You do so because these jobs are stored in the msdb database, and you can't replicate this database by using Database Mirroring (DBM) or Always On Availability Groups (AGs). Create the jobs first in the original primary, then fail over, and create the same jobs in the new primary.
- You might observe elevated latencies compared to a standard DB instance deployment (in a single Availability Zone) because of the synchronous data replication.
- Failover times are affected by the time it takes to complete the recovery process. Large transactions increase the failover time.

The following are some recommendations for working with Multi-AZ deployments for Microsoft SQL Server DB instances:

- For databases used in production or preproduction, we recommend the following options:
 - Multi-AZ deployments for high availability
 - "Provisioned IOPS" for fast, consistent performance
 - "Memory optimized" rather than "General purpose"
- You can't select the Availability Zone (AZ) for the secondary instance, so when you deploy application hosts, take this into account. Your database might fail over to another AZ, and the application hosts might not be in the same AZ as the database. For this reason, we recommend that you balance your application hosts across all AZs in the given AWS Region.
- For best performance, don't enable Database Mirroring or Always On AGs during a large data load operation. If you want your data load to be as fast as possible, finish loading data before you convert your DB instance to a Multi-AZ deployment.
- Applications that access the SQL Server databases should have exception handling that catches connection errors. The following code sample shows a try/catch block that catches a communication error. In this example, the break statement exits the while loop if the connection is successful, but retries up to 10 times if an exception is thrown.

```
int RetryMaxAttempts = 10;
int RetryIntervalPeriodInSeconds = 1;
int iRetryCount = 0;
while (iRetryCount < RetryMaxAttempts)
{
    using (SqlConnection connection = new SqlConnection(DatabaseConnectionString))
    {
        using (SqlCommand command = connection.CreateCommand())
        {
            command.CommandText = "INSERT INTO SOME_TABLE VALUES ('SomeValue');";
            try
            {
                connection.Open();
                command.ExecuteNonQuery();
                break;
            }
            catch (Exception ex)
            {
                Logger(ex.Message);
                iRetryCount++;
            }
        }
    }
}
```

```

        finally {
            connection.Close();
        }
    }
    Thread.Sleep(RetryIntervalPeriodInSeconds * 1000);
}

```

- Don't use the `Set Partner Off` command when working with Multi-AZ instances. For example, don't do the following.

```
--Don't do this
ALTER DATABASE db1 SET PARTNER off
```

- Don't set the recovery mode to `simple`. For example, don't do the following.

```
--Don't do this
ALTER DATABASE db1 SET RECOVERY simple
```

- Don't use the `DEFAULT_DATABASE` parameter when creating new logins on Multi-AZ DB instances, because these settings can't be applied to the standby mirror. For example, don't do the following.

```
--Don't do this
CREATE LOGIN [test_dba] WITH PASSWORD=foo, DEFAULT_DATABASE=[db2]
```

Also, don't do the following.

```
--Don't do this
ALTER LOGIN [test_dba] SET DEFAULT_DATABASE=[db3]
```

Determining the location of the secondary

You can determine the location of the secondary replica by using the AWS Management Console. You need to know the location of the secondary if you are setting up your primary DB instance in a VPC.

Instance		
Configuration	Instance class	Storage
DB instance id database-1	Instance class db.m4.large	Encryption Enabled
Engine version 14.00.3192.2.v1	vCPU 2	KMS key aws/rds
DB name -	RAM 8 GB	Storage type General Purpose (SSD)
License model License Included	Availability	IOPS -
Collation SQL_Latin1_General_CI_AS	Master username admin	Storage 20 GiB
Option groups default:sqlserver-se-14-00	IAM db authentication Not Enabled	Storage autoscaling Enabled
ARN arn:aws:rds:us-west-2: db:database-1	Multi AZ Yes (Mirroring)	Maximum storage threshold 1000 GiB
Resource id db-	Secondary Zone us-west-2c	

You can also view the Availability Zone of the secondary using the AWS CLI command `describe-db-instances` or RDS API operation `DescribeDBInstances`. The output shows the secondary AZ where the standby mirror is located.

Migrating from Database Mirroring to Always On availability groups

In version 14.00.3049.1 of Microsoft SQL Server Enterprise edition, Always On availability groups (AGs) is enabled by default.

To migrate from Database Mirroring (DBM) to AGs, first check your version. If you are using a DB instance with a version prior to Enterprise Edition 13.00.5216.0, modify the instance to patch it to 13.00.5216.0 or later. If you are using a DB instance with a version prior to Enterprise Edition 14.00.3049.1, modify the instance to patch it to 14.00.3049.1 or later.

If you want to upgrade a mirrored DB instance to use AGs, run the upgrade first, modify the instance to remove Multi-AZ, and then modify it again to add Multi-AZ. This converts your instance to use Always On AGs.

Additional features for Microsoft SQL Server on Amazon RDS

In the following sections, you can find information about augmenting Amazon RDS instances running the Microsoft SQL Server DB engine.

Topics

- [Using SSL with a Microsoft SQL Server DB instance \(p. 704\)](#)
- [Configuring security protocols and ciphers \(p. 707\)](#)
- [Using Windows Authentication with an Amazon RDS for SQL Server DB instance \(p. 711\)](#)
- [Integrating an Amazon RDS for SQL Server DB instance with Amazon S3 \(p. 721\)](#)
- [Using Database Mail on Amazon RDS for SQL Server \(p. 734\)](#)
- [Instance store support for the tempdb database on Amazon RDS for SQL Server \(p. 744\)](#)
- [Using extended events with Amazon RDS for Microsoft SQL Server \(p. 746\)](#)

Using SSL with a Microsoft SQL Server DB instance

You can use Secure Sockets Layer (SSL) to encrypt connections between your client applications and your Amazon RDS DB instances running Microsoft SQL Server. SSL support is available in all AWS regions for all supported SQL Server editions.

When you create a SQL Server DB instance, Amazon RDS creates an SSL certificate for it. The SSL certificate includes the DB instance endpoint as the Common Name (CN) for the SSL certificate to guard against spoofing attacks.

There are 2 ways to use SSL to connect to your SQL Server DB instance:

- Force SSL for all connections — this happens transparently to the client, and the client doesn't have to do any work to use SSL.
- Encrypt specific connections — this sets up an SSL connection from a specific client computer, and you must do work on the client to encrypt connections.

For information about Transport Layer Security (TLS) support for SQL Server, see [TLS 1.2 support for Microsoft SQL Server](#).

Forcing connections to your DB instance to use SSL

You can force all connections to your DB instance to use SSL. If you force connections to use SSL, it happens transparently to the client, and the client doesn't have to do any work to use SSL.

If you want to force SSL, use the `rds.force_ssl` parameter. By default, the `rds.force_ssl` parameter is set to 0 (off). Set the `rds.force_ssl` parameter to 1 (on) to force connections to use SSL. The `rds.force_ssl` parameter is static, so after you change the value, you must reboot your DB instance for the change to take effect.

To force all connections to your DB instance to use SSL

1. Determine the parameter group that is attached to your DB instance:
 - a. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
 - b. In the top right corner of the Amazon RDS console, choose the AWS Region of your DB instance.
 - c. In the navigation pane, choose **Databases**, and then choose the name of your DB instance to show its details.
 - d. Choose the **Configuration** tab. Find the **Parameter group** in the section.
2. If necessary, create a new parameter group. If your DB instance uses the default parameter group, you must create a new parameter group. If your DB instance uses a nondefault parameter group, you can choose to edit the existing parameter group or to create a new parameter group. If you edit an existing parameter group, the change affects all DB instances that use that parameter group.

To create a new parameter group, follow the instructions in [Creating a DB parameter group \(p. 229\)](#).

3. Edit your new or existing parameter group to set the `rds.force_ssl` parameter to `true`. To edit the parameter group, follow the instructions in [Modifying parameters in a DB parameter group \(p. 232\)](#).
4. If you created a new parameter group, modify your DB instance to attach the new parameter group. Modify the **DB Parameter Group** setting of the DB instance. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
5. Reboot your DB instance. For more information, see [Rebooting a DB instance \(p. 276\)](#).

Encrypting specific connections

You can force all connections to your DB instance to use SSL, or you can encrypt connections from specific client computers only. To use SSL from a specific client, you must obtain certificates for the client computer, import certificates on the client computer, and then encrypt the connections from the client computer.

Note

All SQL Server instances created after August 5, 2014, use the DB instance endpoint in the Common Name (CN) field of the SSL certificate. Prior to August 5, 2014, SSL certificate verification was not available for VPC-based SQL Server instances. If you have a VPC-based SQL Server DB instance that was created before August 5, 2014, and you want to use SSL certificate verification and ensure that the instance endpoint is included as the CN for the SSL certificate for that DB instance, then rename the instance. When you rename a DB instance, a new certificate is deployed and the instance is rebooted to enable the new certificate.

Obtaining certificates for client computers

To encrypt connections from a client computer to an Amazon RDS DB instance running Microsoft SQL Server, you need a certificate on your client computer.

To obtain that certificate, download the certificate to your client computer. You can download a root certificate that works for all regions. You can also download a certificate bundle that contains both the old and new root certificate. In addition, you can download region-specific intermediate certificates. For more information about downloading certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

After you have downloaded the appropriate certificate, import the certificate into your Microsoft Windows operating system by following the procedure in the section following.

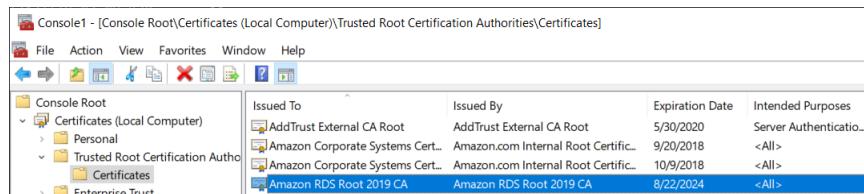
Importing certificates on client computers

You can use the following procedure to import your certificate into the Microsoft Windows operating system on your client computer.

To import the certificate into your Windows operating system:

1. On the **Start** menu, type **Run** in the search box and press **Enter**.
2. In the **Open** box, type **MMC** and then choose **OK**.
3. In the MMC console, on the **File** menu, choose **Add/Remove Snap-in**.
4. In the **Add or Remove Snap-ins** dialog box, for **Available snap-ins**, select **Certificates**, and then choose **Add**.
5. In the **Certificates snap-in** dialog box, choose **Computer account**, and then choose **Next**.
6. In the **Select computer** dialog box, choose **Finish**.
7. In the **Add or Remove Snap-ins** dialog box, choose **OK**.
8. In the MMC console, expand **Certificates**, open the context (right-click) menu for **Trusted Root Certification Authorities**, choose **All Tasks**, and then choose **Import**.
9. On the first page of the Certificate Import Wizard, choose **Next**.
10. On the second page of the Certificate Import Wizard, choose **Browse**. In the browse window, change the file type to **All files (*.*)** because .pem is not a standard certificate extension. Locate the .pem file that you downloaded previously.
11. Choose **Open** to select the certificate file, and then choose **Next**.
12. On the third page of the Certificate Import Wizard, choose **Next**.
13. On the fourth page of the Certificate Import Wizard, choose **Finish**. A dialog box appears indicating that the import was successful.

14. In the MMC console, expand **Certificates**, expand **Trusted Root Certification Authorities**, and then choose **Certificates**. Locate the certificate to confirm it exists, as shown here.



15. Restart your computer.

Encrypting connections to an Amazon RDS DB instance running Microsoft SQL Server

After you have imported a certificate into your client computer, you can encrypt connections from the client computer to an Amazon RDS DB instance running Microsoft SQL Server.

For SQL Server Management Studio, use the following procedure. For more information about SQL Server Management Studio, see [Use SQL Server management studio](#).

To encrypt connections from SQL Server Management Studio

1. Launch SQL Server Management Studio.
2. For **Connect to server**, type the server information, login user name, and password.
3. Choose **Options**.
4. Select **Encrypt connection**.
5. Choose **Connect**.
6. Confirm that your connection is encrypted by running the following query. Verify that the query returns `true` for `encrypt_option`.

```
select ENCRYPT_OPTION from SYS.DM_EXEC_CONNECTIONS where SESSION_ID = @@SPID
```

For any other SQL client, use the following procedure.

To encrypt connections from other SQL clients

1. Append `encrypt=true` to your connection string. This string might be available as an option, or as a property on the connection page in GUI tools.

Note

To enable SSL encryption for clients that connect using JDBC, you might need to add the Amazon RDS SQL certificate to the Java CA certificate (`cacerts`) store. You can do this by using the `keytool` utility.

2. Confirm that your connection is encrypted by running the following query. Verify that the query returns `true` for `encrypt_option`.

```
select ENCRYPT_OPTION from SYS.DM_EXEC_CONNECTIONS where SESSION_ID = @@SPID
```

Configuring security protocols and ciphers

You can turn certain security protocols and ciphers on and off using DB parameters. The security parameters that you can configure (except for TLS version 1.2) are shown in the following table.

A value of **default** means that the operating system default value is used, whether it is enabled or disabled.

Note

You can't disable TLS 1.2, because Amazon RDS uses it internally.

DB parameter	Allowed values (default in bold)	Description
rds.tls10	default , enabled, disabled	TLS 1.0.
rds.tls11	default , enabled, disabled	TLS 1.1.
rds.tls12	default	TLS 1.2. You can't modify this value.
rds.rc4	default , enabled, disabled	RC4 stream cipher.
rds.diffie-hellman	default , enabled, disabled	Diffie-Hellman key-exchange encryption.
rds.diffie-hellman-min-key-bit-length	default , 1024, 2048, 4096	Minimum bit length for Diffie-Hellman keys.
rds.curve25519	default , enabled, disabled	Curve25519 elliptic-curve encryption cipher. This parameter isn't supported for all engine versions.
rds.3des168	default , enabled, disabled	Triple Data Encryption Standard (DES) encryption cipher with a 168-bit key length.

Note

For more information on the default values for SQL Server security protocols and ciphers, see [Protocols in TLS/SSL \(Schannel SSP\)](#) and [Cipher Suites in TLS/SSL \(Schannel SSP\)](#) in the Microsoft documentation.

Use the following process to configure the security protocols and ciphers:

1. Create a custom DB parameter group.
2. Modify the parameters in the parameter group.
3. Associate the DB parameter group with your DB instance.

For more information on DB parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

Creating the security-related parameter group

Create a parameter group for your security-related parameters that corresponds to the SQL Server edition and version of your DB instance.

Console

The following procedure creates a parameter group for SQL Server Standard Edition 2016.

To create the parameter group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. Choose **Create parameter group**.
4. In the **Create parameter group** pane, do the following:
 - a. For **Parameter group family**, choose **sqlserver-se-13.0**.
 - b. For **Group name**, enter an identifier for the parameter group, such as **sqlserver-ciphers-se-13**.
 - c. For **Description**, enter **Parameter group for security protocols and ciphers**.
5. Choose **Create**.

CLI

The following procedure creates a parameter group for SQL Server Standard Edition 2016.

To create the parameter group

- Run one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name sqlserver-ciphers-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "Parameter group for security protocols and ciphers"
```

For Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name sqlserver-ciphers-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "Parameter group for security protocols and ciphers"
```

Modifying security-related parameters

Modify the security-related parameters in the parameter group that corresponds to the SQL Server edition and version of your DB instance.

Console

The following procedure modifies the parameter group that you created for SQL Server Standard Edition 2016. This example turns off TLS version 1.0.

To modify the parameter group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.

2. In the navigation pane, choose **Parameter groups**.
3. Choose the parameter group, such as **sqlserver-ciphers-se-13**.
4. Under **Parameters**, filter the parameter list for **rds**.
5. Choose **Edit parameters**.
6. Choose **rds.tls10**.
7. For **Values**, choose **disabled**.
8. Choose **Save changes**.

CLI

The following procedure modifies the parameter group that you created for SQL Server Standard Edition 2016. This example turns off TLS version 1.0.

To modify the parameter group

- Run one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name sqlserver-ciphers-se-13 \
    --parameters
    "ParameterName='rds.tls10',ParameterValue='disabled',ApplyMethod=pending-reboot"
```

For Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name sqlserver-ciphers-se-13 ^
    --parameters
    "ParameterName='rds.tls10',ParameterValue='disabled',ApplyMethod=pending-reboot"
```

Associating the security-related parameter group with your DB instance

To associate the parameter group with your DB instance, use the AWS Management Console or the AWS CLI.

Console

You can associate the parameter group with a new or existing DB instance:

- For a new DB instance, associate it when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- For an existing DB instance, associate it by modifying the instance. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

CLI

You can associate the parameter group with a new or existing DB instance.

To create a DB instance with the parameter group

- Specify the same DB engine type and major version as you used when creating the parameter group.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --master-user-password secret123 \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --db-parameter-group-name sqlserver-ciphers-se-13
```

For Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --master-user-password secret123 ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --db-parameter-group-name sqlserver-ciphers-se-13
```

To modify a DB instance and associate the parameter group

- Run one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --db-parameter-group-name sqlserver-ciphers-se-13 \
    --apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-parameter-group-name sqlserver-ciphers-se-13 ^
    --apply-immediately
```

Using Windows Authentication with an Amazon RDS for SQL Server DB instance

You can use Microsoft Windows Authentication to authenticate users when they connect to your Amazon RDS for Microsoft SQL Server DB instance. The DB instance works with AWS Directory Service for Microsoft Active Directory, also called AWS Managed Microsoft AD, to enable Windows Authentication. When users authenticate with a SQL Server DB instance joined to the trusting domain, authentication requests are forwarded to the domain directory that you create with AWS Directory Service.

Amazon RDS supports Windows Authentication for SQL Server in all AWS Regions.

Amazon RDS uses mixed mode for Windows Authentication. This approach means that the *master user* (the name and password used to create your SQL Server DB instance) uses SQL Authentication. Because the master user account is a privileged credential, you should restrict access to this account.

To get Windows Authentication using an on-premises or self-hosted Microsoft Active Directory, create a forest trust. The trust can be one-way or two-way. For more information on setting up forest trusts using AWS Directory Service, see [When to create a trust relationship](#) in the *AWS Directory Service Administration Guide*.

To set up Windows authentication for a SQL Server DB instance, do the following steps, explained in greater detail in [Setting up Windows Authentication for SQL Server DB instances \(p. 712\)](#):

1. Use AWS Managed Microsoft AD, either from the AWS Management Console or AWS Directory Service API, to create an AWS Managed Microsoft AD directory.
2. If you use the AWS CLI or Amazon RDS API to create your SQL Server DB instance, create an AWS Identity and Access Management (IAM) role. This role uses the managed IAM policy `AmazonRDSDirectoryServiceAccess` and allows Amazon RDS to make calls to your directory. If you use the console to create your SQL Server DB instance, AWS creates the IAM role for you.
- For the role to allow access, the AWS Security Token Service (AWS STS) endpoint must be activated in the AWS Region for your AWS account. AWS STS endpoints are active by default in all AWS Regions, and you can use them without any further actions. For more information, see [Managing AWS STS in an AWS Region](#) in the *IAM User Guide*.
3. Create and configure users and groups in the AWS Managed Microsoft AD directory using the Microsoft Active Directory tools. For more information about creating users and groups in your Active Directory, see [Manage users and groups in AWS Managed Microsoft AD](#) in the *AWS Directory Service Administration Guide*.
4. If you plan to locate the directory and the DB instance in different VPCs, enable cross-VPC traffic.
5. Use Amazon RDS to create a new SQL Server DB instance either from the console, AWS CLI, or Amazon RDS API. In the create request, you provide the domain identifier ("d-*" identifier) that was generated when you created your directory and the name of the role you created. You can also modify an existing SQL Server DB instance to use Windows Authentication by setting the domain and IAM role parameters for the DB instance.
6. Use the Amazon RDS master user credentials to connect to the SQL Server DB instance as you do any other DB instance. Because the DB instance is joined to the AWS Managed Microsoft AD domain, you can provision SQL Server logins and users from the Active Directory users and groups in their domain. (These are known as SQL Server "Windows" logins.) Database permissions are managed through standard SQL Server permissions granted and revoked to these Windows logins.

Creating the endpoint for Kerberos authentication

Kerberos-based authentication requires that the endpoint be the customer-specified host name, a period, and then the fully qualified domain name (FQDN). For example, the following is an example of an

endpoint you might use with Kerberos-based authentication. In this example, the SQL Server DB instance host name is ad-test and the domain name is corp-ad.company.com.

```
ad-test.corp-ad.company.com
```

If you want to make sure your connection is using Kerberos, run the following query:

```
SELECT net_transport, auth_scheme
  FROM sys.dm_exec_connections
 WHERE session_id = @@SPID;
```

Setting up Windows Authentication for SQL Server DB instances

You use AWS Directory Service for Microsoft Active Directory, also called AWS Managed Microsoft AD, to set up Windows Authentication for a SQL Server DB instance. To set up Windows Authentication, take the following steps.

Step 1: Create a directory using the AWS Directory Service for Microsoft Active Directory

AWS Directory Service creates a fully managed, Microsoft Active Directory in the AWS Cloud. When you create an AWS Managed Microsoft AD directory, AWS Directory Service creates two domain controllers and Domain Name Service (DNS) servers on your behalf. The directory servers are created in two subnets in two different Availability Zones within a VPC. This redundancy helps ensure that your directory remains accessible even if a failure occurs.

When you create an AWS Managed Microsoft AD directory, AWS Directory Service performs the following tasks on your behalf:

- Sets up a Microsoft Active Directory within the VPC.
- Creates a directory administrator account with the user name Admin and the specified password. You use this account to manage your directory.

Note

Be sure to save this password. AWS Directory Service doesn't store this password, and you can't retrieve or reset it.

- Creates a security group for the directory controllers.

When you launch an AWS Directory Service for Microsoft Active Directory, AWS creates an Organizational Unit (OU) that contains all your directory's objects. This OU, which has the NetBIOS name that you typed when you created your directory, is located in the domain root. The domain root is owned and managed by AWS.

The *admin* account that was created with your AWS Managed Microsoft AD directory has permissions for the most common administrative activities for your OU:

- Create, update, or delete users, groups, and computers.
- Add resources to your domain such as file or print servers, and then assign permissions for those resources to users and groups in your OU.
- Create additional OUs and containers.
- Delegate authority.
- Create and link group policies.
- Restore deleted objects from the Active Directory Recycle Bin.
- Run AD and DNS Windows PowerShell modules on the Active Directory Web Service.

The admin account also has rights to perform the following domain-wide activities:

- Manage DNS configurations (add, remove, or update records, zones, and forwarders).
- View DNS event logs.
- View security event logs.

To create a directory with AWS Managed Microsoft AD

1. In the [AWS Directory Service console](#) navigation pane, choose **Directories** and choose **Set up directory**.
2. Choose **AWS Managed Microsoft AD**. This is the only option currently supported for use with Amazon RDS.
3. Choose **Next**.
4. On the **Enter directory information** page, provide the following information:

Edition

Choose the edition that meets your requirements.

Directory DNS name

The fully qualified name for the directory, such as `corp.example.com`. Names longer than 47 characters aren't supported by SQL Server.

Directory NetBIOS name

An optional short name for the directory, such as `CORP`.

Directory description

An optional description for the directory.

Admin password

The password for the directory administrator. The directory creation process creates an administrator account with the user name Admin and this password.

The directory administrator password can't include the word `admin`. The password is case-sensitive and must be 8–64 characters in length. It must also contain at least one character from three of the following four categories:

- Lowercase letters (a-z)
- Uppercase letters (A-Z)
- Numbers (0-9)
- Non-alphanumeric characters (~!@#\$%^&*_+=`|\{};:"<,>,?./)

Confirm password

Retype the administrator password.

5. Choose **Next**.
6. On the **Choose VPC and subnets** page, provide the following information:

VPC

Choose the VPC for the directory.

Note

You can locate the directory and the DB instance in different VPCs, but if you do so, make sure to enable cross-VPC traffic. For more information, see [Step 4: Enable cross-VPC traffic between the directory and the DB instance \(p. 716\)](#).

Subnets

Choose the subnets for the directory servers. The two subnets must be in different Availability Zones.

7. Choose **Next**.
8. Review the directory information. If changes are needed, choose **Previous**. When the information is correct, choose **Create directory**.

Review & create

Review	
Directory type	VPC
Microsoft AD	vpc-8b6b78e9 ([REDACTED])
Directory DNS name	Subnets
corp.example.com	subnet-75128d10 ([REDACTED], us-east-1a) subnet-f51665dd ([REDACTED], us-east-1b)
Directory NetBIOS name	
CORP	
Directory description	
My directory	

Pricing	
Edition	Free trial eligible Learn more 30-day limited trial
Standard	
~USD [REDACTED] *	
* Includes two domain controllers, USD [REDACTED] /mo for each additional domain controller.	

Cancel [Previous](#) **Create directory**

It takes several minutes for the directory to be created. When it has been successfully created, the **Status** value changes to **Active**.

To see information about your directory, choose the directory ID in the directory listing. Make a note of the **Directory ID**. You need this value when you create or modify your SQL Server DB instance.

The screenshot shows the 'Directory details' page for a Microsoft AD Standard edition. The 'Directory ID' field, which contains the value 'd-90670a8d36', is highlighted with a red oval. Other visible details include the VPC ('vpc-6594f31c'), Subnets ('subnet-7d36a227', 'subnet-a2ab49c6'), Status ('Active'), Last updated ('Tuesday, January 7, 2020'), Launch time ('Tuesday, January 7, 2020'), Availability zones ('us-east-1c, us-east-1d'), and DNS address ('[REDACTED]'). Below the main table, there are tabs for 'Application management' (which is selected), 'Scale & share', 'Networking & security', and 'Maintenance'.

Step 2: Create the IAM role for use by Amazon RDS

If you use the console to create your SQL Server DB instance, you can skip this step. If you use the CLI or RDS API to create your SQL Server DB instance, you must create an IAM role that uses the `AmazonRDSDirectoryServiceAccess` managed IAM policy. This role allows Amazon RDS to make calls to the AWS Directory Service for you.

If you are using a custom policy for joining a domain, rather than using the AWS-managed `AmazonRDSDirectoryServiceAccess` policy, make sure that you allow the `ds:GetAuthorizedApplicationDetails` action. This requirement is effective starting July 2019, due to a change in the AWS Directory Service API.

The following IAM policy, `AmazonRDSDirectoryServiceAccess`, provides access to AWS Directory Service.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "ds:DescribeDirectories",  
                "ds:AuthorizeApplication",  
                "ds:UnauthorizeApplication",  
                "ds:GetAuthorizedApplicationDetails"  
            ],  
            "Effect": "Allow",  
            "Resource": "*"  
        }  
    ]  
}
```

```
    ]  
}
```

Create an IAM role using this policy. For more information about creating IAM roles, see [Creating customer managed policies](#) in the *IAM User Guide*.

Step 3: Create and configure users and groups

You can create users and groups with the Active Directory Users and Computers tool. This tool is one of the Active Directory Domain Services and Active Directory Lightweight Directory Services tools. Users represent individual people or entities that have access to your directory. Groups are very useful for giving or denying privileges to groups of users, rather than having to apply those privileges to each individual user.

To create users and groups in an AWS Directory Service directory, you must be connected to a Windows EC2 instance that is a member of the AWS Directory Service directory. You must also be logged in as a user that has privileges to create users and groups. For more information, see [Add users and groups \(Simple AD and AWS Managed Microsoft AD\)](#) in the *AWS Directory Service Administration Guide*.

Step 4: Enable cross-VPC traffic between the directory and the DB instance

If you plan to locate the directory and the DB instance in the same VPC, skip this step and move on to [Step 5: Create or modify a SQL Server DB instance \(p. 716\)](#).

If you plan to locate the directory and the DB instance in different VPCs, configure cross-VPC traffic using VPC peering or [AWS Transit Gateway](#).

The following procedure enables traffic between VPCs using VPC peering. Follow the instructions in [What is VPC peering?](#) in the *Amazon Virtual Private Cloud Peering Guide*.

To enable cross-VPC traffic using VPC peering

1. Set up appropriate VPC routing rules to ensure that network traffic can flow both ways.
2. Ensure that the DB instance's security group can receive inbound traffic from the directory's security group.
3. Ensure that there is no network access control list (ACL) rule to block traffic.

If a different AWS account owns the directory, you must share the directory.

To share the directory between AWS accounts

1. Start sharing the directory with the AWS account that the DB instance will be created in by following the instructions in [Tutorial: Sharing your AWS Managed Microsoft AD directory for seamless EC2 domain-join](#) in the *AWS Directory Service Administration Guide*.
2. Sign in to the AWS Directory Service console using the account for the DB instance, and ensure that the domain has the SHARED status before proceeding.
3. While signed into the AWS Directory Service console using the account for the DB instance, note the **Directory ID** value. You use this directory ID to join the DB instance to the domain.

Step 5: Create or modify a SQL Server DB instance

Create or modify a SQL Server DB instance for use with your directory. You can use the console, CLI, or RDS API to associate a DB instance with a directory. You can do this in one of the following ways:

- Create a new SQL Server DB instance using the console, the [create-db-instance](#) CLI command, or the [CreateDBInstance](#) RDS API operation.

For instructions, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- Modify an existing SQL Server DB instance using the console, the [modify-db-instance](#) CLI command, or the [ModifyDBInstance](#) RDS API operation.

For instructions, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
- Restore a SQL Server DB instance from a DB snapshot using the console, the [restore-db-instance-from-db-snapshot](#) CLI command, or the [RestoreDBInstanceFromDBSnapshot](#) RDS API operation.

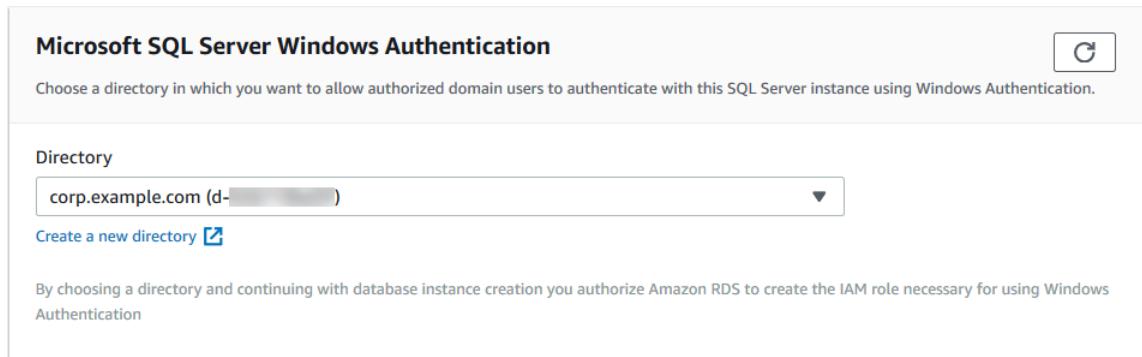
For instructions, see [Restoring from a DB snapshot \(p. 349\)](#).
- Restore a SQL Server DB instance to a point-in-time using the console, the [restore-db-instance-to-point-in-time](#) CLI command, or the [RestoreDBInstanceToPointInTime](#) RDS API operation.

For instructions, see [Restoring a DB instance to a specified time \(p. 389\)](#).

Windows Authentication is only supported for SQL Server DB instances in a VPC.

For the DB instance to be able to use the domain directory that you created, the following is required:

- For **Directory**, you must choose the domain identifier (`d-ID`) generated when you created the directory.
- Make sure that the VPC security group has an outbound rule that lets the DB instance communicate with the directory.



When you use the AWS CLI, the following parameters are required for the DB instance to be able to use the directory that you created:

- For the `--domain` parameter, use the domain identifier (`d-ID`) generated when you created the directory.
- For the `--domain-iam-role-name` parameter, use the role that you created that uses the managed IAM policy `AmazonRDSDirectoryServiceAccess`.

For example, the following CLI command modifies a DB instance to use a directory.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--domain d-ID \
--domain-iam-role-name role-name
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--domain d-ID ^
--domain-iam-role-name role-name
```

Important

If you modify a DB instance to enable Kerberos authentication, reboot the DB instance after making the change.

Step 6: Create Windows Authentication SQL Server logins

Use the Amazon RDS master user credentials to connect to the SQL Server DB instance as you do any other DB instance. Because the DB instance is joined to the AWS Managed Microsoft AD domain, you can provision SQL Server logins and users. You do this from the Active Directory users and groups in your domain. Database permissions are managed through standard SQL Server permissions granted and revoked to these Windows logins.

For an Active Directory user to authenticate with SQL Server, a SQL Server Windows login must exist for the user or a group that the user is a member of. Fine-grained access control is handled through granting and revoking permissions on these SQL Server logins. A user that doesn't have a SQL Server login or belong to a group with such a login can't access the SQL Server DB instance.

The ALTER ANY LOGIN permission is required to create an Active Directory SQL Server login. If you haven't created any logins with this permission, connect as the DB instance's master user using SQL Server Authentication.

Run a data definition language (DDL) command such as the following example to create a SQL Server login for an Active Directory user or group.

Note

Specify users and groups using the pre-Windows 2000 login name in the format *domainName\login_name*. You can't use a user principal name (UPN) in the format *login_name@DomainName*.

```
USE [master]
GO
CREATE LOGIN [mydomain\myuser] FROM WINDOWS WITH DEFAULT_DATABASE = [master],
DEFAULT_LANGUAGE = [us_english];
GO
```

For more information, see [CREATE LOGIN \(Transact-SQL\)](#) in the Microsoft Developer Network documentation.

Users (both humans and applications) from your domain can now connect to the RDS for SQL Server instance from a domain-joined client machine using Windows authentication.

Managing a DB instance in a Domain

You can use the console, AWS CLI, or the Amazon RDS API to manage your DB instance and its relationship with your domain. For example, you can move the DB instance into, out of, or between domains.

For example, using the Amazon RDS API, you can do the following:

- To reattempt a domain join for a failed membership, use the [ModifyDBInstance](#) API operation and specify the current membership's directory ID.
- To update the IAM role name for membership, use the [ModifyDBInstance](#) API operation and specify the current membership's directory ID and the new IAM role.
- To remove a DB instance from a domain, use the [ModifyDBInstance](#) API operation and specify `none` as the domain parameter.
- To move a DB instance from one domain to another, use the [ModifyDBInstance](#) API operation and specify the domain identifier of the new domain as the domain parameter.
- To list membership for each DB instance, use the [DescribeDBInstances](#) API operation.

Understanding Domain membership

After you create or modify your DB instance, the instance becomes a member of the domain. The AWS console indicates the status of the domain membership for the DB instance. The status of the DB instance can be one of the following:

- **joined** – The instance is a member of the domain.
- **joining** – The instance is in the process of becoming a member of the domain.
- **pending-join** – The instance membership is pending.
- **pending-maintenance-join** – AWS will attempt to make the instance a member of the domain during the next scheduled maintenance window.
- **pending-removal** – The removal of the instance from the domain is pending.
- **pending-maintenance-removal** – AWS will attempt to remove the instance from the domain during the next scheduled maintenance window.
- **failed** – A configuration problem has prevented the instance from joining the domain. Check and fix your configuration before reissuing the instance modify command.
- **removing** – The instance is being removed from the domain.

A request to become a member of a domain can fail because of a network connectivity issue or an incorrect IAM role. For example, you might create a DB instance or modify an existing instance and have the attempt fail for the DB instance to become a member of a domain. In this case, either reissue the command to create or modify the DB instance or modify the newly created instance to join the domain.

Connecting to SQL Server with Windows authentication

To connect to SQL Server with Windows Authentication, you must be logged into a domain-joined computer as a domain user. After launching SQL Server Management Studio, choose **Windows Authentication** as the authentication type, as shown following.



Restoring a SQL Server DB instance and then adding it to a domain

You can restore a DB snapshot or do point-in-time recovery (PITR) for a SQL Server DB instance and then add it to a domain. Once the DB instance is restored, modify the instance using the process explained in [Step 5: Create or modify a SQL Server DB instance \(p. 716\)](#) to add the DB instance to a domain.

Integrating an Amazon RDS for SQL Server DB instance with Amazon S3

You can transfer files between a DB instance running Amazon RDS for SQL Server and an Amazon S3 bucket. By doing this, you can use Amazon S3 with SQL Server features such as BULK INSERT. For example, you can download .csv, .xml, .txt, and other files from Amazon S3 to the DB instance host and import the data from D:\S3\ into the database. All files are stored in D:\S3\ on the DB instance.

The following limitations apply:

- Files in the D:\S3 folder are deleted on the standby replica after a failover on Multi-AZ instances. For more information, see [Multi-AZ limitations for S3 integration \(p. 731\)](#).
- The DB instance and the S3 bucket must be in the same AWS Region.
- If you run more than one S3 integration task at a time, the tasks run sequentially, not in parallel.

Note

S3 integration tasks share the same queue as native backup and restore tasks. At maximum, you can have only two tasks in progress at any time in this queue. Therefore, two running native backup and restore tasks will block any S3 integration tasks.

- You must re-enable the S3 integration feature on restored instances. S3 integration isn't propagated from the source instance to the restored instance. Files in D:\S3 are deleted on a restored instance.
- Downloading to the DB instance is limited to 100 files. In other words, there can't be more than 100 files in D:\S3\.
- Only files without file extensions or with the following file extensions are supported for download: .abf, .asdatabase, .bcp, .configsettings, .csv, .dat, .deploymentoptions, .deploymenttargets, .fmt, .info, .isp and .xmla.
- The S3 bucket must have the same owner as the related AWS Identity and Access Management (IAM) role. Therefore, cross-account S3 integration isn't supported.
- The S3 bucket can't be open to the public.
- The file size for uploads from RDS to S3 is limited to 50 GB per file.
- The file size for downloads from S3 to RDS is limited to the maximum supported by S3.

Topics

- [Prerequisites for integrating RDS for SQL Server with S3 \(p. 721\)](#)
- [Enabling RDS for SQL Server integration with S3 \(p. 726\)](#)
- [Transferring files between RDS for SQL Server and Amazon S3 \(p. 727\)](#)
- [Listing files on the RDS DB instance \(p. 728\)](#)
- [Deleting files on the RDS DB instance \(p. 729\)](#)
- [Monitoring the status of a file transfer task \(p. 730\)](#)
- [Canceling a task \(p. 731\)](#)
- [Multi-AZ limitations for S3 integration \(p. 731\)](#)
- [Disabling RDS for SQL Server integration with S3 \(p. 732\)](#)

For more information on working with files in Amazon S3, see [Getting started with Amazon Simple Storage Service](#).

Prerequisites for integrating RDS for SQL Server with S3

Before you begin, find or create the S3 bucket that you want to use. Also, add permissions so that the RDS DB instance can access the S3 bucket. To configure this access, you create both an IAM policy and an IAM role.

Console

To create an IAM policy for access to Amazon S3

1. In the [IAM Management Console](#), choose **Policies** in the navigation pane.
2. Create a new policy, and use the **Visual editor** tab for the following steps.
3. For **Service**, enter **s3** and then choose the **S3** service.
4. For **Actions**, choose the following to grant the access that your DB instance requires:
 - `ListAllMyBuckets` – required
 - `ListBucket` – required
 - `GetBucketACL` – required
 - `GetBucketLocation` – required
 - `GetObject` – required for downloading files from S3 to `D:\S3\`
 - `PutObject` – required for uploading files from `D:\S3\` to S3
 - `ListMultipartUploadParts` – required for uploading files from `D:\S3\` to S3
 - `AbortMultipartUpload` – required for uploading files from `D:\S3\` to S3
5. For **Resources**, the options that display depend on which actions you choose in the previous step. You might see options for **bucket**, **object**, or both. For each of these, add the appropriate Amazon Resource Name (ARN).

For **bucket**, add the ARN for the bucket that you want to use. For example, if your bucket is named `example-bucket`, set the ARN to `arn:aws:s3:::example-bucket`.

For **object**, enter the ARN for the bucket and then choose one of the following:

- To grant access to all files in the specified bucket, choose **Any** for both **Bucket name** and **Object name**.
 - To grant access to specific files or folders in the bucket, provide ARNs for the specific buckets and objects that you want SQL Server to access.
6. Follow the instructions in the console until you finish creating the policy.

The preceding is an abbreviated guide to setting up a policy. For more detailed instructions on creating IAM policies, see [Creating IAM policies](#) in the *IAM User Guide*.

To create an IAM role that uses the IAM policy from the previous procedure

1. In the [IAM Management Console](#), choose **Roles** in the navigation pane.
2. Create a new IAM role, and choose the following options as they appear in the console:
 - **AWS service**
 - **RDS**
 - **RDS – Add Role to Database**

Then choose **Next:Permissions** at the bottom.

3. For **Attach permissions policies**, enter the name of the IAM policy that you previously created. Then choose the policy from the list.
4. Follow the instructions in the console until you finish creating the role.

The preceding is an abbreviated guide to setting up a role. If you want more detailed instructions on creating roles, see [IAM roles](#) in the *IAM User Guide*.

AWS CLI

To grant Amazon RDS access to an Amazon S3 bucket

1. Create an IAM policy that grants Amazon RDS access to an S3 bucket.

Include the appropriate actions to grant the access your DB instance requires:

- `ListAllMyBuckets` – required
- `ListBucket` – required
- `GetBucketACL` – required
- `GetBucketLocation` – required
- `GetObject` – required for downloading files from S3 to D:\S3\
- `PutObject` – required for uploading files from D:\S3\ to S3
- `ListMultipartUploadParts` – required for uploading files from D:\S3\ to S3
- `AbortMultipartUpload` – required for uploading files from D:\S3\ to S3

The following AWS CLI command creates an IAM policy named `rds-s3-integration-policy` with these options. It grants access to a bucket named `your-s3-bucket-arn`.

Example

For Linux, macOS, or Unix:

```
aws iam create-policy \
--policy-name rds-s3-integration-policy \
--policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3>ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3>ListBucket",
                "s3>GetBucketACL",
                "s3>GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::bucket_name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3>GetObject",
                "s3>PutObject",
                "s3>ListMultipartUploadParts",
                "s3>AbortMultipartUpload"
            ],
            "Resource": "arn:aws:s3:::bucket_name/key_prefix/*"
        }
    ]
}'
```

For Windows:

Make sure to change the line endings to the ones supported by your interface (^ instead of \). Also, in Windows, you must escape all double quotes with a \. To avoid the need to escape the quotes in the JSON, you can save it to a file instead and pass that in as a parameter.

First, create the `policy.json` file with the following permission policy:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListAllMyBuckets",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListBucket",  
                "s3:GetBucketACL",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": "arn:aws:s3:::bucket_name"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3GetObject",  
                "s3PutObject",  
                "s3>ListMultipartUploadParts",  
                "s3AbortMultipartUpload"  
            ],  
            "Resource": "arn:aws:s3:::bucket_name/key_prefix/*"  
        }  
    ]  
}
```

Then use the following command to create the policy:

```
aws iam create-policy ^  
    --policy-name rds-s3-integration-policy ^  
    --policy-document file:///policy_file_path
```

2. After the policy is created, note the Amazon Resource Name (ARN) of the policy. You need the ARN for a later step.
3. Create an IAM role that Amazon RDS can assume on your behalf to access your S3 buckets.

The following AWS CLI command creates the `rds-s3-integration-role` for this purpose.

Example

For Linux, macOS, or Unix:

```
aws iam create-role \  
    --role-name rds-s3-integration-role \  
    --assume-role-policy-document '{  
        "Version": "2012-10-17",  
        "Statement": [  
            {  
                "Effect": "Allow",  
                "Principal": {
```

```
        "Service": "rds.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
]
}'
```

For Windows:

Make sure to change the line endings to the ones supported by your interface (^ instead of \). Also, in Windows, you must escape all double quotes with a \. To avoid the need to escape the quotes in the JSON, you can save it to a file instead and pass that in as a parameter.

First, create the `assume_role_policy.json` file with the following policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "rds.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Then use the following command to create the IAM role:

```
aws iam create-role ^
--role-name rds-s3-integration-role ^
--assume-role-policy-document file://assume_role_policy_file_path
```

For more information, see [Creating a role to delegate permissions to an IAM user](#) in the *IAM User Guide*.

4. After the IAM role is created, note the ARN of the role. You need the ARN for a later step.
5. Attach the IAM policy that you created to the IAM role that you created.

The following AWS CLI command attaches the policy to the role named `rds-s3-integration-role`.

Example

For Linux, macOS, or Unix:

```
aws iam attach-role-policy \
--policy-arn your-policy-arn \
--role-name rds-s3-integration-role
```

For Windows:

```
aws iam attach-role-policy ^
--policy-arn your-policy-arn ^
--role-name rds-s3-integration-role
```

Replace *your-policy-arn* with the policy ARN that you noted in a previous step.

Enabling RDS for SQL Server integration with S3

In the following section, you can find how to enable Amazon S3 integration with Amazon RDS for SQL Server. To work with S3 integration, your DB instance must be associated with the IAM role that you previously created before you use the `S3_INTEGRATION` feature-name parameter.

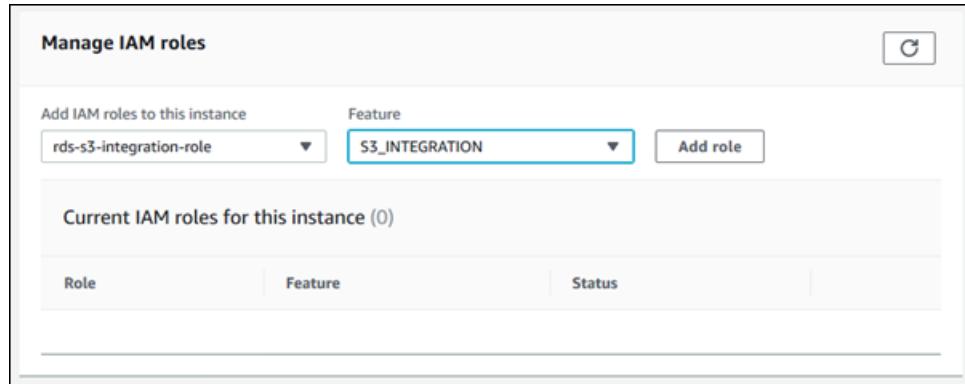
Note

To add an IAM role to a DB instance, the status of the DB instance must be **available**.

Console

To associate your IAM role with your DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose the RDS for SQL Server DB instance name to display its details.
3. On the **Connectivity & security** tab, in the **Manage IAM roles** section, choose the IAM role to add for **Add IAM roles to this instance**.
4. For **Feature**, choose `S3_INTEGRATION`.



5. Choose **Add role**.

AWS CLI

To add the IAM role to the RDS for SQL Server DB instance

- The following AWS CLI command adds your IAM role to an RDS for SQL Server DB instance named *mydbinstance*.

Example

For Linux, macOS, or Unix:

```
aws rds add-role-to-db-instance \
    --db-instance-identifier mydbinstance \
    --feature-name S3_INTEGRATION \
    --role-arn your-role-arn
```

For Windows:

```
aws rds add-role-to-db-instance ^
--db-instance-identifier mydbinstance ^
--feature-name S3_INTEGRATION ^
--role-arn your-role-arn
```

Replace *your-role-arn* with the role ARN that you noted in a previous step. S3_INTEGRATION must be specified for the --feature-name option.

Transferring files between RDS for SQL Server and Amazon S3

You can use Amazon RDS stored procedures to download and upload files between Amazon S3 and your RDS DB instance. You can also use Amazon RDS stored procedures to list and delete files on the RDS instance.

The files that you download from and upload to S3 are stored in the D:\S3 folder. This is the only folder that you can use to access your files. You can organize your files into subfolders, which are created for you when you include the destination folder during download.

Some of the stored procedures require that you provide an Amazon Resource Name (ARN) to your S3 bucket and file. The format for your ARN is arn:aws:s3:::bucket_name/file_name. Amazon S3 doesn't require an account number or AWS Region in ARNs.

S3 integration tasks run sequentially and share the same queue as native backup and restore tasks. At maximum, you can have only two tasks in progress at any time in this queue. It can take up to five minutes for the task to begin processing.

Downloading files from an Amazon S3 bucket to a SQL Server DB instance

To download files from an S3 bucket to an RDS for SQL Server DB instance, use the Amazon RDS stored procedure msdb.dbo.rds_download_from_s3 with the following parameters.

Parameter name	Data type	Default	Required	Description
@s3_arn_of_file	NVARCHAR	–	Required	The S3 ARN of the file to download, for example: arn:aws:s3:::bucket_name/mydata.csv
@rds_file_path	NVARCHAR	–	Optional	The file path for the RDS instance. If not specified, the file path is D:\S3\<filename in s3>. RDS supports absolute paths and relative paths. If you want to create a subfolder, include it in the file path.
@overwrite_file	INT	0	Optional	Overwrite the existing file: 0 = Don't overwrite 1 = Overwrite

You can download files without a file extension and files with the following file extensions: .bcp, .csv, .dat, .fmt, .info, .lst, .tbl, .txt, and .xml.

Note

Files with the .ispac file extension are supported for download when SQL Server Integration Services is enabled. For more information on enabling SSIS, see [SQL Server Integration Services \(p. 773\)](#).

Files with the following file extensions are supported for download when SQL Server Analysis Services is enabled: .abf, .asdatabase, .configsettings, .deploymentoptions, .deploymenttargets, and .xmla. For more information on enabling SSAS, see [SQL Server Analysis Services \(p. 762\)](#).

The following example shows the stored procedure to download files from S3.

```
exec msdb.dbo.rds_download_from_s3
    @s3_arn_of_file='arn:aws:s3:::bucket_name/bulk_data.csv',
    @rds_file_path='D:\S3\seed_data\data.csv',
    @overwrite_file=1;
```

The example `rds_download_from_s3` operation creates a folder named `seed_data` in `D:\S3\`, if the folder doesn't exist yet. Then the example downloads the source file `bulk_data.csv` from S3 to a new file named `data.csv` on the DB instance. If the file previously existed, it's overwritten because the `@overwrite_file` parameter is set to 1.

Uploading files from a SQL Server DB instance to an Amazon S3 bucket

To upload files from an RDS for SQL Server DB instance to an S3 bucket, use the Amazon RDS stored procedure `msdb.dbo.rds_upload_to_s3` with the following parameters.

Parameter name	Data type	Default	Required	Description
<code>@s3_arn_of_file</code>	NVARCHAR	–	Required	The S3 ARN of the file to be created in S3, for example: <code>arn:aws:s3:::bucket_name/mydata.csv</code>
<code>@rds_file_path</code>	NVARCHAR	–	Required	The file path of the file to upload to S3. Absolute and relative paths are supported.
<code>@overwrite_file</code>	INT	–	Optional	Overwrite the existing file: 0 = Don't overwrite 1 = Overwrite

The following example uploads the file named `data.csv` from the specified location in `D:\S3\seed_data\` to a file `new_data.csv` in the S3 bucket specified by the ARN.

```
exec msdb.dbo.rds_upload_to_s3
    @rds_file_path='D:\S3\seed_data\data.csv',
    @s3_arn_of_file='arn:aws:s3:::bucket_name/new_data.csv',
    @overwrite_file=1;
```

If the file previously existed in S3, it's overwritten because the `@overwrite_file` parameter is set to 1.

Listing files on the RDS DB instance

To list the files available on the DB instance, use both a stored procedure and a function. First, run the following stored procedure to gather file details from the files in `D:\S3\`.

```
exec msdb.dbo.rds_gather_file_details;
```

The stored procedure returns the ID of the task. Like other tasks, this stored procedure runs asynchronously. As soon as the status of the task is **SUCCESS**, you can use the task ID in the `rds_fn_list_file_details` function to list the existing files and directories in D:\S3\, as shown following.

```
SELECT * FROM msdb.dbo.rds_fn_list_file_details(TASK_ID);
```

The `rds_fn_list_file_details` function returns a table with the following columns.

Output parameter	Description
<code>filepath</code>	Absolute path of the file (for example, D:\S3\mydata.csv)
<code>size_in_bytes</code>	File size (in bytes)
<code>last_modified_utc</code>	Last modification date and time in UTC format
<code>is_directory</code>	Option that indicates whether the item is a directory (true/false)

Deleting files on the RDS DB instance

To delete the files available on the DB instance, use the Amazon RDS stored procedure `msdb.dbo.rds_delete_from_filesystem` with the following parameters.

Parameter name	Data type	Default	Required	Description
<code>@rds_file_path</code>	NVARCHAR	–	Required	The file path of the file to delete. Absolute and relative paths are supported.
<code>@force_delete</code>	INT	0	Optional	To delete a directory, this flag must be included and set to 1. 1 = delete a directory This parameter is ignored if you are deleting a file.

To delete a directory, the `@rds_file_path` must end with a backslash (\) and `@force_delete` must be set to 1.

The following example deletes the file D:\S3\delete_me.txt.

```
exec msdb.dbo.rds_delete_from_filesystem
@rds_file_path='D:\S3\delete_me.txt';
```

The following example deletes the directory D:\S3\example_folder\.

```
exec msdb.dbo.rds_delete_from_filesystem
@rds_file_path='D:\S3\example_folder\' ,
@force_delete=1;
```

Monitoring the status of a file transfer task

To track the status of your S3 integration task, call the `rds_fn_task_status` function. It takes two parameters. The first parameter should always be `NULL` because it doesn't apply to S3 integration. The second parameter accepts a task ID.

To see a list of all tasks, set the first parameter to `NULL` and the second parameter to 0, as shown in the following example.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

To get a specific task, set the first parameter to `NULL` and the second parameter to the task ID, as shown in the following example.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

The `rds_fn_task_status` function returns the following information.

Output parameter	Description
<code>task_id</code>	The ID of the task.
<code>task_type</code>	For S3 integration, tasks can have the following task types: <ul style="list-style-type: none">• <code>DOWNLOAD_FROM_S3</code>• <code>UPLOAD_TO_S3</code>• <code>LIST_FILES_ON_DISK</code>• <code>DELETE_FILES_ON_DISK</code>
<code>database_name</code>	Not applicable to S3 integration tasks.
<code>% complete</code>	The progress of the task as a percentage.
<code>duration(mins)</code>	The amount of time spent on the task, in minutes.
<code>lifecycle</code>	The status of the task. Possible statuses are the following: <ul style="list-style-type: none">• <code>CREATED</code> – After you call one of the S3 integration stored procedures, a task is created and the status is set to <code>CREATED</code>.• <code>IN_PROGRESS</code> – After a task starts, the status is set to <code>IN_PROGRESS</code>. It can take up to five minutes for the status to change from <code>CREATED</code> to <code>IN_PROGRESS</code>.• <code>SUCCESS</code> – After a task completes, the status is set to <code>SUCCESS</code>.• <code>ERROR</code> – If a task fails, the status is set to <code>ERROR</code>. For more information about the error, see the <code>task_info</code> column.

Output parameter	Description
	<ul style="list-style-type: none"> CANCEL_REQUESTED – After you call <code>rds_cancel_task</code>, the status of the task is set to CANCEL_REQUESTED. CANCELLED – After a task is successfully canceled, the status of the task is set to CANCELLED.
<code>task_info</code>	Additional information about the task. If an error occurs during processing, this column contains information about the error.
<code>last_updated</code>	The date and time that the task status was last updated.
<code>created_at</code>	The date and time that the task was created.
<code>S3_object_arn</code>	The ARN of the S3 object downloaded from or uploaded to.
<code>overwrite_S3_backup_file</code>	Not applicable to S3 integration tasks.
<code>KMS_master_key_arn</code>	Not applicable to S3 integration tasks.
<code>filepath</code>	The file path on the RDS DB instance.
<code>overwrite_file</code>	An option that indicates if an existing file is overwritten.
<code>task_metadata</code>	Not applicable to S3 integration tasks.

Canceling a task

To cancel S3 integration tasks, use the `msdb.dbo.rds_cancel_task` stored procedure with the `task_id` parameter. Delete and list tasks that are in progress can't be cancelled. The following example shows a request to cancel a task.

```
exec msdb.dbo.rds_cancel_task @task_id = 1234;
```

To get an overview of all tasks and their task IDs, use the `rds_fn_task_status` function as described in [Monitoring the status of a file transfer task \(p. 730\)](#).

Multi-AZ limitations for S3 integration

On Multi-AZ instances, files in the `D:\S3` folder are deleted on the standby replica after a failover. A failover can be planned, for example, during DB instance modifications such as changing the instance class or upgrading the engine version. Or a failover can be unplanned, during an outage of the primary.

Note

We don't recommend using the `D:\S3` folder for file storage. The best practice is to upload created files to Amazon S3 to make them durable, and download files when you need to import data.

To determine the last failover time, you can use the `msdb.dbo.rds_failover_time` stored procedure. For more information, see [Determining the last failover time \(p. 817\)](#).

Example of no recent failover

This example shows the output when there is no recent failover in the error logs. No failover has happened since 2020-04-29 23:59:00.01.

Therefore, all files downloaded after that time that haven't been deleted using the `rds_delete_from_filesystem` stored procedure are still accessible on the current host. Files downloaded before that time might also be available.

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	null

Example of recent failover

This example shows the output when there is a failover in the error logs. The most recent failover was at 2020-05-05 18:57:51.89.

All files downloaded after that time that haven't been deleted using the `rds_delete_from_filesystem` stored procedure are still accessible on the current host.

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	2020-05-05 18:57:51.8900000

Disabling RDS for SQL Server integration with S3

Following, you can find how to disable Amazon S3 integration with Amazon RDS for SQL Server. Files in D:\S3\ aren't deleted when disabling S3 integration.

Note

To remove an IAM role from a DB instance, the status of the DB instance must be available.

Console

To disassociate your IAM role from your DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose the RDS for SQL Server DB instance name to display its details.
3. On the **Connectivity & security** tab, in the **Manage IAM roles** section, choose the IAM role to remove.
4. Choose **Delete**.

AWS CLI

To remove the IAM role from the RDS for SQL Server DB instance

- The following AWS CLI command removes the IAM role from a RDS for SQL Server DB instance named `mydbinstance`.

Example

For Linux, macOS, or Unix:

```
aws rds remove-role-from-db-instance \
--db-instance-identifier mydbinstance \
--feature-name S3_INTEGRATION \
--role-arn your-role-arn
```

For Windows:

```
aws rds remove-role-from-db-instance ^
--db-instance-identifier mydbinstance ^
--feature-name S3_INTEGRATION ^
--role-arn your-role-arn
```

Replace *your-role-arn* with the appropriate IAM role ARN for the --feature-name option.

Using Database Mail on Amazon RDS for SQL Server

You can use Database Mail to send email messages to users from your Amazon RDS on SQL Server database instance. The messages can contain files and query results. Database Mail includes the following components:

- **Configuration and security objects** – These objects create profiles and accounts, and are stored in the msdb database.
- **Messaging objects** – These objects include the `sp_send_dbmail` stored procedure used to send messages, and data structures that hold information about messages. They're stored in the msdb database.
- **Logging and auditing objects** – Database Mail writes logging information to the msdb database and the Microsoft Windows application event log.
- **Database Mail executable** – `DatabaseMail.exe` reads from a queue in the msdb database and sends email messages.

RDS supports Database Mail for all SQL Server versions on the Web, Standard, and Enterprise Editions.

Limitations

The following limitations apply to using Database Mail on your SQL Server DB instance:

- Database Mail isn't supported for SQL Server Express Edition.
- Modifying Database Mail configuration parameters isn't supported. To see the preset (default) values, use the `sysmail_help_configure_sp` stored procedure.
- File attachments aren't fully supported. For more information, see [Working with file attachments \(p. 743\)](#).
- The maximum file attachment size is 1 MB.
- Database Mail requires additional configuration on Multi-AZ DB instances. For more information, see [Considerations for Multi-AZ deployments \(p. 743\)](#).
- Configuring SQL Server Agent to send email messages to predefined operators isn't supported.

Enabling Database Mail

Use the following process to enable Database Mail for your DB instance:

1. Create a new parameter group.
2. Modify the parameter group to set the `database mail xps` parameter to 1.
3. Associate the parameter group with the DB instance.

Creating the parameter group for Database Mail

Create a parameter group for the `database mail xps` parameter that corresponds to the SQL Server edition and version of your DB instance.

Note

You can also modify an existing parameter group. Follow the procedure in [Modifying the parameter that enables Database Mail \(p. 735\)](#).

Console

The following example creates a parameter group for SQL Server Standard Edition 2016.

To create the parameter group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. Choose **Create parameter group**.
4. In the **Create parameter group** pane, do the following:
 - a. For **Parameter group family**, choose **sqlserver-se-13.0**.
 - b. For **Group name**, enter an identifier for the parameter group, such as **dbmail-sqlserver-se-13**.
 - c. For **Description**, enter **Database Mail XPs**.
5. Choose **Create**.

CLI

The following example creates a parameter group for SQL Server Standard Edition 2016.

To create the parameter group

- Use one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-parameter-group \
--db-parameter-group-name dbmail-sqlserver-se-13 \
--db-parameter-group-family "sqlserver-se-13.0" \
--description "Database Mail XPs"
```

For Windows:

```
aws rds create-db-parameter-group ^
--db-parameter-group-name dbmail-sqlserver-se-13 ^
--db-parameter-group-family "sqlserver-se-13.0" ^
--description "Database Mail XPs"
```

Modifying the parameter that enables Database Mail

Modify the `database mail xps` parameter in the parameter group that corresponds to the SQL Server edition and version of your DB instance.

To enable Database Mail, set the `database mail xps` parameter to 1.

Console

The following example modifies the parameter group that you created for SQL Server Standard Edition 2016.

To modify the parameter group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.

3. Choose the parameter group, such as **dbmail-sqlserver-se-13**.
4. Under **Parameters**, filter the parameter list for **mail**.
5. Choose **database mail xps**.
6. Choose **Edit parameters**.
7. Enter **1**.
8. Choose **Save changes**.

CLI

The following example modifies the parameter group that you created for SQL Server Standard Edition 2016.

To modify the parameter group

- Use one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name dbmail-sqlserver-se-13 \
--parameters "ParameterName='database mail
xps',ParameterValue=1,ApplyMethod=immediate"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name dbmail-sqlserver-se-13 ^
--parameters "ParameterName='database mail
xps',ParameterValue=1,ApplyMethod=immediate"
```

Associating the parameter group with the DB instance

You can use the AWS Management Console or the AWS CLI to associate the Database Mail parameter group with the DB instance.

Console

You can associate the Database Mail parameter group with a new or existing DB instance.

- For a new DB instance, associate it when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- For an existing DB instance, associate it by modifying the instance. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

CLI

You can associate the Database Mail parameter group with a new or existing DB instance.

To create a DB instance with the Database Mail parameter group

- Specify the same DB engine type and major version as you used when creating the parameter group.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
--db-instance-identifier mydbinstance \
--db-instance-class db.m5.2xlarge \
--engine sqlserver-se \
--engine-version 13.00.5426.0.v1 \
--allocated-storage 100 \
--master-user-password secret123 \
--master-username admin \
--storage-type gp2 \
--license-model li \
--db-parameter-group-name dbmail-sqlserver-se-13
```

For Windows:

```
aws rds create-db-instance ^
--db-instance-identifier mydbinstance ^
--db-instance-class db.m5.2xlarge ^
--engine sqlserver-se ^
--engine-version 13.00.5426.0.v1 ^
--allocated-storage 100 ^
--master-user-password secret123 ^
--master-username admin ^
--storage-type gp2 ^
--license-model li ^
--db-parameter-group-name dbmail-sqlserver-se-13
```

To modify a DB instance and associate the Database Mail parameter group

- Use one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--db-parameter-group-name dbmail-sqlserver-se-13 \
--apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--db-parameter-group-name dbmail-sqlserver-se-13 ^
--apply-immediately
```

Configuring Database Mail

You perform the following tasks to configure Database Mail:

1. Create the Database Mail profile.

2. Create the Database Mail account.
3. Add the Database Mail account to the Database Mail profile.
4. Add users to the Database Mail profile.

Note

To configure Database Mail, make sure that you have `execute` permission on the stored procedures in the `msdb` database.

Creating the Database Mail profile

To create the Database Mail profile, you use the `sysmail_add_profile_sp` stored procedure. The following example creates a profile named `Notifications`.

To create the profile

- Use the following SQL statement.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_profile_sp
    @profile_name      = 'Notifications',
    @description       = 'Profile used for sending outgoing notifications using
Gmail.';
GO
```

Creating the Database Mail account

To create the Database Mail account, you use the `sysmail_add_account_sp` stored procedure. The following example creates an account named `Gmail`.

To create the account

- Use the following SQL statement.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_account_sp
    @account_name      = 'Gmail',
    @description        = 'Mail account for sending outgoing notifications.',
    @email_address     = 'dbmail-test@gmail.com',
    @display_name       = 'Automated Mailer',
    @mailserver_name   = 'smtp.gmail.com',
    @port               = 587,
    @enable_ssl         = 1,
    @username           = 'dbmail-test@gmail.com',
    @password           = 'mypassword';
GO
```

Adding the Database Mail account to the Database Mail profile

To add the Database Mail account to the Database Mail profile, you use the `sysmail_add_profileaccount_sp` stored procedure. The following example adds the `Gmail` account to the `Notifications` profile.

To add the account to the profile

- Use the following SQL statement.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_profileaccount_sp
    @profile_name      = 'Notifications',
    @account_name      = 'Gmail',
    @sequence_number   = 1;
GO
```

Adding users to the Database Mail profile

To grant permission for an msdb database principal to use a Database Mail profile, you use the [sysmail_add_principalprofile_sp](#) stored procedure. A *principal* is an entity that can request SQL Server resources. The database principal must map to a SQL Server authentication user, a Windows Authentication user, or a Windows Authentication group.

The following example grants public access to the Notifications profile.

To add a user to the profile

- Use the following SQL statement.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_principalprofile_sp
    @profile_name      = 'Notifications',
    @principal_name    = 'public',
    @is_default        = 1;
GO
```

Amazon RDS stored procedures and functions for Database Mail

In addition to the [stored procedures](#) provided by Microsoft, RDS provides the stored procedures and functions for Database Mail shown in the following table.

Procedure/Function	Description
rds_fn_sysmail_allitems	Shows sent messages, including those submitted by other users.
rds_fn_sysmail_event_log	Shows events, including those for messages submitted by other users.
rds_fn_sysmail_mailattachments	Shows attachments, including those to messages submitted by other users.
rds_sysmail_control	Starts and stops the mail queue (DatabaseMail.exe process).
rds_sysmail_delete_mailitems_sp	Deletes email messages sent by all users from the Database Mail internal tables.

Sending email messages using Database Mail

You use the [sp_send_dbmail](#) stored procedure to send email messages using Database Mail.

Usage

```
EXEC msdb.dbo.sp_send_dbmail
@profile_name = 'profile_name',
@recipients = 'recipient1@example.com[; recipient2; ... recipientn]',
@subject = 'subject',
@body = 'message_body',
[@body_format = 'HTML'],
[@file_attachments = 'file_path1; file_path2; ... file_pathn'],
[@query = 'SQL_query'],
[@attach_query_result_as_file = 0/1]';
```

The following parameters are required:

- `@profile_name` – The name of the Database Mail profile from which to send the message.
- `@recipients` – The semicolon-delimited list of email addresses to which to send the message.
- `@subject` – The subject of the message.
- `@body` – The body of the message. You can also use a declared variable as the body.

The following parameters are optional:

- `@body_format` – This parameter is used with a declared variable to send email in HTML format.
- `@file_attachments` – The semicolon-delimited list of message attachments. File paths must be absolute paths.
- `@query` – A SQL query to run. The query results can be attached as a file or included in the body of the message.
- `@attach_query_result_as_file` – Whether to attach the query result as a file. Set to 0 for no, 1 for yes. The default is 0.

Examples

The following examples demonstrate how to send email messages.

Example of sending a message to a single recipient

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
    @profile_name      = 'Notifications',
    @recipients        = 'nobody@example.com',
    @subject           = 'Automated DBMail message - 1',
    @body               = 'Database Mail configuration was successful.';
GO
```

Example of sending a message to multiple recipients

```
USE msdb
GO
```

```
EXEC msdb.dbo.sp_send_dbmail
    @profile_name      = 'Notifications',
    @recipients        = 'recipient1@example.com;recipient2@example.com',
    @subject           = 'Automated DBMail message - 2',
    @body               = 'This is a message.';
GO
```

Example of sending a SQL query result as a file attachment

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
    @profile_name      = 'Notifications',
    @recipients        = 'nobody@example.com',
    @subject           = 'Test SQL query',
    @body               = 'This is a SQL query test.',
    @query              = 'SELECT * FROM abc.dbo.test',
    @attach_query_result_as_file = 1;
GO
```

Example of sending a message in HTML format

```
USE msdb
GO

DECLARE @HTML_Body as NVARCHAR(500) = 'Hi, <h4> Heading </h4> <br> See the report. <b>
Regards </b>';

EXEC msdb.dbo.sp_send_dbmail
    @profile_name      = 'Notifications',
    @recipients        = 'nobody@example.com',
    @subject           = 'Test HTML message',
    @body               = @HTML_Body,
    @body_format       = 'HTML';
GO
```

Example of sending a message using a trigger when a specific event occurs in the database

```
USE AdventureWorks2017
GO
IF OBJECT_ID ('Production.iProductNotification', 'TR') IS NOT NULL
DROP TRIGGER Purchasing.iProductNotification
GO

CREATE TRIGGER iProductNotification ON Production.Product
    FOR INSERT
    AS
    DECLARE @ProductInformation nvarchar(255);
    SELECT
        @ProductInformation = 'A new product, ' + Name + ', is now available for $' +
        CAST(StandardCost AS nvarchar(20)) + '!'
    FROM INSERTED i;

EXEC msdb.dbo.sp_send_dbmail
    @profile_name      = 'Notifications',
    @recipients        = 'nobody@example.com',
    @subject           = 'New product information',
    @body               = @ProductInformation;
GO
```

Viewing messages, logs, and attachments

You use RDS stored procedures to view messages, event logs, and attachments.

To view all email messages

- Use the following SQL query.

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_allitems(); --WHERE sent_status='sent' or  
'failed' or 'unsent'
```

To view all email event logs

- Use the following SQL query.

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_event_log();
```

To view all email attachments

- Use the following SQL query.

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_mailattachments();
```

Deleting messages

You use the `rds_sysmail_delete_mailitems_sp` stored procedure to delete messages.

Note

RDS automatically deletes mail table items when DBMail history data reaches 1 GB in size, with a retention period of at least 24 hours.

If you want to keep mail items for a longer period, you can archive them. For more information, see [Create a SQL Server Agent job to archive Database Mail messages and event logs](#) in the Microsoft documentation.

To delete all email messages

- Use the following SQL statement.

```
DECLARE @GETDATE datetime  
SET @GETDATE = GETDATE();  
EXECUTE msdb.dbo.rds_sysmail_delete_mailitems_sp @sent_before = @GETDATE;  
GO
```

To delete all email messages with a particular status

- Use the following SQL statement to delete all failed messages.

```
DECLARE @GETDATE datetime  
SET @GETDATE = GETDATE();  
EXECUTE msdb.dbo.rds_sysmail_delete_mailitems_sp @sent_status = 'failed';  
GO
```

Starting the mail queue

You use the `rds_sysmail_control` stored procedure to start the Database Mail process.

Note

Enabling Database Mail automatically starts the mail queue.

To start the mail queue

- Use the following SQL statement.

```
EXECUTE msdb.dbo.rds_sysmail_control start;
GO
```

Stopping the mail queue

You use the `rds_sysmail_control` stored procedure to stop the Database Mail process.

To stop the mail queue

- Use the following SQL statement.

```
EXECUTE msdb.dbo.rds_sysmail_control stop;
GO
```

Working with file attachments

The following file attachment extensions aren't supported in Database Mail messages from RDS on SQL Server: .ade, .adp, .apk, .appx, .appxbundle, .bat, .bak, .cab, .chm, .cmd, .com, .cpl, .dll, .dmg, .exe, .hta, .inf1, .ins, .isp, .is, and .wsh.

Database Mail uses the Microsoft Windows security context of the current user to control access to files. Users who log in with SQL Server Authentication can't attach files using the `@file_attachments` parameter with the `sp_send_dbmail` stored procedure. Windows doesn't allow SQL Server to provide credentials from a remote computer to another remote computer. Therefore, Database Mail can't attach files from a network share when the command is run from a computer other than the computer running SQL Server.

However, you can use SQL Server Agent jobs to attach files. For more information on SQL Server Agent, see [Using SQL Server Agent \(p. 822\)](#) and [SQL Server Agent](#) in the Microsoft documentation.

Considerations for Multi-AZ deployments

When you configure Database Mail on a Multi-AZ DB instance, the configuration isn't automatically propagated to the secondary. We recommend converting the Multi-AZ instance to a Single-AZ instance, configuring Database Mail, and then converting the DB instance back to Multi-AZ. Then both the primary and secondary nodes have the Database Mail configuration.

If you create a read replica from your Multi-AZ instance that has Database Mail configured, the replica inherits the configuration, but without the password to the SMTP server. Update the Database Mail account with the password.

Instance store support for the tempdb database on Amazon RDS for SQL Server

An *instance store* provides temporary block-level storage for your DB instance. This storage is located on disks that are physically attached to the host computer. These disks have Non-Volatile Memory Express (NVMe) instance storage that is based on solid-state drives (SSDs). This storage is optimized for low latency, very high random I/O performance, and high sequential read throughput.

By placing tempdb data files and tempdb log files on the instance store, you can achieve lower read and write latencies compared to standard storage based on Amazon EBS.

Note

SQL Server database files and database log files aren't placed on the instance store.

Enabling the instance store

When RDS provisions DB instances with one of the following instance classes, the tempdb database is automatically placed onto the instance store:

- db.m5d
- db.r5d

To enable the instance store, do one of the following:

- Create a SQL Server DB instance using one of these instance types. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- Modify an existing SQL Server DB instance to use one of them. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

The instance store is available in all AWS Regions where one or more of these instance types are supported. For more information on the db.m5d and db.r5d instance classes, see [DB instance classes \(p. 7\)](#). For more information on the instance classes supported by Amazon RDS for SQL Server, see [DB instance class support for Microsoft SQL Server \(p. 634\)](#).

File location and size considerations

On instances without an instance store, RDS stores the tempdb data and log files in the D:\rdsdbdata\DATA directory. Both files start at 8 MB by default.

On instances with an instance store, RDS stores the tempdb data and log files in the T:\rdsdbdata\DATA directory.

When tempdb has only one data file (tempdb.mdf) and one log file (templog.ldf), templog.ldf starts at 8 MB by default and tempdb.mdf starts at 80% or more of the instance's storage capacity. Twenty percent of the storage capacity or 200 GB, whichever is less, is kept free to start. Multiple tempdb data files split the 80% disk space evenly, while log files always have an 8-MB initial size.

For example, if you modify your DB instance class from db.m5.2xlarge to db.m5d.2xlarge, the size of tempdb data files increases from 8 MB each to 234 GB in total.

Note

Besides the tempdb data and log files on the instance store (T:\rdsdbdata\DATA), you can still create extra tempdb data and log files on the data volume (D:\rdsdbdata\DATA). Those files always have an 8 MB initial size.

Backup considerations

You might need to retain backups for long periods, incurring costs over time. The `tempdb` data and log blocks can change very often depending on the workload. This can greatly increase the DB snapshot size.

When `tempdb` is on the instance store, snapshots don't include temporary files. This means that snapshot sizes are smaller and consume less of the free backup allocation compared to EBS-only storage.

Disk full errors

If you use all of the available space in the instance store, you might receive errors such as the following:

- The transaction log for database 'tempdb' is full due to 'ACTIVE_TRANSACTION'.
- Could not allocate space for object 'dbo.SORT temporary run storage: 140738941419520' in database 'tempdb' because the 'PRIMARY' filegroup is full. Create disk space by deleting unneeded files, dropping objects in the filegroup, adding additional files to the filegroup, or setting autogrowth on for existing files in the filegroup.

You can do one or more of the following when the instance store is full:

- Adjust your workload or the way you use `tempdb`.
- Scale up to use a DB instance class with more NVMe storage.
- Stop using the instance store, and use an instance class with only EBS storage.
- Use a mixed mode by adding secondary data or log files for `tempdb` on the EBS volume.

Removing the instance store

To remove the instance store, modify your SQL Server DB instance to use an instance type that doesn't support instance store, such as `db.m5` or `db.r5`.

Note

When you remove the instance store, the temporary files are moved to the `D:\rdsdbdata\DATA` directory and reduced in size to 8 MB.

Using extended events with Amazon RDS for Microsoft SQL Server

You can use extended events in Microsoft SQL Server to capture debugging and troubleshooting information for Amazon RDS for SQL Server. Extended events replace SQL Trace and Server Profiler, which have been deprecated by Microsoft. Extended events are similar to profiler traces but with more granular control on the events being traced. Extended events are supported for SQL Server versions 2012 and later on Amazon RDS. For more information, see [Extended events overview](#) in the Microsoft documentation.

Extended events are turned on automatically for users with master user privileges in Amazon RDS for SQL Server.

Topics

- [Limitations and recommendations \(p. 746\)](#)
- [Configuring extended events on RDS for SQL Server \(p. 746\)](#)
- [Considerations for Multi-AZ deployments \(p. 747\)](#)
- [Querying extended event files \(p. 748\)](#)

Limitations and recommendations

When using extended events on RDS for SQL Server, the following limitations apply:

- Extended events are supported only for the Enterprise and Standard Editions.
- You can't alter default extended event sessions.
- Make sure to set the session memory partition mode to `NONE`.
- Session event retention mode can be either `ALLOW_SINGLE_EVENT_LOSS` or `ALLOW_MULTIPLE_EVENT_LOSS`.
- Event Tracing for Windows (ETW) targets aren't supported.
- Make sure that file targets are in the `D:\rdsdbdata\log` directory.
- For pair matching targets, set the `respond_to_memory_pressure` property to 1.
- Ring buffer target memory can't be greater than 4 MB.
- The following actions aren't supported:
 - `debug_break`
 - `create_dump_all_threads`
 - `create_dump_single_threads`
- The `rpc_completed` event is supported on the following versions and later: 15.0.4083.2, 14.0.3370.1, 13.0.5865.1, 12.0.6433.1, 11.0.7507.2.

Configuring extended events on RDS for SQL Server

On RDS for SQL Server, you can configure the values of certain parameters of extended event sessions. The following table describes the configurable parameters.

Parameter name	Description
<code>xe_session_max_memory</code>	Specifies the maximum amount
<code>xe_session_max_event_size</code>	Specifies the maximum memory

Parameter name	Description
<code>xe_session_max_dispatch_latency</code>	Specifies the amount of time that the event session.
<code>xe_file_target_size</code>	Specifies the maximum size of the .xel file.
<code>xe_file_retention</code>	Specifies the retention time in days.

Note

Setting `xe_file_retention` to zero causes .xel files to be removed automatically after the lock on these files is released by SQL Server. The lock is released whenever an .xel file reaches the size limit set in `xe_file_target_size`.

You can use the `rdsadmin.dbo.rds_show_configuration` stored procedure to show the current values of these parameters. For example, use the following SQL statement to view the current setting of `xe_session_max_memory`.

```
exec rdsadmin..rds_show_configuration 'xe_session_max_memory'
```

You can use the `rdsadmin.dbo.rds_set_configuration` stored procedure to modify them. For example, use the following SQL statement to set `xe_session_max_memory` to 4 MB.

```
exec rdsadmin..rds_set_configuration 'xe_session_max_memory', 4
```

Considerations for Multi-AZ deployments

When you create an extended event session on a primary DB instance, it doesn't propagate to the standby replica. You can fail over and create the extended event session on the new primary DB instance. Or you can remove and then re-add the Multi-AZ configuration to propagate the extended event session to the standby replica. RDS stops all nondefault extended event sessions on the standby replica, so that these sessions don't consume resources on the standby. Because of this, after a standby replica becomes the primary DB instance, make sure to manually start the extended event sessions on the new primary.

Note

This approach applies to both Always On Availability Groups and Database Mirroring.

You can also use a SQL Server Agent job to track the standby replica and start the sessions if the standby becomes the primary. For example, use the following query in your SQL Server Agent job step to restart event sessions on a primary DB instance.

```
BEGIN
    IF (DATABASEPROPERTYEX('rdsadmin','Updateability')='READ_WRITE'
        AND DATABASEPROPERTYEX('rdsadmin','status')='ONLINE'
        AND (DATABASEPROPERTYEX('rdsadmin','Collation') IS NOT NULL OR
        DATABASEPROPERTYEX('rdsadmin','IsAutoClose')=1)
    )
    BEGIN
        IF NOT EXISTS (SELECT 1 FROM sys.dm_xe_sessions WHERE name='xe1')
            ALTER EVENT SESSION xe1 ON SERVER STATE=START
        IF NOT EXISTS (SELECT 1 FROM sys.dm_xe_sessions WHERE name='xe2')
            ALTER EVENT SESSION xe2 ON SERVER STATE=START
    END
END
```

This query restarts the event sessions `xe1` and `xe2` on a primary DB instance if these sessions are in a stopped state. You can also add a schedule with a convenient interval to this query.

Querying extended event files

You can either use SQL Server Management Studio or the `sys.fn_xe_file_target_read_file` function to view data from extended events that use file targets. For more information on this function, see [sys.fn_xe_file_target_read_file \(Transact-SQL\)](#) in the Microsoft documentation.

Extended event file targets can only write files to the `D:\rdsdbdata\log` directory on RDS for SQL Server.

As an example, use the following SQL query to list the contents of all files of extended event sessions whose names start with `xe`.

```
SELECT * FROM sys.fn_xe_file_target_read_file('d:\rdsdbdata\log\xe*', null,null,null);
```

Options for the Microsoft SQL Server database engine

In this section, you can find descriptions for options that are available for Amazon RDS instances running the Microsoft SQL Server DB engine. To enable these options, you add them to an option group, and then associate the option group with your DB instance. For more information, see [Working with option groups \(p. 212\)](#).

If you're looking for optional features that aren't added through RDS option groups (such as SSL, Microsoft Windows Authentication, and Amazon S3 integration), see [Additional features for Microsoft SQL Server on Amazon RDS \(p. 703\)](#).

Amazon RDS supports the following options for Microsoft SQL Server DB instances.

Option	Option ID	Engine editions
Native backup and restore (p. 751)	SQLSERVER_BACKUP_RESTORE	SQL Server Enterprise Edition SQL Server Standard Edition SQL Server Web Edition SQL Server Express Edition
Transparent Data Encryption (p. 754)	TRANSPARENT_DATA_ENCRYPTION (RDS console) TDE (AWS CLI and RDS API)	SQL Server 2012–2019 Enterprise Edition SQL Server 2019 Standard Edition
SQL Server Audit (p. 757)	SQLSERVER_AUDIT	In RDS, starting with SQL Server 2012, all editions of SQL Server support server-level audits, and Enterprise Edition also supports database-level audits. Starting with SQL Server SQL Server 2016 (13.x) SP1, all editions support both server-level and database-level audits. For more information, see SQL Server Audit (database engine) in the SQL Server documentation.
SQL Server Analysis Services (p. 762)	SSAS	SQL Server Enterprise Edition

Option	Option ID	Engine editions
		SQL Server Standard Edition
SQL Server Integration Services (p. 773)	SSIS	SQL Server Enterprise Edition
		SQL Server Standard Edition
SQL Server Reporting Services (p. 787)	SSRS	SQL Server Enterprise Edition
		SQL Server Standard Edition
Microsoft Distributed Transaction Coordinator (p. 797)	MSDTC	In RDS, starting with SQL Server 2012, all editions of SQL Server support distributed transactions.

Listing the available options for SQL Server versions and editions

You can use the `describe-option-group-options` AWS CLI command to list the available options for SQL Server versions and editions, and the settings for those options.

The following example shows the options and option settings for SQL Server 2019 Enterprise Edition. The `--engine-name` option is required.

```
aws rds describe-option-group-options --engine-name sqlserver-ee --major-engine-version 15.00
```

The output resembles the following:

```
{
    "OptionGroupOptions": [
        {
            "Name": "MSDTC",
            "Description": "Microsoft Distributed Transaction Coordinator",
            "EngineName": "sqlserver-ee",
            "MajorEngineVersion": "15.00",
            "MinimumRequiredMinorEngineVersion": "4043.16.v1",
            "PortRequired": true,
            "DefaultPort": 5000,
            "OptionsDependedOn": [],
            "OptionsConflictsWith": [],
            "Persistent": false,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "OptionGroupOptionSettings": [
                {
                    "SettingName": "ENABLE_SNA_LU",
                    "SettingDescription": "able support for SNA LU protocol",
                    "DefaultValue": "true",
                    "Value": "true"
                }
            ]
        }
    ]
}
```

```
        "ApplyType": "DYNAMIC",
        "AllowedValues": "true,false",
        "IsModifiable": true,
        "IsRequired": false,
        "MinimumEngineVersionPerAllowedValue": []
    },
    ...
{
    "Name": "TDE",
    "Description": "SQL Server - Transparent Data Encryption",
    "EngineName": "sqlserver-ee",
    "MajorEngineVersion": "15.00",
    "MinimumRequiredMinorEngineVersion": "4043.16.v1",
    "PortRequired": false,
    "OptionsDependedOn": [],
    "OptionsConflictsWith": [],
    "Persistent": true,
    "Permanent": false,
    "RequiresAutoMinorEngineVersionUpgrade": false,
    "VpcOnly": false,
    "OptionGroupOptionSettings": []
}
]
```

Support for native backup and restore in SQL Server

By using native backup and restore for SQL Server databases, you can create a differential or full backup of your on-premises database and store the backup files on Amazon S3. You can then restore to an existing Amazon RDS DB instance running SQL Server. You can also back up an RDS for SQL Server database, store it on Amazon S3, and restore it in other locations. In addition, you can restore the backup to an on-premises server, or a different Amazon RDS DB instance running SQL Server. For more information, see [Importing and exporting SQL Server databases \(p. 671\)](#).

Amazon RDS supports native backup and restore for Microsoft SQL Server databases by using differential and full backup files (.bak files).

Adding the native backup and restore option

The general process for adding the native backup and restore option to a DB instance is the following:

1. Create a new option group, or copy or modify an existing option group.
2. Add the `SQLSERVER_BACKUP_RESTORE` option to the option group.
3. Associate an AWS Identity and Access Management (IAM) role with the option. The IAM role must have access to an S3 bucket to store the database backups.

That is, it must have as its option setting a valid Amazon Resource Name (ARN) in the format `arn:aws:iam::account-id:role/role-name`. For more information, see [Amazon Resource Names \(ARNs\) in the AWS General Reference](#).

4. Associate the option group with the DB instance.

After you add the native backup and restore option, you don't need to restart your DB instance. As soon as the option group is active, you can begin backing up and restoring immediately.

Console

To add the native backup and restore option

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Create a new option group or use an existing option group. For information on how to create a custom DB option group, see [Creating an option group \(p. 214\)](#).
To use an existing option group, skip to the next step.
4. Add the **SQLSERVER_BACKUP_RESTORE** option to the option group. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).
5. Do one of the following:
 - To use an existing IAM role and Amazon S3 settings, choose an existing IAM role for **IAM Role**. If you use an existing IAM role, RDS uses the Amazon S3 settings configured for this role.
 - To create a new role and configure new Amazon S3 settings, do the following:
 1. For **IAM Role**, choose [Create a New Role](#).
 2. For **Select S3 Bucket**, either create an S3 bucket or use an existing one. To create a new bucket, choose [Create a New S3 Bucket](#). To use an existing bucket, choose it from the list.
 3. For **S3 folder path prefix (optional)**, specify a prefix to use for the files stored in your Amazon S3 bucket.

This prefix can include a file path but doesn't have to. If you provide a prefix, RDS attaches that prefix to all backup files. RDS then uses the prefix during a restore to identify related files and ignore irrelevant files. For example, you might use the S3 bucket for purposes besides holding backup files. In this case, you can use the prefix to have RDS perform native backup and restore only on a particular folder and its subfolders.

If you leave the prefix blank, then RDS doesn't use a prefix to identify backup files or files to restore. As a result, during a multiple-file restore, RDS attempts to restore every file in every folder of the S3 bucket.

4. For **Enable Encryption**, choose **Yes** to encrypt the backup file. Choose **No** to leave the backup file unencrypted.

If you choose **Yes**, choose an encryption key for **Master Key**. For more information about encryption keys, see [Getting started in the AWS Key Management Service Developer Guide](#).

6. Choose **Add option**.
7. Apply the option group to a new or existing DB instance:
 - For a new DB instance, apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
 - For an existing DB instance, apply the option group by modifying the instance and attaching the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

CLI

This procedure makes the following assumptions:

- You're adding the **SQLSERVER_BACKUP_RESTORE** option to an option group that already exists. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).
- You're associating the option with an IAM role that already exists and has access to an S3 bucket to store the backups.

- You're applying the option group to a DB instance that already exists. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

To add the native backup and restore option

1. Add the `SQLSERVER_BACKUP_RESTORE` option to the option group.

Example

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \
--apply-immediately \
--option-group-name mybackupgroup \
--options "OptionName=SQLSERVER_BACKUP_RESTORE, \
OptionSettings=[{Name=IAM_ROLE_ARN,Value=arn:aws:iam::account-id:role/role-name}]"
```

For Windows:

```
aws rds add-option-to-option-group ^
--option-group-name mybackupgroup ^
--options "[{\\"OptionName\\": \\"SQLSERVER_BACKUP_RESTORE\\\", ^
\\\"OptionSettings\\\": [{\\\"Name\\\": \\"IAM_ROLE_ARN\\\", ^
\\\"Value\\\": \\"arn:aws:iam::account-id:role/role-name\"}]}]" ^
--apply-immediately
```

Note

When using the Windows command prompt, you must escape double quotes ("") in JSON code by prefixing them with a backslash (\).

2. Apply the option group to the DB instance.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--option-group-name mybackupgroup \
--apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--option-group-name mybackupgroup ^
--apply-immediately
```

Modifying native backup and restore option settings

After you enable the native backup and restore option, you can modify the settings for the option. For more information about how to modify option settings, see [Modifying an option setting \(p. 221\)](#).

Removing the native backup and restore option

You can turn off native backup and restore by removing the option from your DB instance. After you remove the native backup and restore option, you don't need to restart your DB instance.

To remove the native backup and restore option from a DB instance, do one of the following:

- Remove the option from the option group it belongs to. This change affects all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).
- Modify the DB instance and specify a different option group that doesn't include the native backup and restore option. This change affects a single DB instance. You can specify the default (empty) option group, or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Support for Transparent Data Encryption in SQL Server

Amazon RDS supports using Transparent Data Encryption (TDE) to encrypt stored data on your DB instances running Microsoft SQL Server. TDE automatically encrypts data before it is written to storage, and automatically decrypts data when the data is read from storage.

Amazon RDS supports TDE for the following SQL Server versions and editions:

- SQL Server 2019 Standard and Enterprise Editions
- SQL Server 2017 Enterprise Edition
- SQL Server 2016 Enterprise Edition
- SQL Server 2014 Enterprise Edition
- SQL Server 2012 Enterprise Edition

Transparent Data Encryption for SQL Server provides encryption key management by using a two-tier key architecture. A certificate, which is generated from the database master key, is used to protect the data encryption keys. The database encryption key performs the actual encryption and decryption of data on the user database. Amazon RDS backs up and manages the database master key and the TDE certificate.

Note

RDS doesn't support exporting TDE certificates.

Transparent Data Encryption is used in scenarios where you need to encrypt sensitive data. For example, you might want to provide data files and backups to a third party, or address security-related regulatory compliance issues. You can't encrypt the system databases for SQL Server, such as the `model` or `master` databases.

Note

You can create native backups of TDE-enabled databases, but you can't restore those backups to on-premises databases.

A detailed discussion of Transparent Data Encryption is beyond the scope of this guide, but you should understand the security strengths and weaknesses of each encryption algorithm and key. For information about Transparent Data Encryption for SQL Server, see [Transparent Data Encryption \(TDE\)](#) on the Microsoft website.

Enabling TDE

To enable Transparent Data Encryption for an RDS for SQL Server DB instance, specify the TDE option in an RDS option group that is associated with that DB instance.

1. Determine whether your DB instance is already associated with an option group that has the TDE option. To view the option group that a DB instance is associated with, you can use the RDS console, the `describe-db-instance` AWS CLI command, or the API operation `DescribeDBInstances`.
2. If the DB instance isn't associated with an option group that has TDE enabled, you have two choices. You can create an option group and add the TDE option, or you can modify the associated option group to add it.

Note

In the RDS console, the option is named `TRANSPARENT_DATA_ENCRYPTION`. In the AWS CLI and RDS API, it's named `TDE`.

For information about creating or modifying an option group, see [Working with option groups \(p. 212\)](#). For information about adding an option to an option group, see [Adding an option to an option group \(p. 216\)](#).

3. Associate the DB instance with the option group that has the TDE option. For information about associating a DB instance with an option group, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Encrypting data

When the TDE option is added to an option group, Amazon RDS generates a certificate that is used in the encryption process. You can then use the certificate to run SQL statements that encrypt data in a database on the DB instance. The following example uses the RDS-created certificate called `RDSTDECertificateName` to encrypt a database called `customerDatabase`.

```
----- Enabling TDE -----  
  
-- Find a RDSTDECertificate to use  
USE [master]  
GO  
SELECT name FROM sys.certificates WHERE name LIKE 'RDSTDECertificate%'  
GO  
  
USE [customerDatabase]  
GO  
-- Create DEK using one of the certificates from the previous step  
CREATE DATABASE ENCRYPTION KEY  
WITH ALGORITHM = AES_128  
ENCRYPTION BY SERVER CERTIFICATE [RDSTDECertificateName]  
GO  
  
-- Enable encryption on the database  
ALTER DATABASE [customerDatabase]  
SET ENCRYPTION ON  
GO  
  
-- Verify that the database is encrypted  
USE [master]  
GO  
SELECT name FROM sys.databases WHERE is_encrypted = 1  
GO  
SELECT db_name(database_id) as DatabaseName, * FROM sys.dm_database_encryption_keys  
GO
```

The time that it takes to encrypt a SQL Server database using TDE depends on several factors. These include the size of the DB instance, whether PIOPS is enabled for the instance, the amount of data, and other factors.

Option group considerations

The TDE option is a persistent option that you can't remove from an option group unless all DB instances and backups are disassociated from the option group. After you add the TDE option to an option group, the option group can only be associated with DB instances that use TDE. For more information about persistent options in an option group, see [Option groups overview \(p. 212\)](#).

Because the TDE option is a persistent option, you can have a conflict between the option group and an associated DB instance. You can have a conflict between the option group and an associated DB instance in the following situations:

- The current option group has the TDE option, and you replace it with an option group that does not have the TDE option.
- You restore from a DB snapshot to a new DB instance that does not have an option group that contains the TDE option. For more information about this scenario, see [Option group considerations \(p. 356\)](#).

SQL Server performance considerations

The performance of a SQL Server DB instance can be impacted by using Transparent Data Encryption.

Performance for unencrypted databases can also be degraded if the databases are on a DB instance that has at least one encrypted database. As a result, we recommend that you keep encrypted and unencrypted databases on separate DB instances.

Because of the nature of encryption, the database size and the size of the transaction log is larger than for an unencrypted database. You could run over your allocation of free backup space. The nature of TDE causes an unavoidable performance hit. If you need high performance and TDE, measure the impact and make sure that it meets your needs. There is less of an impact on performance if you use Provisioned IOPS and at least an M3.Large DB instance class.

Disabling TDE

To disable TDE for a DB instance, first make sure that there are no encrypted objects left on the DB instance by either decrypting the objects or by dropping them. If any encrypted objects exist on the DB instance, you can't disable TDE for the DB instance. When you use the console to remove the TDE option from an option group, the console indicates that it is processing. In addition, an error event is created if the option group is associated with an encrypted DB instance or DB snapshot.

The following example removes the TDE encryption from a database called `customerDatabase`.

```
----- Removing TDE -----  
  
USE [customerDatabase]  
GO  
  
-- Disable encryption on the database  
ALTER DATABASE [customerDatabase]  
SET ENCRYPTION OFF  
GO  
  
-- Wait until the encryption state of the database becomes 1. The state is 5 (Decryption in progress) for a while
```

```
SELECT db_name(database_id) as DatabaseName, * FROM sys.dm_database_encryption_keys
GO

-- Drop the DEK used for encryption
DROP DATABASE ENCRYPTION KEY
GO

-- Alter to SIMPLE Recovery mode so that your encrypted log gets truncated
USE [master]
GO
ALTER DATABASE [customerDatabase] SET RECOVERY SIMPLE
GO
```

When all objects are decrypted, you can have two options. You can modify the DB instance to be associated with an option group without the TDE option. Or you can remove the TDE option from the option group.

SQL Server Audit

In Amazon RDS, you can audit Microsoft SQL Server databases by using the built-in SQL Server auditing mechanism. You can create audits and audit specifications in the same way that you create them for on-premises database servers.

RDS uploads the completed audit logs to your S3 bucket, using the IAM role that you provide. If you enable retention, RDS keeps your audit logs on your DB instance for the configured period of time.

For more information, see [SQL Server Audit \(database engine\)](#) in the Microsoft SQL Server documentation.

Topics

- [Support for SQL Server Audit \(p. 757\)](#)
- [Adding SQL Server Audit to the DB instance options \(p. 758\)](#)
- [Using SQL Server Audit \(p. 759\)](#)
- [Viewing audit logs \(p. 760\)](#)
- [Using SQL Server Audit with Multi-AZ instances \(p. 760\)](#)
- [Configuring an S3 bucket \(p. 761\)](#)
- [Manually creating an IAM role for SQL Server Audit \(p. 761\)](#)

Support for SQL Server Audit

In Amazon RDS, starting with SQL Server 2012, all editions of SQL Server support server-level audits, and the Enterprise edition also supports database-level audits. Starting with SQL Server 2016 (13.x) SP1, all editions support both server-level and database-level audits. For more information, see [SQL Server Audit \(database engine\)](#) in the SQL Server documentation.

RDS supports configuring the following option settings for SQL Server Audit.

Option setting	Valid values	Description
IAM_ROLE_ARN	A valid Amazon Resource Name (ARN) in the format <code>arn:aws:iam::account-id:role/role-name</code> .	The ARN of the IAM role that grants access to the S3 bucket where you want to store your audit logs. For more information, see Amazon

Option setting	Valid values	Description
		Resource Names (ARNs) in the AWS General Reference .
S3_BUCKET_ARN	A valid ARN in the format <code>arn:aws:s3:::bucket-name</code> or <code>arn:aws:s3:::bucket-name/key-prefix</code>	The ARN for the S3 bucket where you want to store your audit logs.
ENABLE_COMPRESSION	true or false	Controls audit log compression. By default, compression is enabled (set to true).
RETENTION_TIME	0 to 840	The retention time (in hours) that SQL Server audit records are kept on your RDS instance. By default, retention is disabled.

RDS supports SQL Server Audit in all AWS Regions except Middle East (Bahrain).

Adding SQL Server Audit to the DB instance options

Enabling SQL Server Audit requires two steps: enabling the option on the DB instance, and enabling the feature inside SQL Server. The process for adding the SQL Server Audit option to a DB instance is as follows:

1. Create a new option group, or copy or modify an existing option group.
2. Add and configure all required options.
3. Associate the option group with the DB instance.

After you add the SQL Server Audit option, you don't need to restart your DB instance. As soon as the option group is active, you can create audits and store audit logs in your S3 bucket.

To add and configure SQL Server Audit on a DB instance's option group

1. Choose one of the following:
 - Use an existing option group.
 - Create a custom DB option group and use that option group. For more information, see [Creating an option group \(p. 214\)](#).
2. Add the **SQLSERVER_AUDIT** option to the option group, and configure the option settings. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).
 - For **IAM role**, if you already have an IAM role with the required policies, you can choose that role. To create a new IAM role, choose **Create a New Role**. For information about the required policies, see [Manually creating an IAM role for SQL Server Audit \(p. 761\)](#).
 - For **Select S3 destination**, if you already have an S3 bucket that you want to use, choose it. To create an S3 bucket, choose **Create a New S3 Bucket**.
 - For **Enable Compression**, leave this option chosen to compress audit files. Compression is enabled by default. To disable compression, clear **Enable Compression**.
 - For **Audit log retention**, to keep audit records on the DB instance, choose this option. Specify a retention time in hours. The maximum retention time is 35 days.
3. Apply the option group to a new or existing DB instance. Choose one of the following:

- If you are creating a new DB instance, apply the option group when you launch the instance.
- On an existing DB instance, apply the option group by modifying the instance and then attaching the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Modifying the SQL Server Audit option

After you enable the SQL Server Audit option, you can modify the settings. For information about how to modify option settings, see [Modifying an option setting \(p. 221\)](#).

Removing SQL Server Audit from the DB instance options

You can turn off the SQL Server Audit feature by disabling audits and then deleting the option.

To remove auditing

1. Disable all of the audit settings inside SQL Server. To learn where audits are running, query the SQL Server security catalog views. For more information, see [Security catalog views](#) in the Microsoft SQL Server documentation.
2. Delete the SQL Server Audit option from the DB instance. Choose one of the following:
 - Delete the SQL Server Audit option from the option group that the DB instance uses. This change affects all DB instances that use the same option group. For more information, see [Removing an option from an option group \(p. 224\)](#).
 - Modify the DB instance, and then choose an option group without the SQL Server Audit option. This change affects only the DB instance that you modify. You can specify the default (empty) option group, or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
3. After you delete the SQL Server Audit option from the DB instance, you don't need to restart the instance. Remove unneeded audit files from your S3 bucket.

Using SQL Server Audit

You can control server audits, server audit specifications, and database audit specifications the same way that you control them for on-premises database servers.

Creating audits

You create server audits in the same way that you create them for on-premises database servers. For information about how to create server audits, see [CREATE SERVER AUDIT](#) in the Microsoft SQL Server documentation.

To avoid errors, adhere to the following limitations:

- Don't exceed the maximum number of supported server audits per instance of 50.
- Instruct SQL Server to write data to a binary file.
- Don't use RDS_ as a prefix in the server audit name.
- For FILEPATH, specify D:\rdsdbdata\SQLAudit.
- For MAXSIZE, specify a size between 2 MB and 50 MB.
- Don't configure MAX_ROLLOVER_FILES or MAX_FILES.
- Don't configure SQL Server to shut down the DB instance if it fails to write the audit record.

Creating audit specifications

You create server audit specifications and database audit specifications the same way that you create them for on-premises database servers. For information about creating audit specifications, see [CREATE SERVER AUDIT SPECIFICATION](#) and [CREATE DATABASE AUDIT SPECIFICATION](#) in the Microsoft SQL Server documentation.

To avoid errors, don't use `RDS_` as a prefix in the name of the database audit specification or server audit specification.

Viewing audit logs

Your audit logs are stored in `D:\rdsdbdata\SQLAudit`.

After SQL Server finishes writing to an audit log file—when the file reaches its size limit—Amazon RDS uploads the file to your S3 bucket. If retention is enabled, Amazon RDS moves the file into the retention folder: `D:\rdsdbdata\SQLAudit\transmitted`.

For information about configuring retention, see [Adding SQL Server Audit to the DB instance options \(p. 758\)](#).

Audit records are kept on the DB instance until the audit log file is uploaded. You can view the audit records by running the following command.

```
SELECT *
FROM msdb.dbo.rds_fn_get_audit_file
  ('D:\rdsdbdata\SQLAudit\*.sqlaudit'
  , default
  , default )
```

You can use the same command to view audit records in your retention folder by changing the filter to `D:\rdsdbdata\SQLAudit\transmitted*.sqlaudit`.

```
SELECT *
FROM msdb.dbo.rds_fn_get_audit_file
  ('D:\rdsdbdata\SQLAudit\transmitted\*.sqlaudit'
  , default
  , default )
```

Using SQL Server Audit with Multi-AZ instances

For Multi-AZ instances, the process for sending audit log files to Amazon S3 is similar to the process for Single-AZ instances. However, there are some important differences:

- Database audit specification objects are replicated to all nodes.
- Server audits and server audit specifications aren't replicated to secondary nodes. Instead, you have to create or modify them manually.

To capture server audits or a server audit specification from both nodes:

1. Create a server audit or a server audit specification on the primary node.
2. Fail over to the secondary node and create a server audit or a server audit specification with the same name and GUID on the secondary node. Use the `AUDIT_GUID` parameter to specify the GUID.

Configuring an S3 bucket

The audit log files are automatically uploaded from the DB instance to your S3 bucket. The following restrictions apply to the S3 bucket that you use as a target for audit files:

- It must be in the same AWS Region as the DB instance.
- It must not be open to the public.
- It can't use [S3 Object Lock](#).
- The bucket owner must also be the IAM role owner.

The target key that is used to store the data follows this naming schema: `bucket-name/key-prefix/instance-name/audit-name/node_file-name.ext`

Note

You set both the bucket name and the key prefix values with the (`s3_BUCKET_ARN`) option setting.

The schema is composed of the following elements:

- **bucket-name** – The name of your S3 bucket.
- **key-prefix** – The custom key prefix you want to use for audit logs.
- **instance-name** – The name of your Amazon RDS instance.
- **audit-name** – The name of the audit.
- **node** – The identifier of the node that is the source of the audit logs (`node1` or `node2`). There is one node for a Single-AZ instance and two replication nodes for a Multi-AZ instance. These are not primary and secondary nodes, because the roles of primary and secondary change over time. Instead, the node identifier is a simple label.
 - **node1** – The first replication node (Single-AZ has one node only).
 - **node2** – The second replication node (Multi-AZ has two nodes).
- **file-name** – The target file name. The file name is taken as-is from SQL Server.
- **ext** – The extension of the file (`zip` or `sqlaudit`):
 - **zip** – If compression is enabled (default).
 - **sqlaudit** – If compression is disabled.

Manually creating an IAM role for SQL Server Audit

Typically, when you create a new option, the AWS Management Console creates the IAM role and the IAM trust policy for you. However, you can manually create a new IAM role to use with SQL Server Audits, so that you can customize it with any additional requirements you might have. To do this, you create an IAM role and delegate permissions so that the Amazon RDS service can use your Amazon S3 bucket. When you create this IAM role, you attach trust and permissions policies. The trust policy allows Amazon RDS to assume this role. The permission policy defines the actions that this role can do. For more information, see [Creating a role to delegate permissions to an AWS service](#) in the *AWS Identity and Access Management User Guide*.

You can use the examples in this section to create the trust and permissions policies you need.

The following example shows a trust policy for SQL Server Audit. The policy uses the *service principal* `rds.amazonaws.com` to allow RDS to write to the S3 bucket. A *service principal* is an identifier that is used to grant permissions to a service. Anytime you allow access to `rds.amazonaws.com` in this way, you are allowing RDS to perform an action on your behalf. For more information about service principals, see [AWS JSON policy elements: Principal](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

In the following example of a permissions policy for SQL Server Audit, we specify an Amazon Resource Name (ARN) for the Amazon S3 bucket. You can use ARNs to identify a specific account, user, or role that you want grant access to. For more information about using ARNs, see [Amazon resource names \(ARNs\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3>ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3>ListBucket",
                "s3:GetBucketACL",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::bucket_name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3>PutObject",
                "s3>ListMultipartUploadParts",
                "s3>AbortMultipartUpload"
            ],
            "Resource": "arn:aws:s3:::bucket_name/key_prefix/*"
        }
    ]
}
```

Note

The `s3>ListAllMyBuckets` action is required for verifying that the same AWS account owns both the S3 bucket and the SQL Server DB instance. The action lists the names of the buckets in the account.

S3 bucket namespaces are global. If you accidentally delete your bucket, another user can create a bucket with the same name in a different account. Then the SQL Server Audit data is written to the new bucket.

Support for SQL Server Analysis Services in Amazon RDS for SQL Server

Microsoft SQL Server Analysis Services (SSAS) is part of the Microsoft Business Intelligence (MSBI) suite. SSAS is an online analytical processing (OLAP) and data mining tool that is installed within

SQL Server. You use SSAS to analyze data to help make business decisions. SSAS differs from the SQL Server relational database because SSAS is optimized for queries and calculations common in a business intelligence environment. For more information on SSAS, see the Microsoft [Analysis services documentation](#).

Amazon RDS for SQL Server supports running SQL Server Analysis Services (SSAS) in Tabular mode. You can enable SSAS on existing or new DB instances. It's installed on the same DB instance as your database engine.

RDS supports SSAS for SQL Server Standard and Enterprise Editions on the following versions:

- SQL Server 2019, version 15.00.4043.16.v1 and later
- SQL Server 2017, version 14.00.3223.3.v1 and later
- SQL Server 2016, version 13.00.5426.0.v1 and later

Limitations

The following limitations apply to running SSAS on RDS for SQL Server:

- Tabular is the only supported mode for SSAS.
- Multi-AZ instances aren't supported.
- Instances must use AWS Directory Service for Microsoft Active Directory for SSAS authentication.
- Users aren't given SSAS server administrator access, but they can be granted database-level administrator access.
- The only supported port for accessing SSAS is 2383.
- You can't deploy projects directly. We provide an RDS stored procedure to do this. For more information, see [Deploying SSAS projects on Amazon RDS \(p. 767\)](#).
- Processing during deployment isn't supported.
- Using .xmla files for deployment isn't supported.
- SSAS project input files and database backup output files can only be in the D:\S3 folder on the DB instance.

Enabling SSAS

Use the following process to enable SSAS for your DB instance:

1. Create a new option group, or choose an existing option group.
2. Add the SSAS option to the option group.
3. Associate the option group with the DB instance.
4. Allow inbound access to the VPC security group for the SSAS listener port.
5. Enable Amazon S3 integration.

Creating the option group for SSAS

Use the AWS Management Console or the AWS CLI to create an option group that corresponds to the SQL Server engine and version of the DB instance that you plan to use.

Note

You can also use an existing option group if it's for the correct SQL Server engine and version.

Console

The following console procedure creates an option group for SQL Server Standard Edition 2017.

To create the option group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose **Create group**.
4. In the **Create option group** pane, do the following:
 - a. For **Name**, enter a name for the option group that is unique within your AWS account, such as **ssas-se-2017**. The name can contain only letters, digits, and hyphens.
 - b. For **Description**, enter a brief description of the option group, such as **SSAS option group for SQL Server SE 2017**. The description is used for display purposes.
 - c. For **Engine**, choose **sqlserver-se**.
 - d. For **Major engine version**, choose **14.00**.
5. Choose **Create**.

CLI

The following CLI example creates an option group for SQL Server Standard Edition 2017.

To create the option group

- Use one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds create-option-group \
--option-group-name ssas-se-2017 \
--engine-name sqlserver-se \
--major-engine-version 14.00 \
--option-group-description "SSAS option group for SQL Server SE 2017"
```

For Windows:

```
aws rds create-option-group ^
--option-group-name ssas-se-2017 ^
--engine-name sqlserver-se ^
--major-engine-version 14.00 ^
--option-group-description "SSAS option group for SQL Server SE 2017"
```

Adding the SSAS option to the option group

Next, use the AWS Management Console or the AWS CLI to add the SSAS option to the option group.

Console

To add the SSAS option

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.

3. Choose the option group that you just created.
4. Choose **Add option**.
5. Under **Option details**, choose **SSAS** for **Option name**.
6. Under **Option settings**, enter a value from 10–80 for **Max memory**.

Max memory specifies the upper threshold above which SSAS begins releasing memory more aggressively to make room for requests that are running, and also new high-priority requests. The number is a percentage of the total memory of the DB instance. The allowed values are 10–80, and the default is 45.

Note

The port for accessing SSAS, 2383, is prepopulated.

7. For **Security groups**, choose the VPC security group to associate with the option.
8. Under **Scheduling**, choose whether to add the option immediately or at the next maintenance window.
9. Choose **Add option**.

CLI

To add the SSAS option

1. Create a JSON file, for example `ssas-option.json`, with the following parameters:
 - `OptionGroupName` – The name of option group that you created or chose previously (`ssas-se-2017` in the following example).
 - `Port` – The port that you use to access SSAS. The only supported port is 2383.
 - `VpcSecurityGroupMemberships` – VPC security group memberships for your RDS DB instance.
 - `MAX_MEMORY` – The upper threshold above which SSAS should begin releasing memory more aggressively to make room for requests that are running, and also new high-priority requests. The number is a percentage of the total memory of the DB instance. The allowed values are 10–80, and the default is 45.

```
{
  "OptionGroupName": "ssas-se-2017",
  "OptionsToInclude": [
    {
      "OptionName": "SSAS",
      "Port": 2383,
      "VpcSecurityGroupMemberships" : ["sg-0abcdef123"],
      "OptionSettings": [{"Name" : "MAX_MEMORY", "Value" : "60"}]
    },
    "ApplyImmediately": true
  }
}
```

2. Add the SSAS option to the option group.

Example

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \
  --cli-input-json file://ssas-option.json \
  --apply-immediately
```

For Windows:

```
aws rds add-option-to-option-group ^
--cli-input-json file://ssas-option.json ^
--apply-immediately
```

Associating the option group with your DB instance

You can use the AWS Management Console or the AWS CLI to associate the option group with your DB instance.

Console

Associate your option group with a new or existing DB instance:

- For a new DB instance, associate the option group with the DB instance when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- For an existing DB instance, modify the instance and associate the new option group with it. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Note

If you use an existing instance, it must already have an Active Directory domain and IAM role associated with it. If you create a new instance, specify an existing Active Directory domain and IAM role. For more information, see [Using Windows Authentication with an Amazon RDS for SQL Server DB instance \(p. 711\)](#).

CLI

You can associate your option group with a new or existing DB instance.

Note

If you use an existing instance, it must already have an Active Directory domain and IAM role associated with it. If you create a new instance, specify an existing Active Directory domain and IAM role. For more information, see [Using Windows Authentication with an Amazon RDS for SQL Server DB instance \(p. 711\)](#).

To create a DB instance that uses the option group

- Specify the same DB engine type and major version that you used when creating the option group.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
--db-instance-identifier myssasinstance \
--db-instance-class db.m5.2xlarge \
--engine sqlserver-se \
--engine-version 14.00.3223.3.v1 \
--allocated-storage 100 \
--master-user-password secret123 \
--master-username admin \
--storage-type gp2 \
--license-model li \
--domain-iam-role-name my-directory-iam-role \
--domain my-domain-id \
--option-group-name ssas-se-2017
```

For Windows:

```
aws rds create-db-instance ^
--db-instance-identifier myssasinstance ^
--db-instance-class db.m5.2xlarge ^
--engine sqlserver-se ^
--engine-version 14.00.3223.3.v1 ^
--allocated-storage 100 ^
--master-user-password secret123 ^
--master-username admin ^
--storage-type gp2 ^
--license-model li ^
--domain-iam-role-name my-directory-iam-role ^
--domain my-domain-id ^
--option-group-name ssas-se-2017
```

To modify a DB instance to associate the option group

- Use one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier myssasinstance \
--option-group-name ssas-se-2017 \
--apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier myssasinstance ^
--option-group-name ssas-se-2017 ^
--apply-immediately
```

Allowing inbound access to your VPC security group

Create an inbound rule for the specified SSAS listener port in the VPC security group associated with your DB instance. For more information about setting up security groups, see [Provide access to your DB instance in your VPC by creating a security group \(p. 70\)](#).

Enabling S3 integration

To download model configuration files to your host for deployment, use S3 integration. For more information, see [Integrating an Amazon RDS for SQL Server DB instance with Amazon S3 \(p. 721\)](#).

Deploying SSAS projects on Amazon RDS

On RDS, you can't deploy SSAS projects directly by using SQL Server Management Studio (SSMS). To deploy projects, use an RDS stored procedure.

Note

Using .xmla files for deployment isn't supported.

Before you deploy projects, make sure of the following:

- S3 integration is enabled. For more information, see [Integrating an Amazon RDS for SQL Server DB instance with Amazon S3 \(p. 721\)](#).

- The Processing Option configuration setting is set to Do Not Process. This setting means that no processing happens after deployment.
- You have both the `myssasproject.asdatabase` and `myssasproject.deploymentoptions` files. They're automatically generated when you build the SSAS project.

To deploy an SSAS project on RDS

1. Download the .asdatabase (SSAS model) file from your S3 bucket to your DB instance, as shown in the following example. For more information on the download parameters, see [Downloading files from an Amazon S3 bucket to a SQL Server DB instance \(p. 727\)](#).

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/myssasproject.asdatabase',
[@rds_file_path='D:\S3\imyssasproject.asdatabase'],
[@overwrite_file=1];
```

2. Download the .deploymentoptions file from your S3 bucket to your DB instance.

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/myssasproject.deploymentoptions',
[@rds_file_path='D:\S3\imyssasproject.deploymentoptions'],
[@overwrite_file=1];
```

3. Deploy the project.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_DEPLOY_PROJECT',
@file_path='D:\S3\imyssasproject.asdatabase';
```

Monitoring the status of a deployment task

To track the status of your deployment (or download) task, call the `rds_fn_task_status` function. It takes two parameters. The first parameter should always be `NULL` because it doesn't apply to SSAS. The second parameter accepts a task ID.

To see a list of all tasks, set the first parameter to `NULL` and the second parameter to `0`, as shown in the following example.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

To get a specific task, set the first parameter to `NULL` and the second parameter to the task ID, as shown in the following example.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

The `rds_fn_task_status` function returns the following information.

Output parameter	Description
<code>task_id</code>	The ID of the task.
<code>task_type</code>	For SSAS, tasks can have the following task types: <ul style="list-style-type: none"> • <code>SSAS_DEPLOY_PROJECT</code> • <code>SSAS_ADD_DB_ADMIN_MEMBER</code>

Output parameter	Description
	<ul style="list-style-type: none"> • SSAS_BACKUP_DB • SSAS_RESTORE_DB
database_name	Not applicable to SSAS tasks.
% complete	The progress of the task as a percentage.
duration (mins)	The amount of time spent on the task, in minutes.
lifecycle	<p>The status of the task. Possible statuses are the following:</p> <ul style="list-style-type: none"> • CREATED – After you call one of the SSAS stored procedures, a task is created and the status is set to CREATED. • IN_PROGRESS – After a task starts, the status is set to IN_PROGRESS. It can take up to five minutes for the status to change from CREATED to IN_PROGRESS. • SUCCESS – After a task completes, the status is set to SUCCESS. • ERROR – If a task fails, the status is set to ERROR. For more information about the error, see the task_info column. • CANCEL_REQUESTED – After you call rds_cancel_task, the status of the task is set to CANCEL_REQUESTED. • CANCELLED – After a task is successfully canceled, the status of the task is set to CANCELLED.
task_info	Additional information about the task. If an error occurs during processing, this column contains information about the error.
last_updated	The date and time that the task status was last updated.
created_at	The date and time that the task was created.
S3_object_arn	Not applicable to SSAS tasks.
overwrite_S3_backup_file	Not applicable to SSAS tasks.
KMS_master_key_arn	Not applicable to SSAS tasks.
filepath	Not applicable to SSAS tasks.
overwrite_file	Not applicable to SSAS tasks.
task_metadata	Metadata associated with the SSAS task.

Using SSAS on Amazon RDS

After deploying the SSAS project, you can directly process the OLAP database on SSMS.

To use SSAS on RDS

1. In SSMS, connect to SSAS using the user name and password for the Active Directory domain.
2. Expand **Databases**. The newly deployed SSAS database appears.
3. Expand **Connections**, open the context (right-click) menu for the connection object, and then choose **Properties**.
4. In the connection string, update the user name and password to those for the source SQL database. Doing this is required for processing tables.
5. Open the context (right-click) menu for the SSAS database that you created and choose **Process Database**.

Depending on the size of the input data, the processing operation might take several minutes to complete.

Adding a domain user as a database administrator

You can add a domain user as an SSAS database administrator in the following ways:

- A database administrator can use SSMS to create a role with `admin` privileges, then add users to that role.
- You can use the following stored procedure.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_ADD_DB_ADMIN_MEMBER',
@database_name='myssasdb',
@ssas_role_name='exampleRole',
@ssas_role_member='domain_name\domain_user_name';
```

The following parameters are required:

- `@task_type` – The type of the MSBI task, in this case `SSAS_ADD_DB_ADMIN_MEMBER`.
- `@database_name` – The name of the SSAS database to which you're granting administrator privileges.
- `@ssas_role_name` – The SSAS database administrator role name. If the role doesn't already exist, it's created.
- `@ssas_role_member` – The SSAS database user that you're adding to the administrator role.

Backing up an SSAS database

You can create SSAS database backup files only in the `D:\S3` folder on the DB instance. To move the backup files to your S3 bucket, use Amazon S3.

You can back up an SSAS database as follows:

- A domain user with the `admin` role for a particular database can use SSMS to back up the database to the `D:\S3` folder.

For more information, see [Adding a domain user as a database administrator \(p. 770\)](#).

- You can use the following stored procedure.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_BACKUP_DB',
@database_name='myssasdb',
@file_path='D:\S3\ssas_db_backup.abf',
[@ssas_apply_compression=1],
```

```
[@ssas_overwrite_file=1];
```

The following parameters are required:

- @task_type – The type of the MSBI task, in this case SSAS_BACKUP_DB.
- @database_name – The name of the SSAS database that you're backing up.
- @file_path – The path for the SSAS backup file. The .abf extension is required.

The following parameters are optional:

- @ssas_apply_compression – Whether to apply SSAS backup compression. Valid values are 1 (Yes) and 0 (No).
- @ssas_overwrite_file – Whether to overwrite the SSAS backup file. Valid values are 1 (Yes) and 0 (No).

Note

The stored procedure for backup doesn't support encryption.

Restoring an SSAS database

Use the following stored procedure to restore an SSAS database from a backup.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_RESTORE_DB',
@database_name='mynewssasdb',
@file_path='D:\S3\ssas_db_backup.abf';
```

The following parameters are required:

- @task_type – The type of the MSBI task, in this case SSAS_RESTORE_DB.
- @database_name – The name of the new SSAS database that you're restoring to.
- @file_path – The path to the SSAS backup file.

Note

You can't restore a database if there is an existing SSAS database with the same name. The stored procedure for restoring doesn't support encrypted backup files.

Restoring a DB instance to a specified time

Point-in-time recovery (PITR) doesn't apply to SSAS databases. If you do PITR, only the SSAS data in the last snapshot before the requested time is available on the restored instance.

To have up-to-date SSAS databases on a restored DB instance

1. Back up your SSAS databases to the D:\S3 folder on the source instance.
2. Transfer the backup files to the S3 bucket.
3. Transfer the backup files from the S3 bucket to the D:\S3 folder on the restored instance.
4. Run the stored procedure to restore the SSAS databases onto the restored instance.

Note

You can also reprocess the SSAS project to restore the databases.

Disabling SSAS

To disable SSAS, remove the SSAS option from its option group. Before you remove the SSAS option, delete your SSAS databases.

Important

We highly recommend that you back up your SSAS databases before deleting them and removing the SSAS option.

Console

To remove the SSAS option from its option group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose the option group with the SSAS option (ssas-se-2017 in the previous examples).
4. Choose **Delete option**.
5. Under **Deletion options**, choose **SSAS** for **Options to delete**.
6. Under **Apply immediately**, choose **Yes** to delete the option immediately, or **No** to delete it at the next maintenance window.
7. Choose **Delete**.

CLI

To remove the SSAS option from its option group

- Use one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssas-se-2017 \
    --options SSAS \
    --apply-immediately
```

For Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssas-se-2017 ^
    --options SSAS ^
    --apply-immediately
```

Support for SQL Server Integration Services in Amazon RDS for SQL Server

Microsoft SQL Server Integration Services (SSIS) is a component that you can use to perform a broad range of data migration tasks. SSIS is a platform for data integration and workflow applications. It features a data warehousing tool used for data extraction, transformation, and loading (ETL). You can also use this tool to automate maintenance of SQL Server databases and updates to multidimensional cube data.

SSIS projects are organized into packages saved as XML-based .dtsx files. Packages can contain control flows and data flows. You use data flows to represent ETL operations. After deployment, packages are stored in SQL Server in the SSISDB database. SSISDB is an online transaction processing (OLTP) database in the full recovery mode.

Amazon RDS for SQL Server supports running SSIS directly on an RDS DB instance. You can enable SSIS on an existing or new DB instance. SSIS is installed on the same DB instance as your database engine.

RDS supports SSIS for SQL Server Standard and Enterprise Editions on the following versions:

- SQL Server 2019, version 15.00.4043.16.v1 and later
- SQL Server 2017, version 14.00.3223.3.v1 and later
- SQL Server 2016, version 13.00.5426.0.v1 and later

Limitations and recommendations

The following limitations and recommendations apply to running SSIS on RDS for SQL Server:

- The DB instance must use AWS Managed Microsoft AD for SSIS authentication.
- The DB instance must have an associated parameter group with the `clr_enabled` parameter set to 1. For more information, see [Modifying the parameter for SSIS \(p. 777\)](#).

Note

If you enable the `clr_enabled` parameter on SQL Server 2017, you can't use the common language runtime (CLR) on your DB instance.

- The following control flow tasks are supported:
 - Analysis Services Execute DDL Task
 - Analysis Services Processing Task
 - Bulk Insert Task
 - Check Database Integrity Task
 - Data Flow Task
 - Data Mining Query Task
 - Data Profiling Task
 - Execute Package Task
 - Execute SQL Server Agent Job Task
 - Execute SQL Task
 - Execute T-SQL Statement Task
 - Notify Operator Task
 - Rebuild Index Task
 - Reorganize Index Task
 - Shrink Database Task
 - Transfer Database Task

- Transfer Jobs Task
- Transfer Logins Task
- Transfer SQL Server Objects Task
- Update Statistics Task
- Only project deployment is supported.
- Running SSIS packages by using SQL Server Agent is supported.
- Only SQL Server-based logging is supported.
- Use only the D:\S3 folder for working with files. Files placed in any other directory are deleted. Be aware of a few other file location details:
 - Place SSIS project input and output files in the D:\S3 folder.
 - For the Data Flow Task, change the location for BLOBTempStoragePath and BufferTempStoragePath to a file inside the D:\S3 folder.
 - Ensure that all parameters, variables, and expressions used for file connections point to the D:\S3 folder.
 - On Multi-AZ instances, files created by SSIS in the D:\S3 folder are deleted after a failover. For more information, see [Multi-AZ limitations for S3 integration \(p. 731\)](#).
 - Upload the files created by SSIS in the D:\S3 folder to your Amazon S3 bucket to make them durable.
- Import Column and Export Column transformations and the Script component on the Data Flow Task aren't supported.
- You can't enable dump on running SSIS packages, and you can't add data taps on SSIS packages.
- The SSIS Scale Out feature isn't supported.
- You can't deploy projects directly. We provide RDS stored procedures to do this. For more information, see [Deploying an SSIS project \(p. 781\)](#).
- Build SSIS project (.ispac) files with the DoNotSavePasswords protection mode for deploying on RDS.
- SSIS isn't supported on Always On instances with read replicas.
- You can't back up the SSISDB database that is associated with the SSIS option.
- Importing and restoring the SSISDB database from other instances of SSIS isn't supported.

Enabling SSIS

You enable SSIS by adding the SSIS option to your DB instance. Use the following process:

1. Create a new option group, or choose an existing option group.
2. Add the SSIS option to the option group.
3. Create a new parameter group, or choose an existing parameter group.
4. Modify the parameter group to set the clr_enabled parameter to 1.
5. Associate the option group and parameter group with the DB instance.
6. Enable Amazon S3 integration.

Note

If a database with the name SSISDB or a reserved SSIS login already exists on the DB instance, you can't enable SSIS on the instance.

Creating the option group for SSIS

To work with SSIS, create an option group or modify an option group that corresponds to the SQL Server edition and version of the DB instance that you plan to use. To do this, use the AWS Management Console or the AWS CLI.

Console

The following procedure creates an option group for SQL Server Standard Edition 2016.

To create the option group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose **Create group**.
4. In the **Create option group** window, do the following:
 - a. For **Name**, enter a name for the option group that is unique within your AWS account, such as **ssis-se-2016**. The name can contain only letters, digits, and hyphens.
 - b. For **Description**, enter a brief description of the option group, such as **SSIS option group for SQL Server SE 2016**. The description is used for display purposes.
 - c. For **Engine**, choose **sqlserver-se**.
 - d. For **Major engine version**, choose **13.00**.
5. Choose **Create**.

CLI

The following procedure creates an option group for SQL Server Standard Edition 2016.

To create the option group

- Run one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds create-option-group \
    --option-group-name ssis-se-2016 \
    --engine-name sqlserver-se \
    --major-engine-version 13.00 \
    --option-group-description "SSIS option group for SQL Server SE 2016"
```

For Windows:

```
aws rds create-option-group ^
    --option-group-name ssis-se-2016 ^
    --engine-name sqlserver-se ^
    --major-engine-version 13.00 ^
    --option-group-description "SSIS option group for SQL Server SE 2016"
```

Adding the SSIS option to the option group

Next, use the AWS Management Console or the AWS CLI to add the **ssis** option to your option group.

Console

To add the SSIS option

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.

2. In the navigation pane, choose **Option groups**.
3. Choose the option group that you just created, **ssis-se-2016** in this example.
4. Choose **Add option**.
5. Under **Option details**, choose **SSIS** for **Option name**.
6. Under **Scheduling**, choose whether to add the option immediately or at the next maintenance window.
7. Choose **Add option**.

CLI

To add the SSIS option

- Add the SSIS option to the option group.

Example

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \
--option-group-name ssis-se-2016 \
--options OptionName=SSIS \
--apply-immediately
```

For Windows:

```
aws rds add-option-to-option-group ^
--option-group-name ssis-se-2016 ^
--options OptionName=SSIS ^
--apply-immediately
```

Creating the parameter group for SSIS

Create or modify a parameter group for the `clr` enabled parameter that corresponds to the SQL Server edition and version of the DB instance that you plan to use for SSIS.

Console

The following procedure creates a parameter group for SQL Server Standard Edition 2016.

To create the parameter group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. Choose **Create parameter group**.
4. In the **Create parameter group** pane, do the following:
 - a. For **Parameter group family**, choose **sqlserver-se-13.0**.
 - b. For **Group name**, enter an identifier for the parameter group, such as **ssis-sqlserver-se-13**.
 - c. For **Description**, enter **clr enabled parameter group**.
5. Choose **Create**.

CLI

The following procedure creates a parameter group for SQL Server Standard Edition 2016.

To create the parameter group

- Run one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-parameter-group \
--db-parameter-group-name ssis-sqlserver-se-13 \
--db-parameter-group-family "sqlserver-se-13.0" \
--description "clr enabled parameter group"
```

For Windows:

```
aws rds create-db-parameter-group ^
--db-parameter-group-name ssis-sqlserver-se-13 ^
--db-parameter-group-family "sqlserver-se-13.0" ^
--description "clr enabled parameter group"
```

Modifying the parameter for SSIS

Modify the `clr enabled` parameter in the parameter group that corresponds to the SQL Server edition and version of your DB instance. For SSIS, set the `clr enabled` parameter to 1.

Console

The following procedure modifies the parameter group that you created for SQL Server Standard Edition 2016.

To modify the parameter group

- Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
- In the navigation pane, choose **Parameter groups**.
- Choose the parameter group, such as **ssis-sqlserver-se-13**.
- Under **Parameters**, filter the parameter list for `clr`.
- Choose `clr enabled`.
- Choose **Edit parameters**.
- From **Values**, choose **1**.
- Choose **Save changes**.

CLI

The following procedure modifies the parameter group that you created for SQL Server Standard Edition 2016.

To modify the parameter group

- Run one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name ssis-sqlserver-se-13 \
--parameters "ParameterName='clr enabled',ParameterValue=1,ApplyMethod=immediate"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name ssis-sqlserver-se-13 ^
--parameters "ParameterName='clr enabled',ParameterValue=1,ApplyMethod=immediate"
```

Associating the option group and parameter group with your DB instance

To associate the SSIS option group and parameter group with your DB instance, use the AWS Management Console or the AWS CLI

Note

If you use an existing instance, it must already have an Active Directory domain and AWS Identity and Access Management (IAM) role associated with it. If you create a new instance, specify an existing Active Directory domain and IAM role. For more information, see [Using Windows Authentication with an Amazon RDS for SQL Server DB instance \(p. 711\)](#).

Console

To finish enabling SSIS, associate your SSIS option group and parameter group with a new or existing DB instance:

- For a new DB instance, associate them when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- For an existing DB instance, associate them by modifying the instance. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

CLI

You can associate the SSIS option group and parameter group with a new or existing DB instance.

To create an instance with the SSIS option group and parameter group

- Specify the same DB engine type and major version as you used when creating the option group.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
--db-instance-identifier myssisisinstance \
--db-instance-class db.m5.2xlarge \
--engine sqlserver-se \
--engine-version 13.00.5426.0.v1 \
--allocated-storage 100 \
--master-user-password secret123 \
--master-username admin \
--storage-type gp2 \
--license-model li \
```

```
--domain-iam-role-name my-directory-iam-role \
--domain my-domain-id \
--option-group-name ssis-se-2016 \
--db-parameter-group-name ssis-sqlserver-se-13
```

For Windows:

```
aws rds create-db-instance ^
--db-instance-identifier myssisisinstance ^
--db-instance-class db.m5.2xlarge ^
--engine sqlserver-se ^
--engine-version 13.00.5426.0.v1 ^
--allocated-storage 100 ^
--master-user-password secret123 ^
--master-username admin ^
--storage-type gp2 ^
--license-model li ^
--domain-iam-role-name my-directory-iam-role ^
--domain my-domain-id ^
--option-group-name ssis-se-2016 ^
--db-parameter-group-name ssis-sqlserver-se-13
```

To modify an instance and associate the SSIS option group and parameter group

- Run one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier myssisisinstance \
--option-group-name ssis-se-2016 \
--db-parameter-group-name ssis-sqlserver-se-13 \
--apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier myssisisinstance ^
--option-group-name ssis-se-2016 ^
--db-parameter-group-name ssis-sqlserver-se-13 ^
--apply-immediately
```

Enabling S3 integration

To download SSIS project (.ispac) files to your host for deployment, use S3 file integration. For more information, see [Integrating an Amazon RDS for SQL Server DB instance with Amazon S3 \(p. 721\)](#).

Administrative permissions on SSISDB

When the instance is created or modified with the SSIS option, the result is an SSISDB database with the ssis_admin and ssis_logreader roles granted to the master user. The master user has the following privileges in SSISDB:

- alter on ssis_admin role

- alter on ssis_logreader role
- alter any user

Because the master user is a SQL-authenticated user, you can't use the master user for executing SSIS packages. The master user can use these privileges to create new SSISDB users and add them to the ssis_admin and ssis_logreader roles. Doing this is useful for giving access to your domain users for using SSIS.

Setting up a Windows-authenticated user for SSIS

The master user can use the following code example to set up a Windows-authenticated login in SSISDB and grant the required procedure permissions. Doing this grants permissions to the domain user to deploy and run SSIS packages, use S3 file transfer procedures, create credentials, and work with the SQL Server Agent proxy. For more information, see [Credentials \(database engine\)](#) and [Create a SQL Server Agent proxy](#) in the Microsoft documentation.

Note

You can grant some or all of the following permissions as needed to Windows-authenticated users.

Example

```
USE [SSISDB]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]
ALTER ROLE [ssis_admin] ADD MEMBER [mydomain\user_name]
ALTER ROLE [ssis_logreader] ADD MEMBER [mydomain\user_name]
GO

USE [msdb]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]
GRANT EXEC ON msdb.dbo.rds_msbi_task TO [mydomain\user_name] with grant option
GRANT SELECT ON msdb.dbo.rds_fn_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_cancel_task TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_download_from_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_upload_to_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_delete_from_filesystem TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_gather_file_details TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_add_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_update_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_grant_login_to_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_revoke_login_from_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_delete_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_enum_login_for_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_enum_proxy_for_subsystem TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_sqlagent_proxy TO [mydomain\user_name] WITH GRANT OPTION
ALTER ROLE [SQLAgentUserRole] ADD MEMBER [mydomain\user_name]
GO

USE [master]
GO
GRANT ALTER ANY CREDENTIAL TO [mydomain\user_name]
GO
```

Deploying an SSIS project

On RDS, you can't deploy SSIS projects directly by using SQL Server Management Studio (SSMS) or SSIS procedures. To download project files from Amazon S3 and then deploy them, use RDS stored procedures.

To run the stored procedures, log in as any user that you granted permissions for running the stored procedures. For more information, see [Setting up a Windows-authenticated user for SSIS \(p. 780\)](#).

To deploy the SSIS project

1. Download the project (.ispac) file.

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/ssisproject.ispac',
[@rds_file_path='D:\S3\ssisproject.ispac'],
[@overwrite_file=1];
```

2. Submit the deployment task, making sure of the following:

- The folder is present in the SSIS catalog.
- The project name matches the project name that you used while developing the SSIS project.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSIS_DEPLOY_PROJECT',
@folder_name='DEMO',
@project_name='ssisproject',
@file_path='D:\S3\ssisproject.ispac';
```

Monitoring the status of a deployment task

To track the status of your deployment task, call the `rds_fn_task_status` function. It takes two parameters. The first parameter should always be `NULL` because it doesn't apply to SSIS. The second parameter accepts a task ID.

To see a list of all tasks, set the first parameter to `NULL` and the second parameter to 0, as shown in the following example.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

To get a specific task, set the first parameter to `NULL` and the second parameter to the task ID, as shown in the following example.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

The `rds_fn_task_status` function returns the following information.

Output parameter	Description
<code>task_id</code>	The ID of the task.
<code>task_type</code>	<code>SSIS_DEPLOY_PROJECT</code>

Output parameter	Description
database_name	Not applicable to SSIS tasks.
% complete	The progress of the task as a percentage.
duration (mins)	The amount of time spent on the task, in minutes.
lifecycle	<p>The status of the task. Possible statuses are the following:</p> <ul style="list-style-type: none"> • CREATED – After you call the <code>msdb.dbo.rds_msbi_task</code> stored procedure, a task is created and the status is set to CREATED. • IN_PROGRESS – After a task starts, the status is set to IN_PROGRESS. It can take up to five minutes for the status to change from CREATED to IN_PROGRESS. • SUCCESS – After a task completes, the status is set to SUCCESS. • ERROR – If a task fails, the status is set to ERROR. For more information about the error, see the <code>task_info</code> column. • CANCEL_REQUESTED – After you call <code>rds_cancel_task</code>, the status of the task is set to CANCEL_REQUESTED. • CANCELLED – After a task is successfully canceled, the status of the task is set to CANCELLED.
task_info	Additional information about the task. If an error occurs during processing, this column contains information about the error.
last_updated	The date and time that the task status was last updated.
created_at	The date and time that the task was created.
S3_object_arn	Not applicable to SSIS tasks.
overwrite_S3_backup_file	Not applicable to SSIS tasks.
KMS_master_key_arn	Not applicable to SSIS tasks.
filepath	Not applicable to SSIS tasks.
overwrite_file	Not applicable to SSIS tasks.
task_metadata	Metadata associated with the SSIS task.

Using SSIS

After deploying the SSIS project into the SSIS catalog, you can run packages directly from SSMS or schedule them by using SQL Server Agent. You must use a Windows-authenticated login for executing SSIS packages. For more information, see [Setting up a Windows-authenticated user for SSIS \(p. 780\)](#).

Setting database connection managers for SSIS projects

When you use a connection manager, you can use these types of authentication:

- For local database connections, you can use SQL authentication or Windows authentication. For Windows authentication, use `DB_instance_name.fully_qualified_domain_name` as the server name of the connection string.

An example is `myssisinstance.corp-ad.example.com`, where `myssisinstance` is the DB instance name and `corp-ad.example.com` is the fully qualified domain name.

- For remote connections, always use SQL authentication.

Creating an SSIS proxy

To be able to schedule SSIS packages using SQL Server Agent, create an SSIS credential and an SSIS proxy. Run these procedures as a Windows-authenticated user.

To create the SSIS credential

- Create the credential for the proxy. To do this, you can use SSMS or the following SQL statement.

```
USE [master]
GO
CREATE CREDENTIAL [SSIS_Credential] WITH IDENTITY = N'mydomain\user_name', SECRET =
N'mysecret'
GO
```

Note

`IDENTITY` must be a domain-authenticated login. Replace `mysecret` with the password for the domain-authenticated login.

Whenever the SSISDB primary host is changed, alter the SSIS proxy credentials to allow the new host to access them.

To create the SSIS proxy

1. Use the following SQL statement to create the proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_proxy
    @proxy_name=N'SSIS_Proxy',@credential_name=N'SSIS_Credential',@description=N''
GO
```

2. Use the following SQL statement to grant access to the proxy to other users.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_grant_login_to_proxy
    @proxy_name=N'SSIS_Proxy',@login_name=N'mydomain\user_name'
GO
```

3. Use the following SQL statement to give the SSIS subsystem access to the proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
    @task_type='GRANT_SUBSYSTEM_ACCESS',@proxy_name='SSIS_Proxy',@proxy_subsystem='SSIS'
```

GO

To view the proxy and grants on the proxy

1. Use the following SQL statement to view the grantees of the proxy.

```
USE [msdb]
GO
EXEC sp_help_proxy
GO
```

2. Use the following SQL statement to view the subsystem grants.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_enum_proxy_for_subsystem
GO
```

Scheduling an SSIS package using SQL Server Agent

After you create the credential and proxy and grant SSIS access to the proxy, you can create a SQL Server Agent job to schedule the SSIS package.

To schedule the SSIS package

- You can use SSMS or T-SQL for creating the SQL Server Agent job. The following example uses T-SQL.

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'MYSSISJob',
@enabled=1,
@notify_level_eventlog=0,
@notify_level_email=2,
@notify_level_page=2,
@delete_level=0,
@category_name=N'[Uncategorized (Local)]',
@job_id = @jobId OUTPUT
GO
EXEC msdb.dbo.sp_add_jobserver @job_name=N'MYSSISJob',@server_name=N'(local)'
GO
EXEC msdb.dbo.sp_add_jobstep @job_name=N'MYSSISJob',@step_name=N'ExecuteSSISPackage',
@step_id=1,
@cmdexec_success_code=0,
@on_success_action=1,
@on_fail_action=2,
@retry_attempts=0,
@retry_interval=0,
@os_run_priority=0,
@subsystem=N'SSIS',
@command=N'/ISSERVER "\"\\SSISDB\\MySSISFolder\\MySSISProject\\MySSISPackage.dtsx\" /'
SERVER "\"my-rds-ssis-instance.corp-ad.company.com\/\""
/Par "\"$ServerOption::LOGGING_LEVEL(Int16)\";1 /Par
"\"$ServerOption::SYNCHRONIZED(Boolean)\"";True /CALLERINFO SQLAGENT /REPORTING E',
@database_name=N'master',
@flags=0,
@proxy_name=N'SSIS_Proxy'
GO
```

Revoking SSIS access from the proxy

You can revoke access to the SSIS subsystem and delete the SSIS proxy using the following stored procedures.

To revoke access and delete the proxy

1. Revoke subsystem access.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
    @task_type='REVOKE_SUBSYSTEM_ACCESS',@proxy_name='SSIS_Proxy',@proxy_subsystem='SSIS'
GO
```

2. Revoke the grants on the proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_revoke_login_from_proxy
    @proxy_name=N'SSIS_Proxy',@name=N'mydomain\user_name'
GO
```

3. Delete the proxy.

```
USE [msdb]
GO
EXEC dbo.sp_delete_proxy @proxy_name = N'SSIS_Proxy'
GO
```

Disabling SSIS

To disable SSIS, remove the `SSIS` option from its option group.

Important

Removing the option doesn't delete the SSISDB database, so you can safely remove the option without losing the SSIS projects.

You can re-enable the `SSIS` option after removal to reuse the SSIS projects that were previously deployed to the SSIS catalog.

Console

The following procedure removes the `SSIS` option.

To remove the SSIS option from its option group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose the option group with the `SSIS` option (`ssis-se-2016` in the previous examples).
4. Choose **Delete option**.
5. Under **Deletion options**, choose **SSIS** for **Options to delete**.
6. Under **Apply immediately**, choose **Yes** to delete the option immediately, or **No** to delete it at the next maintenance window.
7. Choose **Delete**.

CLI

The following procedure removes the SSIS option.

To remove the SSIS option from its option group

- Run one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssis-se-2016 \
    --options SSIS \
    --apply-immediately
```

For Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssis-se-2016 ^
    --options SSIS ^
    --apply-immediately
```

Dropping the SSISDB database

After removing the SSIS option, the SSISDB database isn't deleted. To drop the SSISDB database, use the `rds_drop_ssdb_database` stored procedure after removing the SSIS option.

To drop the SSIS database

- Use the following stored procedure.

```
USE [msdb]
GO
EXEC dbo.rds_drop_ssdb_database
GO
```

After dropping the SSISDB database, if you re-enable the SSIS option you get a fresh SSISDB catalog.

Support for SQL Server Reporting Services in Amazon RDS for SQL Server

Microsoft SQL Server Reporting Services (SSRS) is a server-based application used for report generation and distribution. It's part of a suite of SQL Server services that also includes SQL Server Analysis Services (SSAS) and SQL Server Integration Services (SSIS). SSRS is a service built on top of SQL Server. You can use it to collect data from various data sources and present it in a way that's easily understandable and ready for analysis.

Amazon RDS for SQL Server supports running SSRS directly on RDS DB instances. You can enable SSRS for existing or new DB instances.

RDS supports SSRS for SQL Server Standard and Enterprise Editions on the following versions:

- SQL Server 2019, version 15.00.4043.16.v1 and later
- SQL Server 2017, version 14.00.3223.3.v1 and later
- SQL Server 2016, version 13.00.5820.21.v1 and later

Limitations and recommendations

The following limitations and recommendations apply to running SSRS on RDS for SQL Server:

- Instances must use AWS Managed Microsoft AD for SSRS web portal and web server authentication.
- Importing and restoring report server databases from other instances of SSRS isn't supported.

Make sure to use the databases that are created when the `SSRS` option is added to the RDS DB instance. For more information, see [Report server databases \(p. 791\)](#).

- You can't configure SSRS to listen on the default SSL port (443). The allowed values are 1150–49511, except 1234, 1434, 3260, 3343, 3389, and 47001.
- Subscriptions through email or a Microsoft Windows file share aren't supported.
- Using Reporting Services Configuration Manager isn't supported.
- Creating and modifying roles isn't supported.
- Modifying report server properties isn't supported.
- System administrator and system user roles aren't granted.
- You can't edit system-level role assignments through the web portal.

Enabling SSRS

Use the following process to enable SSRS on your DB instance:

1. Create a new option group, or choose an existing option group.
2. Add the `SSRS` option to the option group.
3. Associate the option group with the DB instance.
4. Allow inbound access to the virtual private cloud (VPC) security group for the SSRS listener port.

Creating an option group for SSRS

To work with SSRS, create an option group that corresponds to the SQL Server engine and version of the DB instance that you plan to use. To do this, use the AWS Management Console or the AWS CLI.

Note

You can also use an existing option group if it's for the correct SQL Server engine and version.

Console

The following procedure creates an option group for SQL Server Standard Edition 2017.

To create the option group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose **Create group**.
4. In the **Create option group** pane, do the following:
 - a. For **Name**, enter a name for the option group that is unique within your AWS account, such as **ssrs-se-2017**. The name can contain only letters, digits, and hyphens.
 - b. For **Description**, enter a brief description of the option group, such as **SSRS option group for SQL Server SE 2017**. The description is used for display purposes.
 - c. For **Engine**, choose **sqlserver-se**.
 - d. For **Major engine version**, choose **14.00**.
5. Choose **Create**.

CLI

The following procedure creates an option group for SQL Server Standard Edition 2017.

To create the option group

- Run one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds create-option-group \
--option-group-name ssrs-se-2017 \
--engine-name sqlserver-se \
--major-engine-version 14.00 \
--option-group-description "SSRS option group for SQL Server SE 2017"
```

For Windows:

```
aws rds create-option-group ^
--option-group-name ssrs-se-2017 ^
--engine-name sqlserver-se ^
--major-engine-version 14.00 ^
--option-group-description "SSRS option group for SQL Server SE 2017"
```

Adding the SSRS option to your option group

Next, use the AWS Management Console or the AWS CLI to add the SSRS option to your option group.

Console

To add the SSRS option

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose the option group that you just created.
4. Choose **Add option**.
5. Under **Option details**, choose **SSRS** for **Option name**.
6. Under **Option settings**, do the following:
 - a. Enter the port for the SSRS service to listen on. The default is 8443. For a list of allowed values, see [Limitations and recommendations \(p. 787\)](#).
 - b. Enter a value for **Max memory**.

Max memory specifies the upper threshold above which no new memory allocation requests are granted to report server applications. The number is a percentage of the total memory of the DB instance. The allowed values are 10–80.

7. For **Security groups**, choose the VPC security group to associate with the option. Use the same security group that is associated with your DB instance.
8. Under **Scheduling**, choose whether to add the option immediately or at the next maintenance window.
9. Choose **Add option**.

CLI

To add the SSRS option

1. Create a JSON file, for example `ssrs-option.json`, with the following parameters:
 - `OptionGroupName` – The name of option group that you created or chose previously (`ssrs-se-2017` in the following example).
 - `Port` – The port for the SSRS service to listen on. The default is 8443. For a list of allowed values, see [Limitations and recommendations \(p. 787\)](#).
 - `VpcSecurityGroupMemberships` – VPC security group memberships for your RDS DB instance.
 - `MAX_MEMORY` – The upper threshold above which no new memory allocation requests are granted to report server applications. The number is a percentage of the total memory of the DB instance. The allowed values are 10–80.

```
{  
  "OptionGroupName": "ssrs-se-2017",  
  "OptionsToInclude": [  
    {  
      "OptionName": "SSRS",  
      "Port": 8443,  
      "VpcSecurityGroupMemberships": ["sg-0abcdef123"],  
      "OptionSettings": [{"Name": "MAX_MEMORY", "Value": "60"}]  
    }],  
  "ApplyImmediately": true  
}
```

2. Add the SSRS option to the option group.

Example

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \
    --cli-input-json file://ssrs-option.json \
    --apply-immediately
```

For Windows:

```
aws rds add-option-to-option-group ^
    --cli-input-json file://ssrs-option.json ^
    --apply-immediately
```

Associating your option group with your DB instance

Use the AWS Management Console or the AWS CLI to associate your option group with your DB instance.

If you use an existing DB instance, it must already have an Active Directory domain and AWS Identity and Access Management (IAM) role associated with it. If you create a new instance, specify an existing Active Directory domain and IAM role. For more information, see [Using Windows Authentication with an Amazon RDS for SQL Server DB instance \(p. 711\)](#).

Console

You can associate your option group with a new or existing DB instance:

- For a new DB instance, associate the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- For an existing DB instance, modify the instance and associate the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

CLI

You can associate your option group with a new or existing DB instance.

To create a DB instance that uses your option group

- Specify the same DB engine type and major version as you used when creating the option group.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
    --db-instance-identifier myssrsinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 14.00.3223.3.v1 \
    --allocated-storage 100 \
    --master-user-password secret123 \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
```

```
--option-group-name ssrs-se-2017
```

For Windows:

```
aws rds create-db-instance ^
--db-instance-identifier myssrsinstance ^
--db-instance-class db.m5.2xlarge ^
--engine sqlserver-se ^
--engine-version 14.00.3223.3.v1 ^
--allocated-storage 100 ^
--master-user-password secret123 ^
--master-username admin ^
--storage-type gp2 ^
--license-model li ^
--domain-iam-role-name my-directory-iam-role ^
--domain my-domain-id ^
--option-group-name ssrs-se-2017
```

To modify a DB instance to use your option group

- Run one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier myssrsinstance \
--option-group-name ssrs-se-2017 \
--apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier myssrsinstance ^
--option-group-name ssrs-se-2017 ^
--apply-immediately
```

Allowing inbound access to your VPC security group

To allow inbound access to the VPC security group associated with your DB instance, create an inbound rule for the specified SSRS listener port. For more information about setting up security groups, see [Provide access to your DB instance in your VPC by creating a security group \(p. 70\)](#).

Report server databases

When your DB instance is associated with the SSRS option, two new databases are created on your DB instance: rdsadmin_ReportServer and rdsadmin_ReportServerTempDB. These databases act as the ReportServer and ReportServerTempDB databases. SSRS stores its data in the ReportServer database and caches its data in the ReportServerTempDB database. RDS owns and manages these databases, so database operations on them such as ALTER and DROP aren't permitted.

Accessing the SSRS web portal

Use the following process to access the SSRS web portal:

1. Enable Secure Sockets Layer (SSL).
2. Grant access to domain users.
3. Access the web portal using a browser and the domain user credentials.

Enabling SSL on RDS

SSRS uses the HTTPS SSL protocol for its connections. To work with this protocol, import an SSL certificate into the Microsoft Windows operating system on your client computer.

For more information on SSL certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#). For more information about using SSL with SQL Server, see [Using SSL with a Microsoft SQL Server DB instance \(p. 704\)](#).

Granting access to domain users

In a new SSRS activation, there are no role assignments in SSRS. To give a domain user or user group access to the web portal, RDS provides a stored procedure.

To grant access to a domain user on the web portal

- Use the following stored procedure.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSRS_GRANT_PORTAL_PERMISSION',
@ssrs_group_or_username=N'AD_domain\user';
```

The domain user or user group is granted the `RDS_SSRS_ROLE` system role. This role has the following system-level tasks granted to it:

- Execute report definitions
- Manage jobs
- Manage shared schedules
- View shared schedules

The item-level role of `Content Manager` on the root folder is also granted.

Accessing the web portal

After the `SSRS_GRANT_PORTAL_PERMISSION` task finishes successfully, you have access to the portal using a web browser. The web portal URL has the following format.

```
https://rds_endpoint:port/Reports
```

In this format, the following applies:

- `rds_endpoint` – The endpoint for the RDS DB instance that you're using with SSRS.

You can find the endpoint on the **Connectivity & security** tab for your DB instance. For more information, see [Connecting to a DB instance running the Microsoft SQL Server database engine \(p. 656\)](#).

- `port` – The listener port for SSRS that you set in the `SSRS` option.

To access the web portal

1. Enter the web portal URL in your browser.

```
https://myssrsinstance.cg034itsfake.us-east-1.rds.amazonaws.com:8443/Reports
```

2. Log in with the credentials for a domain user that you granted access with the SSRS_GRANT_PORTAL_PERMISSION task.

Deploying reports to SSRS

After you have access to the web portal, you can deploy reports to it. You can use the Upload tool in the web portal to upload reports, or deploy directly from [SQL Server data tools \(SSDT\)](#). When deploying from SSDT, ensure the following:

- The user who launched SSDT has access to the SSRS web portal.
- The TargetServerURL value in the SSRS project properties is set to the HTTPS endpoint of the RDS DB instance suffixed with ReportServer, for example:

```
https://myssrsinstance.cg034itsfake.us-east-1.rds.amazonaws.com:8443/ReportServer
```

Revoking system-level permissions

The RDS_SSRS_ROLE system role doesn't have sufficient permissions to delete system-level role assignments. To remove a user or user group from RDS_SSRS_ROLE, use the same stored procedure that you used to grant the role but use the SSRS_REVOKER_PORTAL_PERMISSION task type.

To revoke access from a domain user for the web portal

- Use the following stored procedure.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSRS_REVOKER_PORTAL_PERMISSION',
@ssrs_group_or_username=N'AD_domain\user';
```

Doing this deletes the user from the RDS_SSRS_ROLE system role. It also deletes the user from the Content Manager item-level role if the user has it.

Monitoring the status of a task

To track the status of your granting or revoking task, call the rds_fn_task_status function. It takes two parameters. The first parameter should always be `NULL` because it doesn't apply to SSRS. The second parameter accepts a task ID.

To see a list of all tasks, set the first parameter to `NULL` and the second parameter to 0, as shown in the following example.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

To get a specific task, set the first parameter to `NULL` and the second parameter to the task ID, as shown in the following example.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL, 42);
```

The `rds_fn_task_status` function returns the following information.

Output parameter	Description
<code>task_id</code>	The ID of the task.
<code>task_type</code>	For SSRS, tasks can have the following task types: <ul style="list-style-type: none"> • <code>SSRS_GRANT_PORTAL_PERMISSION</code> • <code>SSRS_REVOKE_PORTAL_PERMISSION</code>
<code>database_name</code>	Not applicable to SSRS tasks.
<code>% complete</code>	The progress of the task as a percentage.
<code>duration (mins)</code>	The amount of time spent on the task, in minutes.
<code>lifecycle</code>	The status of the task. Possible statuses are the following: <ul style="list-style-type: none"> • <code>CREATED</code> – After you call one of the SSRS stored procedures, a task is created and the status is set to <code>CREATED</code>. • <code>IN_PROGRESS</code> – After a task starts, the status is set to <code>IN_PROGRESS</code>. It can take up to five minutes for the status to change from <code>CREATED</code> to <code>IN_PROGRESS</code>. • <code>SUCCESS</code> – After a task completes, the status is set to <code>SUCCESS</code>. • <code>ERROR</code> – If a task fails, the status is set to <code>ERROR</code>. For more information about the error, see the <code>task_info</code> column. • <code>CANCEL_REQUESTED</code> – After you call the <code>rds_cancel_task</code> stored procedure, the status of the task is set to <code>CANCEL_REQUESTED</code>. • <code>CANCELLED</code> – After a task is successfully canceled, the status of the task is set to <code>CANCELLED</code>.
<code>task_info</code>	Additional information about the task. If an error occurs during processing, this column contains information about the error.
<code>last_updated</code>	The date and time that the task status was last updated.
<code>created_at</code>	The date and time that the task was created.
<code>S3_object_arn</code>	Not applicable to SSRS tasks.
<code>overwrite_S3_backup_file</code>	Not applicable to SSRS tasks.
<code>KMS_master_key_arn</code>	Not applicable to SSRS tasks.
<code>filepath</code>	Not applicable to SSRS tasks.

Output parameter	Description
overwrite_file	Not applicable to SSRS tasks.
task_metadata	Metadata associated with the SSRS task.

Disabling SSRS

To disable SSRS, remove the `SSRS` option from its option group. Removing the option doesn't delete the SSRS databases. For more information, see [Deleting the SSRS databases \(p. 796\)](#).

You can re-enable SSRS by adding back the `SSRS` option. If you have also deleted the SSRS databases, re-enabling SSRS on the same DB instance creates new report server databases.

Console

To remove the SSRS option from its option group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose the option group with the `SSRS` option (`ssrs-se-2017` in the previous examples).
4. Choose **Delete option**.
5. Under **Deletion options**, choose **SSRS** for **Options to delete**.
6. Under **Apply immediately**, choose **Yes** to delete the option immediately, or **No** to delete it at the next maintenance window.
7. Choose **Delete**.

CLI

To remove the SSRS option from its option group

- Run one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds remove-option-from-option-group \
--option-group-name ssrs-se-2017 \
--options SSRS \
--apply-immediately
```

For Windows:

```
aws rds remove-option-from-option-group ^
--option-group-name ssrs-se-2017 ^
--options SSRS ^
--apply-immediately
```

Deleting the SSRS databases

Removing the SSRS option doesn't delete the report server databases. To delete them, use the following stored procedure.

To delete the report server databases, be sure to remove the SSRS option first.

To delete the SSRS databases

- Use the following stored procedure.

```
exec msdb.dbo.rds_drop_ssrs_databases
```

Support for Microsoft Distributed Transaction Coordinator in SQL Server

A *distributed transaction* is a database transaction in which two or more network hosts are involved. Amazon RDS for SQL Server supports distributed transactions among hosts, where a single host can be one of the following:

- RDS for SQL Server DB instance
- On-premises SQL Server host
- Amazon EC2 host with SQL Server installed
- Any other EC2 host or RDS DB instance with a database engine that supports distributed transactions

In RDS, starting with SQL Server 2012 (version 11.00.5058.0.v1 and later), all editions of SQL Server support distributed transactions. The support is provided using Microsoft Distributed Transaction Coordinator (MSDTC). For in-depth information about MSDTC, see [Distributed Transaction Coordinator](#) in the Microsoft documentation.

Limitations

The following limitations apply to using MSDTC on RDS for SQL Server:

- MSDTC isn't supported on instances using SQL Server Database Mirroring. For more information, see [Transactions - availability groups and database mirroring](#).
- The `in-doubt xact resolution` parameter must be set to 1 or 2. For more information, see [Modifying the parameter for MSDTC \(p. 801\)](#).
- MSDTC requires all host names participating in distributed transactions to be resolvable using their computer names. RDS automatically maintains this functionality for domain-joined instances. However, for standalone instances make sure to configure the DNS server manually.
- Distributed transactions that depend on client dynamic link libraries (DLLs) on RDS instances aren't supported.

Enabling MSDTC

Use the following process to enable MSDTC for your DB instance:

1. Create a new option group, or choose an existing option group.
2. Add the `MSDTC` option to the option group.
3. Create a new parameter group, or choose an existing parameter group.
4. Modify the parameter group to set the `in-doubt xact resolution` parameter to 1 or 2.
5. Associate the option group and parameter group with the DB instance.

Creating the option group for MSDTC

Use the AWS Management Console or the AWS CLI to create an option group that corresponds to the SQL Server engine and version of your DB instance.

Note

You can also use an existing option group if it's for the correct SQL Server engine and version.

Console

The following procedure creates an option group for SQL Server Standard Edition 2016.

To create the option group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose **Create group**.
4. In the **Create option group** pane, do the following:
 - a. For **Name**, enter a name for the option group that is unique within your AWS account, such as **msdtc-se-2016**. The name can contain only letters, digits, and hyphens.
 - b. For **Description**, enter a brief description of the option group, such as **MSDTC option group for SQL Server SE 2016**. The description is used for display purposes.
 - c. For **Engine**, choose **sqlserver-se**.
 - d. For **Major engine version**, choose **13.00**.
5. Choose **Create**.

CLI

The following example creates an option group for SQL Server Standard Edition 2016.

To create the option group

- Use one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds create-option-group \
--option-group-name msdtc-se-2016 \
--engine-name sqlserver-se \
--major-engine-version 13.00 \
--option-group-description "MSDTC option group for SQL Server SE 2016"
```

For Windows:

```
aws rds create-option-group ^
--option-group-name msdtc-se-2016 ^
--engine-name sqlserver-se ^
--major-engine-version 13.00 ^
--option-group-description "MSDTC option group for SQL Server SE 2016"
```

Adding the MSDTC option to the option group

Next, use the AWS Management Console or the AWS CLI to add the **MSDTC** option to the option group.

The following option settings are required:

- **Port** – The port that you use to access MSDTC. Allowed values are 1150–49151 except for 1234, 1434, 3260, 3343, 3389, and 47001. The default value is 5000.

Make sure that the port you want to use is enabled in your firewall rules. Also, make sure as needed that this port is enabled in the inbound and outbound rules for the security group associated with your DB instance. For more information, see [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).

- **Security groups** – The VPC or DB security group memberships for your RDS DB instance.
- **Authentication type** – The authentication mode between hosts. The following authentication types are supported:
 - Mutual – The RDS instances are mutually authenticated to each other using integrated authentication. If this option is selected, all instances associated with this option group must be domain-joined.
 - None – No authentication is performed between hosts. We don't recommend using this mode in production environments.
- **Transaction log size** – The size of the MSDTC transaction log. Allowed values are 4–1024 MB. The default size is 4 MB.

The following option settings are optional:

- **Enable inbound connections** – Whether to allow inbound MSDTC connections to instances associated with this option group.
- **Enable outbound connections** – Whether to allow outbound MSDTC connections from instances associated with this option group.
- **Enable XA** – Whether to allow XA transactions. For more information on the XA protocol, see [XA specification](#).

Note

Using custom XA dynamic link libraries isn't supported.

- **Enable SNA LU** – Whether to allow the SNA LU protocol to be used for distributed transactions. For more information on SNA LU protocol support, see [Managing IBM CICS LU 6.2 transactions](#) in the Microsoft documentation.

Console

To add the MSDTC option

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose the option group that you just created.
4. Choose **Add option**.
5. Under **Option details**, choose **MSDTC** for **Option name**.
6. Under **Option settings**:
 - a. For **Port**, enter the port number for accessing MSDTC. The default is **5000**.
 - b. For **Security groups**, choose the VPC or DB security group to associate with the option.
 - c. For **Authentication type**, choose **Mutual** or **None**.
 - d. For **Transaction log size**, enter a value from 4–1024. The default is **4**.
7. Under **Additional configuration**, do the following:
 - a. For **Connections**, as needed choose **Enable inbound connections** and **Enable outbound connections**.
 - b. For **Allowed protocols**, as needed choose **Enable XA** and **Enable SNA LU**.
8. Under **Scheduling**, choose whether to add the option immediately or at the next maintenance window.
9. Choose **Add option**.

To add this option, no reboot is required.

CLI

To add the MSDTC option

1. Create a JSON file, for example `msdtc-option.json`, with the following required parameters.

```
{  
    "OptionGroupName": "msdtc-se-2016",  
    "OptionsToInclude": [  
        {  
            "OptionName": "MSDTC",  
            "Port": 5000,  
            "VpcSecurityGroupMemberships": ["sg-0abcdef123"],  
            "OptionSettings": [{"Name": "AUTHENTICATION", "Value": "MUTUAL"},  
                {"Name": "TRANSACTION_LOG_SIZE", "Value": "4"}]  
        },  
        "ApplyImmediately": true  
    ]  
}
```

2. Add the MSDTC option to the option group.

Example

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \  
    --cli-input-json file://msdtc-option.json \  
    --apply-immediately
```

For Windows:

```
aws rds add-option-to-option-group ^  
    --cli-input-json file://msdtc-option.json ^  
    --apply-immediately
```

No reboot is required.

Creating the parameter group for MSDTC

Create or modify a parameter group for the `in-doubt xact resolution` parameter that corresponds to the SQL Server edition and version of your DB instance.

Console

The following example creates a parameter group for SQL Server Standard Edition 2016.

To create the parameter group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. Choose **Create parameter group**.
4. In the **Create parameter group** pane, do the following:
 - a. For **Parameter group family**, choose `sqlserver-se-13.0`.
 - b. For **Group name**, enter an identifier for the parameter group, such as `msdtc-sqlserver-se-13`.

- c. For **Description**, enter **in-doubt xact resolution**.
5. Choose **Create**.

CLI

The following example creates a parameter group for SQL Server Standard Edition 2016.

To create the parameter group

- Use one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "in-doubt xact resolution"
```

For Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "in-doubt xact resolution"
```

Modifying the parameter for MSDTC

Modify the **in-doubt xact resolution** parameter in the parameter group that corresponds to the SQL Server edition and version of your DB instance.

For MSDTC, set the **in-doubt xact resolution** parameter to one of the following:

- 1 – **Presume commit**. Any MSDTC in-doubt transactions are presumed to have committed.
- 2 – **Presume abort**. Any MSDTC in-doubt transactions are presumed to have stopped.

For more information, see [in-doubt xact resolution server configuration option](#) in the Microsoft documentation.

Console

The following example modifies the parameter group that you created for SQL Server Standard Edition 2016.

To modify the parameter group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. Choose the parameter group, such as **msdtc-sqlserver-se-13**.
4. Under **Parameters**, filter the parameter list for **xact**.
5. Choose **in-doubt xact resolution**.
6. Choose **Edit parameters**.

7. Enter **1** or **2**.
8. Choose **Save changes**.

CLI

The following example modifies the parameter group that you created for SQL Server Standard Edition 2016.

To modify the parameter group

- Use one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name msdtc-sqlserver-se-13 \
--parameters "ParameterName='in-doubt xact
resolution',ParameterValue=1,ApplyMethod=immediate"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name msdtc-sqlserver-se-13 ^
--parameters "ParameterName='in-doubt xact
resolution',ParameterValue=1,ApplyMethod=immediate"
```

Associating the option group and parameter group with the DB instance

You can use the AWS Management Console or the AWS CLI to associate the MSDTC option group and parameter group with the DB instance.

Console

You can associate the MSDTC option group and parameter group with a new or existing DB instance.

- For a new DB instance, associate them when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- For an existing DB instance, associate them by modifying the instance. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Note

If you use an domain-joined existing DB instance, it must already have an Active Directory domain and AWS Identity and Access Management (IAM) role associated with it. If you create a new domain-joined instance, specify an existing Active Directory domain and IAM role. For more information, see [Using Windows Authentication with an Amazon RDS for SQL Server DB instance \(p. 711\)](#).

CLI

You can associate the MSDTC option group and parameter group with a new or existing DB instance.

Note

If you use an existing domain-joined DB instance, it must already have an Active Directory domain and IAM role associated with it. If you create a new domain-joined instance, specify

an existing Active Directory domain and IAM role. For more information, see [Using Windows Authentication with an Amazon RDS for SQL Server DB instance \(p. 711\)](#).

To create a DB instance with the MSDTC option group and parameter group

- Specify the same DB engine type and major version as you used when creating the option group.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --master-user-password secret123 \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name msdtc-se-2016 \
    --db-parameter-group-name msdtc-sqlserver-se-13
```

For Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --master-user-password secret123 ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name msdtc-se-2016 ^
    --db-parameter-group-name msdtc-sqlserver-se-13
```

To modify a DB instance and associate the MSDTC option group and parameter group

- Use one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --option-group-name msdtc-se-2016 \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--option-group-name msdtc-se-2016 ^
--db-parameter-group-name msdtc-sqlserver-se-13 ^
--apply-immediately
```

Using distributed transactions

In Amazon RDS for SQL Server, you run distributed transactions in the same way as distributed transactions running on-premises:

- Using .NET framework `System.Transactions` promotable transactions, which optimizes distributed transactions by deferring their creation until they're needed.

In this case, promotion is automatic and doesn't require you to make any intervention. If there's only one resource manager within the transaction, no promotion is performed. For more information about implicit transaction scopes, see [Implementing an implicit transaction using transaction scope](#) in the Microsoft documentation.

Promotable transactions are supported with these .NET implementations:

- Starting with ADO.NET 2.0, `System.Data.SqlClient` supports promotable transactions with SQL Server. For more information, see [System.Transactions integration with SQL Server](#) in the Microsoft documentation.
- ODP.NET supports `System.Transactions`. A local transaction is created for the first connection opened in the `TransactionsScope` scope to Oracle Database 11g release 1 (version 11.1) and later. When a second connection is opened, this transaction is automatically promoted to a distributed transaction. For more information about distributed transaction support in ODP.NET, see [Microsoft Distributed Transaction Coordinator integration](#) in the Microsoft documentation.
- Using the `BEGIN DISTRIBUTED TRANSACTION` statement. For more information, see [BEGIN DISTRIBUTED TRANSACTION \(Transact-SQL\)](#) in the Microsoft documentation.

Using transaction tracing

RDS supports controlling MSDTC transaction traces and downloading them from the RDS DB instance for troubleshooting. You can control transaction tracing sessions by running the following RDS stored procedure.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'trace_action',  
[@traceall='0/1'],  
[@traceaborted='0/1'],  
[@tracelong='0/1'];
```

The following parameter is required:

- `trace_action` – The tracing action. It can be `START`, `STOP`, or `STATUS`.

The following parameters are optional:

- `@traceall` – Set to 1 to trace all distributed transactions. The default is 0.
- `@traceaborted` – Set to 1 to trace canceled distributed transactions. The default is 0.
- `@tracelong` – Set to 1 to trace long-running distributed transactions. The default is 0.

Example of START tracing action

To start a new transaction tracing session, run the following example statement.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'START',
@traceall='0',
@traceaborted='1',
@tracelong='1';
```

Note

Only one transaction tracing session can be active at one time. If a new tracing session START command is issued while a tracing session is active, an error is returned and the active tracing session remains unchanged.

Example of STOP tracing action

To stop a transaction tracing session, run the following statement.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'STOP'
```

This statement stops the active transaction tracing session and saves the transaction trace data into the log directory on the RDS DB instance. The first row of the output contains the overall result, and the following lines indicate details of the operation.

The following is an example of a successful tracing session stop.

```
OK: Trace session has been successfully stopped.
Setting log file to: D:\rdsdbdata\MSDTC\Trace\dtctrace.log
Examining D:\rdsdbdata\MSDTC\Trace\msdtctr.mof for message formats, 8 found.
Searching for TMF files on path: (null)
LogFile D:\rdsdbdata\MSDTC\Trace\dtctrace.log:
OS version      10.0.14393 (Currently running on 6.2.9200)
Start Time       <timestamp>
End Time         <timestamp>
Timezone is     @tzres.dll,-932 (Bias is 0mins)
BufferSize       16384 B
Maximum File Size    10 MB
Buffers Written   Not set (Logger may not have been stopped).
Logger Mode Settings (11000002) ( circular paged
ProcessorCount     1
Processing completed Buffers: 1, Events: 3, EventsLost: 0 :: Format Errors: 0, Unknowns:
3
Event traces dumped to d:\rdsdbdata\Log\msdtc_<timestamp>.log
```

You can use the detailed information to query the name of the generated log file. For more information about downloading log files from the RDS DB instance, see [Accessing Amazon RDS database log files \(p. 504\)](#).

The trace session logs remain on the instance for 35 days. Any older trace session logs are automatically deleted.

Example of STATUS tracing action

To trace the status of a transaction tracing session, run the following statement.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'STATUS'
```

This statement outputs the following as separate rows of the result set.

```
OK
SessionStatus: <Started/Stopped>
TraceAll: <True/False>
TraceAborted: <True/False>
TraceLongLived: <True/False>
```

The first line indicates the overall result of the operation: OK or ERROR with details, if applicable. The subsequent lines indicate details about the tracing session status:

- SessionStatus can be one of the following:
 - Started if a tracing session is running.
 - Stopped if no tracing session is running.
- The tracing session flags can be True or False depending on how they were set in the START command.

Modifying the MSDTC option

After you enable the MSDTC option, you can modify its settings. For information about how to modify option settings, see [Modifying an option setting \(p. 221\)](#).

Note

Some changes to MSDTC option settings require the MSDTC service to be restarted. This requirement can affect running distributed transactions.

Disabling MSDTC

To disable MSDTC, remove the MSDTC option from its option group.

Console

To remove the MSDTC option from its option group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose the option group with the MSDTC option (msdtc-se-2016 in the previous examples).
4. Choose **Delete option**.
5. Under **Deletion options**, choose **MSDTC for Options to delete**.
6. Under **Apply immediately**, choose **Yes** to delete the option immediately, or **No** to delete it at the next maintenance window.
7. Choose **Delete**.

CLI

To remove the MSDTC option from its option group

- Use one of the following commands.

Example

For Linux, macOS, or Unix:

```
aws rds remove-option-from-option-group \
--option-group-name msdtc-se-2016 \
```

```
--options MSDTC \
--apply-immediately
```

For Windows:

```
aws rds remove-option-from-option-group ^
--option-group-name msdtc-se-2016 ^
--options MSDTC ^
--apply-immediately
```

Troubleshooting MSDTC for RDS for SQL Server

In some cases, you might have trouble establishing a connection between MSDTC running on a client computer and the MSDTC service running on an RDS for SQL Server DB instance. If so, make sure of the following:

- The inbound rules for the security group associated with the DB instance are configured correctly. For more information, see [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).
- Your client computer is configured correctly.
- The MSDTC firewall rules on your client computer are enabled.

To configure the client computer

1. Open **Component Services**.
Or, in **Server Manager**, choose **Tools**, and then choose **Component Services**.
2. Expand **Component Services**, expand **Computers**, expand **My Computer**, and then expand **Distributed Transaction Coordinator**.
3. Open the context (right-click) menu for **Local DTC** and choose **Properties**.
4. Choose the **Security** tab.
5. Choose all of the following:
 - **Network DTC Access**
 - **Allow Inbound**
 - **Allow Outbound**
6. Make sure that the correct authentication mode is chosen:
 - **Mutual Authentication Required** – The client machine is joined to the same domain as other nodes participating in distributed transaction, or there is a trust relationship configured between domains.
 - **No Authentication Required** – All other cases.
7. Choose **OK** to save your changes.
8. If prompted to restart the service, choose **Yes**.

To enable MSDTC firewall rules

1. Open Windows Firewall, then choose **Advanced settings**.

Or, in **Server Manager**, choose **Tools**, and then choose **Windows Firewall with Advanced Security**.

Note

Depending on your operating system, Windows Firewall might be called Windows Defender Firewall.

2. Choose **Inbound Rules** in the left pane.
3. Enable the following firewall rules, if they are not already enabled:
 - **Distributed Transaction Coordinator (RPC)**
 - **Distributed Transaction Coordinator (RPC)-EPMAP**
 - **Distributed Transaction Coordinator (TCP-In)**
4. Close Windows Firewall.

Common DBA tasks for Microsoft SQL Server

This section describes the Amazon RDS-specific implementations of some common DBA tasks for DB instances that are running the Microsoft SQL Server database engine. In order to deliver a managed service experience, Amazon RDS does not provide shell access to DB instances, and it restricts access to certain system procedures and tables that require advanced privileges.

Note

When working with a SQL Server DB instance, you can run scripts to modify a newly created database, but you cannot modify the [model] database, the database used as the model for new databases.

Topics

- [Accessing the tempdb database on Microsoft SQL Server DB instances on Amazon RDS \(p. 810\)](#)
- [Analyzing your database workload on an Amazon RDS DB instance with SQL Server Tuning Advisor \(p. 812\)](#)
- [Collations and character sets for Microsoft SQL Server \(p. 814\)](#)
- [Determining a recovery model for your Microsoft SQL Server database \(p. 817\)](#)
- [Determining the last failover time \(p. 817\)](#)
- [Disabling fast inserts during bulk loading \(p. 818\)](#)
- [Dropping a Microsoft SQL Server database \(p. 818\)](#)
- [Renaming a Microsoft SQL Server database in a Multi-AZ deployment \(p. 818\)](#)
- [Resetting the db_owner role password \(p. 819\)](#)
- [Restoring license-terminated DB instances \(p. 819\)](#)
- [Transitioning a Microsoft SQL Server database from OFFLINE to ONLINE \(p. 820\)](#)
- [Using change data capture \(p. 820\)](#)
- [Using SQL Server Agent \(p. 822\)](#)
- [Working with Microsoft SQL Server logs \(p. 823\)](#)
- [Working with trace and dump files \(p. 824\)](#)

Accessing the tempdb database on Microsoft SQL Server DB instances on Amazon RDS

You can access the tempdb database on your Microsoft SQL Server DB instances on Amazon RDS. You can run code on tempdb by using Transact-SQL through Microsoft SQL Server Management Studio (SSMS), or any other standard SQL client application. For more information about connecting to your DB instance, see [Connecting to a DB instance running the Microsoft SQL Server database engine \(p. 656\)](#).

The master user for your DB instance is granted CONTROL access to tempdb so that this user can modify the tempdb database options. The master user isn't the database owner of the tempdb database. If necessary, the master user can grant CONTROL access to other users so that they can also modify the tempdb database options.

Note

You can't run Database Console Commands (DBCC) on the tempdb database.

Modifying tempdb database options

You can modify the database options on the tempdb database on your Amazon RDS DB instances. For more information about which options can be modified, see [tempdb database](#) in the Microsoft documentation.

Database options such as the maximum file size options are persistent after you restart your DB instance. You can modify the database options to optimize performance when importing data, and to prevent running out of storage.

Optimizing performance when importing data

To optimize performance when importing large amounts of data into your DB instance, set the SIZE and FILEGROWTH properties of the tempdb database to large numbers. For more information about how to optimize tempdb, see [Optimizing tempdb performance](#) in the Microsoft documentation.

The following example demonstrates setting the size to 100 GB and file growth to 10 percent.

```
alter database[tempdb] modify file (NAME = N'templog', SIZE=100GB, FILEGROWTH = 10%)
```

Preventing storage problems

To prevent the tempdb database from using all available disk space, set the MAXSIZE property. The following example demonstrates setting the property to 2048 MB.

```
alter database [tempdb] modify file (NAME = N'templog', MAXSIZE = 2048MB)
```

Shrinking the tempdb database

There are two ways to shrink the tempdb database on your Amazon RDS DB instance. You can use the `rds_shrink_tempdbfile` procedure, or you can set the SIZE property,

Using the `rds_shrink_tempdbfile` procedure

You can use the Amazon RDS procedure `msdb.dbo.rds_shrink_tempdbfile` to shrink the tempdb database. You can only call `rds_shrink_tempdbfile` if you have CONTROL access to tempdb. When you call `rds_shrink_tempdbfile`, there is no downtime for your DB instance.

The `rds_shrink_tempdbfile` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>@temp_filename</code>	SYSNAME	—	required	The logical name of the file to shrink.
<code>@target_size</code>	int	null	optional	The new size for the file, in megabytes.

The following example gets the names of the files for the `tempdb` database.

```
use tempdb;
GO

select name, * from sys.sysfiles;
GO
```

The following example shrinks a `tempdb` database file named `test_file`, and requests a new size of 10 megabytes:

```
exec msdb.dbo.rds_shrink_tempdbfile @temp_filename = N'test_file', @target_size = 10;
```

Setting the SIZE property

You can also shrink the `tempdb` database by setting the `SIZE` property and then restarting your DB instance. For more information about restarting your DB instance, see [Rebooting a DB instance \(p. 276\)](#).

The following example demonstrates setting the `SIZE` property to 1024 MB.

```
alter database [tempdb] modify file (NAME = N'templog', SIZE = 1024MB)
```

Considerations for Multi-AZ deployments

If your Amazon RDS DB instance is in a Multi-AZ Deployment for Microsoft SQL Server with Database Mirroring (DBM) or Always On Availability Groups (AGs), there are some things to consider.

The `tempdb` database can't be replicated. No data that you store on your primary instance is replicated to your secondary instance.

If you modify any database options on the `tempdb` database, you can capture those changes on the secondary by using one of the following methods:

- First modify your DB instance and turn Multi-AZ off, then modify `tempdb`, and finally turn Multi-AZ back on. This method doesn't involve any downtime.

For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

- First modify `tempdb` in the original primary instance, then fail over manually, and finally modify `tempdb` in the new primary instance. This method involves downtime.

For more information, see [Rebooting a DB instance \(p. 276\)](#).

Analyzing your database workload on an Amazon RDS DB instance with SQL Server Tuning Advisor

The Database Engine Tuning Advisor is a client application provided by Microsoft that analyzes database workload and recommends an optimal set of indexes for your Microsoft SQL Server databases based on the kinds of queries you run. Like SQL Server Management Studio, you run Tuning Advisor from a client computer that connects to your Amazon RDS DB instance that is running SQL Server. The client computer can be a local computer that you run on premises within your own network or it can be an Amazon EC2 Windows instance that is running in the same region as your Amazon RDS DB instance.

This section shows how to capture a workload for Tuning Advisor to analyze. This is the preferred process for capturing a workload because Amazon RDS restricts host access to the SQL Server instance. The full documentation on Tuning Advisor can be found on [MSDN](#).

To use Tuning Advisor, you must provide what is called a workload to the advisor. A workload is a set of Transact-SQL statements that run against a database or databases that you want to tune. Database Engine Tuning Advisor uses trace files, trace tables, Transact-SQL scripts, or XML files as workload input when tuning databases. When working with Amazon RDS, a workload can be a file on a client computer or a database table on an Amazon RDS for SQL Server DB accessible to your client computer. The file or the table must contain queries against the databases you want to tune in a format suitable for replay.

For Tuning Advisor to be most effective, a workload should be as realistic as possible. You can generate a workload file or table by performing a trace against your DB instance. While a trace is running, you can either simulate a load on your DB instance or run your applications with a normal load.

There are two types of traces: client-side and server-side. A client-side trace is easier to set up and you can watch trace events being captured in real-time in SQL Server Profiler. A server-side trace is more complex to set up and requires some Transact-SQL scripting. In addition, because the trace is written to a file on the Amazon RDS DB instance, storage space is consumed by the trace. It is important to track of how much storage space a running server-side trace uses because the DB instance could enter a storage-full state and would no longer be available if it runs out of storage space.

For a client-side trace, when a sufficient amount of trace data has been captured in the SQL Server Profiler, you can then generate the workload file by saving the trace to either a file on your local computer or in a database table on a DB instance that is available to your client computer. The main disadvantage of using a client-side trace is that the trace may not capture all queries when under heavy loads. This could weaken the effectiveness of the analysis performed by the Database Engine Tuning Advisor. If you need to run a trace under heavy loads and you want to ensure that it captures every query during a trace session, you should use a server-side trace.

For a server-side trace, you must get the trace files on the DB instance into a suitable workload file or you can save the trace to a table on the DB instance after the trace completes. You can use the SQL Server Profiler to save the trace to a file on your local computer or have the Tuning Advisor read from the trace table on the DB instance.

Running a client-side trace on a SQL Server DB instance

To run a client-side trace on a SQL Server DB instance

1. Start SQL Server Profiler. It is installed in the Performance Tools folder of your SQL Server instance folder. You must load or define a trace definition template to start a client-side trace.
2. In the SQL Server Profiler File menu, choose **New Trace**. In the **Connect to Server** dialog box, enter the DB instance endpoint, port, master user name, and password of the database you would like to run a trace on.
3. In the **Trace Properties** dialog box, enter a trace name and choose a trace definition template. A default template, `TSQL_Replay`, ships with the application. You can edit this template to define your

trace. Edit events and event information under the **Events Selection** tab of the **Trace Properties** dialog box. For more information about trace definition templates and using the SQL Server Profiler to specify a client-side trace see the documentation in [MSDN](#).

4. Start the client-side trace and watch SQL queries in real-time as they run against your DB instance.
5. Select **Stop Trace** from the **File** menu when you have completed the trace. Save the results as a file or as a trace table on your DB instance.

Running a server-side trace on a SQL Server DB instance

Writing scripts to create a server-side trace can be complex and is beyond the scope of this document. This section contains sample scripts that you can use as examples. As with a client-side trace, the goal is to create a workload file or trace table that you can open using the Database Engine Tuning Advisor.

The following is an abridged example script that starts a server-side trace and captures details to a workload file. The trace initially saves to the file RDSTrace.trc in the D:\RDSDDBDATA\Log directory and rolls-over every 100 MB so subsequent trace files are named RDSTrace_1.trc, RDSTrace_2.trc, etc.

```
DECLARE @file_name NVARCHAR(245) = 'D:\RDSDDBDATA\Log\RDSTrace';
DECLARE @max_file_size BIGINT = 100;
DECLARE @on BIT = 1
DECLARE @rc INT
DECLARE @traceid INT

EXEC @rc = sp_trace_create @traceid OUTPUT, 2, @file_name, @max_file_size
IF (@rc = 0) BEGIN
    EXEC sp_trace_setevent @traceid, 10, 1, @on
    EXEC sp_trace_setevent @traceid, 10, 2, @on
    EXEC sp_trace_setevent @traceid, 10, 3, @on
    .
    .
    EXEC sp_trace_setfilter @traceid, 10, 0, 7, N'SQL Profiler'
    EXEC sp_trace_setstatus @traceid, 1
END
```

The following example is a script that stops a trace. Note that a trace created by the previous script continues to run until you explicitly stop the trace or the process runs out of disk space.

```
DECLARE @traceid INT
SELECT @traceid = traceid FROM ::fn_trace_getinfo(default)
WHERE property = 5 AND value = 1 AND traceid <> 1

IF @traceid IS NOT NULL BEGIN
    EXEC sp_trace_setstatus @traceid, 0
    EXEC sp_trace_setstatus @traceid, 2
END
```

You can save server-side trace results to a database table and use the database table as the workload for the Tuning Advisor by using the fn_trace_gettable function. The following commands load the results of all files named RDSTrace.trc in the D:\rdsdbdata\Log directory, including all rollover files like RDSTrace_1.trc, into a table named RDSTrace in the current database.

```
SELECT * INTO RDSTrace
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace.trc', default);
```

To save a specific rollover file to a table, for example the RDSTrace_1.trc file, specify the name of the rollover file and substitute 1 instead of default as the last parameter to fn_trace_gettable.

```
SELECT * INTO RDSTrace_1
```

```
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace_1.trc', 1);
```

Running Tuning Advisor with a trace

Once you create a trace, either as a local file or as a database table, you can then run Tuning Advisor against your DB instance. Microsoft includes documentation on using the Database Engine Tuning Advisor in [MSDN](#). Using Tuning Advisor with Amazon RDS is the same process as when working with a standalone, remote SQL Server instance. You can either use the Tuning Advisor UI on your client machine or use the dta.exe utility from the command line. In both cases, you must connect to the Amazon RDS DB instance using the endpoint for the DB instance and provide your master user name and master user password when using Tuning Advisor.

The following code example demonstrates using the dta.exe command line utility against an Amazon RDS DB instance with an endpoint of `dta.cnazcmklsdei.us-east-1.rds.amazonaws.com`. The example includes the master user name `admin` and the master user password `test`, the example database to tune is named `RDSDTA` and the input workload is a trace file on the local machine named `C:\RDSTrace.trc`. The example command line code also specifies a trace session named `RDSTrace1` and specifies output files to the local machine named `RDSTrace.sql` for the SQL output script, `RDSTrace.txt` for a result file, and `RDSTrace.xml` for an XML file of the analysis. There is also an error table specified on the RDSDTA database named `RDSTraceErrors`.

```
dta -S dta.cnazcmklsdei.us-east-1.rds.amazonaws.com -U admin -P test -D RDSDTA -if C:\RDSTrace.trc -s RDSTrace1 -of C:\ RDSTrace.sql -or C:\ RDSTrace.txt -ox C:\ RDSTrace.xml -e RDSDTA.dbo.RDSTraceErrors
```

Here is the same example command line code except the input workload is a table on the remote Amazon RDS instance named `RDSTrace` which is on the `RDSDTA` database.

```
dta -S dta.cnazcmklsdei.us-east-1.rds.amazonaws.com -U admin -P test -D RDSDTA -it RDSDTA.dbo.RDSTrace -s RDSTrace1 -of C:\ RDSTrace.sql -or C:\ RDSTrace.txt -ox C:\ RDSTrace.xml -e RDSDTA.dbo.RDSTraceErrors
```

A full list of dta utility command-line parameters can be found in [MSDN](#).

Collations and character sets for Microsoft SQL Server

SQL Server supports collations at multiple levels. You set the default server collation when you create the DB instance. You can override the collation in the database, table, or column level.

Topics

- [Server-level collation for Microsoft SQL Server \(p. 814\)](#)
- [Database-level collation for Microsoft SQL Server \(p. 816\)](#)

Server-level collation for Microsoft SQL Server

When you create a Microsoft SQL Server DB instance, you can set the server collation that you want to use. If you don't choose a different collation, the server-level collation defaults to `SQL_Latin1_General_CI_AS`. The server collation is applied by default to all databases and database objects.

Note

You can't change the collation when you restore from a DB snapshot.

Currently, Amazon RDS supports the following server collations:

Collation	Description
Chinese_PRC_CI_AS	Chinese-PRC, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Chinese_Taiwan_Stroke_CI_AS	Chinese-Taiwan-Stroke, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Finnish_Swedish_CI_AS	Finnish, Swedish, and Swedish (Finland), case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
French_CI_AS	French, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Hebrew_BIN	Hebrew, binary sort
Japanese_CI_AS	Japanese, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Korean_Wansung_CI_AS	Korean-Wansung, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Latin1_General_100_BIN	Latin1-General-100, binary sort
Latin1_General_100_BIN2	Latin1-General-100, binary code point comparison sort
Latin1_General_100_CI_AS	Latin1-General-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Latin1_General_BIN	Latin1-General, binary sort
Latin1_General_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive
Latin1_General_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Latin1_General_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Modern_Spanish_CI_AS	Modern-Spanish, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
SQL_Latin1_General_CP1_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 54 on Code Page 1252 for non-Unicode Data
SQL_Latin1_General_CP1_CI_AS (default)	Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 52 on Code Page 1252 for non-Unicode Data
SQL_Latin1_General_CP1_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive for

Collation	Description
	Unicode Data, SQL Server Sort Order 51 on Code Page 1252 for non-Unicode Data
SQL_Latin1_General_CI_AS	Latin1-General, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 34 on Code Page 437 for non-Unicode Data
SQL_Latin1_General_CI_AS_BIN	Latin1-General, binary code point comparison sort for Unicode Data, SQL Server Sort Order 40 on Code Page 850 for non-Unicode Data
SQL_Latin1_General_CI_AS_WS	Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 42 on Code Page 850 for non-Unicode Data

To choose the collation:

- If you're using the Amazon RDS console, when creating a new DB instance choose **Additional configuration**, then choose the collation from the **Collation** menu under **Database options**. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- If you're using the AWS CLI, use the `--character-set-name` option with the `create-db-instance` command. For more information, see [create-db-instance](#).
- If you're using the Amazon RDS API, use the `CharacterSetName` parameter with the `CreateDBInstance` operation. For more information, see [CreateDBInstance](#).

Database-level collation for Microsoft SQL Server

You can change the default collation at the database, table, or column level by overriding the collation when creating a new database or database object. For example, if your default server collation is `SQL_Latin1_General_CI_AS`, you can change it to `Mohawk_100_CI_AS` for Mohawk collation support. Even arguments in a query can be type-cast to use a different collation if necessary.

For example, the following query would change the default collation for the `AccountName` column to `Mohawk_100_CI_AS`

```
CREATE TABLE [dbo].[Account]
(
    [AccountID] [nvarchar](10) NOT NULL,
    [AccountName] [nvarchar](100) COLLATE Mohawk_100_CI_AS NOT NULL
) ON [PRIMARY];
```

The Microsoft SQL Server DB engine supports Unicode by the built-in `NCHAR`, `NVARCHAR`, and `NTEXT` data types. For example, if you need CJK support, use these Unicode data types for character storage and override the default server collation when creating your databases and tables. Here are several links from Microsoft covering collation and Unicode support for SQL Server:

- [Working with collations](#)
- [Collation and international terminology](#)
- [Using SQL Server collations](#)
- [International considerations for databases and database engine applications](#)

Determining a recovery model for your Microsoft SQL Server database

In Amazon RDS, the recovery model, retention period, and database status are linked.

It's important to understand the consequences before making a change to one of these settings. Each setting can affect the others. For example:

- If you change a database's recovery model to SIMPLE or BULK_LOGGED while backup retention is enabled, Amazon RDS resets the recovery model to FULL within five minutes. This also results in RDS taking a snapshot of the DB instance.
- If you set backup retention to 0 days, RDS sets the recovery mode to SIMPLE.
- If you change a database's recovery model from SIMPLE to any other option while backup retention is set to 0 days, RDS resets the recovery model to SIMPLE.

Important

Never change the recovery model on Multi-AZ instances, even if it seems you can do so—for example, by using ALTER DATABASE. Backup retention, and therefore FULL recovery mode, is required for Multi-AZ. If you alter the recovery model, RDS immediately changes it back to FULL. This automatic reset forces RDS to completely rebuild the mirror. During this rebuild, the availability of the database is degraded for about 30-90 minutes until the mirror is ready for failover. The DB instance also experiences performance degradation in the same way it does during a conversion from Single-AZ to Multi-AZ. How long performance is degraded depends on the database storage size—the bigger the stored database, the longer the degradation.

For more information on SQL Server recovery models, see [Recovery models \(SQL Server\)](#) in the Microsoft documentation.

Determining the last failover time

To determine the last failover time, use the following stored procedure:

```
execute msdb.dbo.rds_failover_time;
```

This procedure returns the following information.

Output parameter	Description
errorlog_available_from	Shows the time from when error logs are available in the log directory.
recent_failover_time	Shows the last failover time if it's available from the error logs. Otherwise it shows null.

Note

The stored procedure searches all of the available SQL Server error logs in the log directory to retrieve the most recent failover time. If the failover messages have been overwritten by SQL Server, then the procedure doesn't retrieve the failover time.

Example of no recent failover

This example shows the output when there is no recent failover in the error logs. No failover has happened since 2020-04-29 23:59:00.01.

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	null

Example of recent failover

This example shows the output when there is a failover in the error logs. The most recent failover was at 2020-05-05 18:57:51.89.

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	2020-05-05 18:57:51.8900000

Disabling fast inserts during bulk loading

Starting with SQL Server 2016, fast inserts are enabled by default. Fast inserts leverage the minimal logging that occurs while the database is in the simple or bulk logged recovery model to optimize insert performance. With fast inserts, each bulk load batch acquires new extents, bypassing the allocation lookup for existing extents with available free space to optimize insert performance.

However, with fast inserts bulk loads with small batch sizes can lead to increased unused space consumed by objects. If increasing batch size isn't feasible, enabling trace flag 692 can help reduce unused reserved space, but at the expense of performance. Enabling this trace flag disables fast inserts while bulk loading data into heap or clustered indexes.

You enable trace flag 692 as a startup parameter using DB parameter groups. For more information, see [Working with DB parameter groups \(p. 228\)](#).

Trace flag 692 is supported for Amazon RDS on SQL Server 2016 and later. For more information on trace flags, see [DBCC TRACEON - trace flags](#) in the Microsoft documentation.

Dropping a Microsoft SQL Server database

You can drop a database on an Amazon RDS DB instance running Microsoft SQL Server in a Single-AZ or Multi-AZ deployment. To drop the database, use the following command:

```
--replace your-database-name with the name of the database you want to drop
EXECUTE msdb.dbo.rds_drop_database N'your-database-name'
```

Note

Use straight single quotes in the command. Smart quotes will cause an error.

After you use this procedure to drop the database, Amazon RDS drops all existing connections to the database and removes the database's backup history.

Renaming a Microsoft SQL Server database in a Multi-AZ deployment

To rename a Microsoft SQL Server database instance that uses Multi-AZ, use the following procedure:

1. First, turn off Multi-AZ for the DB instance.
2. Rename the database by running `rdsadmin.dbo.rds_modify_db_name`.

3. Then, turn on Multi-AZ Mirroring or Always On Availability Groups for the DB instance, to return it to its original state.

For more information, see [Adding Multi-AZ to a Microsoft SQL Server DB instance \(p. 699\)](#).

Note

If your instance doesn't use Multi-AZ, you don't need to change any settings before or after running `rdsadmin.dbo.rds_modify_db_name`.

Example: In the following example, the `rdsadmin.dbo.rds_modify_db_name` stored procedure renames a database from `MOO` to `ZAR`. This is similar to running the statement `DDL ALTER DATABASE [MOO] MODIFY NAME = [ZAR]`.

```
EXEC rdsadmin.dbo.rds_modify_db_name N'MOO', N'ZAR'  
GO
```

Resetting the db_owner role password

If you lock yourself out of the `db_owner` role on your Microsoft SQL Server database, you can reset the `db_owner` role password by modifying the DB instance master password. By changing the DB instance master password, you can regain access to the DB instance, access databases using the modified password for the `db_owner`, and restore privileges for the `db_owner` role that may have been accidentally revoked. You can change the DB instance password by using the Amazon RDS console, the AWS CLI command [modify-db-instance](#), or by using the [ModifyDBInstance](#) operation. For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Restoring license-terminated DB instances

Microsoft has requested that some Amazon RDS customers who did not report their Microsoft License Mobility information terminate their DB instance. Amazon RDS takes snapshots of these DB instances, and you can restore from the snapshot to a new DB instance that has the License Included model.

You can restore from a snapshot of Standard Edition to either Standard Edition or Enterprise Edition.

You can restore from a snapshot of Enterprise Edition to either Standard Edition or Enterprise Edition.

To restore from a SQL Server snapshot after Amazon RDS has created a final snapshot of your instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. Choose the snapshot of your SQL Server DB instance. Amazon RDS creates a final snapshot of your DB instance. The name of the terminated instance snapshot is in the format `instance_name-final-snapshot`. For example, if your DB instance name is `mytest.cdxgahs1ksma.us-east-1.rds.com`, the final snapshot is called `mytest-final-snapshot` and is located in the same AWS Region as the original DB instance.
4. For **Actions**, choose **Restore Snapshot**.
The **Restore DB Instance** window appears.
5. For **License Model**, choose **license-included**.
6. Choose the SQL Server DB engine that you want to use.
7. For **DB Instance Identifier**, enter the name for the restored DB instance.
8. Choose **Restore DB Instance**.

For more information about restoring from a snapshot, see [Restoring from a DB snapshot \(p. 349\)](#).

Transitioning a Microsoft SQL Server database from OFFLINE to ONLINE

You can transition your Microsoft SQL Server database on an Amazon RDS DB instance from OFFLINE to ONLINE.

SQL Server method	Amazon RDS method
ALTER DATABASE <code>db_name</code> SET ONLINE;	EXEC rdsadmin.dbo.rds_set_database_online <code>db_name</code>

Using change data capture

Amazon RDS supports change data capture (CDC) for your DB instances running Microsoft SQL Server. CDC captures changes that are made to the data in your tables. It stores metadata about each change, which you can access later. For more information about how CDC works, see [Change data capture](#) in the Microsoft documentation.

Before you use CDC with your Amazon RDS DB instances, enable it in the database by running `msdb.dbo.rds_cdc_enable_db`. You must have master user privileges to enable CDC in the Amazon RDS DB instance. After CDC is enabled, any user who is `db_owner` of that database can enable or disable CDC on tables in that database.

Important

During restores, CDC will be disabled. All of the related metadata is automatically removed from the database. This applies to snapshot restores, point-in-time restores, and SQL Server Native restores from S3. After performing one of these types of restores, you can re-enable CDC and re-specify tables to track.

To enable CDC for a DB instance, run the `msdb.dbo.rds_cdc_enable_db` stored procedure.

```
exec msdb.dbo.rds_cdc_enable_db 'database_name'
```

To disable CDC for a DB instance, run the `msdb.dbo.rds_cdc_disable_db` stored procedure.

```
exec msdb.dbo.rds_cdc_disable_db 'database_name'
```

Topics

- [Tracking tables with change data capture \(p. 820\)](#)
- [Change data capture jobs \(p. 821\)](#)
- [Change data capture for Multi-AZ instances \(p. 821\)](#)

Tracking tables with change data capture

After CDC is enabled on the database, you can start tracking specific tables. You can choose the tables to track by running [`sys.sp_cdc_enable_table`](#).

```
--Begin tracking a table
```

```
exec sys.sp_cdc_enable_table
    @source_schema      = N'source_schema'
    , @source_name       = N'source_name'
    , @role_name         = N'role_name'

--The following parameters are optional:

--, @capture_instance     = 'capture_instance'
--, @supports_net_changes = supports_net_changes
--, @index_name           = 'index_name'
--, @captured_column_list = 'captured_column_list'
--, @filegroup_name        = 'filegroup_name'
--, @allow_partition_switch = 'allow_partition_switch'
;
```

To view the CDC configuration for your tables, run [sys.sp_cdc_help_change_data_capture](#).

```
--View CDC configuration
exec sys.sp_cdc_help_change_data_capture

--The following parameters are optional and must be used together.
--  'schema_name', 'table_name'
;
```

For more information on CDC tables, functions, and stored procedures in SQL Server documentation, see the following:

- [Change data capture stored procedures \(Transact-SQL\)](#)
- [Change data capture functions \(Transact-SQL\)](#)
- [Change data capture tables \(Transact-SQL\)](#)

Change data capture jobs

When you enable CDC, SQL Server creates the CDC jobs. Database owners (`db_owner`) can view, create, modify, and delete the CDC jobs. However, the RDS system account owns them. Therefore, the jobs aren't visible from native views, procedures, or in SQL Server Management Studio.

To control behavior of CDC in a database, use native SQL Server procedures such as `sp_cdc_enable_table` and `sp_cdc_start_job`. To change CDC job parameters, like `maxtrans` and `maxscans`, you can use `sp_cdc_change_job`.

To get more information regarding the CDC jobs, you can query the following dynamic management views:

- `sys.dm_cdc_errors`
- `sys.dm_cdc_log_scan_sessions`
- `sysjobs`
- `sysjobhistory`

Change data capture for Multi-AZ instances

If you use CDC on a Multi-AZ instance, make sure the mirror's CDC job configuration matches the one on the principal. CDC jobs are mapped to the `database_id`. If the database IDs on the secondary are different from the principal, then the jobs won't be associated with the correct database. To try to prevent errors after failover, RDS drops and recreates the jobs on the new principal. The recreated jobs use the parameters that the principal recorded before failover.

Although this process runs quickly, it's still possible that the CDC jobs might run before RDS can correct them. Here are three ways to force parameters to be consistent between primary and secondary replicas:

- Use the same job parameters for all the databases that have CDC enabled.
- Before you change the CDC job configuration, convert the Multi-AZ instance to Single-AZ.
- Manually transfer the parameters whenever you change them on the principal.

To view and define the CDC parameters that are used to recreate the CDC jobs after a failover, use `rds_show_configuration` and `rds_set_configuration`.

The following example returns the value set for `cdc_capture_maxtrans`. For any parameter that is set to `RDS_DEFAULT`, RDS automatically configures the value.

```
-- Show configuration for each parameter on either primary and secondary replicas.  
exec rdsadmin.dbo.rds_show_configuration 'cdc_capture_maxtrans'
```

To set the configuration on the secondary, run `rdsadmin.dbo.rds_set_configuration`. This procedure sets the parameter values for all of the databases on the secondary server. These settings are used only after a failover. The following example sets the `maxtrans` for all CDC capture jobs to `1000`:

```
--To set values on secondary. These are used after failover.  
exec rdsadmin..rds_set_configuration 'cdc_capture_maxtrans' , 1000
```

To set the CDC job parameters on the principal, use `sys.sp_cdc_change_job` instead.

Using SQL Server Agent

With Amazon RDS, you can use SQL Server Agent on a DB instance running Microsoft SQL Server Standard, Web Edition, or Enterprise Edition. SQL Server Agent is a Microsoft Windows service that runs scheduled administrative tasks, which are called jobs. You can use SQL Server Agent to run T-SQL jobs to rebuild indexes, run corruption checks, and aggregate data in a SQL Server DB instance.

SQL Server Agent can run a job on a schedule, in response to a specific event, or on demand. For more information, see [SQL Server Agent](#) in the SQL Server documentation. You should avoid scheduling jobs to run during the maintenance and backup windows for your DB instance because these maintenance and backup processes that are launched by AWS could interrupt the job or cause it to be cancelled. Because Amazon RDS backs up your DB instance, you do not use SQL Server Agent to create backups.

To view the history of an individual SQL Server Agent job in the SQL Server Management Studio, you open Object Explorer, right-click the job, and then click **View History**.

Because SQL Server Agent is running on a managed host in a DB instance, there are some actions that are not supported. Running replication jobs and running command-line scripts by using ActiveX, Windows command shell, or Windows PowerShell are not supported. In addition, you cannot manually start, stop, or restart SQL Server Agent because its operation is managed by the host. Email notifications through SQL Server Agent are not available from a DB instance.

When you create a SQL Server DB instance, the master user name is enrolled in the `SQLAgentUserRole` role. To allow an additional login/user to use SQL Server Agent, you must log in as the master user and do the following.

1. Create another server-level login by using the `CREATE LOGIN` command.
2. Create a user in `msdb` using `CREATE USER` command, and then link this user to the login that you created in the previous step.
3. Add the user to the `SQLAgentUserRole` using the `sp_addrolemember` system stored procedure.

For example, suppose your master user name is **admin** and you want to give access to SQL Server Agent to a user named **theirname** with a password **theirpassword**. You would log in using the master user name and run the following commands.

```
--Initially set context to master database
USE [master];
GO
--Create a server-level login named theirname with password theirpassword
CREATE LOGIN [theirname] WITH PASSWORD = 'theirpassword';
GO
--Set context to msdb database
USE [msdb];
GO
--Create a database user named theirname and link it to server-level login theirname
CREATE USER [theirname] FOR LOGIN [theirname];
GO
--Added database user theirname in msdb to SQLAgentUserRole in msdb
EXEC sp_addrolemember [SQLAgentUserRole], [theirname];
```

To delete a SQL Server Agent job, run the following T-SQL statement.

```
EXEC msdb..sp_delete_job @job_name = 'job_name';
```

Note

Don't use the UI in SQL Server Management Console (SSMS) to delete a SQL Server Agent job. If you do, you get an error message similar to the following:

```
The EXECUTE permission was denied on the object 'xp_regrid', database
'mssqlsystemresource', schema 'sys'.
```

This error occurs because, as a managed service, RDS is restricted from running procedures that access the Windows registry. When you use SSMS to delete the job, it tries to run a process (*xp_regrid*) that RDS isn't authorized to do.

Working with Microsoft SQL Server logs

You can use the Amazon RDS console to view, watch, and download SQL Server Agent logs and Microsoft SQL Server error logs.

Watching log files

If you view a log in the Amazon RDS console, you can see its contents as they exist at that moment. Watching a log in the console opens it in a dynamic state so that you can see updates to it in near-real time.

Only the latest log is active for watching. For example, suppose you have the following logs shown:

Name	Last Written	Size	view	watch	download
log/ERROR	January 14, 2015 at 5:17:35 AM UTC-8	6.1 kB	view	watch	download
log/ERROR.1	January 13, 2015 at 3:59:00 PM UTC-8	53.3 kB	view	watch	download
log/ERROR.2	January 12, 2015 at 3:59:00 PM UTC-8	5.9 kB	view	watch	download
log/ERROR.3	January 11, 2015 at 3:59:00 PM UTC-8	5.9 kB	view	watch	download
log/ERROR.4	January 10, 2015 at 3:59:00 PM UTC-8	5.9 kB	view	watch	download

Only log/ERROR, as the most recent log, is being actively updated. You can choose to watch others, but they are static and will not update.

Archiving log files

The Amazon RDS console shows logs for the past week through the current day. You can download and archive logs to keep them for reference past that time. One way to archive logs is to load them into an Amazon S3 bucket. For instructions on how to set up an Amazon S3 bucket and upload a file, see [Amazon S3 basics](#) in the *Amazon Simple Storage Service Getting Started Guide* and click **Get Started**.

Viewing error and agent logs

To view Microsoft SQL Server error and agent logs, use the Amazon RDS stored procedure `rds_read_error_log` with the following parameters:

- `@index` – the version of the log to retrieve. The default value is 0, which retrieves the current error log. Specify 1 to retrieve the previous log, specify 2 to retrieve the one before that, and so on.
- `@type` – the type of log to retrieve. Specify 1 to retrieve an error log. Specify 2 to retrieve an agent log.

Example

The following example requests the current error log.

```
EXEC rdsadmin.dbo.rds_read_error_log @index = 0, @type = 1;
```

For more information on SQL Server errors, see [Database engine errors](#) in the Microsoft documentation.

Working with trace and dump files

This section describes working with trace files and dump files for your Amazon RDS DB instances running Microsoft SQL Server.

Generating a trace SQL query

```
declare @rc int
declare @TraceID int
declare @maxfilesize bigint

set @maxfilesize = 5

exec @rc = sp_trace_create @TraceID output, 0, N'D:\rdsdbdata\log\rdstest', @maxfilesize,
NULL
```

Viewing an open trace

```
select * from ::fn_trace_getinfo(default)
```

Viewing trace contents

```
select * from ::fn_trace_gettable('D:\rdsdbdata\log\rdstest.trc', default)
```

Setting the retention period for trace and dump files

Trace and dump files can accumulate and consume disk space. By default, Amazon RDS purges trace and dump files that are older than seven days.

To view the current trace and dump file retention period, use the `rds_show_configuration` procedure, as shown in the following example.

```
exec rdsadmin..rds_show_configuration;
```

To modify the retention period for trace files, use the `rds_set_configuration` procedure and set the `tracefile retention` in minutes. The following example sets the trace file retention period to 24 hours.

```
exec rdsadmin..rds_set_configuration 'tracefile retention', 1440;
```

To modify the retention period for dump files, use the `rds_set_configuration` procedure and set the `dumpfile retention` in minutes. The following example sets the dump file retention period to 3 days.

```
exec rdsadmin..rds_set_configuration 'dumpfile retention', 4320;
```

For security reasons, you cannot delete a specific trace or dump file on a SQL Server DB instance. To delete all unused trace or dump files, set the retention period for the files to 0.

MySQL on Amazon RDS

Amazon RDS supports DB instances running several versions of MySQL. You can use the following major versions:

- MySQL 8.0
- MySQL 5.7
- MySQL 5.6
- MySQL 5.5

For more information about minor version support, see [MySQL on Amazon RDS versions \(p. 828\)](#).

You first use the Amazon RDS management tools or interfaces to create an RDS for MySQL DB instance. You can then resize the DB instance, authorize connections to the DB instance, create and restore from backups or snapshots, create Multi-AZ secondaries, create read replicas, and monitor the performance of the DB instance. You use standard MySQL utilities and applications to store and access the data in the DB instance.

Amazon RDS for MySQL is compliant with many industry standards. For example, you can use Amazon RDS for MySQL databases to build HIPAA-compliant applications and to store healthcare related information, including protected health information (PHI) under a Business Associate Agreement (BAA) with AWS. Amazon RDS for MySQL also meets Federal Risk and Authorization Management Program (FedRAMP) security requirements and has received a FedRAMP Joint Authorization Board (JAB) Provisional Authority to Operate (P-ATO) at the FedRAMP HIGH Baseline within the AWS GovCloud (US) Regions. For more information on supported compliance standards, see [AWS cloud compliance](#).

For information about the features in each version of MySQL, see [The main features of MySQL](#) in the MySQL documentation.

Common management tasks for MySQL on Amazon RDS

The following are the common management tasks you perform with an RDS for MySQL DB instance, with links to relevant documentation for each task.

Task area	Relevant documentation
Understanding Amazon RDS Understand key Amazon RDS components, including DB instances, AWS Regions, Availability Zones, security groups, parameter groups, and option groups.	What is Amazon Relational Database Service (Amazon RDS)? (p. 1)
Setting up Amazon RDS for first time use Set up Amazon RDS so that you can create MySQL DB instances in Amazon Web Services (AWS).	Setting up for Amazon RDS (p. 67)
Understanding Amazon RDS DB instances	Amazon RDS DB instances (p. 5)

Task area	Relevant documentation
<p>Create virtual MySQL server instances that run in AWS. Because DB instances are the building blocks of Amazon RDS, we recommend that you understand their principles.</p>	
<p>Creating a DB instance for production</p> <p>Create a DB instance for production purposes. Creating an instance includes choosing a DB instance class with appropriate processing power and memory capacity and choosing a storage type that supports the way you expect to use your database.</p>	<p>DB instance classes (p. 7)</p> <p>Amazon RDS storage types (p. 40)</p> <p>Creating an Amazon RDS DB instance (p. 141)</p>
<p>Managing security for your DB instance</p> <p>By default, DB instances are created with a firewall that prevents access to them. You must create a security group with the correct IP addresses and network configuration to access the DB instance. You can also use AWS Identity and Access Management (IAM) policies to assign permissions that determine who is allowed to manage RDS resources.</p>	<p>Security in Amazon RDS (p. 1627)</p> <p>Managing access using policies (p. 1646)</p> <p>Controlling access with security groups (p. 1699)</p> <p>Determining whether you are using the EC2-VPC or EC2-Classic platform (p. 1718)</p>
<p>Connecting to your DB instance</p> <p>Connect to your DB instance using a standard SQL client application such as the MySQL command line utility or MySQL Workbench.</p>	<p>Connecting to a DB instance running the MySQL database engine (p. 840)</p>
<p>Configuring high availability for a production DB instance</p> <p>Provide high availability with synchronous standby replication in a different Availability Zone, automatic failover, fault tolerance for DB instances using Multi-AZ deployments, and read replicas.</p>	<p>High availability (Multi-AZ) for Amazon RDS (p. 53)</p>
<p>Configuring a DB instance in an Amazon Virtual Private Cloud</p> <p>Configure a virtual private cloud (VPC) in the Amazon VPC service. An Amazon VPC is a virtual network logically isolated from other virtual networks in AWS.</p>	<p>Determining whether you are using the EC2-VPC or EC2-Classic platform (p. 1718)</p> <p>Working with a DB instance in a VPC (p. 1727)</p>
<p>Configuring specific MySQL database parameters and features</p> <p>Configure specific MySQL database parameters with a parameter group that can be associated with many DB instances. You can also configure specific MySQL database features with an option group that can be associated with many DB instances.</p>	<p>Working with DB parameter groups (p. 228)</p> <p>Working with option groups (p. 212)</p> <p>Options for MySQL DB instances (p. 925)</p>

Task area	Relevant documentation
Modifying a DB instance running the MySQL database engine Change the settings of a DB instance to accomplish tasks such as adding additional storage or changing the DB instance class.	Modifying an Amazon RDS DB instance (p. 250)
Configuring database backup and restore Configure your DB instance to take automated backups. You can also back up and restore your databases manually by using full backup files.	Working with backups (p. 328) Backing up and restoring an Amazon RDS DB instance (p. 327)
Importing and exporting data Import data from other MySQL DB instances, MySQL instances running external to Amazon RDS, and other types of data sources, and export data to MySQL instances running external to Amazon RDS.	Restoring a backup into a MySQL DB instance (p. 871)
Monitoring a MySQL DB instance Monitor your MySQL DB instance by using Amazon CloudWatch RDS metrics, events, and Enhanced Monitoring. View log files for your MySQL DB instance.	Monitoring an Amazon RDS DB instance (p. 399) Viewing DB instance metrics (p. 548) Viewing Amazon RDS events (p. 503) Accessing Amazon RDS database log files (p. 504) Accessing MySQL database log files (p. 519)
Replicating your data Create a MySQL read replica, in the same AWS Region or a different one. You can use read replicas for load balancing, disaster recovery, and processing read-heavy database workloads, such as for analysis and reporting.	Working with read replicas (p. 278) Replication with a MySQL or MariaDB instance running external to Amazon RDS (p. 914)

There are also several sections with useful information about working with MySQL DB instances:

- [Common DBA tasks for MySQL DB instances \(p. 933\)](#)
- [Options for MySQL DB instances \(p. 925\)](#)
- [MySQL on Amazon RDS SQL reference \(p. 952\)](#)

MySQL on Amazon RDS versions

For MySQL, version numbers are organized as version = X.Y.Z. In Amazon RDS terminology, X.Y denotes the major version, and Z is the minor version number. For Amazon RDS implementations, a version change is considered major if the major version number changes—for example, going from version 5.7 to 8.0. A version change is considered minor if only the minor version number changes—for example, going from version 5.7.16 to 5.7.21.

Amazon RDS currently supports the following versions of MySQL:

Major version	Minor version
MySQL 8.0	<ul style="list-style-type: none">• 8.0.23• 8.0.21• 8.0.20• 8.0.19• 8.0.17• 8.0.16• 8.0.15• 8.0.13• 8.0.11
MySQL 5.7	<ul style="list-style-type: none">• 5.7.33• 5.7.31• 5.7.30• 5.7.28• 5.7.26• 5.7.25• 5.7.24• 5.7.23• 5.7.22• 5.7.21• 5.7.19• 5.7.17• 5.7.16
MySQL 5.6	<ul style="list-style-type: none">• 5.6.51• 5.6.49• 5.6.48• 5.6.46• 5.6.44• 5.6.43• 5.6.41• 5.6.40• 5.6.39• 5.6.37• 5.6.35• 5.6.34
MySQL 5.5	<ul style="list-style-type: none">• 5.5.62• 5.5.61• 5.5.59• 5.5.57• 5.5.54• 5.5.53• 5.5.46

You can specify any currently supported MySQL version when creating a new DB instance. You can specify the major version (such as MySQL 5.7), and any supported minor version for the specified major version. If no version is specified, Amazon RDS defaults to a supported version, typically the most recent version. If a major version is specified but a minor version is not, Amazon RDS defaults to a recent release of the major version you have specified. To see a list of supported versions, as well as defaults for newly created DB instances, use the [describe-db-engine-versions](#) AWS CLI command.

The default MySQL version might vary by AWS Region. To create a DB instance with a specific minor version, specify the minor version during DB instance creation. You can determine the default minor version for an AWS Region using the following AWS CLI command:

```
aws rds describe-db-engine-versions --default-only --engine mysql --engine-version major-engine-version --region region --query "*[].{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

Replace *major-engine-version* with the major engine version, and replace *region* with the AWS Region. For example, the following AWS CLI command returns the default MySQL minor engine version for the 5.7 major version and the US West (Oregon) AWS Region (us-west-2):

```
aws rds describe-db-engine-versions --default-only --engine mysql --engine-version 5.7 --region us-west-2 --query '*[].{Engine:Engine,EngineVersion:EngineVersion}' --output text
```

With Amazon RDS, you control when to upgrade your MySQL instance to a new major version supported by Amazon RDS. You can maintain compatibility with specific MySQL versions, test new versions with your application before deploying in production, and perform major version upgrades at times that best fit your schedule.

When automatic minor version upgrade is enabled, your DB instance will be upgraded automatically to new MySQL minor versions as they are supported by Amazon RDS. This patching occurs during your scheduled maintenance window. You can modify a DB instance to enable or disable automatic minor version upgrades.

If you opt out of automatically scheduled upgrades, you can manually upgrade to a supported minor version release by following the same procedure as you would for a major version update. For information, see [Upgrading a DB instance engine version \(p. 271\)](#).

Amazon RDS currently supports the major version upgrades from MySQL version 5.5 to version 5.6, from MySQL version 5.6 to version 5.7, and from MySQL version 5.7 to version 8.0. Because major version upgrades involve some compatibility risk, they do not occur automatically; you must make a request to modify the DB instance. You should thoroughly test any upgrade before upgrading your production instances. For information about upgrading a MySQL DB instance, see [Upgrading the MySQL DB engine \(p. 853\)](#).

You can test a DB instance against a new version before upgrading by creating a DB snapshot of your existing DB instance, restoring from the DB snapshot to create a new DB instance, and then initiating a version upgrade for the new DB instance. You can then experiment safely on the upgraded clone of your DB instance before deciding whether or not to upgrade your original DB instance.

Deprecation of MySQL version 5.6

On August 3, 2021, Amazon RDS plans to deprecate support for MySQL 5.6 using the following schedule, which includes upgrade recommendations. For more information, see [Upgrading the MySQL DB engine \(p. 853\)](#).

Action or recommendation	Dates
We recommend that you upgrade MySQL 5.6 DB instances manually to the version of your choice.	Now–August 3, 2021
We recommend that you upgrade MySQL 5.6 snapshots manually to the version of your choice.	Now–August 3, 2021
You can no longer create new MySQL 5.6 DB instances.	April 1, 2021
Amazon RDS starts automatic upgrades of your MySQL 5.6 DB instances to version 5.7.	August 3, 2021
Amazon RDS starts automatic upgrades to version 5.7 for any MySQL 5.6 DB instances restored from snapshots.	August 3, 2021
Amazon RDS automatically upgrades any remaining MySQL 5.6 DB instances to version 5.7 whether or not they are in a maintenance window.	September 1, 2021

For more information, see [Announcement: Amazon Relational Database Service \(RDS\) for MySQL 5.6 End-of-Life date is August 3, 2021](#).

Deprecation of MySQL version 5.5

On May 25, 2021, Amazon RDS plans to deprecate support for MySQL 5.5 using the following schedule, which includes upgrade recommendations. For more information, see [Upgrading the MySQL DB engine \(p. 853\)](#).

Action or recommendation	Dates
We recommend that you upgrade MySQL 5.5 DB instances manually to the version of your choice.	Now–May 25, 2021
We recommend that you upgrade MySQL 5.5 snapshots manually to the version of your choice.	Now–May 25, 2021
You can no longer create new MySQL 5.5 DB instances.	December 3, 2020
Amazon RDS starts automatic upgrades of your MySQL 5.5 DB instances to version 5.6.	March 29, 2021
Amazon RDS starts automatic upgrades to version 5.6 for any MySQL 5.5 DB instances restored from snapshots.	March 29, 2021
Amazon RDS automatically upgrades any remaining MySQL 5.5 DB instances to version 5.6 whether or not they are in a maintenance window.	May 25, 2021

For more information, see [Announcement: Extending end-of-life process for Amazon RDS for MySQL 5.5](#).

MySQL features not supported by Amazon RDS

Amazon RDS doesn't currently support the following MySQL features:

- Authentication Plugin
- Error Logging to the System Log
- Group Replication Plugin
- InnoDB Tablespace Encryption
- MariaDB Audit Plugin (not supported for RDS for MySQL version 8.0 only)

The MariaDB Audit Plugin is supported for RDS for MySQL version 5.5, 5.6, and 5.7.

- Password Strength Plugin
- Persisted system variables
- Semisynchronous replication
- Transportable tablespace
- X Plugin

Note

Global transaction IDs are supported for MySQL 5.7.23 and later MySQL 5.7 versions. Global transaction IDs are not supported for RDS for MySQL 5.5, 5.6, or 8.0.

IAM database authentication is supported for MySQL for MySQL 5.6, 5.7, and 8.0. IAM database authentication is not supported for MySQL 5.5.

Amazon RDS Performance Insights is supported for MySQL 5.6, 5.7, and 8.0. Amazon RDS Performance Insights is not supported for MySQL 5.5.

To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances. It also restricts access to certain system procedures and tables that require advanced privileges. Amazon RDS supports access to databases on a DB instance using any standard SQL client application. Amazon RDS doesn't allow direct host access to a DB instance by using Telnet, Secure Shell (SSH), or Windows Remote Desktop Connection. When you create a DB instance, you are assigned to the *db_owner* role for all databases on that instance, and you have all database-level permissions except for those used for backups. Amazon RDS manages backups for you.

Supported storage engines for MySQL on Amazon RDS

While MySQL supports multiple storage engines with varying capabilities, not all of them are optimized for recovery and data durability. Amazon RDS fully supports the InnoDB storage engine for MySQL DB instances. Amazon RDS features such as Point-In-Time restore and snapshot restore require a recoverable storage engine and are supported for the InnoDB storage engine only. You must be running an instance of MySQL 5.6 or later to use the InnoDB memcached interface. For more information, see [MySQL memcached support \(p. 929\)](#).

The Federated Storage Engine is currently not supported by Amazon RDS for MySQL.

For user-created schemas, the MyISAM storage engine does not support reliable recovery and can result in lost or corrupt data when MySQL is restarted after a recovery, preventing Point-In-Time restore or snapshot restore from working as intended. However, if you still choose to use MyISAM with Amazon RDS, snapshots can be helpful under some conditions.

Note

System tables in the `mysql` schema can be in MyISAM storage.

If you want to convert existing MyISAM tables to InnoDB tables, you can use the `ALTER TABLE` command (for example, `alter table TABLE_NAME engine=innodb;`). Bear in mind that MyISAM and InnoDB have different strengths and weaknesses, so you should fully evaluate the impact of making this switch on your applications before doing so.

MySQL 5.1 is no longer supported in Amazon RDS. However, you can restore existing MySQL 5.1 snapshots. When you restore a MySQL 5.1 snapshot, the instance is automatically upgraded to MySQL 5.5.

MySQL security on Amazon RDS

Security for MySQL DB instances is managed at three levels:

- AWS Identity and Access Management controls who can perform Amazon RDS management actions on DB instances. When you connect to AWS using IAM credentials, your IAM account must have IAM policies that grant the permissions required to perform Amazon RDS management operations. For more information, see [Identity and access management in Amazon RDS \(p. 1644\)](#).
- When you create a DB instance, you use either a VPC security group or a DB security group to control which devices and Amazon EC2 instances can open connections to the endpoint and port of the DB instance. These connections can be made using Secure Sockets Layer (SSL). In addition, firewall rules at your company can control whether devices running at your company can open connections to the DB instance.
- To authenticate login and permissions for a MySQL DB instance, you can take either of the following approaches, or a combination of them.

You can take the same approach as with a stand-alone instance of MySQL. Commands such as `CREATE USER`, `RENAME USER`, `GRANT`, `REVOKE`, and `SET PASSWORD` work just as they do in on-premises databases, as does directly modifying database schema tables. For information, see [Access control and account management](#) in the MySQL documentation.

You can also use IAM database authentication. With IAM database authentication, you authenticate to your DB instance by using an IAM user or IAM role and an authentication token. An *authentication token* is a unique value that is generated using the Signature Version 4 signing process. By using IAM database authentication, you can use the same credentials to control access to your AWS resources and your databases. For more information, see [IAM database authentication for MySQL and PostgreSQL \(p. 1660\)](#).

When you create an Amazon RDS DB instance, the master user has the following default privileges:

- `alter`
- `alter routine`
- `create`
- `create routine`
- `create temporary tables`
- `create user`
- `create view`
- `delete`
- `drop`

- event
- execute
- grant option
- index
- insert
- lock tables
- process
- references
- replication client
- replication slave (MySQL 5.6 and later)
- select
- show databases
- show view
- trigger
- update

Note

Although it is possible to delete the master user on the DB instance, it is not recommended. To recreate the master user, use the [ModifyDBInstance](#) RDS API operation or the [modify-db-instance](#) AWS CLI command and specify a new master user password with the appropriate parameter. If the master user does not exist in the instance, the master user is created with the specified password.

To provide management services for each DB instance, the `rdsadmin` user is created when the DB instance is created. Attempting to drop, rename, change the password, or change privileges for the `rdsadmin` account will result in an error.

To allow management of the DB instance, the standard `kill` and `kill_query` commands have been restricted. The Amazon RDS commands `rds_kill` and `rds_kill_query` are provided to allow you to end user sessions or queries on DB instances.

Using the Password Validation Plugin

MySQL provides the `validate_password` plugin for improved security. The plugin enforces password policies using parameters in the DB parameter group for your MySQL DB instance. The plugin is supported for DB instances running MySQL version 5.6, 5.7, and 8.0. For more information about the `validate_password` plugin, see [The Password Validation Plugin](#) in the MySQL documentation.

To enable the `validate_password` plugin for a MySQL DB instance

1. Connect to your MySQL DB instance and run the following command.

```
INSTALL PLUGIN validate_password SONAME 'validate_password.so';
```

2. Configure the parameters for the plugin in the DB parameter group used by the DB instance.

For more information about the parameters, see [Password Validation Plugin options and variables](#) in the MySQL documentation.

For more information about modifying DB instance parameters, see [Modifying parameters in a DB parameter group \(p. 232\)](#).

After installing and enabling the `password_validate` plugin, reset existing passwords to comply with your new validation policies.

Amazon RDS doesn't validate passwords. The MySQL DB instance performs password validation. If you set a user password with the AWS Management Console, the `modify-db-instance` AWS CLI command, or the `ModifyDBInstance` RDS API operation, the change can succeed even if the new password doesn't satisfy your password policies. However, a new password is set in the MySQL DB instance only if it satisfies the password policies. In this case, Amazon RDS records the following event.

```
"RDS-EVENT-0067" - An attempt to reset the master password for the DB instance has failed.
```

For more information about Amazon RDS events, see [Using Amazon RDS event notification \(p. 487\)](#).

Using SSL with a MySQL DB instance

Amazon RDS supports Secure Sockets Layer (SSL) connections with DB instances running the MySQL database engine.

Amazon RDS creates an SSL certificate and installs the certificate on the DB instance when Amazon RDS provisions the instance. These certificates are signed by a certificate authority. The SSL certificate includes the DB instance endpoint as the Common Name (CN) for the SSL certificate to guard against spoofing attacks.

An SSL certificate created by Amazon RDS is the trusted root entity and should work in most cases but might fail if your application does not accept certificate chains. If your application does not accept certificate chains, you might need to use an intermediate certificate to connect to your AWS Region. For example, you must use an intermediate certificate to connect to the AWS GovCloud (US) Regions using SSL.

For information about downloading certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#). For more information about using SSL with MySQL, see [Updating applications to connect to MySQL DB instances using new SSL/TLS certificates \(p. 848\)](#).

MySQL uses yaSSL for secure connections in the following versions:

- MySQL version 5.7.19 and earlier 5.7 versions
- MySQL version 5.6.37 and earlier 5.6 versions
- MySQL version 5.5.57 and earlier 5.5 versions

MySQL uses OpenSSL for secure connections in the following versions:

- MySQL version 8.0
- MySQL version 5.7.21 and later 5.7 versions
- MySQL version 5.6.39 and later 5.6 versions
- MySQL version 5.5.59 and later 5.5 versions

Amazon RDS for MySQL supports Transport Layer Security (TLS) versions 1.0, 1.1, and 1.2. The following table shows the TLS support for MySQL versions.

MySQL version	TLS 1.0	TLS 1.1	TLS 1.2
MySQL 8.0	Supported	Supported	Supported
MySQL 5.7	Supported	Supported	Supported for MySQL 5.7.21 and later
MySQL 5.6	Supported	Supported for MySQL 5.6.46 and later	Supported for MySQL 5.6.46 and later
MySQL 5.5	Supported	Not supported	Not supported

To encrypt connections using the default `mysql` client, launch the `mysql` client using the `--ssl-ca` parameter to reference the public key, as shown in the examples following.

The following example shows how to launch the client using the `--ssl-ca` parameter for MySQL 5.7 and later.

```
mysql -h myinstance.c9akciq32.rds-us-east-1.amazonaws.com
--ssl-ca=[full path]rds-combined-ca-bundle.pem --ssl-mode=VERIFY_IDENTITY
```

The following example shows how to launch the client using the `--ssl-ca` parameter for MySQL 5.6 and earlier.

```
mysql -h myinstance.c9akciq32.rds-us-east-1.amazonaws.com
--ssl-ca=[full path]rds-combined-ca-bundle.pem --ssl-verify-server-cert
```

You can require SSL connections for specific user accounts. For example, you can use one of the following statements, depending on your MySQL version, to require SSL connections on the user account `encrypted_user`.

For MySQL 5.7 and later, use the following statement.

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

For MySQL 5.6 and earlier, use the following statement.

```
GRANT USAGE ON *.* TO 'encrypted_user'@'%' REQUIRE SSL;
```

For more information on SSL connections with MySQL, see the [Using encrypted connections](#) in the MySQL documentation.

Using memcached and other options with MySQL

Most Amazon RDS DB engines support option groups that allow you to select additional features for your DB instance. DB instances on MySQL version 5.6 and later support the `memcached` option, a simple,

key-based cache. For more information about memcached and other options, see [Options for MySQL DB instances \(p. 925\)](#). For more information about working with option groups, see [Working with option groups \(p. 212\)](#).

InnoDB cache warming

InnoDB cache warming can provide performance gains for your MySQL DB instance by saving the current state of the buffer pool when the DB instance is shut down, and then reloading the buffer pool from the saved information when the DB instance starts up. This bypasses the need for the buffer pool to "warm up" from normal database use and instead preloads the buffer pool with the pages for known common queries. The file that stores the saved buffer pool information only stores metadata for the pages that are in the buffer pool, and not the pages themselves. As a result, the file does not require much storage space. The file size is about 0.2 percent of the cache size. For example, for a 64 GiB cache, the cache warming file size is 128 MiB. For more information on InnoDB cache warming, see [Saving and restoring the buffer pool state](#) in the MySQL documentation.

MySQL on Amazon RDS supports InnoDB cache warming for MySQL version 5.6 and later. To enable InnoDB cache warming, set the `innodb_buffer_pool_dump_at_shutdown` and `innodb_buffer_pool_load_at_startup` parameters to 1 in the parameter group for your DB instance. Changing these parameter values in a parameter group will affect all MySQL DB instances that use that parameter group. To enable InnoDB cache warming for specific MySQL DB instances, you might need to create a new parameter group for those instances. For information on parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

InnoDB cache warming primarily provides a performance benefit for DB instances that use standard storage. If you use PIOPS storage, you do not commonly see a significant performance benefit.

Important

If your MySQL DB instance does not shut down normally, such as during a failover, then the buffer pool state will not be saved to disk. In this case, MySQL loads whatever buffer pool file is available when the DB instance is restarted. No harm is done, but the restored buffer pool might not reflect the most recent state of the buffer pool prior to the restart. To ensure that you have a recent state of the buffer pool available to warm the InnoDB cache on startup, we recommend that you periodically dump the buffer pool "on demand." You can dump or load the buffer pool on demand if your DB instance is running MySQL version 5.6.19 or later.

You can create an event to dump the buffer pool automatically and on a regular interval. For example, the following statement creates an event named `periodic_buffer_pool_dump` that dumps the buffer pool every hour.

```
CREATE EVENT periodic_buffer_pool_dump
ON SCHEDULE EVERY 1 HOUR
DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

For more information on MySQL events, see [Event syntax](#) in the MySQL documentation.

Dumping and loading the buffer pool on demand

For MySQL version 5.6.19 and later, you can save and load the InnoDB cache "on demand."

- To dump the current state of the buffer pool to disk, call the [mysql.rds_innodb_buffer_pool_dump_now \(p. 971\)](#) stored procedure.
- To load the saved state of the buffer pool from disk, call the [mysql.rds_innodb_buffer_pool_load_now \(p. 971\)](#) stored procedure.
- To cancel a load operation in progress, call the [mysql.rds_innodb_buffer_pool_load_abort \(p. 972\)](#) stored procedure.

Local time zone for MySQL DB instances

By default, the time zone for a MySQL DB instance is Universal Time Coordinated (UTC). You can set the time zone for your DB instance to the local time zone for your application instead.

To set the local time zone for a DB instance, set the `time_zone` parameter in the parameter group for your DB instance to one of the supported values listed later in this section. When you set the `time_zone` parameter for a parameter group, all DB instances and read replicas that are using that parameter group change to use the new local time zone. For information on setting parameters in a parameter group, see [Working with DB parameter groups \(p. 228\)](#).

After you set the local time zone, all new connections to the database reflect the change. If you have any open connections to your database when you change the local time zone, you won't see the local time zone update until after you close the connection and open a new connection.

You can set a different local time zone for a DB instance and one or more of its read replicas. To do this, use a different parameter group for the DB instance and the replica or replicas and set the `time_zone` parameter in each parameter group to a different local time zone.

If you are replicating across AWS Regions, then the source DB instance and the read replica use different parameter groups (parameter groups are unique to an AWS Region). To use the same local time zone for each instance, you must set the `time_zone` parameter in the instance's and read replica's parameter groups.

When you restore a DB instance from a DB snapshot, the local time zone is set to UTC. You can update the time zone to your local time zone after the restore is complete. If you restore a DB instance to a point in time, then the local time zone for the restored DB instance is the time zone setting from the parameter group of the restored DB instance.

You can set your local time zone to one of the following values.

Africa/Cairo	Asia/Bangkok	Australia/Darwin
Africa/Casablanca	Asia/Beirut	Australia/Hobart
Africa/Harare	Asia/Calcutta	Australia/Perth
Africa/Monrovia	Asia/Damascus	Australia/Sydney
Africa/Nairobi	Asia/Dhaka	Brazil/East
Africa/Tripoli	Asia/Irkutsk	Canada/Newfoundland
Africa/Windhoek	Asia/Jerusalem	Canada/Saskatchewan
America/Araguaina	Asia/Kabul	Europe/Amsterdam
America/Asuncion	Asia/Karachi	Europe/Athens
America/Bogota	Asia/Kathmandu	Europe/Dublin
America/Caracas	Asia/Krasnoyarsk	Europe/Helsinki
America/Chihuahua	Asia/Magadan	Europe/Istanbul
America/Cuiaba	Asia/Muscat	Europe/Kaliningrad
America/Denver	Asia/Novosibirsk	Europe/Moscow
America/Fortaleza	Asia/Riyadh	Europe/Paris

America/Guatemala	Asia/Seoul	Europe/Prague
America/Halifax	Asia/Shanghai	Europe/Sarajevo
America/Manaus	Asia/Singapore	Pacific/Auckland
America/Matamoros	Asia/Taipei	Pacific/Fiji
America/Monterrey	Asia/Tehran	Pacific/Guam
America/Montevideo	Asia/Tokyo	Pacific/Honolulu
America/Phoenix	Asia/Ulaanbaatar	Pacific/Samoa
America/Santiago	Asia/Vladivostok	US/Alaska
America/Tijuana	Asia/Yakutsk	US/Central
Asia/Amman	Asia/Yerevan	US/Eastern
Asia/Ashgabat	Atlantic/Azores	US/East-Indiana
Asia/Baghdad	Australia/Adelaide	US/Pacific
Asia/Baku	Australia/Brisbane	UTC

Known issues and limitations for MySQL on Amazon RDS

There are some known issues and limitations for working with MySQL on Amazon RDS. For more information, see [Known issues and limitations for MySQL on Amazon RDS \(p. 948\)](#).

Deprecated MySQL on Amazon RDS versions

MySQL on Amazon RDS version 5.1 is deprecated.

For information about the Amazon RDS deprecation policy for MySQL, see [Amazon RDS FAQs](#).

Connecting to a DB instance running the MySQL database engine

Before you can connect to a DB instance running the MySQL database engine, you must create a DB instance. For information, see [Creating an Amazon RDS DB instance \(p. 141\)](#). After Amazon RDS provisions your DB instance, you can use any standard MySQL client application or utility to connect to the instance. In the connection string, you specify the DNS address from the DB instance endpoint as the host parameter, and specify the port number from the DB instance endpoint as the port parameter.

To authenticate to your RDS DB instance, you can use one of the authentication methods for MySQL and AWS Identity and Access Management (IAM) database authentication:

- To learn how to authenticate to MySQL using one of the authentication methods for MySQL, see [Authentication method](#) in the MySQL documentation.
- To learn how to authenticate to MySQL using IAM database authentication, see [IAM database authentication for MySQL and PostgreSQL \(p. 1660\)](#).

You can connect to a MySQL DB instance by using tools like the MySQL command line utility. For more information on using the MySQL client, see [mysql - the MySQL command-line client](#) in the MySQL documentation. One GUI-based application you can use to connect is MySQL Workbench. For more information, see the [Download MySQL Workbench](#) page. For information about installing MySQL (including the MySQL client), see [Installing and upgrading MySQL](#).

To connect to a DB instance from outside of its Amazon VPC, the DB instance must be publicly accessible, access must be granted using the inbound rules of the DB instance's security group, and other requirements must be met. For more information, see [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).

You can use Secure Sockets Layer (SSL) encryption on connections to a MySQL DB instance. For information, see [Using SSL with a MySQL DB instance \(p. 835\)](#). If you are using AWS Identity and Access Management (IAM) database authentication, make sure to use an SSL connection. For information, see [IAM database authentication for MySQL and PostgreSQL \(p. 1660\)](#).

You can also connect to a DB instance from a web server. For more information, see [Tutorial: Create a web server and an Amazon RDS DB instance \(p. 108\)](#).

Note

For information on connecting to a MariaDB DB instance, see [Connecting to a DB instance running the MariaDB database engine \(p. 588\)](#).

Topics

- [Finding the connection information for a MySQL DB instance \(p. 840\)](#)
- [Connecting from the MySQL client \(p. 843\)](#)
- [Connecting with SSL \(p. 844\)](#)
- [Connecting from MySQL Workbench \(p. 844\)](#)
- [Troubleshooting connections to your MySQL DB instance \(p. 846\)](#)

Finding the connection information for a MySQL DB instance

The connection information for a DB instance includes its endpoint, port, and a valid database user, such as the master user. For example, suppose that an endpoint value is `mydb.123456789012.us-`

`east-1.rds.amazonaws.com`. In this case, the port value is 3306, and the database user is `admin`. Given this information, you specify the following values in a connection string:

- For host or host name or DNS name, specify `mydb.123456789012.us-east-1.rds.amazonaws.com`.
- For port, specify 3306.
- For user, specify `admin`.

To connect to a DB instance, use any client for a DB engine. For example, you might use the `mysql` utility to connect to a MariaDB or MySQL DB instance. You might use Microsoft SQL Server Management Studio to connect to a SQL Server DB instance. You might use Oracle SQL Developer to connect to an Oracle DB instance, or the `psql` command line utility to connect to a PostgreSQL DB instance.

To find the connection information for a DB instance, you can use the AWS Management Console, the AWS CLI `describe-db-instances` command, or the Amazon RDS API `DescribeDBInstances` operation to list its details.

Console

To find the connection information for a DB instance in the AWS Management Console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases** to display a list of your DB instances.
3. Choose the name of the MySQL DB instance to display its details.
4. On the **Connectivity & security** tab, copy the endpoint. Also, note the port number. You need both the endpoint and the port number to connect to the DB instance.

The screenshot shows the AWS RDS console for a database named 'mydb'. The 'Summary' tab is selected, displaying basic information like DB identifier, Role, and instance status. Below it, the 'Connectivity & security' tab is active, showing the endpoint and port details which are highlighted with red circles.

Category	Value	Notes
DB identifier	mydb	
Role	Instance	
CPU	2.33%	Current activity: 0 Connections
Connectivity & security	Endpoint: mydb. [REDACTED] .us-east-1.rds.amazonaws.com Port: 3306	Network: [REDACTED] , Availability: us-east-1, VPC: vpc-6f..., Subnet: default

5. If you need to find the master user name, choose the **Configuration** tab and view the **Master username** value.

AWS CLI

To find the connection information for a MySQL DB instance by using the AWS CLI, call the [describe-db-instances](#) command. In the call, query for the DB instance ID, endpoint, port, and master user name.

For Linux, macOS, or Unix:

```
aws rds describe-db-instances \
--filters "Name=engine,Values=mysql" \
--query "[].{DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername}"
```

For Windows:

```
aws rds describe-db-instances ^
--filters "Name=engine,Values=mysql" ^
--query "[].{DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername}"
```

Your output should be similar to the following.

```
[  
  [  
    "mydb1",  
    "mydb1.123456789012.us-east-1.rds.amazonaws.com",  
    3306,  
    "admin"  
  ],  
  [  
    "mydb2",  
    "mydb2.123456789012.us-east-1.rds.amazonaws.com",  
    3306,  
    "admin"  
  ]  
]
```

RDS API

To find the connection information for a DB instance by using the Amazon RDS API, call the [DescribeDBInstances](#) operation. In the output, find the values for the endpoint address, endpoint port, and master user name.

Connecting from the MySQL client

To connect to a DB instance using the MySQL client, enter the following command at a command prompt to connect to a DB instance using the MySQL client. For the -h parameter, substitute the DNS name (endpoint) for your DB instance. For the -P parameter, substitute the port for your DB instance. For the -u parameter, substitute the user name of a valid database user, such as the master user. Enter the master user password when prompted.

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com -P 3306 -u mymasteruser -  
p
```

After you enter the password for the user, you should see output similar to the following.

```
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 350  
Server version: 5.6.40-log MySQL Community Server (GPL)  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql>
```

Connecting with SSL

Amazon RDS creates an SSL certificate for your DB instance when the instance is created. If you enable SSL certificate verification, then the SSL certificate includes the DB instance endpoint as the Common Name (CN) for the SSL certificate to guard against spoofing attacks. To connect to your DB instance using SSL, you can use native password authentication or IAM database authentication. To connect to your DB instance using IAM database authentication, see [IAM database authentication for MySQL and PostgreSQL \(p. 1660\)](#). To connect to your DB instance using native password authentication, you can follow these steps:

To connect to a DB instance with SSL using the MySQL client

1. Download a root certificate that works for all AWS Regions.

For information about downloading certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

2. Enter the following command at a command prompt to connect to a DB instance with SSL using the MySQL client. For the -h parameter, substitute the DNS name (endpoint) for your DB instance. For the --ssl-ca parameter, substitute the SSL certificate file name as appropriate. For the -P parameter, substitute the port for your DB instance. For the -u parameter, substitute the user name of a valid database user, such as the master user. Enter the master user password when prompted.

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=rds-ca-2015-root.pem -P 3306 -u mymasteruser -p
```

3. You can require that the SSL connection verifies the DB instance endpoint against the endpoint in the SSL certificate.

For MySQL 5.7 and later:

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=rds-ca-2015-root.pem --ssl-mode=VERIFY_IDENTITY -P 3306 -u mymasteruser -p
```

For MySQL 5.6 and earlier:

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=rds-ca-2015-root.pem --ssl-verify-server-cert -P 3306 -u mymasteruser -p
```

4. Enter the master user password when prompted.

You will see output similar to the following.

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 350
Server version: 5.6.40-log MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Connecting from MySQL Workbench

To connect from MySQL Workbench

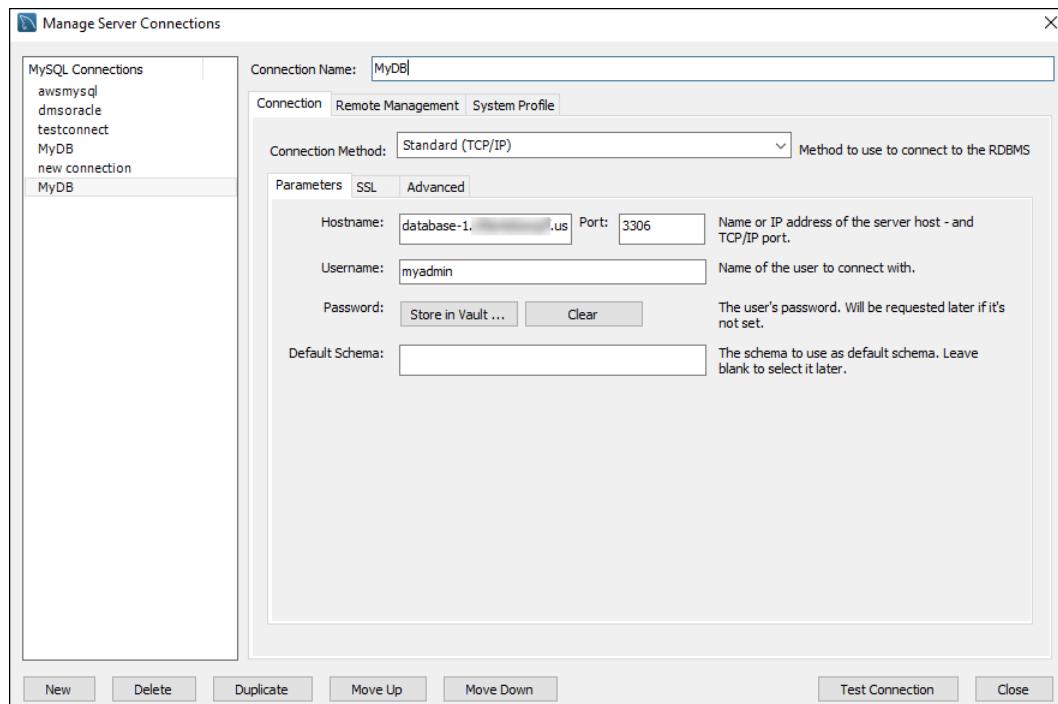
1. Download and install MySQL Workbench at [Download MySQL Workbench](#).

2. Open MySQL Workbench.



3. From **Database**, choose **Manage Connections**.
4. In the **Manage Server Connections** window, choose **New**.
5. In the **Connect to Database** window, enter the following information:
 - **Stored Connection** – Enter a name for the connection, such as **MyDB**.
 - **Hostname** – Enter the DB instance endpoint.
 - **Port** – Enter the port used by the DB instance.
 - **Username** – Enter the user name of a valid database user, such as the master user.
 - **Password** – Optionally, choose **Store in Vault** and then enter and save the password for the user.

The window looks similar to the following:



You can use the features of MySQL Workbench to customize connections. For example, you can use the **SSL** tab to configure SSL connections. For information about using MySQL Workbench, see the [MySQL Workbench documentation](#).

6. Optionally, choose **Test Connection** to confirm that the connection to the DB instance is successful.
7. Choose **Close**.
8. From **Database**, choose **Connect to Database**.
9. From **Stored Connection**, choose your connection.
10. Choose **OK**.

Troubleshooting connections to your MySQL DB instance

Two common causes of connection failures to a new DB instance are:

- The DB instance was created using a security group that doesn't authorize connections from the device or Amazon EC2 instance where the MySQL application or utility is running. If the DB instance was created in a VPC, it must have a VPC security group that authorizes the connections. For more information, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).

You can add or edit an inbound rule in the security group. For **Source**, choose **My IP**. This allows access to the DB instance from the IP address detected in your browser.

If the DB instance was created outside of a VPC, it must have a DB security group that authorizes the connections.

- The DB instance was created using the default port of 3306, and your company has firewall rules blocking connections to that port from devices in your company network. To fix this failure, recreate the instance with a different port.

For more information on connection issues, see [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).

Updating applications to connect to MySQL DB instances using new SSL/TLS certificates

As of September 19, 2019, Amazon RDS has published new Certificate Authority (CA) certificates for connecting to your RDS DB instances using Secure Socket Layer or Transport Layer Security (SSL/TLS). Following, you can find information about updating your applications to use the new certificates.

This topic can help you to determine whether any client applications use SSL/TLS to connect to your DB instances. If they do, you can further check whether those applications require certificate verification to connect.

Note

Some applications are configured to connect to MySQL DB instances only if they can successfully verify the certificate on the server. For such applications, you must update your client application trust stores to include the new CA certificates.

You can specify the following SSL modes: `disabled`, `preferred`, and `required`. When you use the `preferred` SSL mode and the CA certificate doesn't exist or isn't up to date, the following behavior applies:

- For newer MySQL minor versions, the connection falls back to not using SSL and still connects successfully.

Because these later versions use the OpenSSL protocol, an expired server certificate doesn't prevent successful connections unless the `required` SSL mode is specified.

The following MySQL minor versions use the OpenSSL protocol:

- All MySQL 8.0 versions
- MySQL 5.7.21 and later MySQL 5.7 versions
- MySQL 5.6.39 and later MySQL 5.6 versions
- MySQL 5.5.59 and later MySQL 5.5 versions
- For older MySQL minor versions, an error is returned.

Because these older versions use the yaSSL protocol, certificate verification is strictly enforced and the connection is unsuccessful.

The following MySQL minor versions use the yaSSL protocol:

- MySQL 5.7.19 and earlier MySQL 5.7 versions
- MySQL 5.6.37 and earlier MySQL 5.6 versions
- MySQL 5.5.57 and earlier MySQL 5.5 versions

After you update your CA certificates in the client application trust stores, you can rotate the certificates on your DB instances. We strongly recommend testing these procedures in a development or staging environment before implementing them in your production environments.

For more information about certificate rotation, see [Rotating your SSL/TLS certificate \(p. 1636\)](#). For more information about downloading certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#). For information about using SSL/TLS with MySQL DB instances, see [Using SSL with a MySQL DB instance \(p. 835\)](#).

Topics

- [Determining whether any applications are connecting to your MySQL DB instance using SSL \(p. 849\)](#)
- [Determining whether a client requires certificate verification to connect \(p. 849\)](#)

- [Updating your application trust store \(p. 850\)](#)
- [Example Java code for establishing SSL connections \(p. 851\)](#)

Determining whether any applications are connecting to your MySQL DB instance using SSL

If you are using Amazon RDS for MySQL version 5.7 or 8.0 and the Performance Schema is enabled, run the following query to check if connections are using SSL/TLS. For information about enabling the Performance Schema, see [Performance Schema quick start](#) in the MySQL documentation.

```
mysql> SELECT id, user, host, connection_type
    FROM performance_schema.threads pst
    INNER JOIN information_schema.processlist isp
    ON pst.processlist_id = isp.id;
```

In this sample output, you can see both your own session (`admin`) and an application logged in as `webapp1` are using SSL.

```
+---+-----+-----+-----+
| id | user           | host          | connection_type |
+---+-----+-----+-----+
|  8 | admin          | 10.0.4.249:42590 | SSL/TLS        |
|  4 | event_scheduler | localhost      | NULL          |
| 10 | webapp1        | 159.28.1.1:42189 | SSL/TLS        |
+---+-----+-----+-----+
3 rows in set (0.00 sec)
```

If you are using Amazon RDS for MySQL versions 5.5 or 5.6, then you can't determine from the server side whether applications are connecting with or without SSL. For those versions, you can determine whether SSL is used by examining the application's connection method. In the following section, you can find more information on examining the client connection configuration.

Determining whether a client requires certificate verification to connect

You can check whether JDBC clients and MySQL clients require certificate verification to connect.

JDBC

The following example with MySQL Connector/J 8.0 shows one way to check an application's JDBC connection properties to determine whether successful connections require a valid certificate. For more information on all of the JDBC connection options for MySQL, see [Configuration properties](#) in the MySQL documentation.

When using the MySQL Connector/J 8.0, an SSL connection requires verification against the server CA certificate if your connection properties have `sslMode` set to `VERIFY_CA` or `VERIFY_IDENTITY`, as in the following example.

```
Properties properties = new Properties();
properties.setProperty("sslMode", "VERIFY_IDENTITY");
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
```

Note

If you use either the MySQL Java Connector v5.1.38 or later, or the MySQL Java Connector v8.0.9 or later to connect to your databases, even if you haven't explicitly configured your applications to use SSL/TLS when connecting to your databases, these client drivers default to using SSL/TLS. In addition, when using SSL/TLS, they perform partial certificate verification and fail to connect if the database server certificate is expired.

MySQL

The following examples with the MySQL Client show two ways to check a script's MySQL connection to determine whether successful connections require a valid certificate. For more information on all of the connection options with the MySQL Client, see [Client-side configuration for encrypted connections](#) in the MySQL documentation.

When using the MySQL 5.7 or MySQL 8.0 Client, an SSL connection requires verification against the server CA certificate if for the --ssl-mode option you specify VERIFY_CA or VERIFY_IDENTITY, as in the following example.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem --
ssl-mode=VERIFY_CA
```

When using the MySQL 5.6 Client, an SSL connection requires verification against the server CA certificate if you specify the --ssl-verify-server-cert option, as in the following example.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem --
ssl-verify-server-cert
```

Updating your application trust store

For information about updating the trust store for MySQL applications, see [Installing SSL certificates](#) in the MySQL documentation.

Note

When you update the trust store, you can retain older certificates in addition to adding the new certificates.

Updating your application trust store for JDBC

You can update the trust store for applications that use JDBC for SSL/TLS connections.

To update the trust store for JDBC applications

1. Download the 2019 root certificate that works for all AWS Regions and put the file in the trust store directory.

For information about downloading the root certificate, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

2. Convert the certificate to .der format using the following command.

```
openssl x509 -outform der -in rds-ca-2019-root.pem -out rds-ca-2019-root.der
```

Replace the file name with the one that you downloaded.

3. Import the certificate into the key store using the following command.

```
keytool -import -alias rds-root -keystore clientkeystore -file rds-ca-2019-root.der
```

4. Confirm that the key store was updated successfully.

```
keytool -list -v -keystore clientkeystore.jks
```

Enter the key store password when you are prompted for it.

Your output should contain the following.

```
rds-root, date, trustedCertEntry,  
Certificate fingerprint (SHA1):  
D4:0D:DB:29:E3:75:0D:FF:A6:71:C3:14:0B:BF:5F:47:8D:1C:80:96  
# This fingerprint should match the output from the below command  
openssl x509 -fingerprint -in rds-ca-2019-root.pem -noout
```

If you are using the mysql JDBC driver in an application, set the following properties in the application.

```
System.setProperty("javax.net.ssl.trustStore", certs);  
System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

When you start the application, set the following properties.

```
java -Djavax.net.ssl.trustStore=/path_to_truststore/MyTruststore.jks -  
Djavax.net.ssl.trustStorePassword=my_truststore_password com.companyName.MyApplication
```

Example Java code for establishing SSL connections

The following code example shows how to set up the SSL connection that validates the server certificate using JDBC.

```
public class MySQLSSLTest {
```

```
private static final String DB_USER = "username";
private static final String DB_PASSWORD = "password";
// This key store has only the prod root ca.
private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";
private static final String KEY_STORE_PASS = "keystore-password";

public static void test(String[] args) throws Exception {
    Class.forName("com.mysql.jdbc.Driver");

    System.setProperty("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
    System.setProperty("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);

    Properties properties = new Properties();
    properties.setProperty("sslMode", "VERIFY_IDENTITY");
    properties.put("user", DB_USER);
    properties.put("password", DB_PASSWORD);

    Connection connection = null;
    Statement stmt = null;
    ResultSet rs = null;
    try {
        connection =
DriverManager.getConnection("jdbc:mysql://mydatabase.123456789012.us-
east-1.rds.amazonaws.com:3306",properties);
        stmt = connection.createStatement();
        rs=stmt.executeQuery("SELECT 1 from dual");
    } finally {
        if (rs != null) {
            try {
                rs.close();
            } catch (SQLException e) {
            }
        }
        if (stmt != null) {
            try {
                stmt.close();
            } catch (SQLException e) {
            }
        }
        if (connection != null) {
            try {
                connection.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
    return;
}
```

Important

After you have determined that your database connections use SSL/TLS and have updated your application trust store, you can update your database to use the rds-ca-2019 certificates. For instructions, see step 3 in [Updating your CA certificate by modifying your DB instance \(p. 1636\)](#).

Upgrading the MySQL DB engine

When Amazon RDS supports a new version of a database engine, you can upgrade your DB instances to the new version. There are two kinds of upgrades for MySQL DB instances: major version upgrades and minor version upgrades.

Major version upgrades can contain database changes that are not backward-compatible with existing applications. As a result, you must manually perform major version upgrades of your DB instances. You can initiate a major version upgrade by modifying your DB instance. However, before you perform a major version upgrade, we recommend that you follow the instructions in [Major version upgrades for MySQL \(p. 854\)](#).

In contrast, *minor version upgrades* include only changes that are backward-compatible with existing applications. You can initiate a minor version upgrade manually by modifying your DB instance. Or you can enable the **Auto minor version upgrade** option when creating or modifying a DB instance. Doing so means that your DB instance is automatically upgraded after Amazon RDS tests and approves the new version. For information about performing an upgrade, see [Upgrading a DB instance engine version \(p. 271\)](#).

If your MySQL DB instance is using read replicas, you must upgrade all of the read replicas before upgrading the source instance. If your DB instance is in a Multi-AZ deployment, both the primary and standby replicas are upgraded. Your DB instance will not be available until the upgrade is complete.

Topics

- [Overview of upgrading \(p. 853\)](#)
- [Major version upgrades for MySQL \(p. 854\)](#)
- [Testing an upgrade \(p. 858\)](#)
- [Upgrading a MySQL DB instance \(p. 858\)](#)
- [Automatic minor version upgrades for MySQL \(p. 858\)](#)
- [Using a read replica to reduce downtime when upgrading a MySQL database \(p. 860\)](#)

Overview of upgrading

When you use the AWS Management Console to upgrade a DB instance, it shows the valid upgrade targets for the DB instance. You can also use the following AWS CLI command to identify the valid upgrade targets for a DB instance:

For Linux, macOS, or Unix:

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version version-number \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

For Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version version-number ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

For example, to identify the valid upgrade targets for a MySQL version 5.6.43 DB instance, run the following AWS CLI command:

For Linux, macOS, or Unix:

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version 5.6.43 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

For Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version 5.6.43 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Amazon RDS takes two DB snapshots during the upgrade process. The first DB snapshot is of the DB instance before any upgrade changes have been made. If the upgrade doesn't work for your databases, you can restore this snapshot to create a DB instance running the old version. The second DB snapshot is taken when the upgrade completes.

Note

Amazon RDS only takes DB snapshots if you have set the backup retention period for your DB instance to a number greater than 0. To change your backup retention period, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

After the upgrade is complete, you can't revert to the previous version of the database engine. If you want to return to the previous version, restore the first DB snapshot taken to create a new DB instance.

You control when to upgrade your DB instance to a new version supported by Amazon RDS. This level of control helps you maintain compatibility with specific database versions and test new versions with your application before deploying in production. When you are ready, you can perform version upgrades at the times that best fit your schedule.

If your DB instance is using read replication, you must upgrade all of the Read Replicas before upgrading the source instance.

If your DB instance is in a Multi-AZ deployment, both the primary and standby DB instances are upgraded. The primary and standby DB instances are upgraded at the same time and you will experience an outage until the upgrade is complete. The time for the outage varies based on your database engine, engine version, and the size of your DB instance.

Major version upgrades for MySQL

Amazon RDS supports the following in-place upgrades for major versions of the MySQL database engine:

- MySQL 5.5 to MySQL 5.6
- MySQL 5.6 to MySQL 5.7
- MySQL 5.7 to MySQL 8.0

Note

You can only create MySQL version 5.7 and 8.0 DB instances with latest-generation and current-generation DB instance classes, in addition to the db.m3 previous-generation DB instance class. In some cases, you want to upgrade a MySQL version 5.6 DB instance running on a previous-generation DB instance class (other than db.m3) to a MySQL version 5.7 DB instance. In

these cases, first modify the DB instance to use a latest-generation or current-generation DB instance class. After you do this, you can then modify the DB instance to use the MySQL version 5.7 database engine. For information on Amazon RDS DB instance classes, see [DB instance classes \(p. 7\)](#).

Topics

- [Overview of MySQL major version upgrades \(p. 855\)](#)
- [Upgrades to MySQL version 5.7 might be slow \(p. 855\)](#)
- [Prechecks for upgrades from MySQL 5.7 to 8.0 \(p. 856\)](#)
- [Rollback after failure to upgrade from MySQL 5.7 to 8.0 \(p. 857\)](#)

Overview of MySQL major version upgrades

Major version upgrades can contain database changes that are not backward-compatible with existing applications. As a result, Amazon RDS doesn't apply major version upgrades automatically; you must manually modify your DB instance. We recommend that you thoroughly test any upgrade before applying it to your production instances.

To perform a major version upgrade for a MySQL version 5.5 DB instance on Amazon RDS to MySQL version 5.6 or later, first perform any available OS updates. After OS updates are complete, upgrade to each major version: 5.5 to 5.6, then 5.6 to 5.7, and then 5.7 to 8.0. MySQL DB instances created before April 24, 2014, show an available OS update until the update has been applied. For more information on OS updates, see [Applying updates for a DB instance \(p. 266\)](#).

During a major version upgrade of MySQL, Amazon RDS runs the MySQL binary `mysql_upgrade` to upgrade tables, if necessary. Also, Amazon RDS empties the `slow_log` and `general_log` tables during a major version upgrade. To preserve log information, save the log contents before the major version upgrade.

MySQL major version upgrades typically complete in about 10 minutes. Some upgrades might take longer because of the DB instance class size or because the instance doesn't follow certain operational guidelines in [Best practices for Amazon RDS \(p. 128\)](#). If you upgrade a DB instance from the Amazon RDS console, the status of the DB instance indicates when the upgrade is complete. If you upgrade using the AWS Command Line Interface (AWS CLI), use the `describe-db-instances` command and check the `Status` value.

Upgrades to MySQL version 5.7 might be slow

MySQL version 5.6.4 introduced a new date and time format for the `datetime`, `time`, and `timestamp` columns that allows fractional components in date and time values. When upgrading a DB instance to MySQL version 5.7, MySQL forces the conversion of all date and time column types to the new format.

Because this conversion rebuilds your tables, it might take a considerable amount of time to complete the DB instance upgrade. The forced conversion occurs for any DB instances that are running a version before MySQL version 5.6.4. It also occurs for any DB instances that were upgraded from a version before MySQL version 5.6.4 to a version other than 5.7.

If your DB instance runs a version before MySQL version 5.6.4, or was upgraded from a version before 5.6.4, we recommend an extra step. In these cases, we recommend that you convert the `datetime`, `time`, and `timestamp` columns in your database before upgrading your DB instance to MySQL version 5.7. This conversion can significantly reduce the amount of time required to upgrade the DB instance to MySQL version 5.7. To upgrade your date and time columns to the new format, issue the `ALTER TABLE <table_name> FORCE`; command for each table that contains date or time columns. Because altering a table locks the table as read-only, we recommend that you perform this update during a maintenance window.

To find all tables in your database that have `datetime`, `time`, or `timestamp` columns and create an `ALTER TABLE <table_name> FORCE;` command for each table, use the following query.

```
SELECT DISTINCT CONCAT('ALTER TABLE `',
    REPLACE(is_tables.TABLE_SCHEMA, '`', ``), `.``,
    REPLACE(is_tables.TABLE_NAME, '`', ``), `` FORCE;')
FROM information_schema.TABLES is_tables
INNER JOIN information_schema.COLUMNS col ON col.TABLE_SCHEMA =
is_tables.TABLE_SCHEMA
    AND col.TABLE_NAME = is_tables.TABLE_NAME
LEFT OUTER JOIN information_schema.INNODB_SYS_TABLES systables ON
    SUBSTRING_INDEX(systables.NAME, '#', 1) =
CONCAT(is_tables.TABLE_SCHEMA,'/',is_tables.TABLE_NAME)
    LEFT OUTER JOIN information_schema.INNODB_SYS_COLUMNS syscolumns ON
        syscolumns.TABLE_ID = systables.TABLE_ID AND syscolumns.NAME = col.COLUMN_NAME
WHERE col.COLUMN_TYPE IN ('time','timestamp','datetime')
    AND is_tables.TABLE_TYPE = 'BASE TABLE'
    AND is_tables.TABLE_SCHEMA NOT IN ('mysql','information_schema','performance_schema')
    AND (is_tables.ENGINE = 'InnoDB' AND syscolumns.MTYPE = 6);
```

Prechecks for upgrades from MySQL 5.7 to 8.0

MySQL 8.0 includes a number of incompatibilities with MySQL 5.7. These incompatibilities can cause problems during an upgrade from MySQL 5.7 to MySQL 8.0. So, some preparation might be required on your database for the upgrade to be successful. The following is a general list of these incompatibilities:

- There must be no tables that use obsolete data types or functions.
- There must be no orphan *.frm files.
- Triggers must not have a missing or empty definer or an invalid creation context.
- There must be no partitioned table that uses a storage engine that does not have native partitioning support.
- There must be no keyword or reserved word violations. Some keywords might be reserved in MySQL 8.0 that were not reserved previously.

For more information, see [Keywords and reserved words](#) in the MySQL documentation.

- There must be no tables in the MySQL 5.7 mysql system database that have the same name as a table used by the MySQL 8.0 data dictionary.
- There must be no obsolete SQL modes defined in your `sql_mode` system variable setting.
- There must be no tables or stored procedures with individual `ENUM` or `SET` column elements that exceed 255 characters or 1020 bytes in length.
- Before upgrading to MySQL 8.0.13 or higher, there must be no table partitions that reside in shared InnoDB tablespaces.
- There must be no queries and stored program definitions from MySQL 8.0.12 or lower that use `ASC` or `DESC` qualifiers for `GROUP BY` clauses.
- Your MySQL 5.7 installation must not use features that are not supported in MySQL 8.0.

For more information, see [Features removed in MySQL 8.0](#) in the MySQL documentation.

- There must be no foreign key constraint names longer than 64 characters.
- For improved Unicode support, consider converting objects that use the `utf8mb3` charset to use the `utf8mb4` charset. The `utf8mb3` character set is deprecated. Also, consider using `utf8mb4` for character set references instead of `utf8`, because currently `utf8` is an alias for the `utf8mb3` charset.

For more information, see [The utf8mb3 character set \(3-byte UTF-8 unicode encoding\)](#) in the MySQL documentation.

When you start an upgrade from MySQL 5.7 to 8.0, Amazon RDS runs prechecks automatically to detect these incompatibilities. For information about upgrading to MySQL 8.0, see [Upgrading MySQL](#) in the MySQL documentation.

These prechecks are mandatory. You can't choose to skip them. The prechecks provide the following benefits:

- They enable you to avoid unplanned downtime during the upgrade.
- If there are incompatibilities, Amazon RDS prevents the upgrade and provides a log for you to learn about them. You can then use the log to prepare your database for the upgrade to MySQL 8.0 by eliminating the incompatibilities. For detailed information about removing incompatibilities, see [Preparing your installation for upgrade](#) in the MySQL documentation and [Upgrading to MySQL 8.0? Here is what you need to know...](#) on the MySQL Server Blog.

The prechecks include some that are included with MySQL and some that were created specifically by the Amazon RDS team. For information about the prechecks provided by MySQL, see [Upgrade checker utility](#).

The prechecks run before the DB instance is stopped for the upgrade, meaning that they don't cause any downtime when they run. If the prechecks find an incompatibility, Amazon RDS automatically cancels the upgrade before the DB instance is stopped. Amazon RDS also generates an event for the incompatibility. For more information about Amazon RDS events, see [Using Amazon RDS event notification \(p. 487\)](#).

Amazon RDS records detailed information about each incompatibility in the log file `PrePatchCompatibility.log`. In most cases, the log entry includes a link to the MySQL documentation for correcting the incompatibility. For more information about viewing log files, see [Viewing and listing database log files \(p. 504\)](#).

Due to the nature of the prechecks, they analyze the objects in your database. This analysis results in resource consumption and increases the time for the upgrade to complete.

Note

Amazon RDS runs prechecks only for an upgrade from MySQL 5.7 to MySQL 8.0. They aren't run for upgrades to releases lower than MySQL 8.0. For example, prechecks aren't run for an upgrade from MySQL 5.6 to MySQL 5.7.

[Rollback after failure to upgrade from MySQL 5.7 to 8.0](#)

When you upgrade a DB instance from MySQL version 5.7 to MySQL version 8.0, the upgrade can fail. In particular, it can fail if the data dictionary contains incompatibilities that weren't captured by the prechecks. In this case, the database fails to start up successfully in the new MySQL 8.0 version. At this point, Amazon RDS rolls back the changes performed for the upgrade. After the rollback, the MySQL DB instance is running MySQL version 5.7. When an upgrade fails and is rolled back, Amazon RDS generates an event with the event ID RDS-EVENT-0188.

Typically, an upgrade fails because there are incompatibilities in the metadata between the databases in your DB instance and the target MySQL version. When an upgrade fails, you can view the details about these incompatibilities in the `upgradeFailure.log` file. Resolve the incompatibilities before attempting to upgrade again.

During an unsuccessful upgrade attempt and rollback, your DB instance is restarted. Any pending parameter changes are applied during the restart and persist after the rollback.

For more information about upgrading to MySQL 8.0, see the following topics in the MySQL documentation:

- [Preparing Your Installation for Upgrade](#)
- [Upgrading to MySQL 8.0? Here is what you need to know...](#)

Note

Currently, automatic rollback after upgrade failure is supported only for MySQL 5.7 to 8.0 major version upgrades.

Testing an upgrade

Before you perform a major version upgrade on your DB instance, thoroughly test your database for compatibility with the new version. In addition, thoroughly test all applications that access the database for compatibility with the new version. We recommend that you use the following procedure.

To test a major version upgrade

1. Review the upgrade documentation for the new version of the database engine to see if there are compatibility issues that might affect your database or applications:
 - [Changes in MySQL 5.6](#)
 - [Changes in MySQL 5.7](#)
 - [Changes in MySQL 8.0](#)
2. If your DB instance is a member of a custom DB parameter group, create a new DB parameter group with your existing settings that is compatible with the new major version. Specify the new DB parameter group when you upgrade your test instance, so your upgrade testing ensures that it works correctly. For more information about creating a DB parameter group, see [Working with DB parameter groups \(p. 228\)](#).
3. Create a DB snapshot of the DB instance to be upgraded. For more information, see [Creating a DB snapshot \(p. 346\)](#).
4. Restore the DB snapshot to create a new test DB instance. For more information, see [Restoring from a DB snapshot \(p. 349\)](#).
5. Modify this new test DB instance to upgrade it to the new version, using one of the methods detailed following. If you created a new parameter group in step 2, specify that parameter group.
6. Evaluate the storage used by the upgraded instance to determine if the upgrade requires additional storage.
7. Run as many of your quality assurance tests against the upgraded DB instance as needed to ensure that your database and application work correctly with the new version. Implement any new tests needed to evaluate the impact of any compatibility issues that you identified in step 1. Test all stored procedures and functions. Direct test versions of your applications to the upgraded DB instance.
8. If all tests pass, then perform the upgrade on your production DB instance. We recommend that you don't allow write operations to the DB instance until you confirm that everything is working correctly.

Upgrading a MySQL DB instance

For information about manually or automatically upgrading a MySQL DB instance, see [Upgrading a DB instance engine version \(p. 271\)](#).

Automatic minor version upgrades for MySQL

If you specify the following settings when creating or modifying a DB instance, you can have your DB instance automatically upgraded.

- The **Auto minor version upgrade** setting is enabled.
- The **Backup retention period** setting is greater than 0.

For more information about these settings, see [Settings for DB instances \(p. 251\)](#).

For some RDS for MySQL major versions in some AWS Regions, one minor version is designated by RDS as the automatic upgrade version. After a minor version has been tested and approved by Amazon RDS, the minor version upgrade occurs automatically during your maintenance window. RDS doesn't automatically set newer released minor versions as the automatic upgrade version. Before RDS designates a newer automatic upgrade version, several criteria are considered, such as the following:

- Known security issues
 - Bugs in the MySQL community version
 - Overall fleet stability since the minor version was released

You can use the following AWS CLI command and script to determine the current automatic minor upgrade target version for a specified MySQL minor version in a specific AWS Region.

```
aws rds describe-db-engine-versions --output=table --engine mysql --engine-version minor-version --region region
```

For example, the following AWS CLI command determines the automatic minor upgrade target for MySQL minor version 5.7.19 in the US East (Ohio) AWS Region (us-east-2).

```
aws rds describe-db-engine-versions --output=table --engine mysql --engine-version 5.7.19  
--region us-east-2
```

Your output is similar to the following.

```
|                                     DescribeDBEngineVersions
+-----+
||                                         DBEngineVersions
+-----+
|| DBEngineDescription                  | MySQL Community Edition
|| DBEngineVersionDescription          | mysql 5.7.19
|| DBParameterGroupFamily             | mysql5.7
|| Engine                            | mysql
|| EngineVersion                     | 5.7.19
|| Status                            | available
|| SupportsGlobalDatabases           | False
|| SupportsLogExportsToCloudwatchLogs| True
|| SupportsParallelQuery             | False
|| SupportsReadReplica               | True
+-----+
||                                         ExportableLogTypes
+-----+
|| audit
|| error
|| general
|| slowquery
|| +-----+
||                                         ValidUpgradeTarget
|| +-----+-----+-----+-----+
|| AutoUpgrade | Description   | Engine    | EngineVersion | IsMajorVersionUpgrade
|| +-----+-----+-----+-----+
|| False       | MySQL 5.7.21  | mysql     | 5.7.21      | False
|| False       | MySQL 5.7.22  | mysql     | 5.7.22      | False
|| False       | MySQL 5.7.23  | mysql     | 5.7.23      | False
|| False       | MySQL 5.7.24  | mysql     | 5.7.24      | False
|| False       | MySQL 5.7.25  | mysql     | 5.7.25      | False
|| True        | MySQL 5.7.26  | mysql     | 5.7.26      | False
```

	False	MySQL 5.7.28	mysql	5.7.28	False	
	False	MySQL 5.7.30	mysql	5.7.30	False	
	False	MySQL 5.7.31	mysql	5.7.31	False	
	False	MySQL 8.0.11	mysql	8.0.11	True	
	False	MySQL 8.0.13	mysql	8.0.13	True	
	False	MySQL 8.0.15	mysql	8.0.15	True	
	False	MySQL 8.0.16	mysql	8.0.16	True	
	False	MySQL 8.0.17	mysql	8.0.17	True	
	False	MySQL 8.0.19	mysql	8.0.19	True	
	False	MySQL 8.0.20	mysql	8.0.20	True	
	False	MySQL 8.0.21	mysql	8.0.21	True	
+-----+-----+-----+-----+-----+-----+						

In this example, the `AutoUpgrade` value is `True` for MySQL version 5.7.26. So, the automatic minor upgrade target is MySQL version 5.7.26, which is highlighted in the output.

A MySQL DB instance is automatically upgraded during your maintenance window if the following criteria are met:

- The **Auto minor version upgrade** setting is enabled.
- The **Backup retention period** setting is greater than 0.
- The DB instance is running a minor DB engine version that is less than the current automatic upgrade minor version.

For more information, see [Automatically upgrading the minor engine version \(p. 273\)](#).

Using a read replica to reduce downtime when upgrading a MySQL database

If your MySQL DB instance is currently in use with a production application, you can use the following procedure to upgrade the database version for your DB instance. This procedure can reduce the amount of downtime for your application.

By using a read replica, you can perform most of the maintenance steps ahead of time and minimize the necessary changes during the actual outage. With this technique, you can test and prepare the new DB instance without making any changes to your existing DB instance.

The following procedure shows an example of upgrading from MySQL version 5.7 to MySQL version 8.0. You can use the same general steps for upgrades to other major versions.

Note

When you are upgrading from MySQL version 5.7 to MySQL version 8.0, complete the prechecks before performing the upgrade. For more information, see [Prechecks for upgrades from MySQL 5.7 to 8.0 \(p. 856\)](#).

To upgrade an MySQL database while a DB instance is in use

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Create a read replica of your MySQL 5.7 DB instance. This process creates an upgradable copy of your database. Other read replicas of the DB instance might also exist.
 - a. In the console, choose **Databases**, and then choose the DB instance that you want to upgrade.
 - b. For **Actions**, choose **Create read replica**.
 - c. Provide a value for **DB instance identifier** for your read replica and ensure that the **DB instance class** and other settings match your MySQL 5.7 DB instance.

- d. Choose **Create read replica**.
3. (Optional) When the read replica has been created and **Status** shows **Available**, convert the read replica into a Multi-AZ deployment and enable backups.

By default, a read replicas is created as a single-AZ deployment with backups disabled. Because the read replica will ultimately become the production DB instance, it is a best practice to enable configure a Multi-AZ deployment and enable backups now.

- a. In the console, choose **Databases**, and then choose the read replica that you just created.
- b. Choose **Modify**.
- c. For **Multi-AZ deployment**, choose **Create a standby instance**.
- d. For **Backup Retention Period**, choose a positive nonzero value, for example 3 days, and then choose **Continue**.
- e. For **Scheduling of modifications**, choose **Apply immediately**.
- f. Choose **Modify DB instance**.
4. When the read replica **Status** shows **Available**, upgrade the read replica to MySQL 8.0:
 - a. In the console, choose **Databases**, and then choose the read replica that you just created.
 - b. Choose **Modify**.
 - c. For **DB engine version**, choose the MySQL 8.0 version to upgrade to, and then choose **Continue**.
 - d. For **Scheduling of modifications**, choose **Apply immediately**.
 - e. Choose **Modify DB instance** to start the upgrade.
5. When the upgrade is complete and **Status** shows **Available**, verify that the upgraded read replica is up-to-date with the source MySQL 5.7 DB instance. You can do this by connecting to the read replica and issuing the `SHOW SLAVE STATUS` command. If the `Seconds_Behind_Master` field is 0, then replication is up-to-date.
6. (Optional) Create a read replica of your read replica.

If you want the DB instance to have a read replica after it is promoted to a standalone DB instance, you can create the read replica now.

- a. In the console, choose **Databases**, and then choose the read replica that you just upgraded.
- b. For **Actions**, choose **Create read replica**.
- c. Provide a value for **DB instance identifier** for your read replica and ensure that the **DB instance class** and other settings match your MySQL 5.7 DB instance.
- d. Choose **Create read replica**.
7. (Optional) Configure a custom DB parameter group for the read replica.

If you want the DB instance to use a custom parameter group after it is promoted to a standalone DB instance, you can create the DB parameter group now can associate it with the read replica.

- a. Create a custom DB parameter group for MySQL 8.0. For instructions, see [Creating a DB parameter group \(p. 229\)](#).
- b. Modify the parameters that you want to change in the DB parameter group you just created. For instructions, see [Modifying parameters in a DB parameter group \(p. 232\)](#).
- c. In the console, choose **Databases**, and then choose the read replica.
- d. Choose **Modify**.
- e. For **DB parameter group**, choose the MySQL 8.0 DB parameter group you just created, and then choose **Continue**.
- f. For **Scheduling of modifications**, choose **Apply immediately**.
- g. Choose **Modify DB instance** to start the upgrade.
8. Make your MySQL 8.0 read replica a standalone DB instance.

Important

When you promote your MySQL 8.0 read replica to a standalone DB instance, it no longer is a replica of your MySQL 5.7 DB instance. We recommend that you promote your MySQL 8.0 read replica during a maintenance window when your source MySQL 5.7 DB instance is in read-only mode and all write operations are suspended. When the promotion is completed, you can direct your write operations to the upgraded MySQL 8.0 DB instance to ensure that no write operations are lost.

In addition, we recommend that, before promoting your MySQL 8.0 read replica, you perform all necessary data definition language (DDL) operations on your MySQL 8.0 read replica. An example is creating indexes. This approach avoids negative effects on the performance of the MySQL 8.0 read replica after it has been promoted. To promote a read replica, use the following procedure.

- a. In the console, choose **Databases**, and then choose the read replica that you just upgraded.
 - b. For **Actions**, choose **Promote**.
 - c. Choose **Yes** to enable automated backups for the read replica instance. For more information, see [Working with backups \(p. 328\)](#).
 - d. Choose **Continue**.
 - e. Choose **Promote Read Replica**.
9. You now have an upgraded version of your MySQL database. At this point, you can direct your applications to the new MySQL 8.0 DB instance.

Upgrading a MySQL DB snapshot

With Amazon RDS, you can create a storage volume DB snapshot of your MySQL DB instance. When you create a DB snapshot, the snapshot is based on the engine version used by your Amazon RDS instance. In addition to upgrading the DB engine version of your DB instance, you can also upgrade the engine version for your DB snapshots. For example, you can upgrade DB snapshots created from the MySQL 5.1 engine to DB snapshots for the MySQL 5.5 engine. After restoring a DB snapshot upgraded to a new engine version, you should test that the upgrade was successful. To learn how to test a major version upgrade, see [Testing an upgrade \(p. 858\)](#). To learn how to restore a DB snapshot, see [Restoring from a DB snapshot \(p. 349\)](#).

Amazon RDS supports upgrading a MySQL DB snapshot from MySQL 5.1 to MySQL 5.5.

You can upgrade manual DB snapshots, which can be encrypted or not encrypted, from MySQL 5.1 to MySQL 5.5 within the same AWS Region. You can't upgrade automated DB snapshots that are created during the automated backup process.

Console

To upgrade a DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. For **Actions**, choose **Upgrade snapshot**. The **Upgrade snapshot** page appears.
4. Choose the **New engine version** to upgrade to.
5. Choose **Save changes** to upgrade the snapshot.

During the upgrade process, all snapshot actions are disabled for this DB snapshot. Also, the DB snapshot status changes from **available** to **upgrading**, and then changes to **active** upon completion. If the DB snapshot can't be upgraded because of snapshot corruption issues, the status changes to **unavailable**. You can't recover the snapshot from this state.

Note

If the DB snapshot upgrade fails, the snapshot is rolled back to the original state with the original version.

AWS CLI

To upgrade a DB snapshot to a new database engine version, use the AWS CLI `modify-db-snapshot` command.

Parameters

- `--db-snapshot-identifier` – The identifier of the DB snapshot to upgrade. The identifier must be a unique Amazon Resource Name (ARN). For more information, see [Working with Amazon Resource Names \(ARNs\) in Amazon RDS \(p. 309\)](#).
- `--engine-version` – The engine version to upgrade the DB snapshot to.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-snapshot \
```

```
--db-snapshot-identifier <mydbsnapshot> \
--engine-version <new_version>
```

For Windows:

```
aws rds modify-db-snapshot ^
--db-snapshot-identifier <mydbsnapshot> ^
--engine-version <new_version>
```

RDS API

To upgrade a DB snapshot to a new database engine version, call the Amazon RDS API [ModifyDBSnapshot](#) operation.

- **DBSnapshotIdentifier** – The identifier of the DB snapshot to upgrade. The identifier must be a unique Amazon Resource Name (ARN). For more information, see [Working with Amazon Resource Names \(ARNs\) in Amazon RDS \(p. 309\)](#).
- **EngineVersion** – The engine version to upgrade the DB snapshot to.

Importing data into a MySQL DB instance

You can use several different techniques to import data into an Amazon RDS for MySQL DB instance. The best approach depends on the source of the data, the amount of data, and whether the import is done one time or is ongoing. If you are migrating an application along with the data, also consider the amount of downtime that you are willing to experience.

Overview

Find techniques to import data into an Amazon RDS for MySQL DB instance in the following table.

Source	Amount of data	One time or ongoing	Application downtime	Technique	More information
Existing MySQL database on premises or on Amazon EC2	Any	One time	Some	Create a backup of your on-premises database, store it on Amazon S3, and then restore the backup file to a new Amazon RDS DB instance running MySQL.	Restoring a backup into a MySQL DB instance (p. 871)
Any existing database	Any	One time or ongoing	Minimal	Use AWS Database Migration Service to migrate the database with minimal downtime and, for many database DB engines, continue ongoing replication.	What is AWS Database Migration Service in the AWS Database Migration Service User Guide
Existing MySQL DB instance	Any	One time or ongoing	Minimal	Create a read replica for ongoing replication. Promote the read replica for one-time creation of a new DB instance.	Working with read replicas (p. 278)
Existing MySQL or MariaDB database	Small	One time	Some	Copy the data directly to your MySQL DB instance using a command-line utility.	Importing data from a MySQL or MariaDB DB to a MySQL or MariaDB DB instance (p. 879)
Data not stored	Medium	One time	Some	Create flat files and import them using the <code>mysqlimport</code> utility.	Importing data

Source	Amount of data	One time or ongoing	Application downtime	Technique	More information
in an existing database					from any source to a MySQL or MariaDB DB instance (p. 894)
Existing MySQL or MariaDB database on premises or on Amazon EC2	Any	Ongoing	Minimal	Configure replication with an existing MySQL database as the replication source.	Replication with a MySQL or MariaDB instance running external to Amazon RDS (p. 914) or Importing data to an Amazon RDS MySQL or MariaDB DB instance with reduced downtime (p. 881)

Note

The 'mysql' system database contains authentication and authorization information required to log in to your DB instance and access your data. Dropping, altering, renaming, or truncating tables, data, or other contents of the 'mysql' database in your DB instance can result in error and might render the DB instance and your data inaccessible. If this occurs, you can restore the DB instance from a snapshot using the AWS CLI `restore-db-instance-from-db-snapshot` command. You can recover the DB instance using the AWS CLI `restore-db-instance-to-point-in-time` command.

Importing data considerations

Following, you can find additional technical information related to loading data into MySQL. This information is intended for advanced users who are familiar with the MySQL server architecture. All comments related to `LOAD DATA LOCAL INFILE` also apply to `mysqldump`.

Binary log

Data loads incur a performance penalty and require additional free disk space (up to four times more) when binary logging is enabled versus loading the same data with binary logging turned off. The severity of the performance penalty and the amount of free disk space required is directly proportional to the size of the transactions used to load the data.

Transaction size

Transaction size plays an important role in MySQL data loads. It has a major influence on resource consumption, disk space utilization, resume process, time to recover, and input format (flat files or SQL). This section describes how transaction size affects binary logging and makes the case for disabling binary logging during large data loads. As noted earlier, binary logging is enabled and disabled by setting the Amazon RDS automated backup retention period. Non-zero values enable binary logging, and zero disables it. We also describe the impact of large transactions on InnoDB and why it's important to keep transaction sizes small.

Small transactions

For small transactions, binary logging doubles the number of disk writes required to load the data. This effect can severely degrade performance for other database sessions and increase the time required to load the data. The degradation experienced depends in part upon the upload rate, other database activity taking place during the load, and the capacity of your Amazon RDS DB instance.

The binary logs also consume disk space roughly equal to the amount of data loaded until they are backed up and removed. Fortunately, Amazon RDS minimizes this by backing up and removing binary logs on a frequent basis.

Large transactions

Large transactions incur a 3X penalty for IOPS and disk consumption with binary logging enabled. This is due to the binary log cache spilling to disk, consuming disk space and incurring additional IO for each write. The cache cannot be written to the binlog until the transaction commits or rolls back, so it consumes disk space in proportion to the amount of data loaded. When the transaction commits, the cache must be copied to the binlog, creating a third copy of the data on disk.

Because of this, there must be at least three times as much free disk space available to load the data compared to loading with binary logging disabled. For example, 10 GiB of data loaded as a single transaction consumes at least 30 GiB disk space during the load. It consumes 10 GiB for the table + 10 GiB for the binary log cache + 10 GiB for the binary log itself. The cache file remains on disk until the session that created it terminates or the session fills its binary log cache again during another transaction. The binary log must remain on disk until backed up, so it might be some time before the extra 20 GiB is freed.

If the data was loaded using LOAD DATA LOCAL INFILE, yet another copy of the data is created if the database has to be recovered from a backup made before the load. During recovery, MySQL extracts the data from the binary log into a flat file. MySQL then runs LOAD DATA LOCAL INFILE, just as in the original transaction. However, this time the input file is local to the database server. Continuing with the example preceding, recovery fails unless there is at least 40 GiB free disk space available.

Disable binary logging

Whenever possible, disable binary logging during large data loads to avoid the resource overhead and addition disk space requirements. In Amazon RDS, disabling binary logging is as simple as setting the backup retention period to zero. If you do this, we recommend that you take a DB snapshot of the database instance immediately before the load. By doing this, you can quickly and easily undo changes made during loading if you need to.

After the load, set the backup retention period back to an appropriate (no zero) value.

You can't set the backup retention period to zero if the DB instance is a source DB instance for read replicas.

InnoDB

The information in this section provides a strong argument for keeping transaction sizes small when using InnoDB.

Undo

InnoDB generates undo to support features such as transaction rollback and MVCC. Undo is stored in the InnoDB system tablespace (usually `ibdata1`) and is retained until removed by the purge thread. The purge thread cannot advance beyond the undo of the oldest active transaction, so it is effectively blocked until the transaction commits or completes a rollback. If the database is processing other transactions during the load, their undo also accumulates in the system tablespace and cannot be removed even if they commit and no other transaction needs the undo for MVCC. In this situation, all transactions (including read-only transactions) that access any of the rows changed by any transaction (not just the load transaction) slow down. The slowdown occurs because transactions scan through undo that could have been purged if not for the long-running load transaction.

Undo is stored in the system tablespace, and the system tablespace never shrinks in size. Thus, large data load transactions can cause the system tablespace to become quite large, consuming disk space that you can't reclaim without recreating the database from scratch.

Rollback

InnoDB is optimized for commits. Rolling back a large transaction can take a very, very long time. In some cases, it might be faster to perform a point-in-time recovery or restore a DB snapshot.

Input data format

MySQL can accept incoming data in one of two forms: flat files and SQL. This section points out some key advantages and disadvantages of each.

Flat files

Loading flat files with `LOAD DATA LOCAL INFILE` can be the fastest and least costly method of loading data as long as transactions are kept relatively small. Compared to loading the same data with SQL, flat files usually require less network traffic, lowering transmission costs and load much faster due to the reduced overhead in the database.

One big transaction

`LOAD DATA LOCAL INFILE` loads the entire flat file as one transaction. This isn't necessarily a bad thing. If the size of the individual files can be kept small, this has a number of advantages:

- Resume capability – Keeping track of which files have been loaded is easy. If a problem arises during the load, you can pick up where you left off with little effort. Some data might have to be retransmitted to Amazon RDS, but with small files, the amount retransmitted is minimal.
- Load data in parallel – If you've got IOPS and network bandwidth to spare with a single file load, loading in parallel might save time.
- Throttle the load rate – Data load having a negative impact on other processes? Throttle the load by increasing the interval between files.

Be careful

The advantages of LOAD DATA LOCAL INFILE diminish rapidly as transaction size increases. If breaking up a large set of data into smaller ones isn't an option, SQL might be the better choice.

SQL

SQL has one main advantage over flat files: it's easy to keep transaction sizes small. However, SQL can take significantly longer to load than flat files and it can be difficult to determine where to resume the load after a failure. For example, mysqldump files are not restartable. If a failure occurs while loading a mysqldump file, the file requires modification or replacement before the load can resume. The alternative is to restore to the point in time before the load and replay the file after the cause of the failure has been corrected.

Take checkpoints using Amazon RDS snapshots

If you have a load that's going to take several hours or even days, loading without binary logging isn't a very attractive prospect unless you can take periodic checkpoints. This is where the Amazon RDS DB snapshot feature comes in very handy. A DB snapshot creates a point-in-time consistent copy of your database instance which can be used to restore the database to that point in time after a crash or other mishap.

To create a checkpoint, simply take a DB snapshot. Any previous DB snapshots taken for checkpoints can be removed without affecting durability or restore time.

Snapshots are fast too, so frequent checkpointing doesn't add significantly to load time.

Decreasing load time

Here are some additional tips to reduce load times:

- Create all secondary indexes before loading. This is counter-intuitive for those familiar with other databases. Adding or modifying a secondary index causes MySQL to create a new table with the index changes, copy the data from the existing table to the new table, and drop the original table.
- Load data in PK order. This is particularly helpful for InnoDB tables, where load times can be reduced by 75–80 percent and data file size cut in half.
- Disable foreign key constraints `foreign_key_checks=0`. For flat files loaded with LOAD DATA LOCAL INFILE, this is required in many cases. For any load, disabling FK checks provides significant performance gains. Just be sure to enable the constraints and verify the data after the load.
- Load in parallel unless already near a resource limit. Use partitioned tables when appropriate.
- Use multi-value inserts when loading with SQL to minimize overhead when running statements. When using mysqldump, this is done automatically.
- Reduce InnoDB log IO `innodb_flush_log_at_trx_commit=0`
- If you are loading data into a DB instance that does not have read replicas, set the `sync_binlog` parameter to 0 while loading data. When data loading is complete, set the `sync_binlog` parameter back to 1.
- Load data before converting the DB instance to a Multi-AZ deployment. However, if the DB instance already uses a Multi-AZ deployment, switching to a Single-AZ deployment for data loading is not recommended, because doing so only provides marginal improvements.

Note

Using `innodb_flush_log_at_trx_commit=0` causes InnoDB to flush its logs every second instead of at each commit. This provides a significant speed advantage, but can lead to data loss during a crash. Use with caution.

Topics

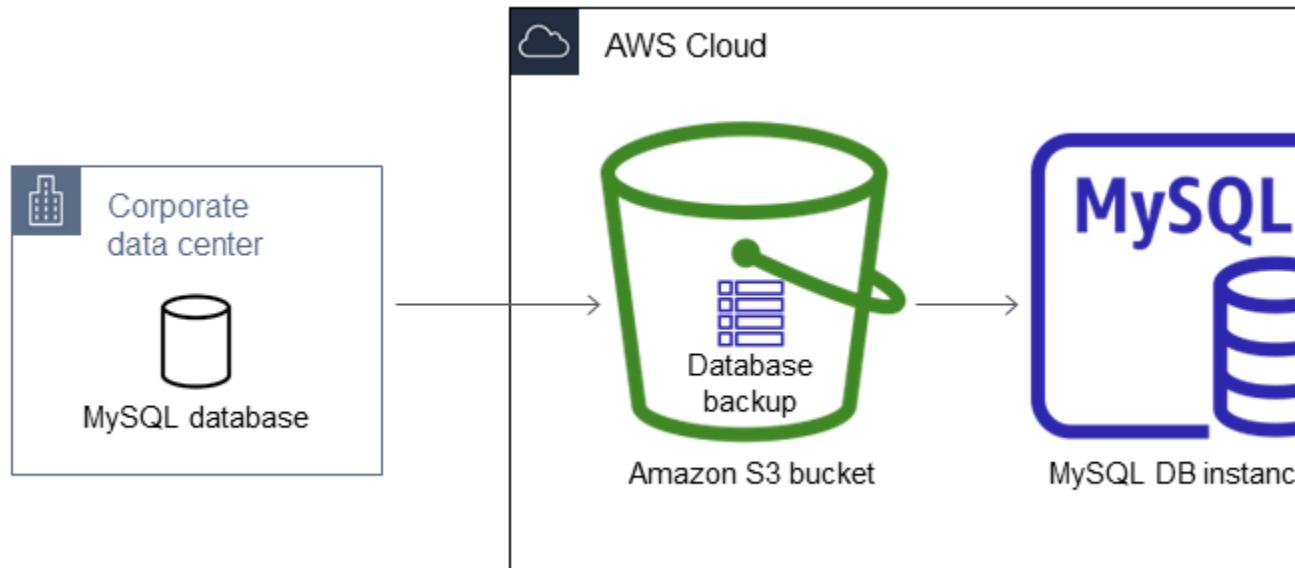
- [Restoring a backup into a MySQL DB instance \(p. 871\)](#)
- [Importing data from a MySQL or MariaDB DB to a MySQL or MariaDB DB instance \(p. 879\)](#)
- [Importing data to an Amazon RDS MySQL or MariaDB DB instance with reduced downtime \(p. 881\)](#)
- [Importing data from any source to a MySQL or MariaDB DB instance \(p. 894\)](#)

Restoring a backup into a MySQL DB instance

Amazon RDS supports importing MySQL databases by using backup files. You can create a backup of your database, store it on Amazon S3, and then restore the backup file onto a new Amazon RDS DB instance running MySQL.

The scenario described in this section restores a backup of an on-premises database. You can use this technique for databases in other locations, such as Amazon EC2 or non-AWS cloud services, as long as the database is accessible.

You can find the supported scenario in the following diagram.



Importing backup files from Amazon S3 is supported for MySQL version 5.6, 5.7, and 8.0. Importing backup files from Amazon S3 is available in all AWS Regions.

We recommend that you import your database to Amazon RDS by using backup files if your on-premises database can be offline while the backup file is created, copied, and restored. If your database can't be offline, you can use binary log (binlog) replication to update your database after you have migrated to Amazon RDS through Amazon S3 as explained in this topic. For more information, see [Replication with a MySQL or MariaDB instance running external to Amazon RDS \(p. 914\)](#). You can also use the AWS Database Migration Service to migrate your database to Amazon RDS. For more information, see [What is AWS Database Migration Service?](#)

Limitations and recommendations for importing backup files from Amazon S3 to Amazon RDS

The following are some limitations and recommendations for importing backup files from Amazon S3:

- You can only import your data to a new DB instance, not an existing DB instance.
- You must use Percona XtraBackup to create the backup of your on-premises database.
- You can't migrate from a source database that has tables defined outside of the default MySQL data directory.
- You must import your data to the default minor version of your MySQL major version in your AWS Region. For example, if your major version is MySQL 5.6, and the default minor version for your AWS Region is 5.6.44, then you must import your data into a MySQL version 5.6.44 DB instance. You can

upgrade your DB instance after importing. For information about determining the default minor version, see [MySQL on Amazon RDS versions \(p. 828\)](#).

- Backward migration is not supported for both major versions and minor versions. For example, you can't migrate from version 5.7 to version 5.6, and you can't migrate from version 5.6.39 to version 5.6.37.
- You can't import a MySQL 5.5 database.
- You can't import an on-premises MySQL database from one major version to another. For example, you can't import a MySQL 5.6 database to an Amazon RDS MySQL 5.7 or 8.0 database. Similarly, you can't import a MySQL 5.7 database to an RDS for MySQL 8.0 database. You can upgrade your DB instance after you complete the import.
- You can't restore from an encrypted source database, but you can restore to an encrypted Amazon RDS DB instance.
- You can't restore from an encrypted backup in the Amazon S3 bucket.
- You can't restore from an Amazon S3 bucket in a different AWS Region than your Amazon RDS DB instance.
- Importing from Amazon S3 is not supported on the db.t2.micro DB instance class. However, you can restore to a different DB instance class, and change the DB instance class later. For more information about instance classes, see [Hardware specifications for DB instance classes \(p. 33\)](#).
- Amazon S3 limits the size of a file uploaded to an Amazon S3 bucket to 5 TB. If a backup file exceeds 5 TB, then you must split the backup file into smaller files.
- When you restore the database, the backup is copied and then extracted on your DB instance. Therefore, provision storage space for your DB instance that is equal to or greater than the sum of the backup size, plus the original database's size on disk.
- Amazon RDS limits the number of files uploaded to an Amazon S3 bucket to 1 million. If the backup data for your database, including all full and incremental backups, exceeds 1 million files, use a Gzip (.gz), tar (.tar.gz), or Percona xbstream (.xbstream) file to store full and incremental backup files in the Amazon S3 bucket. Percona XtraBackup 8.0 only supports Percona xbstream for compression.
- User accounts are not imported automatically. Save your user accounts from your source database and add them to your new DB instance later.
- Functions are not imported automatically. Save your functions from your source database and add them to your new DB instance later.
- Stored procedures are not imported automatically. Save your stored procedures from your source database and add them to your new DB instance later.
- Time zone information is not imported automatically. Record the time zone information for your source database, and set the time zone of your new DB instance later. For more information, see [Local time zone for MySQL DB instances \(p. 838\)](#).
- The `innodb_data_file_path` parameter must be configured with only one data file that uses the default data file name "ibdata1:12M:autoextend". Databases with two data files, or with a data file with a different name, can't be migrated using this method.

The following are examples of file names that are not allowed:

"`innodb_data_file_path=ibdata1:50M; ibdata2:50M:autoextend`" and
"`innodb_data_file_path=ibdata01:50M:autoextend`".

- The maximum size of the restored database is the maximum database size supported minus the size of the backup. So, if the maximum database size supported is 64 TiB, and the size of the backup is 30 TiB, then the maximum size of the restored database is 34 TiB, as in the following example:

$64 \text{ TiB} - 30 \text{ TiB} = 34 \text{ TiB}$

For information about the maximum database size supported by Amazon RDS for MySQL, see [General Purpose SSD storage \(p. 40\)](#) and [Provisioned IOPS SSD storage \(p. 42\)](#).

Overview of setting up to import backup files from Amazon S3 to Amazon RDS

These are the components you need to set up to import backup files from Amazon S3 to Amazon RDS:

- An Amazon S3 bucket to store your backup files.
- A backup of your on-premises database created by Percona XtraBackup.
- An AWS Identity and Access Management (IAM) role to allow Amazon RDS to access the bucket.

If you already have an Amazon S3 bucket, you can use that. If you don't have an Amazon S3 bucket, you can create a new one. If you want to create a new bucket, see [Creating a bucket](#).

Use the Percona XtraBackup tool to create your backup. For more information, see [Creating your database backup \(p. 873\)](#).

If you already have an IAM role, you can use that. If you don't have an IAM role, you can create a new one manually. Alternatively, you can choose to have a new IAM role created for you in your account by the wizard when you restore the database by using the AWS Management Console. If you want to create a new IAM role manually, or attach trust and permissions policies to an existing IAM role, see [Creating an IAM role manually \(p. 875\)](#). If you want to have a new IAM role created for you, follow the procedure in [Console \(p. 876\)](#).

Creating your database backup

Use the Percona XtraBackup software to create your backup. You can install Percona XtraBackup from [Download Percona XtraBackup](#).

Note

For MySQL 8.0 migration, you must use Percona XtraBackup 8.0. Percona XtraBackup 8.0.12 and higher versions support migration of all versions of MySQL. If you are migrating to RDS for MySQL 8.0.20 or higher, you must use Percona XtraBackup 8.0.12 or higher.

For MySQL 5.7 migrations, you can also use Percona XtraBackup 2.4. For migrations of earlier MySQL versions, you can also use Percona XtraBackup 2.3 or 2.4.

You can create a full backup of your MySQL database files using Percona XtraBackup. Alternatively, if you already use Percona XtraBackup to back up your MySQL database files, you can upload your existing full and incremental backup directories and files.

For more information about backing up your database with Percona XtraBackup, see [Percona XtraBackup - documentation](#) and [The xtrabackup binary](#) on the Percona website.

Creating a full backup with Percona XtraBackup

To create a full backup of your MySQL database files that can be restored from Amazon S3, use the Percona XtraBackup utility (`xtrabackup`) to back up your database.

For example, the following command creates a backup of a MySQL database and stores the files in the folder `/on-premises/s3-restore/backup` folder.

```
xtrabackup --backup --user=<myuser> --password=<password> --target-dir=</on-premises/s3-restore/backup>
```

If you want to compress your backup into a single file (which can be split later, if needed), you can save your backup in one of the following formats:

- Gzip (.gz)
- tar (.tar)
- Percona xbstream (.xbstream)

Note

Percona XtraBackup 8.0 only supports Percona xbstream for compression.

The following command creates a backup of your MySQL database split into multiple Gzip files.

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=tar \  
--target-dir=</on-premises/s3-restore/backup> | gzip - | split -d --bytes=500MB \  
- </on-premises/s3-restore/backup/backup>.tar.gz
```

The following command creates a backup of your MySQL database split into multiple tar files.

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=tar \  
--target-dir=</on-premises/s3-restore/backup> | split -d --bytes=500MB \  
- </on-premises/s3-restore/backup/backup>.tar
```

The following command creates a backup of your MySQL database split into multiple xbstream files.

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=xbstream \  
--target-dir=</on-premises/s3-restore/backup> | split -d --bytes=500MB \  
- </on-premises/s3-restore/backup/backup>.xbstream
```

Using incremental backups with Percona XtraBackup

If you already use Percona XtraBackup to perform full and incremental backups of your MySQL database files, you don't need to create a full backup and upload the backup files to Amazon S3. Instead, you can save a significant amount of time by copying your existing backup directories and files to your Amazon S3 bucket. For more information about creating incremental backups using Percona XtraBackup, see [Incremental backup](#).

When copying your existing full and incremental backup files to an Amazon S3 bucket, you must recursively copy the contents of the base directory. Those contents include the full backup and also all incremental backup directories and files. This copy must preserve the directory structure in the Amazon S3 bucket. Amazon RDS iterates through all files and directories. Amazon RDS uses the `xtrabackup-checkpoints` file that is included with each incremental backup to identify the base directory, and to order incremental backups by log sequence number (LSN) range.

Backup considerations for Percona XtraBackup

Amazon RDS consumes your backup files based on the file name. Name your backup files with the appropriate file extension based on the file format—for example, `.xbstream` for files stored using the Percona xbstream format.

Amazon RDS consumes your backup files in alphabetical order and also in natural number order. Use the `split` option when you issue the `xtrabackup` command to ensure that your backup files are written and named in the proper order.

Amazon RDS doesn't support partial backups created using Percona XtraBackup. You can't use the following options to create a partial backup when you back up the source files for your database: `--tables`, `--tables-exclude`, `--tables-file`, `--databases`, `--databases-exclude`, or `--databases-file`.

Amazon RDS supports incremental backups created using Percona XtraBackup. For more information about creating incremental backups using Percona XtraBackup, see [Incremental backup](#).

Creating an IAM role manually

If you don't have an IAM role, you can create a new one manually. Alternatively, you can choose to have a new IAM role created for you by the wizard when you restore the database by using the AWS Management Console. If you want to have a new IAM role created for you, follow the procedure in [Console \(p. 876\)](#).

To manually create a new IAM role for importing your database from Amazon S3, create a role to delegate permissions from Amazon RDS to your Amazon S3 bucket. When you create an IAM role, you attach trust and permissions policies. To import your backup files from Amazon S3, use trust and permissions policies similar to the examples following. For more information about creating the role, see [Creating a role to delegate permissions to an AWS service](#).

Alternatively, you can choose to have a new IAM role created for you by the wizard when you restore the database by using the AWS Management Console. If you want to have a new IAM role created for you, follow the procedure in [Console \(p. 876\)](#)

The trust and permissions policies require that you provide an Amazon Resource Name (ARN). For more information about ARN formatting, see [Amazon Resource Names \(ARNs\) and AWS service namespaces](#).

Example Trust policy for importing from Amazon S3

```
{  
    "Version": "2012-10-17",  
    "Statement":  
    [ {  
        "Effect": "Allow",  
        "Principal": {"Service": "rds.amazonaws.com"},  
        "Action": "sts:AssumeRole"  
    } ]  
}
```

Example Permissions policy for importing from Amazon S3 — IAM user permissions

```
{  
    "Version": "2012-10-17",  
    "Statement":  
    [ {  
        "Sid": "AllowS3AccessRole",  
        "Effect": "Allow",  
        "Action": "iam:PassRole",  
        "Resource": "arn:aws:iam::IAM User ID:role/S3Access"  
    } ]  
}
```

Example Permissions policy for importing from Amazon S3 — role permissions

```
{  
    "Version": "2012-10-17",  
    "Statement":  
    [ {  
        "Effect": "Allow",  
        "Action": "s3:PutObject",  
        "Resource": "arn:aws:s3:::mybucket/*"  
    } ]  
}
```

```
"Action":  
  [  
    "s3>ListBucket",  
    "s3:GetBucketLocation"  
  ],  
  "Resource": "arn:aws:s3:::bucket_name"  
},  
{  
  "Effect": "Allow",  
  "Action":  
    [  
      "s3.GetObject"  
    ],  
  "Resource": "arn:aws:s3:::bucket_name/prefix*"  
}  
]  
}
```

Note

If you include a file name prefix, include the asterisk (*) after the prefix. If you don't want to specify a prefix, specify only an asterisk.

Importing data from Amazon S3 to a new MySQL DB instance

You can import data from Amazon S3 to a new MySQL DB instance using the AWS Management Console, AWS CLI, or RDS API.

Console

To import data from Amazon S3 to a new MySQL DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the top right corner of the Amazon RDS console, choose the AWS Region in which to create your DB instance. Choose the same AWS Region as the Amazon S3 bucket that contains your database backup.
3. In the navigation pane, choose **Databases**.
4. Choose **Restore from S3**.

The **Create database by restoring from S3** page appears.

Create database by restoring from S3

S3 destination



Write audit logs to S3

Enter a destination in Amazon S3 where your audit logs will be stored. Amazon S3 is object storage build to store and retrieve any amount of data from anywhere

S3 bucket

test-eu1-bucket



S3 prefix (optional) [Info](#)

Engine options

Engine type [Info](#)

Amazon Aurora



MySQL



Edition

MySQL Community

Source engine version [Info](#)

5.6



Version [Info](#)

MySQL 5.6.44



Known Issues/Limitations

Review the [Known Issues/Limitations](#) to learn about potential compatibility issues with specific database versions.

IAM role



IAM role

Choose or create an IAM role to grant write access to your S3 bucket.

[Choose an option](#)



Settings

877

DB instance identifier [Info](#)

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

5. Under **S3 destination**:

- a. Choose the **S3 bucket** where to write audit logs.
- b. (Optional) For **S3 folder path prefix**, enter a file path prefix for the files stored in your Amazon S3 bucket.

If you don't specify a prefix, then RDS creates your DB instance using all of the files and folders in the root folder of the S3 bucket. If you do specify a prefix, then RDS creates your DB instance using the files and folders in the S3 bucket where the path for the file begins with the specified prefix.

For example, suppose that you store your backup files on S3 in a subfolder named backups, and you have multiple sets of backup files, each in its own directory (gzip_backup1, gzip_backup2, and so on). In this case, you specify a prefix of backups/gzip_backup1 to restore from the files in the gzip_backup1 folder.

6. Under **Engine options**:

- a. For **Engine type**, choose **MySQL**.
 - b. For **Source engine version**, choose the MySQL major version of your source database.
 - c. For **Version**, choose the MySQL engine version for your restored DB instance.
7. For **IAM role**, you can choose an existing IAM role.
8. (Optional) You can also have a new IAM role created for you by choosing **Create a new role** and entering the **IAM role name**.
9. Specify your DB instance information. For information about each setting, see [Settings for DB instances \(p. 145\)](#).

Note

Be sure to allocate enough memory for your new DB instance so that the restore operation can succeed.

You can also choose **Enable storage autoscaling** to allow for future growth automatically.

10. Choose additional settings as needed.

11. Choose **Create database**.

AWS CLI

To import data from Amazon S3 to a new MySQL DB instance by using the AWS CLI, call the [restore-db-instance-from-s3](#) command with the following parameters. For information about each setting, see [Settings for DB instances \(p. 145\)](#).

Note

Be sure to allocate enough memory for your new DB instance so that the restore operation can succeed.

You can also use the `--max-allocated-storage` parameter to enable storage autoscaling and allow for future growth automatically.

- `--allocated-storage`
- `--db-instance-identifier`
- `--db-instance-class`
- `--engine`
- `--master-username`
- `--master-user-password`
- `--s3-bucket-name`
- `--s3-ingestion-role-arn`

- `--s3-prefix`
- `--source-engine`
- `--source-engine-version`

Example

For Linux, macOS, or Unix:

```
aws rds restore-db-instance-from-s3 \
    --allocated-storage 250 \
    --db-instance-identifier myidentifier \
    --db-instance-class db.m5.large \
    --engine mysql \
    --master-username admin \
    --master-user-password mypassword \
    --s3-bucket-name mybucket \
    --s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename \
    --s3-prefix bucketprefix \
    --source-engine mysql \
    --source-engine-version 5.6.44 \
    --max-allocated-storage 1000
```

For Windows:

```
aws rds restore-db-instance-from-s3 ^
    --allocated-storage 250 ^
    --db-instance-identifier myidentifier ^
    --db-instance-class db.m5.large ^
    --engine mysql ^
    --master-username admin ^
    --master-user-password mypassword ^
    --s3-bucket-name mybucket ^
    --s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename ^
    --s3-prefix bucketprefix ^
    --source-engine mysql ^
    --source-engine-version 5.6.44 ^
    --max-allocated-storage 1000
```

RDS API

To import data from Amazon S3 to a new MySQL DB instance by using the Amazon RDS API, call the [RestoreDBInstanceFromS3](#) operation.

Importing data from a MySQL or MariaDB DB to a MySQL or MariaDB DB instance

If your scenario supports it, it is easier to move data in and out of Amazon RDS by using backup files and Amazon S3. For more information, see [Restoring a backup into a MySQL DB instance \(p. 871\)](#).

You can also import data from an existing MySQL or MariaDB database to a MySQL or MariaDB DB instance. You do so by copying the database with [mysqldump](#) and piping it directly into the MySQL or MariaDB DB instance. The [mysqldump](#) command-line utility is commonly used to make backups and transfer data from one MySQL or MariaDB server to another. It is included with MySQL and MariaDB client software.

A typical [mysqldump](#) command to move data from an external database to an Amazon RDS DB instance looks similar to the following:

```
mysqldump -u local_user \
    --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary \
    -plocal_password | mysql -u RDS_user \
        --port=port_number \
        --host=host_name \
        -pRDS_password
```

Important

Make sure not to leave a space between the `-p` option and the entered password.

Note

- Exclude the following schemas from the dump file: `sys`, `performance_schema`, and `information_schema`. The `mysqldump` utility excludes these schemas by default.
- If you need to migrate users and privileges, consider using a tool that generates the data control language (DCL) for recreating them, such as the [pt-show-grants](#) utility.

The parameters used are as follows:

- `-u local_user` – Use to specify a user name. In the first usage of this parameter, you specify the name of a user account on the local MySQL or MariaDB database identified by the `--databases` parameter.
- `--databases database_name` – Use to specify the name of the database on the local MySQL or MariaDB instance that you want to import into Amazon RDS.
- `--single-transaction` – Use to ensure that all of the data loaded from the local database is consistent with a single point in time. If there are other processes changing the data while `mysqldump` is reading it, using this option helps maintain data integrity.
- `--compress` – Use to reduce network bandwidth consumption by compressing the data from the local database before sending it to Amazon RDS.
- `--order-by-primary` – Use to reduce load time by sorting each table's data by its primary key.
- `-plocal_password` – Use to specify a password. In the first usage of this parameter, you specify the password for the user account identified by the first `-u` parameter.
- `-u RDS_user` – Use to specify a user name. In the second usage of this parameter, you specify the name of a user account on the default database for the MySQL or MariaDB DB instance identified by the `--host` parameter.
- `--port port_number` – Use to specify the port for your MySQL or MariaDB DB instance. By default, this is 3306 unless you changed the value when creating the instance.
- `--host host_name` – Use to specify the DNS name from the Amazon RDS DB instance endpoint, for example, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. You can find the endpoint value in the instance details in the Amazon RDS Management Console.
- `-pRDS_password` – Use to specify a password. In the second usage of this parameter, you specify the password for the user account identified by the second `-u` parameter.

You must create any stored procedures, triggers, functions, or events manually in your Amazon RDS database. If you have any of these objects in the database that you are copying, then exclude them when you run `mysqldump` by including the following parameters with your `mysqldump` command: `--routines=0 --triggers=0 --events=0`.

The following example copies the `world` sample database on the local host to a MySQL DB instance.

For Linux, macOS, or Unix:

```
sudo mysqldump -u localuser \
--databases world \
--single-transaction \
--compress \
--order-by-primary \
-plocalpassword | mysql -u rdsuser \
--port=3306 \
--host=myinstance.123456789012.us-east-1.rds.amazonaws.com \
-prdspassword
```

For Windows, the following command needs to be run in a command prompt that has been opened by right-clicking **Command Prompt** on the Windows programs menu and choosing **Run as administrator**:

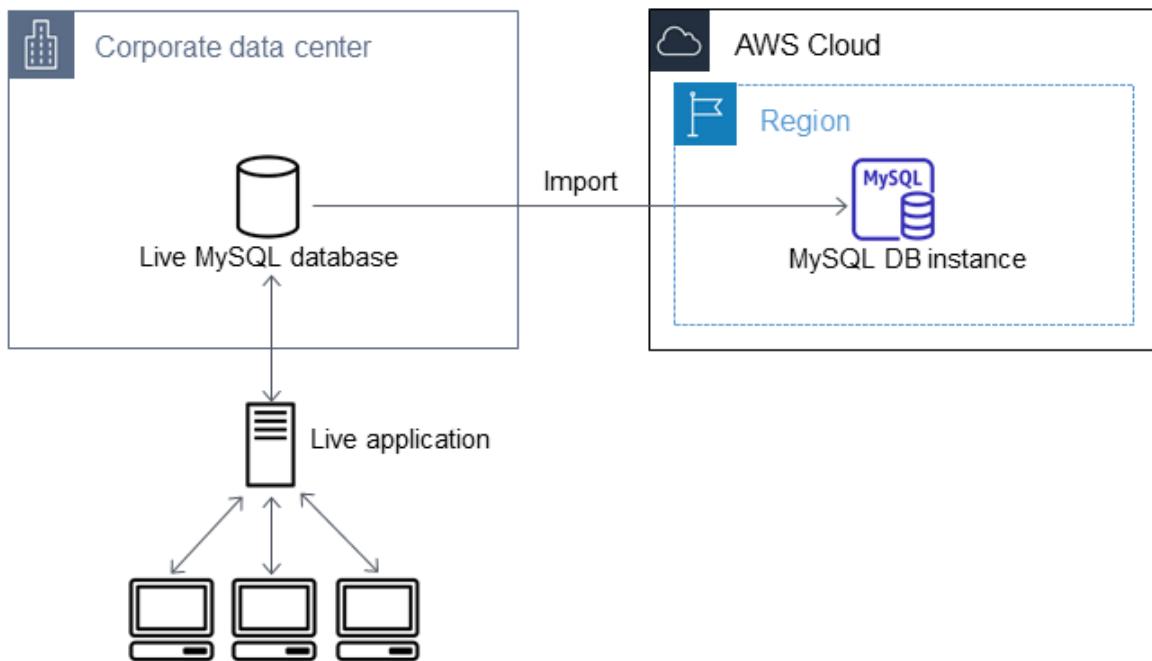
```
mysqldump -u localuser ^
--databases world ^
--single-transaction ^
--compress ^
--order-by-primary ^
-plocalpassword | mysql -u rdsuser ^
--port=3306 ^
--host=myinstance.123456789012.us-east-1.rds.amazonaws.com ^
-prdspassword
```

Importing data to an Amazon RDS MySQL or MariaDB DB instance with reduced downtime

If your scenario supports it, it is easier to move data in and out of Amazon RDS by using backup files and Amazon S3. For more information, see [Restoring a backup into a MySQL DB instance \(p. 871\)](#).

In some cases, you might need to import data from an external MySQL or MariaDB database that supports a live application to a MySQL or MariaDB DB instance. In these cases, you can use the following procedure to minimize the impact on application availability. This procedure can also help if you are working with a very large database. Here, the procedure helps because you can reduce the cost of the import by reducing the amount of data that is passed across the network to AWS.

In this procedure, you transfer a copy of your database data to an Amazon EC2 instance and import the data into a new Amazon RDS DB instance. You then use replication to bring the Amazon RDS DB instance up-to-date with your live external instance, before redirecting your application to the Amazon RDS DB instance. You configure MariaDB replication based on global transaction identifiers (GTIDs) if the external instance is MariaDB 10.0.2 or greater and the target instance is RDS for MariaDB; otherwise, you configure replication based on binary log coordinates. We recommend GTID-based replication if your external database supports it due to its enhanced crash-safety features. For more information, see [Global transaction ID](#) in the MariaDB documentation.

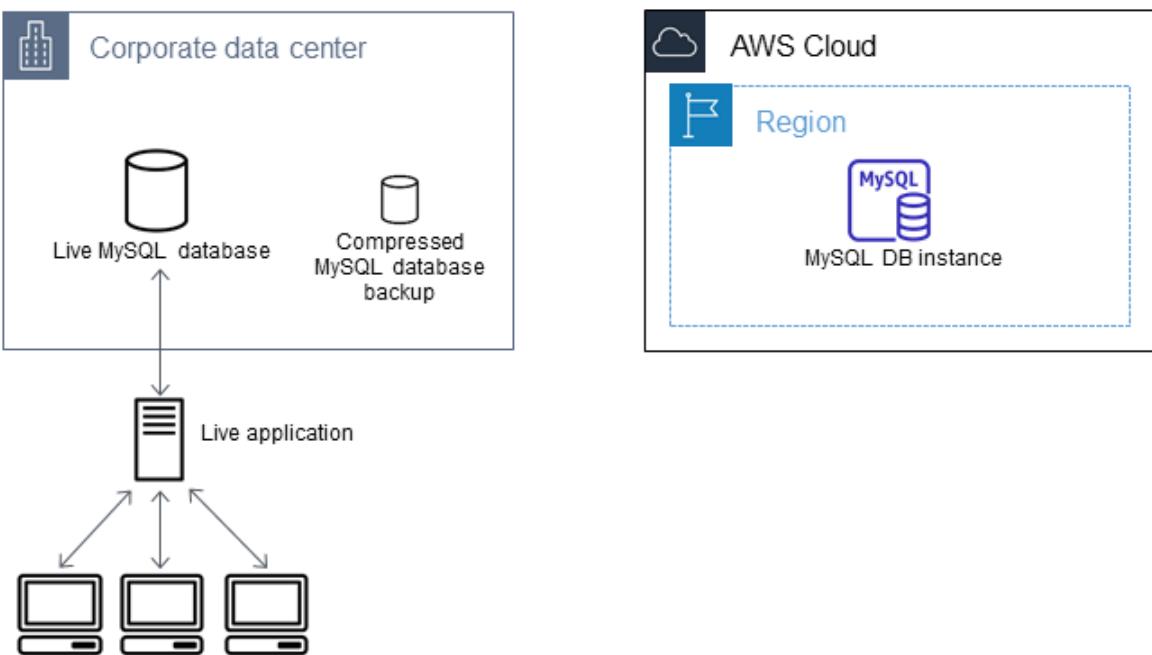


Note

We don't recommend that you use this procedure with source MySQL databases from MySQL versions earlier than version 5.1, due to potential replication issues. For more information, see [Replication compatibility between MySQL versions](#) in the MySQL documentation.

Create a copy of your existing database

The first step in the process of migrating a large amount of data to an Amazon RDS MySQL or MariaDB DB instance with minimal downtime is to create a copy of the source data.



You can use the `mysqldump` utility to create a database backup in either SQL or delimited-text format. You should do a test run with each format in a nonproduction environment to see which method minimizes the amount of time that `mysqldump` runs.

You should also weigh `mysqldump` performance against the benefit offered by using the delimited-text format for loading. A backup using delimited-text format creates a tab-separated text file for each table being dumped. You can load these files in parallel using the `LOAD DATA LOCAL INFILE` command to reduce the amount of time required to import your database. For more information about choosing a `mysqldump` format and then loading the data, see [Using mysqldump for backups](#) in the MySQL documentation.

Before you start the backup operation, you must set the replication options on the MySQL or MariaDB database that you are copying to Amazon RDS. The replication options include enabling binary logging and setting a unique server ID. Setting these options causes your server to start logging database transactions and prepares it to be a source replication instance later in this process.

Note

- Use the `--single-transaction` option with `mysqldump` because it dumps a consistent state of the database. To ensure a valid dump file, don't run data definition language (DDL) statements while `mysqldump` is running. You can schedule a maintenance window for these operations.
- Exclude the following schemas from the dump file: `sys`, `performance_schema`, and `information_schema`. The `mysqldump` utility excludes these schemas by default.
- If you need to migrate users and privileges, consider using a tool that generates the data control language (DCL) for recreating them, such as the [pt-show-grants](#) utility.

To set replication options

1. Edit the `my.cnf` file (this file is usually under `/etc`).

```
sudo vi /etc/my.cnf
```

Add the `log_bin` and `server_id` options to the `[mysqld]` section. The `log_bin` option provides a file name identifier for binary log files. The `server_id` option provides a unique identifier for the server in source-replica relationships.

The following example shows the updated `[mysqld]` section of a `my.cnf` file:

```
[mysqld]
log-bin=mysql-bin
server-id=1
```

For more information, see [the MySQL documentation](#).

2. Restart the `mysql` service.

```
sudo service mysql restart
```

To create a backup copy of your existing database

1. Create a backup of your data using the `mysqldump` utility, specifying either SQL or delimited-text format.

You must specify `--master-data=2` in order to create a backup file that can be used to start replication between servers. For more information, see the [mysqldump](#) documentation.

To improve performance and ensure data integrity, use the `--order-by-primary` and `--single-transaction` options of `mysqldump`.

To avoid including the MySQL system database in the backup, do not use the `--all-databases` option with `mysqldump`. For more information, see [Creating a data snapshot using mysqldump](#) in the MySQL documentation.

Use `chmod` if necessary to make sure that the directory where the backup file is being created is writeable.

Important

On Windows, run the command window as an administrator.

- To produce SQL output, use the following command.

For Linux, macOS, or Unix:

```
sudo mysqldump \  
    --databases database_name \  
    --master-data=2 \  
    --single-transaction \  
    --order-by-primary \  
    -r backup.sql \  
    -u local_user \  
    -p password
```

For Windows:

```
mysqldump ^  
    --databases database_name ^  
    --master-data=2 ^  
    --single-transaction ^  
    --order-by-primary ^  
    -r backup.sql ^  
    -u local_user ^  
    -p password
```

- To produce delimited-text output, use the following command.

For Linux, macOS, or Unix:

```
sudo mysqldump \  
    --tab=target_directory \  
    --fields-terminated-by ',' \  
    --fields-enclosed-by '"' \  
    --lines-terminated-by 0x0d0a \  
    database_name \  
    --master-data=2 \  
    --single-transaction \  
    --order-by-primary \  
    -p password
```

For Windows:

```
mysqldump ^  
    --tab=target_directory ^  
    --fields-terminated-by "," ^  
    --fields-enclosed-by "\"" ^  
    --lines-terminated-by 0x0d0a ^  
    database_name ^
```

```
--master-data=2 ^  
--single-transaction ^  
--order-by-primary ^  
-p password
```

Note

You must create any stored procedures, triggers, functions, or events manually in your Amazon RDS database. If you have any of these objects in the database that you are copying, exclude them when you run `mysqldump` by including the following arguments with your `mysqldump` command: `--routines=0 --triggers=0 --events=0`.

When using the delimited-text format, a `CHANGE MASTER TO` comment is returned when you run `mysqldump`. This comment contains the master log file name and position. If the external instance is other than MariaDB version 10.0.2 or greater, note the values for `MASTER_LOG_FILE` and `MASTER_LOG_POS`. You need these values when setting up replication.

```
-- Position to start replication or point-in-time recovery from  
--  
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
```

If you are using SQL format, you can get the master log file name and position in step 4 of the procedure at [Replicate between your external database and new Amazon RDS DB instance \(p. 890\)](#). If the external instance is MariaDB version 10.0.2 or greater, you can get the GTID in the next step.

2. If the external instance you are using is MariaDB version 10.0.2 or greater, you use GTID-based replication. Run `SHOW MASTER STATUS` on the external MariaDB instance to get the binary log file name and position, then convert them to a GTID by running `BINLOG_GTID_POS` on the external MariaDB instance.

```
SELECT BINLOG_GTID_POS('binary log file name', binary log file position);
```

Note the GTID returned; you need it to configure replication.

3. Compress the copied data to reduce the amount of network resources needed to copy your data to the Amazon RDS DB instance. Take note of the size of the backup file; you need this information when determining how large an Amazon EC2 instance to create. When you are done, compress the backup file using GZIP or your preferred compression utility.

- To compress SQL output, use the following command.

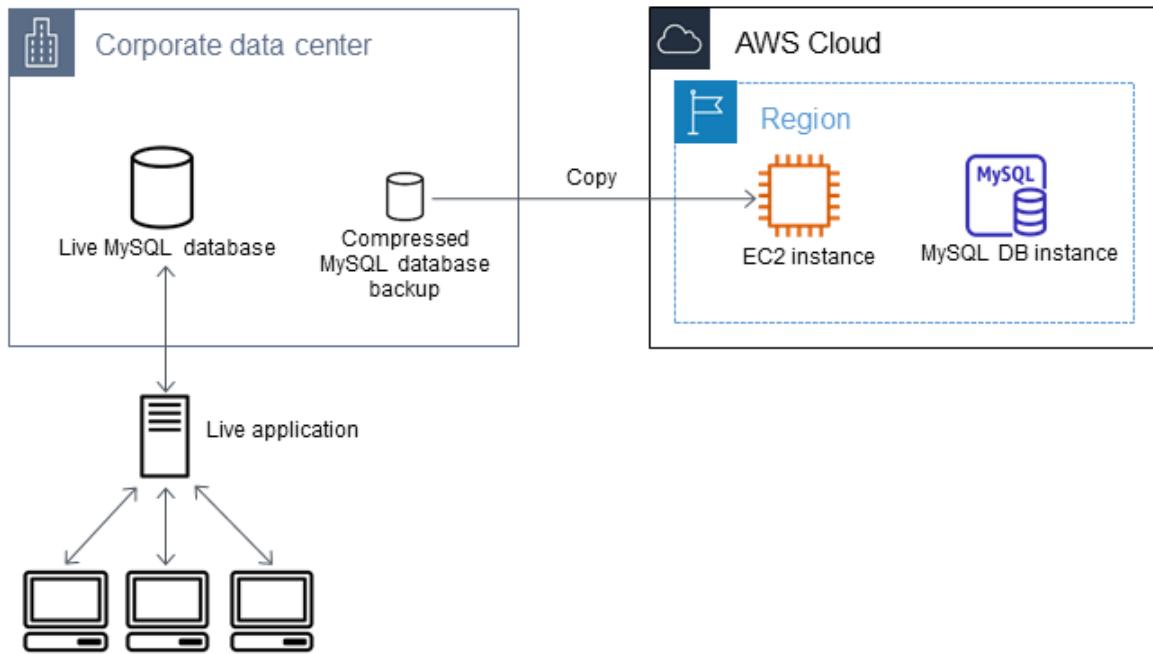
```
gzip backup.sql
```

- To compress delimited-text output, use the following command.

```
tar -zcvf backup.tar.gz target_directory
```

Create an Amazon EC2 instance and copy the compressed database

Copying your compressed database backup file to an Amazon EC2 instance takes fewer network resources than doing a direct copy of uncompressed data between database instances. After your data is in Amazon EC2, you can copy it from there directly to your MySQL or MariaDB DB instance. For you to save on the cost of network resources, your Amazon EC2 instance must be in the same AWS Region as your Amazon RDS DB instance. Having the Amazon EC2 instance in the same AWS Region as your Amazon RDS DB instance also reduces network latency during the import.



To create an Amazon EC2 instance and copy your data

1. In the AWS Region where you plan to create the RDS DB instance to run your MySQL database engine, create a VPC, a VPC security group, and a VPC subnet. Ensure that the inbound rules for your VPC security group allow the IP addresses required for your application to connect to AWS. This can be a range of IP addresses (for example, 203.0.113.0/24), or another VPC security group. You can use the [Amazon VPC management console](#) to create and manage VPCs, subnets, and security groups. For more information, see [Getting started with Amazon VPC](#) in the *Amazon Virtual Private Cloud Getting Started Guide*.

Note

Older AWS accounts can also launch instances in Amazon EC2-Classic mode. In this case, make sure that the inbound rules in the DB security group for your Amazon RDS instance allow access for your EC2-Classic instance using the Amazon EC2 private IP address. For more information, see [Working with DB security groups \(EC2-Classic platform\) \(p. 1704\)](#).

2. Open the [Amazon EC2 management console](#) and select the AWS Region to contain both your Amazon EC2 instance and your Amazon RDS DB instance. Launch an Amazon EC2 instance using the VPC, subnet, and security group that you created in Step 1. Ensure that you select an instance type with enough storage for your database backup file when it is uncompressed. For details on Amazon EC2 instances, see [Getting started with Amazon EC2 Linux instances](#) in the *Amazon Elastic Compute Cloud User Guide for Linux*.
3. To connect to your Amazon RDS DB instance from your Amazon EC2 instance, you need to edit your VPC security group, and add an inbound rule specifying the private IP address of your EC2 instance. You can find the private IP address on the **Details** tab of the **Instance** pane in the EC2 console window. To edit the VPC security group and add an inbound rule, choose **Security Groups** in the EC2 console navigation pane, choose your security group, and then add an inbound rule for MySQL/Aurora specifying the private IP address of your EC2 instance. To learn how to add an inbound rule to a VPC security group, see [Adding and removing rules](#).
4. Copy your compressed database backup file from your local system to your Amazon EC2 instance. Use `chmod` if necessary to make sure you have write permission for the target directory of the Amazon EC2 instance. You can use `scp` or an SSH client to copy the file. The following is an example:

```
$ scp -r -i key_pair.pem backup.sql.gz ec2-user@EC2_DNS:/target_directory/backup.sql.gz
```

Important

Be sure to copy sensitive data using a secure network transfer protocol.

5. Connect to your Amazon EC2 instance and install the latest updates and the MySQL client tools using the following commands:

```
sudo yum update -y  
sudo yum install mysql -y
```

For more information, see [Connect to your instance](#) in the *Amazon Elastic Compute Cloud User Guide for Linux*.

Important

This example installs the MySQL client on an Amazon Linux AMI distribution. If you want to install the MySQL client on a different distribution, such as Ubuntu or RedHat Enterprise Linux, this example won't work. For information about installing MySQL, see [Installing and Upgrading MySQL](#) in the MySQL documentation.

6. While connected to your Amazon EC2 instance, decompress your database backup file. For example:
 - To decompress SQL output, use the following command:

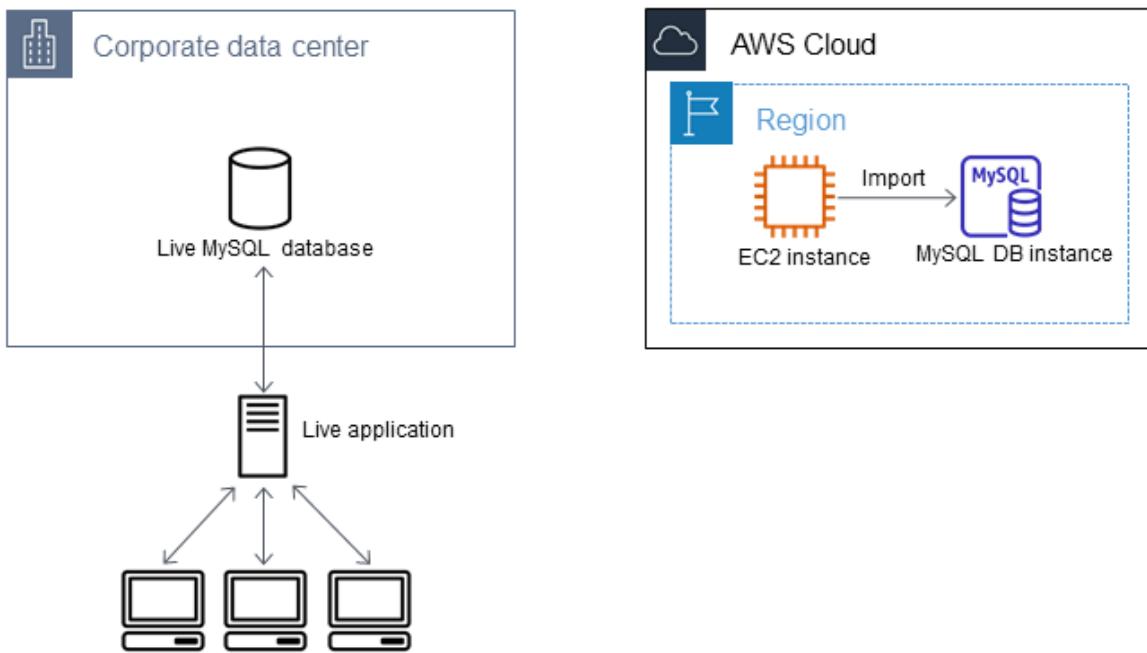
```
gzip backup.sql.gz -d
```

- To decompress delimited-text output, use the following command:

```
tar xzvf backup.tar.gz
```

Create a MySQL or MariaDB DB instance and import data from your Amazon EC2 instance

By creating a MySQL or MariaDB DB instance in the same AWS Region as your Amazon EC2 instance, you can import the database backup file from EC2 faster than over the internet.



To create a MySQL or MariaDB DB instance and import your data

1. Determine which DB instance class and what amount of storage space is required to support the expected workload for this Amazon RDS DB instance. As part of this process, decide what is sufficient space and processing capacity for your data load procedures, and also what is required to handle the production workload. You can estimate this based on the size and resources of the source MySQL or MariaDB database. For more information, see [DB instance classes \(p. 7\)](#).
2. Determine if Amazon RDS provisioned input/output operations per second (IOPS) is required to support the workloads. Provisioned IOPS storage delivers fast throughput for online transaction processing (OLTP) workloads, which are I/O intensive. For more information, see [Provisioned IOPS SSD storage \(p. 42\)](#).
3. Open the [Amazon RDS console](#). In the upper-right corner, choose the AWS Region that contains your Amazon EC2 instance.
4. In the navigation pane, choose **Databases**.
5. Choose **Create database**, and then go through the steps to choose options for your DB instance:
 - a. Make sure that **Standard Create** is chosen.
 - b. In the **Engine options** section, choose **MySQL** or **MariaDB**, as appropriate.
 - c. For **Version**, choose the version that is compatible with your source MySQL instance, as follows:
 - If your source instance is MySQL 5.1.x, the Amazon RDS DB instance must be MySQL 5.5.x.
 - If your source instance is MySQL 5.5.x, the Amazon RDS DB instance must be MySQL 5.5.x or greater.
 - If your source instance is MySQL 5.6.x, the Amazon RDS DB instance must be MySQL 5.6.x or MariaDB.
 - If your source instance is MySQL 5.7.x, the Amazon RDS DB instance must be MySQL 5.7.x, 5.6.x, or MariaDB.
 - If your source instance is MySQL 8.0.x, the Amazon RDS DB instance must be MySQL 8.0.x.
 - If your source instance is MariaDB 5.1, 5.2, or 5.3, the Amazon RDS DB instance must be MySQL 5.1.x.

- If your source instance is MariaDB 5.5 or greater, the Amazon RDS DB instance must be MariaDB.
- d. In the **Templates** section, choose **Dev/Test** to skip configuring Multi-AZ deployment and provisioned IOPS storage.
 - e. In the **Settings** section, specify the requested **DB instance identifier** and user information.
 - f. In the **DB instance class** and **Storage** sections, specify the DB instance class and allocated storage size that you want.
 - g. In the **Availability & durability** section, choose **Do not create a standby instance for Multi-AZ deployment**.
 - h. In the **Connectivity** section, choose the same virtual private cloud (VPC) and VPC security group as for your Amazon EC2 instance. This approach ensures that your Amazon EC2 instance and your Amazon RDS instance are visible to each other over the network. Set **Publicly accessible** to **Yes**. To set up replication with your source database as described later, your DB instance must be publicly accessible.

Use the default values for the other settings in this section.

In the **Backup** section, set the backup retention period to **0 days**.

Use the default values for the other settings in this section.

- i. Open the **Additional configuration** section, and specify an **Initial database name**.

Set the **Backup retention period to 0 days**

Use the default values for the other settings in this section.

- j. Choose **Create database**.

Your new DB instance appears in the **Databases** list with the status **Creating**. Wait for the **Status** of your new DB instance to show as **Available**.

Don't configure multiple Availability Zones, backup retention, or read replicas until after you have imported the database backup. When that import is done, you can set Multi-AZ and backup retention the way you want them for the production instance. For a detailed walkthrough of creating a DB instance, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

6. Review the default configuration options for the Amazon RDS DB instance. In the RDS console navigation pane, choose **Parameter groups**, and then choose the magnifying glass icon next to the **default.mysqlx.x** or **default.mariadb.x** parameter group. If this parameter group doesn't have the configuration options that you want, find a different one that does or create a new parameter group. For more information on creating a parameter group, see [Working with DB parameter groups \(p. 228\)](#).

If you decide to use a different parameter group than the default, associate it with your Amazon RDS DB instance. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

7. Connect to the new Amazon RDS DB instance as the master user, and create the users required to support the administrators, applications, and services that need to access the instance. The host name for the Amazon RDS DB instance is the **Endpoint** value for this instance without including the port number, for example `mysampledb.claxc2oy9ak1.us-west-2.rds.amazonaws.com`. You can find the endpoint value in the instance details in the Amazon RDS Management Console.
8. Connect to your Amazon EC2 instance. For more information, see [Connect to your instance](#) in the [Amazon Elastic Compute Cloud User Guide for Linux](#).
9. Connect to your Amazon RDS DB instance as a remote host from your Amazon EC2 instance using the `mysql` command. The following is an example.

```
mysql -h host_name -P 3306 -u db_master_user -p
```

The host name is the DNS name from the Amazon RDS DB instance endpoint.

10. At the mysql prompt, run the source command and pass it the name of your database dump file to load the data into the Amazon RDS DB instance:

- For SQL format, use the following command.

```
mysql> source backup.sql;
```

- For delimited-text format, first create the database (if it isn't the default database you created when setting up the Amazon RDS DB instance).

```
mysql> create database database_name;
$ mysql> use database_name;
```

Then create the tables.

```
mysql> source table1.sql
$ mysql> source table2.sql
etc...
```

Then import the data.

```
mysql> LOAD DATA LOCAL INFILE 'table1.txt' INTO TABLE table1 FIELDS TERMINATED BY ',' 
ENCLOSED BY '\"' LINES TERMINATED BY '0x0d0a';
$ mysql> LOAD DATA LOCAL INFILE 'table2.txt' INTO TABLE table2 FIELDS TERMINATED BY ',' 
ENCLOSED BY '\"' LINES TERMINATED BY '0x0d0a';
etc...
```

To improve performance, you can perform these operations in parallel from multiple connections so that all of your tables get created and then loaded at the same time.

Note

If you used any data-formatting options with mysqldump when you initially dumped the table, you must use the same options with mysqlimport or LOAD DATA LOCAL INFILE to ensure proper interpretation of the data file contents.

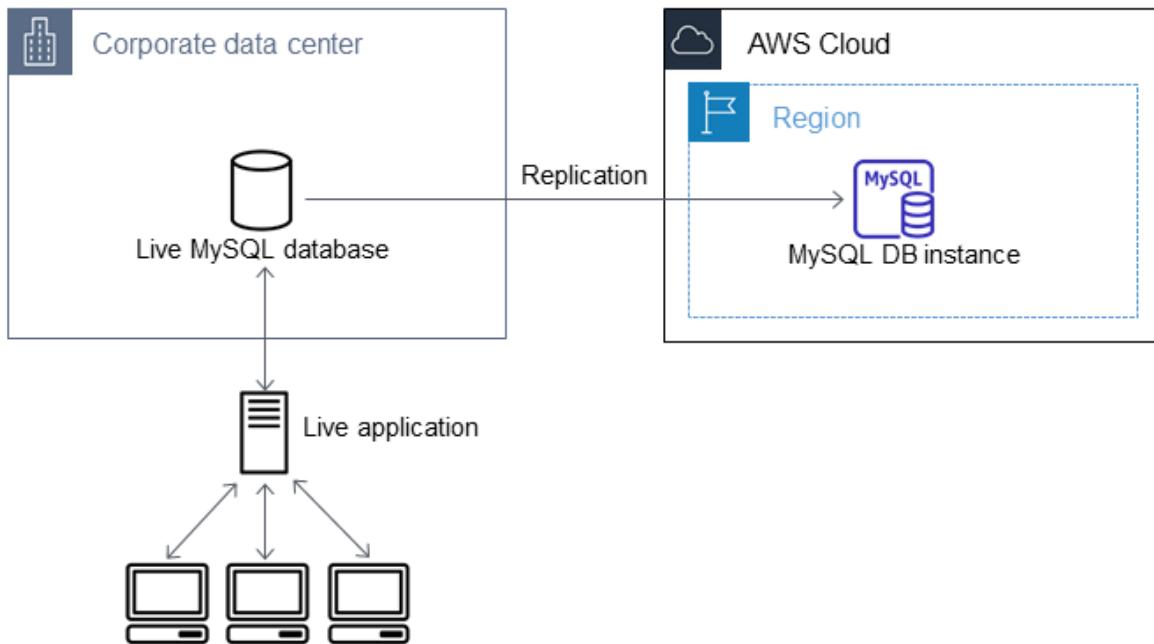
11. Run a simple SELECT query against one or two of the tables in the imported database to verify that the import was successful.

Note

If you no longer need the Amazon EC2 instance used in this procedure, terminate the EC2 instance to reduce your AWS resource usage. To terminate an EC2 instance, see [Terminating an instance](#).

Replicate between your external database and new Amazon RDS DB instance

Your source database was likely updated during the time that it took to copy and transfer the data to the MySQL or MariaDB DB instance. That being the case, you can use replication to bring the copied database up-to-date with the source database.



Note

The permissions required to start replication on an Amazon RDS DB instance are restricted and not available to your Amazon RDS master user. Because of this, you must use either the Amazon RDS [mysql.rds_set_external_master \(p. 954\)](#) command or the [mysql.rds_set_external_master_gtid \(p. 626\)](#) command to configure replication, and the [mysql.rds_start_replication \(p. 964\)](#) command to start replication between your live database and your Amazon RDS database.

To start replication

Earlier, you enabled binary logging and set a unique server ID for your source database. Now you can set up your Amazon RDS DB instance as a replica with your live database as the source replication instance.

1. In the Amazon RDS Management Console, add the IP address of the server that hosts the source database to the VPC security group for the Amazon RDS DB instance. For more information on modifying a VPC security group, see [Security groups for your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

You might also need to configure your local network to permit connections from the IP address of your Amazon RDS DB instance, so that it can communicate with your source instance. To find the IP address of the Amazon RDS DB instance, use the host command.

```
host db_instance_endpoint
```

The host name is the DNS name from the Amazon RDS DB instance endpoint, for example `myinstance.123456789012.us-east-1.rds.amazonaws.com`. You can find the endpoint value in the instance details in the Amazon RDS Management Console.

2. Using the client of your choice, connect to the source instance and create a user to be used for replication. This account is used solely for replication and must be restricted to your domain to improve security. The following is an example.

MySQL 5.5, 5.6, and 5.7

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

MySQL 8.0

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY  
'password';
```

3. For the source instance, grant REPLICATION CLIENT and REPLICATION SLAVE privileges to your replication user. For example, to grant the REPLICATION CLIENT and REPLICATION SLAVE privileges on all databases for the 'repl_user' user for your domain, issue the following command.

MySQL 5.5, 5.6, and 5.7

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'  
IDENTIFIED BY 'password';
```

MySQL 8.0

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

4. If you used SQL format to create your backup file and the external instance is not MariaDB 10.0.2 or greater, look at the contents of that file.

```
cat backup.sql
```

The file includes a CHANGE MASTER TO comment that contains the master log file name and position. This comment is included in the backup file when you use the --master-data option with mysqldump. Note the values for MASTER_LOG_FILE and MASTER_LOG_POS.

```
--  
-- Position to start replication or point-in-time recovery from  
--  
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
```

If you used delimited text format to create your backup file and the external instance is not MariaDB 10.0.2 or greater, you should already have binary log coordinates from step 1 of the procedure at [To create a backup copy of your existing database \(p. 883\)](#).

If the external instance is MariaDB 10.0.2 or greater, you should already have the GTID from which to start replication from step 2 of the procedure at [To create a backup copy of your existing database \(p. 883\)](#).

5. Make the Amazon RDS DB instance the replica. If the external instance is not MariaDB 10.0.2 or greater, connect to the Amazon RDS DB instance as the master user and identify the source database as the source replication instance by using the [mysql.rds_set_external_master \(p. 954\)](#) command. Use the master log file name and master log position that you determined in the previous step if you have a SQL format backup file. Alternatively, use the name and position that you determined when creating the backup files if you used delimited-text format. The following is an example.

```
CALL mysql.rds_set_external_master ('myserver.mydomain.com', 3306,  
'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```

If the external instance is MariaDB 10.0.2 or greater, connect to the Amazon RDS DB instance as the master user and identify the source database as the source replication instance by using the

[mysql.rds_set_external_master_gtid \(p. 626\)](#) command. Use the GTID that you determined in step 2 of the procedure at [To create a backup copy of your existing database \(p. 883\)](#). The following is an example.

```
CALL mysql.rds_set_external_master_gtid ('source_server_ip_address', 3306,  
'ReplicationUser', 'password', 'GTID', 0);
```

The *source_server_ip_address* is the IP address of source replication instance. An EC2 private DNS address is currently not supported.

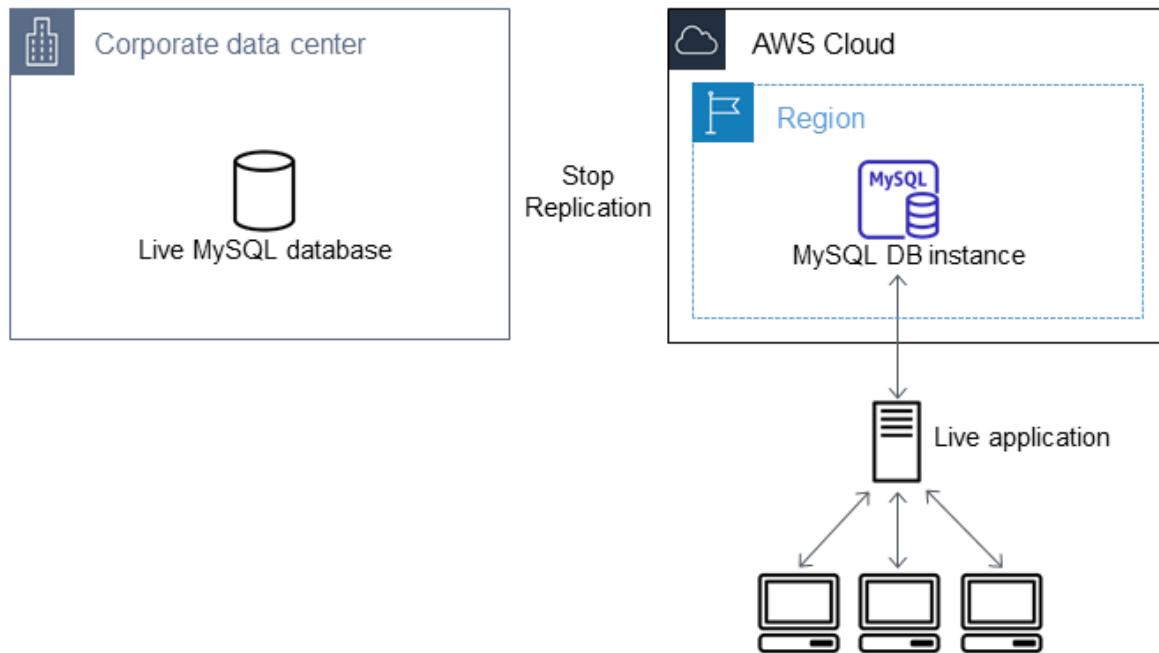
6. On the Amazon RDS DB instance, issue the [mysql.rds_start_replication \(p. 964\)](#) command to start replication.

```
CALL mysql.rds_start_replication;
```

7. On the Amazon RDS DB instance, run the [SHOW SLAVE STATUS](#) command to determine when the replica is up-to-date with the source replication instance. The results of the `SHOW SLAVE STATUS` command include the `Seconds_Behind_Master` field. When the `Seconds_Behind_Master` field returns 0, then the replica is up-to-date with the source replication instance.
8. After the Amazon RDS DB instance is up-to-date, enable automated backups so you can restore that database if needed. You can enable or modify automated backups for your Amazon RDS DB instance using the [Amazon RDS management console](#). For more information, see [Working with backups \(p. 328\)](#).

Redirect your live application to your Amazon RDS instance

After the MySQL or MariaDB DB instance is up-to-date with the source replication instance, you can now update your live application to use the Amazon RDS instance.



To redirect your live application to your MySQL or MariaDB DB instance and stop replication

1. To add the VPC security group for the Amazon RDS DB instance, add the IP address of the server that hosts the application. For more information on modifying a VPC security group, see [Security groups for your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.
2. Verify that the Seconds_Behind_Master field in the [SHOW SLAVE STATUS](#) command results is 0, which indicates that the replica is up-to-date with the source replication instance.

```
SHOW SLAVE STATUS;
```

3. Close all connections to the source when their transactions complete.
4. Update your application to use the Amazon RDS DB instance. This update typically involves changing the connection settings to identify the host name and port of the Amazon RDS DB instance, the user account and password to connect with, and the database to use.
5. Stop replication for the Amazon RDS instance using the [mysql.rds_stop_replication](#) (p. 967) command.

```
CALL mysql.rds_stop_replication;
```

6. Run the [mysql.rds_reset_external_master](#) (p. 961) command on your Amazon RDS DB instance to reset the replication configuration so this instance is no longer identified as a replica.

```
CALL mysql.rds_reset_external_master;
```

7. Enable additional Amazon RDS features such as Multi-AZ support and read replicas. For more information, see [High availability \(Multi-AZ\) for Amazon RDS](#) (p. 53) and [Working with read replicas](#) (p. 278).

Importing data from any source to a MySQL or MariaDB DB instance

If you have more than 1 GiB of data to load, or if your data is coming from somewhere other than a MySQL or MariaDB database, we recommend creating flat files and loading them with `mysqldump`. `mysqldump` is another command line utility bundled with the MySQL and MariaDB client software whose purpose is to load flat files into MySQL or MariaDB. For information about `mysqldump`, see [mysqldump - a data import program](#) in the MySQL documentation.

We also recommend creating DB snapshots of the target Amazon RDS DB instance before and after the data load. Amazon RDS DB snapshots are complete backups of your DB instance that can be used to restore your DB instance to a known state. When you initiate a DB snapshot, I/O operations to your database instance are momentarily suspended while your database is backed up.

Creating a DB snapshot immediately before the load lets you restore the database to its state before the load, if you need to. A DB snapshot taken immediately after the load protects you from having to load the data again in case of a mishap and can also be used to seed new database instances.

The following list shows the steps to take. Each step is discussed in more detail below.

1. Create flat files containing the data to be loaded.
2. Stop any applications accessing the target DB instance.
3. Create a DB snapshot.
4. Consider disabling Amazon RDS automated backups.

5. Load the data using mysqlimport.
6. Enable automated backups again.

Step 1: Create flat files containing the data to be loaded

Use a common format, such as CSV (Comma-Separated Values), to store the data to be loaded. Each table must have its own file; data for multiple tables cannot be combined in the same file. Give each file the same name as the table it corresponds to. The file extension can be anything you like. For example, if the table name is "sales", the file name could be "sales.csv" or "sales.txt", but not "sales_01.csv".

Whenever possible, order the data by the primary key of the table being loaded. This drastically improves load times and minimizes disk storage requirements.

The speed and efficiency of this procedure is dependent upon keeping the size of the files small. If the uncompressed size of any individual file is larger than 1 GiB, split it into multiple files and load each one separately.

On Unix-like systems (including Linux), use the 'split' command. For example, the following command splits the sales.csv file into multiple files of less than 1 GiB, splitting only at line breaks (-C 1024m). The new files are named sales.part_00, sales.part_01, and so on.

```
split -C 1024m -d sales.csv sales.part_
```

Similar utilities are available on other operating systems.

Step 2: Stop any applications accessing the target DB instance

Before starting a large load, stop all application activity accessing the target DB instance that you plan to load to. We recommend this particularly if other sessions will be modifying the tables being loaded or tables they reference. Doing this reduces the risk of constraint violations occurring during the load and improves load performance. It also makes it possible to restore the database instance to the point just before the load without losing changes made by processes not involved in the load.

Of course, this might not be possible or practical. If you are unable to stop applications from accessing the DB instance before the load, take steps to ensure the availability and integrity of your data. The specific steps required vary greatly depending upon specific use cases and site requirements.

Step 3: Create a DB snapshot

If you plan to load data into a new DB instance that contains no data, you can skip this step. Otherwise, creating a DB snapshot of your DB instance allows you to restore the DB instance to the point just before the load, if it becomes necessary. As previously mentioned, when you initiate a DB snapshot, I/O operations to your database instance are suspended for a few minutes while the database is backed up.

In the example below, we use the AWS CLI `create-db-snapshot` command to create a DB snapshot of our AcmeRDS instance and give the DB snapshot the identifier "preload".

For Linux, macOS, or Unix:

```
aws rds create-db-snapshot \  
--db-instance-identifier AcmeRDS \  
--db-snapshot-identifier preload
```

For Windows:

```
aws rds create-db-snapshot ^
```

```
--db-instance-identifier AcmeRDS ^  
--db-snapshot-identifier preload
```

You can also use the restore from DB snapshot functionality in order to create test database instances for dry runs or to "undo" changes made during the load.

Keep in mind that restoring a database from a DB snapshot creates a new DB instance that, like all DB instances, has a unique identifier and endpoint. If you need to restore the database instance without changing the endpoint, you must first delete the DB instance so that the endpoint can be reused.

For example, to create a DB instance for dry runs or other testing, you would give the DB instance its own identifier. In the example, "AcmeRDS-2" is the identifier and we would connect to the database instance using the endpoint associated with AcmeRDS-2.

For Linux, macOS, or Unix:

```
aws rds restore-db-instance-from-db-snapshot \  
  --db-instance-identifier AcmeRDS-2 \  
  --db-snapshot-identifier preload
```

For Windows:

```
aws rds restore-db-instance-from-db-snapshot ^  
  --db-instance-identifier AcmeRDS-2 ^  
  --db-snapshot-identifier preload
```

To reuse the existing endpoint, we must first delete the database instance and then give the restored database the same identifier.

For Linux, macOS, or Unix:

```
aws rds delete-db-instance \  
  --db-instance-identifier AcmeRDS \  
  --final-db-snapshot-identifier AcmeRDS-Final  
  
aws rds restore-db-instance-from-db-snapshot \  
  --db-instance-identifier AcmeRDS \  
  --db-snapshot-identifier preload
```

For Windows:

```
aws rds delete-db-instance ^  
  --db-instance-identifier AcmeRDS ^  
  --final-db-snapshot-identifier AcmeRDS-Final  
  
aws rds restore-db-instance-from-db-snapshot ^  
  --db-instance-identifier AcmeRDS ^  
  --db-snapshot-identifier preload
```

The example takes a final DB snapshot of the database instance before deleting it. This is optional, but recommended.

Step 4: Consider disabling Amazon RDS automated backups

Warning

Do not disable automated backups if you need the ability to perform point-in-time recovery.

Disabling automated backups erases all existing backups, so point-in-time recovery is not possible after automated backups have been disabled. Disabling automated backups is a performance optimization

and is not required for data loads. Manual DB snapshots are not affected by disabling automated backups. All existing manual DB snapshots are still available for restore.

Disabling automated backups reduces load time by about 25 percent and reduce the amount of storage space required during the load. If you plan to load data into a new DB instance that contains no data, disabling backups is an easy way to speed up the load and avoid using the additional storage needed for backups. However, if you plan to load into a DB instance that already contains data, weigh the benefits of disabling backups against the impact of losing the ability to perform point-in-time-recovery.

DB instances have automated backups enabled by default (with a one day retention period). In order to disable automated backups, you must set the backup retention period to zero. After the load, you can re-enable backups by setting the backup retention period to a non-zero value. In order to enable or disable backups, Amazon RDS must shut the DB instance down and restart it in order to turn MySQL or MariaDB logging on or off.

Use the AWS CLI `modify-db-instance` command to set the backup retention to zero and apply the change immediately. Setting the retention period to zero requires a DB instance restart, so wait until the restart has completed before proceeding.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier AcmeRDS \
--apply-immediately \
--backup-retention-period 0
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier AcmeRDS ^
--apply-immediately ^
--backup-retention-period 0
```

You can check the status of your DB instance with the AWS CLI `describe-db-instances` command. The example displays the DB instance status of the AcmeRDS DB instance.

```
aws rds describe-db-instances --db-instance-identifier AcmeRDS --query "*[].{DBInstanceStatus:DBInstanceState}"
```

When the DB instance status is available, you're ready to proceed.

Step 5: Load the data

Use the `mysqlimport` utility to load the flat files into Amazon RDS. In the example we tell `mysqlimport` to load all of the files named "sales" with an extension starting with "part_". This is a convenient way to load all of the files created in the "split" example. Use the `--compress` option to minimize network traffic. The `--fields-terminated-by=,` option is used for CSV files and the `--local` option specifies that the incoming data is located on the client. Without the `--local` option, the Amazon RDS DB instance looks for the data on the database host, so always specify the `--local` option.

For Linux, macOS, or Unix:

```
mysqlimport --local \
--compress \
--user=username \
--password \
--host=hostname \
```

```
--fields-terminated-by=',' Acme sales.part_*
```

For Windows:

```
mysqlimport --local ^
--compress ^
--user=username ^
--password ^
--host=hostname ^
--fields-terminated-by="," Acme sales.part_*
```

For very large data loads, take additional DB snapshots periodically between loading files and note which files have been loaded. If a problem occurs, you can easily resume from the point of the last DB snapshot, avoiding lengthy reloads.

Step 6: Enable Amazon RDS automated backups

After the load is finished, re-enable Amazon RDS automated backups by setting the backup retention period back to its pre-load value. As noted earlier, Amazon RDS restarts the DB instance, so be prepared for a brief outage.

In the example, we use the AWS CLI modify-db-instance command to enable automated backups for the AcmeRDS DB instance and set the retention period to 1 day.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier AcmeRDS \
--backup-retention-period 1 \
--apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier AcmeRDS ^
--backup-retention-period 1 ^
--apply-immediately
```

Working with MySQL replication in Amazon RDS

You usually use read replicas to configure replication between Amazon RDS DB instances. For general information about read replicas, see [Working with read replicas \(p. 278\)](#). For specific information about working with read replicas on Amazon RDS for MySQL, see [Working with MySQL read replicas \(p. 899\)](#).

You can use global transaction identifiers (GTIDs) for replication with Amazon RDS for MySQL. For more information, see [Using GTID-based replication for RDS for MySQL \(p. 910\)](#).

You can also set up replication between an Amazon RDS for MySQL DB instance and a MySQL or MariaDB instance that is external to Amazon RDS. For information about configuring replication with an external source, see [Replication with a MySQL or MariaDB instance running external to Amazon RDS \(p. 914\)](#).

For any of these replication options, you can use either row-based replication, statement-based, or mixed replication. Row-based replication only replicates the changed rows that result from a SQL statement. Statement-based replication replicates the entire SQL statement. Mixed replication uses statement-based replication when possible, but switches to row-based replication when SQL statements that are unsafe for statement-based replication are run. In most cases, mixed replication is recommended. The binary log format of the DB instance determines whether replication is row-based, statement-based, or mixed. For information about setting the binary log format, see [Setting the binary logging format \(p. 524\)](#).

Note

You can configure replication to import databases from a MySQL or MariaDB instance that is external to Amazon RDS, or to export databases to such instances. For more information, see [Importing data to an Amazon RDS MySQL or MariaDB DB instance with reduced downtime \(p. 881\)](#) and [Exporting data from a MySQL DB instance by using replication \(p. 921\)](#).

Topics

- [Working with MySQL read replicas \(p. 899\)](#)
- [Using GTID-based replication for RDS for MySQL \(p. 910\)](#)
- [Replication with a MySQL or MariaDB instance running external to Amazon RDS \(p. 914\)](#)

Working with MySQL read replicas

Following, you can find specific information about working with read replicas on Amazon RDS for MySQL. For general information about read replicas and instructions for using them, see [Working with read replicas \(p. 278\)](#).

Topics

- [Read replica configuration with MySQL \(p. 899\)](#)
- [Configuring replication filters with MySQL \(p. 900\)](#)
- [Configuring delayed replication with MySQL \(p. 905\)](#)
- [Read replica updates with MySQL \(p. 907\)](#)
- [Multi-AZ read replica deployments with MySQL \(p. 907\)](#)
- [Monitoring MySQL read replicas \(p. 908\)](#)
- [Starting and stopping replication with MySQL read replicas \(p. 908\)](#)
- [Troubleshooting a MySQL read replica problem \(p. 908\)](#)

Read replica configuration with MySQL

Before a MySQL DB instance can serve as a replication source, make sure to enable automatic backups on the source DB instance. To do this, set the backup retention period to a value other than 0. This

requirement also applies to a read replica that is the source DB instance for another read replica. Automatic backups are supported only for read replicas running any version of MySQL 5.6 and later. You can configure replication based on binary log coordinates for a MySQL DB instance.

On Amazon RDS for MySQL version 5.7.23 and later MySQL 5.7 versions, you can configure replication using global transaction identifiers (GTIDs). For more information, see [Using GTID-based replication for RDS for MySQL \(p. 910\)](#).

You can create up to five read replicas from one DB instance. For replication to operate effectively, each read replica should have the same amount of compute and storage resources as the source DB instance. If you scale the source DB instance, also scale the read replicas.

If a read replica is running any version of MySQL 5.6 and later, you can specify it as the source DB instance for another read replica. For example, you can create ReadReplica1 from MyDBInstance, and then create ReadReplica2 from ReadReplica1. Updates made to MyDBInstance are replicated to ReadReplica1 and then replicated from ReadReplica1 to ReadReplica2. You can't have more than four instances involved in a replication chain. For example, you can create ReadReplica1 from MySourceDBInstance, and then create ReadReplica2 from ReadReplica1, and then create ReadReplica3 from ReadReplica2, but you can't create a ReadReplica4 from ReadReplica3.

If you promote a MySQL read replica that is in turn replicating to other read replicas, those read replicas remain active. Consider an example where MyDBInstance1 replicates to MyDBInstance2, and MyDBInstance2 replicates to MyDBInstance3. If you promote MyDBInstance2, replication from MyDBInstance1 to MyDBInstance2 no longer occurs, but MyDBInstance2 still replicates to MyDBInstance3.

To enable automatic backups on a read replica for Amazon RDS for MySQL version 5.6 and later, first create the read replica. Then modify the read replica to enable automatic backups.

You can run multiple read replica create or delete actions at the same time that reference the same source DB instance. To do this, stay within the limit of five read replicas for each source instance.

A read replica of a MySQL DB instance can't use a lower DB engine version than its source DB instance.

Preparing MySQL DB instances that use MyISAM

If your MySQL DB instance uses a nontransactional engine such as MyISAM, you need to perform the following steps to successfully set up your read replica. These steps are required to make sure that the read replica has a consistent copy of your data. These steps are not required if all of your tables use a transactional engine such as InnoDB.

1. Stop all data manipulation language (DML) and data definition language (DDL) operations on non-transactional tables in the source DB instance and wait for them to complete. SELECT statements can continue running.
2. Flush and lock the tables in the source DB instance.
3. Create the read replica using one of the methods in the following sections.
4. Check the progress of the read replica creation using, for example, the `DescribeDBInstances` API operation. Once the read replica is available, unlock the tables of the source DB instance and resume normal database operations.

Configuring replication filters with MySQL

You can use replication filters to specify which databases and tables are replicated with a read replica. Replication filters can include databases and tables in replication or exclude them from replication.

The following are some use cases for replication filters:

- To reduce the size of a read replica. With replication filtering, you can exclude the databases and tables that aren't needed on the read replica.

- To exclude databases and tables from read replicas for security reasons.
- To replicate different databases and tables for specific use cases at different read replicas. For example, you might use specific read replicas for analytics or sharding.
- For a DB instance that has read replicas in different AWS Regions, to replicate different databases or tables in different AWS Regions.

Topics

- [Replication filtering parameters for Amazon RDS for MySQL \(p. 901\)](#)
- [Replication filtering limitations for Amazon RDS for MySQL \(p. 902\)](#)
- [Replication filtering examples for Amazon RDS for MySQL \(p. 902\)](#)
- [Viewing the replication filters for a read replica \(p. 905\)](#)

Replication filtering parameters for Amazon RDS for MySQL

To configure replication filters, set the following replication filtering parameters on the read replica:

- `replicate-do-db` – Replicate changes to the specified databases. When you set this parameter for a read replica, only the databases specified in the parameter are replicated.
- `replicate-ignore-db` – Don't replicate changes to the specified databases. When the `replicate-do-db` parameter is set for a read replica, this parameter isn't evaluated.
- `replicate-do-table` – Replicate changes to the specified tables. When you set this parameter for a read replica, only the tables specified in the parameter are replicated. Also, when the `replicate-do-db` or `replicate-ignore-db` parameter is set, make sure to include the database that includes the specified tables in replication with the read replica.
- `replicate-ignore-table` – Don't replicate changes to the specified tables. When the `replicate-do-table` parameter is set for a read replica, this parameter isn't evaluated.
- `replicate-wild-do-table` – Replicate tables based on the specified database and table name patterns. The % and _ wildcard characters are supported. When the `replicate-do-db` or `replicate-ignore-db` parameter is set, make sure to include the database that includes the specified tables in replication with the read replica.
- `replicate-wild-ignore-table` – Don't replicate tables based on the specified database and table name patterns. The % and _ wildcard characters are supported. When the `replicate-do-table` or `replicate-wild-do-table` parameter is set for a read replica, this parameter isn't evaluated.

The parameters are evaluated in the order that they are listed. For more information about how these parameters work, see the MySQL documentation:

- For general information, see [Replica Server Options and Variables](#).
- For information about how database replication filtering parameters are evaluated, see [Evaluation of Database-Level Replication and Binary Logging Options](#).
- For information about how table replication filtering parameters are evaluated, see [Evaluation of Table-Level Replication Options](#).

By default, each of these parameters has an empty value. On each read replica, you can use these parameters to set, change, and delete replication filters. When you set one of these parameters, separate each filter from others with a comma.

You can use the % and _ wildcard characters in the `replicate-wild-do-table` and `replicate-wild-ignore-table` parameters. The % wildcard matches any number of characters, and the _ wildcard matches only one character.

The binary logging format of the source DB instance is important for replication because it determines the record of data changes. The setting of the `binlog_format` parameter determines whether the replication is row-based or statement-based. For more information, see [Setting the binary logging format \(p. 524\)](#).

Note

All data definition language (DDL) statements are replicated as statements, regardless of the `binlog_format` setting on the source DB instance.

Replication filtering limitations for Amazon RDS for MySQL

The following limitations apply to replication filtering for Amazon RDS for MySQL:

- Each replication filtering parameter has a 2,000-character limit.
- Commas aren't supported in replication filters.
- The MySQL `--binlog-do-db` and `--binlog-ignore-db` options for binary log filtering aren't supported.
- Replication filtering doesn't support XA transactions.

For more information, see [Restrictions on XA Transactions](#) in the MySQL documentation.

- Replication filtering is supported for Amazon RDS for MySQL version 8.0.17 and higher 8.0 versions and version 5.7.26 and higher 5.7 versions.
- Replication filtering isn't supported for Amazon RDS for MySQL version 5.5 or 5.6.

Replication filtering examples for Amazon RDS for MySQL

To configure replication filtering for a read replica, modify the replication filtering parameters in the parameter group associated with the read replica.

Note

You can't modify a default parameter group. If the read replica is using a default parameter group, create a new parameter group and associate it with the read replica. For more information on DB parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

You can set parameters in a parameter group using the AWS Management Console, AWS CLI, or RDS API. For information about setting parameters, see [Modifying parameters in a DB parameter group \(p. 232\)](#). When you set parameters in a parameter group, all of the DB instances associated with the parameter group use the parameter settings. If you set the replication filtering parameters in a parameter group, make sure that the parameter group is associated only with read replicas. Leave the replication filtering parameters empty for source DB instances.

The following examples set the parameters using the AWS CLI. These examples set `ApplyMethod` to `immediate` so that the parameter changes occur immediately after the CLI command completes. If you want a pending change to be applied after the read replica is rebooted, set `ApplyMethod` to `pending-reboot`.

The following examples set replication filters:

- [Including databases in replication](#)
- [Including tables in replication](#)
- [Including tables in replication with wildcard characters](#)
- [Escaping wildcard characters in names](#)
- [Excluding databases from replication](#)
- [Excluding tables from replication](#)
- [Excluding tables from replication using wildcard characters](#)

Example Including databases in replication

The following example includes the `mydb1` and `mydb2` databases in replication.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
"ApplyMethod":"immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
"ApplyMethod":"immediate"}]"
```

Example Including tables in replication

The following example includes the `table1` and `table2` tables in database `mydb1` in replication.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-do-table", "ParameterValue":
"mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-do-table", "ParameterValue":
"mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example Including tables in replication using wildcard characters

The following example includes tables with names that begin with `orders` and `returns` in database `mydb` in replication.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue":
"mydb.orders%,mydb_returns%", "ApplyMethod":"immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue":
"mydb.orders%,mydb_returns%", "ApplyMethod":"immediate"}]"
```

Example Escaping wildcard characters in names

The following example shows you how to use the escape character \ to escape a wildcard character that is part of a name.

Assume that you have several table names in database mydb1 that start with my_table, and you want to include these tables in replication. The table names include an underscore, which is also a wildcard character, so the example escapes the underscore in the table names.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my\_table%", "ApplyMethod": "immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my\_table%", "ApplyMethod": "immediate"}]"
```

Example Excluding databases from replication

The following example excludes the mydb1 and mydb2 databases from replication.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue": "mydb1,mydb2", "ApplyMethod": "immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue": "mydb1,mydb2", "ApplyMethod": "immediate"}]"
```

Example Excluding tables from replication

The following example excludes tables table1 and table2 in database mydb1 from replication.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue": "mydb1.table1,mydb1.table2", "ApplyMethod": "immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue": "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example Excluding tables from replication using wildcard characters

The following example excludes tables with names that begin with `orders` and `returns` in database `mydb` from replication.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue": "mydb.orders%,mydb_returns%", "ApplyMethod":"immediate"}]"
```

For Windows:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name myparametergroup ^
--parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue": "mydb.orders%,mydb_returns%", "ApplyMethod":"immediate"}]"
```

Viewing the replication filters for a read replica

You can view the replication filters for a read replica in the following ways:

- Check the settings of the replication filtering parameters in the parameter group associated with the read replica.

For instructions, see [Viewing parameter values for a DB parameter group \(p. 239\)](#).

- In a MySQL client, connect to the read replica and run the `SHOW SLAVE STATUS` statement.

In the output, the following fields show the replication filters for the read replica:

- `Replicate_Do_DB`
- `Replicate_Ignore_DB`
- `Replicate_Do_Table`
- `Replicate_Ignore_Table`
- `Replicate_Wild_Do_Table`
- `Replicate_Wild_Ignore_Table`

For more information about these fields, see [Checking Replication Status](#) in the MySQL documentation.

Configuring delayed replication with MySQL

You can use delayed replication as a strategy for disaster recovery. With delayed replication, you specify the minimum amount of time, in seconds, to delay replication from the source to the read replica. In the event of a disaster, such as a table deleted unintentionally, you complete the following steps to recover from the disaster quickly:

- Stop replication to the read replica before the change that caused the disaster is sent to it.

Use the [mysql.rds_stop_replication \(p. 967\)](#) stored procedure to stop replication.

- Start replication and specify that replication stops automatically at a log file location.

You specify a location just before the disaster using the [mysql.rds_start_replication_until \(p. 965\)](#) stored procedure.

- Promote the read replica to be the new source DB instance by using the instructions in [Promoting a read replica to be a standalone DB instance \(p. 285\)](#).

Note

- On Amazon RDS for MySQL 5.7, delayed replication is supported for MySQL 5.7.22 and later. On Amazon RDS for MySQL 5.6, delayed replication is supported for MySQL 5.6.40 and later. Delayed replication is not supported on Amazon RDS for MySQL 8.0.
- Use stored procedures to configure delayed replication. You can't configure delayed replication with the AWS Management Console, the AWS CLI, or the Amazon RDS API.
- On Amazon RDS for MySQL 5.7.23 and later MySQL 5.7 versions, you can use replication based on global transaction identifiers (GTIDs) in a delayed replication configuration. If you use GTID-based replication, use the [mysql.rds_start_replication_until_gtid \(p. 966\)](#) stored procedure instead of the [mysql.rds_start_replication_until \(p. 965\)](#) stored procedure. For more information about GTID-based replication, see [Using GTID-based replication for RDS for MySQL \(p. 910\)](#).

Topics

- [Configuring delayed replication during read replica creation \(p. 906\)](#)
- [Modifying delayed replication for an existing read replica \(p. 906\)](#)
- [Setting a location to stop replication to a read replica \(p. 907\)](#)

Configuring delayed replication during read replica creation

To configure delayed replication for any future read replica created from a DB instance, run the [mysql.rds_set_configuration \(p. 972\)](#) stored procedure with the `target_delay` parameter.

To configure delayed replication during read replica creation

- Using a MySQL client, connect to the MySQL DB instance to be the source for read replicas as the master user.
- Run the [mysql.rds_set_configuration \(p. 972\)](#) stored procedure with the `target_delay` parameter.

For example, run the following stored procedure to specify that replication is delayed by at least one hour (3,600 seconds) for any read replica created from the current DB instance.

```
call mysql.rds_set_configuration('target_delay', 3600);
```

Note

After running this stored procedure, any read replica you create using the AWS CLI or Amazon RDS API is configured with replication delayed by the specified number of seconds.

Modifying delayed replication for an existing read replica

To modify delayed replication for an existing read replica, run the [mysql.rds_set_source_delay \(p. 964\)](#) stored procedure.

To modify delayed replication for an existing read replica

1. Using a MySQL client, connect to the read replica as the master user.
2. Use the [mysql.rds_stop_replication \(p. 967\)](#) stored procedure to stop replication.
3. Run the [mysql.rds_set_source_delay \(p. 964\)](#) stored procedure.

For example, run the following stored procedure to specify that replication to the read replica is delayed by at least one hour (3600 seconds).

```
call mysql.rds_set_source_delay(3600);
```

4. Use the [mysql.rds_start_replication \(p. 964\)](#) stored procedure to start replication.

Setting a location to stop replication to a read replica

After stopping replication to the read replica, you can start replication and then stop it at a specified binary log file location using the [mysql.rds_start_replication_until \(p. 965\)](#) stored procedure.

To start replication to a read replica and stop replication at a specific location

1. Using a MySQL client, connect to the source MySQL DB instance as the master user.
2. Run the [mysql.rds_start_replication_until \(p. 965\)](#) stored procedure.

The following example initiates replication and replicates changes until it reaches location 120 in the `mysql-bin-changelog.000777` binary log file. In a disaster recovery scenario, assume that location 120 is just before the disaster.

```
call mysql.rds_start_replication_until(
    'mysql-bin-changelog.000777',
    120);
```

Replication stops automatically when the stop point is reached. The following RDS event is generated:
`Replication has been stopped since the replica reached the stop point specified by the rds_start_replication_until stored procedure.`

After replication is stopped, in a disaster recovery scenario, you can [Promoting a read replica to be a standalone DB instance \(p. 285\)](#) promote the read replica to be the new source DB instance. For information about promoting the read replica, see [Promoting a read replica to be a standalone DB instance \(p. 285\)](#).

Read replica updates with MySQL

Read replicas are designed to support read queries, but you might need occasional updates. For example, you might need to add an index to optimize the specific types of queries accessing the replica. You can enable updates by setting the `read_only` parameter to 0 in the DB parameter group for the read replica. Be careful when disabling read-only on a read replica because it can cause problems if the read replica becomes incompatible with the source DB instance. Change the value of the `read_only` parameter back to 1 as soon as possible.

Multi-AZ read replica deployments with MySQL

You can create a read replica from either single-AZ or Multi-AZ DB instance deployments. You use Multi-AZ deployments to improve the durability and availability of critical data, but you can't use the Multi-AZ secondary to serve read-only queries. Instead, you can create read replicas from high-traffic Multi-AZ DB

instances to offload read-only queries. If the source instance of a Multi-AZ deployment fails over to the secondary, any associated read replicas automatically switch to use the secondary (now primary) as their replication source. For more information, see [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).

You can create a read replica as a Multi-AZ DB instance. Amazon RDS creates a standby of your replica in another Availability Zone for failover support for the replica. Creating your read replica as a Multi-AZ DB instance is independent of whether the source database is a Multi-AZ DB instance.

Note

To create a read replica as a Multi-AZ DB instance, the DB instance must be MySQL 5.6 or later.

Monitoring MySQL read replicas

For MySQL read replicas, you can monitor replication lag in Amazon CloudWatch by viewing the Amazon RDS `ReplicaLag` metric. The `ReplicaLag` metric reports the value of the `Seconds_Behind_Master` field of the `SHOW SLAVE STATUS` command.

Common causes for replication lag for MySQL are the following:

- A network outage.
- Writing to tables that have different indexes on a read replica. If the `read_only` parameter is set to 0 on the read replica, replication can break if the read replica becomes incompatible with the source DB instance. After you've performed maintenance tasks on the read replica, we recommend that you set the `read_only` parameter back to 1.
- Using a nontransactional storage engine such as MyISAM. Replication is only supported for the InnoDB storage engine on MySQL.

When the `ReplicaLag` metric reaches 0, the replica has caught up to the source DB instance. If the `ReplicaLag` metric returns -1, then replication is currently not active. `ReplicaLag = -1` is equivalent to `Seconds_Behind_Master = NULL`.

Starting and stopping replication with MySQL read replicas

You can stop and restart the replication process on an Amazon RDS DB instance by calling the system stored procedures [mysql.rds_stop_replication \(p. 967\)](#) and [mysql.rds_start_replication \(p. 964\)](#). You can do this when replicating between two Amazon RDS instances for long-running operations such as creating large indexes. You also need to stop and start replication when importing or exporting databases. For more information, see [Importing data to an Amazon RDS MySQL or MariaDB DB instance with reduced downtime \(p. 881\)](#) and [Exporting data from a MySQL DB instance by using replication \(p. 921\)](#).

If replication is stopped for more than 30 consecutive days, either manually or due to a replication error, Amazon RDS terminates replication between the source DB instance and all read replicas. It does so to prevent increased storage requirements on the source DB instance and long failover times. The read replica DB instance is still available. However, replication can't be resumed because the binary logs required by the read replica are deleted from the source DB instance after replication is terminated. You can create a new read replica for the source DB instance to reestablish replication.

Troubleshooting a MySQL read replica problem

For MySQL DB instances, in some cases read replicas present replication errors or data inconsistencies between the read replica and its source DB instance. This problem occurs when some binary log (binlog) events or InnoDB redo logs aren't flushed during a failure of the read replica or the source DB instance. In these cases, manually delete and recreate the read replicas. You can reduce the chance of this happening by setting the following parameter values: `sync_binlog=1` and `innodb_flush_log_at_trx_commit=1`. These settings might reduce performance, so test their

impact before implementing the changes in a production environment. For MySQL 5.5, `sync_binlog` defaults to 0, but in MySQL 5.6 and later, problems are less likely to occur because these parameters are all set to the recommended values by default.

The replication technologies for MySQL are asynchronous. Because they are asynchronous, occasional `BinLogDiskUsage` increases on the source DB instance and `ReplicaLag` on the read replica are to be expected. For example, a high volume of write operations to the source DB instance can occur in parallel. In contrast, write operations to the read replica are serialized using a single I/O thread, which can lead to a lag between the source instance and read replica. For more information about read-only replicas in the MySQL documentation, see [Replication implementation details](#).

You can do several things to reduce the lag between updates to a source DB instance and the subsequent updates to the read replica, such as the following:

- Sizing a read replica to have a storage size and DB instance class comparable to the source DB instance.
- Ensuring that parameter settings in the DB parameter groups used by the source DB instance and the read replica are compatible. For more information and an example, see the discussion of the `max_allowed_packet` parameter later in this section.

Amazon RDS monitors the replication status of your read replicas and updates the `Replication State` field of the read replica instance to `Error` if replication stops for any reason. An example might be if DML queries run on your read replica conflict with the updates made on the source DB instance.

You can review the details of the associated error thrown by the MySQL engine by viewing the `Replication Error` field. Events that indicate the status of the read replica are also generated, including [RDS-EVENT-0045 \(p. 492\)](#), [RDS-EVENT-0046 \(p. 492\)](#), and [RDS-EVENT-0047 \(p. 491\)](#). For more information about events and subscribing to events, see [Using Amazon RDS event notification \(p. 487\)](#). If a MySQL error message is returned, review the error number in the [MySQL error message documentation](#).

One common issue that can cause replication errors is when the value for the `max_allowed_packet` parameter for a read replica is less than the `max_allowed_packet` parameter for the source DB instance. The `max_allowed_packet` parameter is a custom parameter that you can set in a DB parameter group. You use `max_allowed_packet` to specify the maximum size of DML code that can be run on the database. In some cases, the `max_allowed_packet` value in the DB parameter group associated with a read replica is smaller than the `max_allowed_packet` value in the DB parameter group associated with the source DB instance. In these cases, the replication process can throw the error `Packet bigger than 'max_allowed_packet' bytes` and stop replication. To fix the error, have the source DB instance and read replica use DB parameter groups with the same `max_allowed_packet` parameter values.

Other common situations that can cause replication errors include the following:

- Writing to tables on a read replica. In some cases, you might create indexes on a read replica that are different from the indexes on the source DB instance. If you do, set the `read_only` parameter to 0 to create the indexes. If you write to tables on the read replica, it might break replication if the read replica becomes incompatible with the source DB instance. After you perform maintenance tasks on the read replica, we recommend that you set the `read_only` parameter back to 1.
- Using a non-transactional storage engine such as MyISAM. Read replicas require a transactional storage engine. Replication is only supported for the InnoDB storage engine on MySQL.
- Using unsafe nondeterministic queries such as `SYSDATE()`. For more information, see [Determination of safe and unsafe statements in binary logging](#).

If you decide that you can safely skip an error, you can follow the steps described in the section [Skipping the current replication error \(p. 933\)](#). Otherwise, you can first delete the read replica. Then you create

an instance using the same DB instance identifier so that the endpoint remains the same as that of your old read replica. If a replication error is fixed, the `Replication State` changes to *replicating*.

Using GTID-based replication for RDS for MySQL

Following, you can learn how to use global transaction identifiers (GTIDs) with binary log (binlog) replication among RDS for MySQL DB instances.

If you use binlog replication and aren't familiar with GTID-based replication with MySQL, see [Replication with global transaction identifiers](#) in the MySQL documentation for background.

Note

GTID-based replication is supported for RDS for MySQL version 5.7.23 and later MySQL 5.7 versions. All MySQL DB instances in a replication configuration must meet this requirement. GTID-based replication isn't supported for RDS for MySQL 5.5, 5.6, or 8.0.

Topics

- [Overview of global transaction identifiers \(GTIDs\) \(p. 910\)](#)
- [Parameters for GTID-based replication \(p. 910\)](#)
- [Configuring GTID-based replication for new read replicas \(p. 911\)](#)
- [Configuring GTID-based replication for existing read replicas \(p. 912\)](#)
- [Disabling GTID-based replication for a MySQL DB instance with read replicas \(p. 913\)](#)

Note

For information about configuring GTID-based replication with an external database, see [Replication with a MySQL or MariaDB instance running external to Amazon RDS \(p. 914\)](#).

Overview of global transaction identifiers (GTIDs)

Global transaction identifiers (GTIDs) are unique identifiers generated for committed MySQL transactions. You can use GTIDs to make binlog replication simpler and easier to troubleshoot.

MySQL uses two different types of transactions for binlog replication:

- *GTID transactions* – Transactions that are identified by a GTID.
- *Anonymous transactions* – Transactions that don't have a GTID assigned.

In a replication configuration, GTIDs are unique across all DB instances. GTIDs simplify replication configuration because when you use them, you don't have to refer to log file positions. GTIDs also make it easier to track replicated transactions and determine whether the source instance and replicas are consistent.

You can use GTID-based replication to replicate data with Amazon RDS MySQL read replicas or with an external MySQL database. For RDS for MySQL read replicas, you can configure GTID-based replication when you are creating new read replicas, or you can convert existing read replicas to use GTID-based replication.

You can also use GTID-based replication in a delayed replication configuration with RDS for MySQL . For more information, see [Configuring delayed replication with MySQL \(p. 905\)](#).

Parameters for GTID-based replication

Use the following parameters to configure GTID-based replication.

Parameter	Valid values	Description
gtid_mode	OFF, OFF_PERMISSIVE, ON_PERMISSIVE, ON	<p>OFF specifies that new transactions are anonymous transactions (that is, don't have GTIDs), and a transaction must be anonymous to be replicated.</p> <p>OFF_PERMISSIVE specifies that new transactions are anonymous transactions, but all transactions can be replicated.</p> <p>ON_PERMISSIVE specifies that new transactions are GTID transactions, but all transactions can be replicated.</p> <p>ON specifies that new transactions are GTID transactions, and a transaction must be a GTID transaction to be replicated.</p>
enforce_gtid_consistency	OFF, ON, WARN	<p>OFF allows transactions to violate GTID consistency.</p> <p>ON prevents transactions from violating GTID consistency.</p> <p>WARN allows transactions to violate GTID consistency but generates a warning when a violation occurs.</p>

Note

In the AWS Management Console, the `gtid_mode` parameter appears as `gtid-mode`.

For GTID-based replication, use these settings for the parameter group for your DB instance or read replica:

- ON and ON_PERMISSIVE apply only to outgoing replication from an RDS DB instance or Aurora MySQL cluster. Both of these values cause your RDS DB instance or Aurora DB cluster to use GTIDs for transactions that are replicated to an external database. ON requires that the external database also use GTID-based replication. ON_PERMISSIVE makes GTID-based replication optional on the external database.
- OFF_PERMISSIVE, if set, means that your RDS DB instances or Aurora DB cluster can accept incoming replication from an external database. It can do this whether the external database uses GTID-based replication or not.
- OFF, if set, means that your RDS DB instances or Aurora DB cluster only accept incoming replication from external databases that don't use GTID-based replication.

For more information about parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

Configuring GTID-based replication for new read replicas

When GTID-based replication is enabled for an RDS for MySQL DB instance, GTID-based replication is configured automatically for read replicas of the DB instance.

To enable GTID-based replication for new read replicas

1. Make sure that the parameter group associated with the DB instance has the following parameter settings:

- `gtid_mode` – ON or ON_PERMISSIVE
- `enforce_gtid_consistency` – ON

For more information about setting configuration parameters using parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

2. If you changed the parameter group of the DB instance, reboot the DB instance. For more information on how to do so, see [Rebooting a DB instance \(p. 276\)](#).
3. Create one or more read replicas of the DB instance. For more information on how to do so, see [Creating a read replica \(p. 283\)](#).

Amazon RDS attempts to establish GTID-based replication between the MySQL DB instance and the read replicas using the `MASTER_AUTO_POSITION`. If the attempt fails, Amazon RDS uses log file positions for replication with the read replicas. For more information about the `MASTER_AUTO_POSITION`, see [GTID auto-positioning](#) in the MySQL documentation.

Configuring GTID-based replication for existing read replicas

For an existing MySQL DB instance with read replicas that doesn't use GTID-based replication, you can configure GTID-based replication between the DB instance and the read replicas.

To enable GTID-based replication for existing read replicas

1. If the DB instance or any read replica is using RDS for MySQL version 5.7.22 or lower, upgrade the DB instance or read replica. Upgrade to RDS for MySQL version 5.7.23 or a later MySQL 5.7 version.

For more information, see [Upgrading the MySQL DB engine \(p. 853\)](#).

2. (Optional) Reset the GTID parameters and test the behavior of the DB instance and read replicas:

- a. Make sure that the parameter group associated with the DB instance and each read replica has the `enforce_gtid_consistency` parameter set to `WARN`.

For more information about setting configuration parameters using parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

- b. If you changed the parameter group of the DB instance, reboot the DB instance. If you changed the parameter group for a read replica, reboot the read replica.

For more information, see [Rebooting a DB instance \(p. 276\)](#).

- c. Run your DB instance and read replicas with your normal workload and monitor the log files.

If you see warnings about GTID-incompatible transactions, adjust your application so that it only uses GTID-compatible features. Make sure that the DB instance is not generating any warnings about GTID-incompatible transactions before proceeding to the next step.

3. Reset the GTID parameters for GTID-based replication that allows anonymous transactions until the read replicas have processed all of them.

- a. Make sure that the parameter group associated with the DB instance and each read replica has the following parameter settings:

- `gtid_mode` – ON_PERMISSIVE
- `enforce_gtid_consistency` – ON

- b. If you changed the parameter group of the DB instance, reboot the DB instance. If you changed the parameter group for a read replica, reboot the read replica.

4. Wait for all of your anonymous transactions to be replicated. To check that these are replicated, do the following:

- a. Run the following statement on your source DB instance.

```
SHOW MASTER STATUS;
```

Note the values in the `File` and `Position` columns.

- b. On each read replica, use the file and position information from its source instance in the previous step to run the following query.

```
SELECT MASTER_POS_WAIT('file', position);
```

For example, if the file name is `mysql-bin-changelog.000031` and the position is `107`, run the following statement.

```
SELECT MASTER_POS_WAIT('mysql-bin-changelog.000031', 107);
```

If the read replica is past the specified position, the query returns immediately. Otherwise, the function waits. Proceed to the next step when the query returns for all read replicas.

5. Reset the GTID parameters for GTID-based replication only.

- a. Make sure that the parameter group associated with the DB instance and each read replica has the following parameter settings:

- `gtid_mode` – ON
- `enforce_gtid_consistency` – ON

- b. Reboot the DB instance and each read replica.

6. On each read replica, run the following procedure.

```
CALL mysql.rds_set_master_auto_position(1);
```

Disabling GTID-based replication for a MySQL DB instance with read replicas

You can disable GTID-based replication for a MySQL DB instance with read replicas.

To disable GTID-based replication for a MySQL DB instance with read replicas

1. On each read replica, run the following procedure.

```
CALL mysql.rds_set_master_auto_position(0);
```

2. Reset the `gtid_mode` to `ON_PERMISSIVE`.

- a. Make sure that the parameter group associated with the MySQL DB instance and each read replica has `gtid_mode` set to `ON_PERMISSIVE`.

For more information about setting configuration parameters using parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

- b. Reboot the MySQL DB instance and each read replica. For more information about rebooting, see [Rebooting a DB instance \(p. 276\)](#).

3. Reset the `gtid_mode` to `OFF_PERMISSIVE`:

- a. Make sure that the parameter group associated with the MySQL DB instance and each read replica has `gtid_mode` set to `OFF_PERMISSIVE`.
 - b. Reboot the MySQL DB instance and each read replica.
4. Wait for all of the GTID transactions to be applied on all of the read replicas. To check that these are applied, do the following:
- Wait for all of the GTID transactions to be applied on the Aurora primary instance. To check that these are applied, do the following:
- a. On the MySQL DB instance, run the `SHOW MASTER STATUS` command.

Your output should be similar to the following.

File	Position
mysql-bin-changelog.000031	107

Note the file and position in your output.

- b. On each read replica, use the file and position information from its source instance in the previous step to run the following query.

```
SELECT MASTER_POS_WAIT('file', position);
```

For example, if the file name is `mysql-bin-changelog.000031` and the position is `107`, run the following statement.

```
SELECT MASTER_POS_WAIT('mysql-bin-changelog.000031', 107);
```

If the read replica is past the specified position, the query returns immediately. Otherwise, the function waits. When the query returns for all read replicas, go to the next step.

5. Reset the GTID parameters to disable GTID-based replication:
- a. Make sure that the parameter group associated with the MySQL DB instance and each read replica has the following parameter settings:
 - `gtid_mode` – OFF
 - `enforce_gtid_consistency` – OFF
 - b. Reboot the MySQL DB instance and each read replica.

Replication with a MySQL or MariaDB instance running external to Amazon RDS

You can set up replication between an Amazon RDS for MySQL or MariaDB DB instance and a MySQL or MariaDB instance that is external to Amazon RDS.

Topics

- [Before you begin \(p. 915\)](#)
- [Configuring binary log file position replication with an external source instance \(p. 915\)](#)
- [Configuring GTID-based replication with an external source instance \(p. 918\)](#)

Before you begin

You can configure replication using the binary log file position of replicated transactions. On Amazon RDS MySQL 5.7.23 and later MySQL 5.7 versions, you can also configure replication using global transaction identifiers (GTIDs).

The permissions required to start replication on an Amazon RDS DB instance are restricted and not available to your Amazon RDS master user. Because of this, make sure that you use the Amazon RDS [mysql.rds_set_external_master \(p. 954\)](#) and [mysql.rds_start_replication \(p. 964\)](#) commands to set up replication between your live database and your Amazon RDS database.

To set the binary logging format for a MySQL or MariaDB database, update the `binlog_format` parameter. If your DB instance uses the default DB instance parameter group, create a new DB parameter group to modify `binlog_format` settings. We recommend that you use the default setting for `binlog_format`, which is `MIXED`. However, you can also set `binlog_format` to `ROW` or `STATEMENT` if you need a specific binlog format. Reboot your DB instance for the change to take effect.

For information about setting the `binlog_format` parameter, see [Setting the binary logging format \(p. 524\)](#). For information about the implications of different MySQL replication types, see [Advantages and disadvantages of statement-based and row-based replication](#) in the MySQL documentation.

Note

Use the procedure in this topic to configure replication in all cases except when the external instance is MariaDB version 10.0.2 or greater and the Amazon RDS instance is MariaDB.

In that case, use the procedure at [Configuring GTID-based replication into a MariaDB DB instance \(p. 613\)](#) to set up GTID-based replication.

Configuring binary log file position replication with an external source instance

Follow these guidelines when you set up an external source instance and a replica on Amazon RDS:

- Monitor failover events for the Amazon RDS DB instance that is your replica. If a failover occurs, then the DB instance that is your replica might be recreated on a new host with a different network address. For information on how to monitor failover events, see [Using Amazon RDS event notification \(p. 487\)](#).
- Maintain the binary logs (binlogs) on your source instance until you have verified that they have been applied to the replica. This maintenance makes sure that you can restore your source instance in the event of a failure.
- Turn on automated backups on your Amazon RDS DB instance. Turning on automated backups makes sure that you can restore your replica to a particular point in time if you need to re-synchronize your source instance and replica. For information on backups and point-in-time restore, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#).

To configure binary log file replication with an external source instance

1. Make the source MySQL or MariaDB instance read-only.

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

2. Run the `SHOW MASTER STATUS` command on the source MySQL or MariaDB instance to determine the binlog location.

You receive output similar to the following example.

File	Position
mysql-bin-changelog.000031	107

3. Copy the database from the external instance to the Amazon RDS DB instance using `mysqldump`. For very large databases, you might want to use the procedure in [Importing data to an Amazon RDS MySQL or MariaDB DB instance with reduced downtime \(p. 881\)](#).

For Linux, macOS, or Unix:

```
mysqldump --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql \
        --host=hostname \
        --port=3306 \
        -u RDS_user_name \
        -pRDS_password
```

For Windows:

```
mysqldump --databases database_name ^
    --single-transaction ^
    --compress ^
    --order-by-primary ^
    -u local_user ^
    -plocal_password | mysql ^
        --host=hostname ^
        --port=3306 ^
        -u RDS_user_name ^
        -pRDS_password
```

Note

Make sure that there isn't a space between the `-p` option and the entered password.

To specify the host name, user name, port, and password to connect to your Amazon RDS DB instance, use the `--host`, `--user` (`-u`), `--port` and `-p` options in the `mysql` command. The host name is the Domain Name Service (DNS) name from the Amazon RDS DB instance endpoint, for example, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. You can find the endpoint value in the instance details in the AWS Management Console.

4. Make the source MySQL or MariaDB instance writeable again.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

For more information on making backups for use with replication, see [the MySQL documentation](#).

5. In the AWS Management Console, add the IP address of the server that hosts the external database to the VPC security group for the Amazon RDS DB instance. For more information on modifying a VPC security group, see [Security groups for your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

The IP address can change when the following conditions are met:

- You are using a public IP address for communication between the external source instance and the DB instance.

- The external source instance was stopped and restarted.

If these conditions are met, verify the IP address before adding it.

You might also need to configure your local network to permit connections from the IP address of your Amazon RDS DB instance. You do this so that your local network can communicate with your external MySQL or MariaDB instance. To find the IP address of the Amazon RDS DB instance, use the host command.

```
host db_instance_endpoint
```

The host name is the DNS name from the Amazon RDS DB instance endpoint.

6. Using the client of your choice, connect to the external instance and create a user to use for replication. Use this account solely for replication. and restrict it to your domain to improve security. The following is an example.

MySQL 5.5, 5.6, and 5.7

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

MySQL 8.0

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY  
'password';
```

7. For the external instance, grant REPLICATION CLIENT and REPLICATION SLAVE privileges to your replication user. For example, to grant the REPLICATION CLIENT and REPLICATION SLAVE privileges on all databases for the 'repl_user' user for your domain, issue the following command.

MySQL 5.5, 5.6, and 5.7

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'  
IDENTIFIED BY 'password';
```

MySQL 8.0

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Make the Amazon RDS DB instance the replica. To do so, first connect to the Amazon RDS DB instance as the master user. Then identify the external MySQL or MariaDB database as the source instance by using the [mysql.rds_set_external_master \(p. 954\)](#) command. Use the master log file name and master log position that you determined in step 2. The following is an example.

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306, 'repl_user',  
'password', 'mysql-bin-changelog.000031', 107, 0);
```

Note

On Amazon RDS for MySQL, you can choose to use delayed replication by running the [mysql.rds_set_external_master_with_delay \(p. 956\)](#) stored procedure instead.

One reason to use delayed replication is to enable disaster recovery with the [mysql.rds_start_replication_until \(p. 965\)](#) stored procedure. Currently, delayed replication is not supported on RDS for MariaDB.

9. On the Amazon RDS DB instance, issue the [mysql.rds_start_replication \(p. 964\)](#) command to start replication:

```
CALL mysql.rds_start_replication;
```

Configuring GTID-based replication with an external source instance

When you set up an external source instance and a replica on Amazon RDS, monitor failover events for the Amazon RDS DB instance that is your replica. If a failover occurs, then the DB instance that is your replica might be recreated on a new host with a different network address. For information on how to monitor failover events, see [Using Amazon RDS event notification \(p. 487\)](#).

Important

GTID-based replication is only supported on Amazon RDS for MySQL version 5.7.23 and later MySQL 5.7 versions. GTID-based replication is not supported for Amazon RDS for MySQL 5.5, 5.6, or 8.0.

To configure GTID-based replication with an external source instance

1. Prepare for GTID-based replication:

- a. Make sure that the external MySQL or MariaDB database has GTID-based replication enabled. To do so, make sure that the external database has the following parameters set to the specified values:

```
gtid_mode - ON  
enforce_gtid_consistency - ON
```

For more information, see [Replication with global transaction identifiers](#) in the MySQL documentation or [Global transaction ID](#) in the MariaDB documentation.

- b. Make sure that the parameter group associated with the DB instance has the following parameter settings:

- `gtid_mode` – ON, ON_PERMISSIVE, or OFF_PERMISSIVE
- `enforce_gtid_consistency` – ON

For more information about parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

- c. If you changed the parameter group of the DB instance, reboot the DB instance. For more information, see [Rebooting a DB instance \(p. 276\)](#).

2. Make the source MySQL or MariaDB instance read-only.

```
mysql> FLUSH TABLES WITH READ LOCK;  
mysql> SET GLOBAL read_only = ON;
```

3. Copy the database from the external instance to the Amazon RDS DB instance using `mysqldump`. For very large databases, you might want to use the procedure in [Importing data to an Amazon RDS MySQL or MariaDB DB instance with reduced downtime \(p. 881\)](#).

For Linux, macOS, or Unix:

```
mysqldump --databases database_name \  
--single-transaction \  
--compress \  
--order-by-primary \  
-u local_user \  
-
```

```
-plocal_password | mysql \
--host=hostname \
--port=3306 \
-u RDS_user_name \
-pRDS_password
```

For Windows:

```
mysqldump --databases database_name ^
--single-transaction ^
--compress ^
--order-by-primary ^
-u local_user ^
-plocal_password | mysql ^
--host=hostname ^
--port=3306 ^
-u RDS_user_name ^
-pRDS_password
```

Note

Make sure that there is not a space between the `-p` option and the entered password.

To specify the host name, user name, port, and password to connect to your Amazon RDS DB instance, use the `--host`, `--user` (`-u`), `--port` and `-p` options in the `mysql` command. The host name is the DNS name from the Amazon RDS DB instance endpoint, for example, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. You can find the endpoint value in the instance details in the AWS Management Console.

4. Make the source MySQL or MariaDB instance writeable again.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

For more information on making backups for use with replication, see [the MySQL documentation](#).

5. In the AWS Management Console, add the IP address of the server that hosts the external database to the VPC security group for the Amazon RDS DB instance. For more information on modifying a VPC security group, see [Security groups for your VPC](#) in the *Amazon Virtual Private Cloud User Guide*.

The IP address can change when the following conditions are met:

- You are using a public IP address for communication between the external source instance and the DB instance.
- The external source instance was stopped and restarted.

If these conditions are met, verify the IP address before adding it.

You might also need to configure your local network to permit connections from the IP address of your Amazon RDS DB instance. You do this so that your local network can communicate with your external MySQL or MariaDB instance. To find the IP address of the Amazon RDS DB instance, use the `host` command.

```
host db_instance_endpoint
```

The host name is the DNS name from the Amazon RDS DB instance endpoint.

6. Using the client of your choice, connect to the external instance and create a user to use for replication. Use this account solely for replication, and restrict it to your domain to improve security. The following is an example.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

7. For the external instance, grant REPLICATION CLIENT and REPLICATION SLAVE privileges to your replication user. For example, to grant the REPLICATION CLIENT and REPLICATION SLAVE privileges on all databases for the 'repl_user' user for your domain, issue the following command.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'  
IDENTIFIED BY 'password';
```

8. Make the Amazon RDS DB instance the replica. To do so, first connect to the Amazon RDS DB instance as the master user. Then identify the external MySQL or MariaDB database as the replication primary instance by using the [mysql.rds_set_external_master_with_auto_position \(p. 959\)](#) command. The following is an example.

```
CALL mysql.rds_set_external_master_with_auto_position ('mymasterserver.mydomain.com',  
3306, 'repl_user', 'password', 0, 0);
```

Note

On Amazon RDS for MySQL, you can choose to use delayed replication by running the [mysql.rds_set_external_master_with_delay \(p. 956\)](#) stored procedure instead. One reason to use delayed replication is to enable disaster recovery with the [mysql.rds_start_replication_until_gtid \(p. 966\)](#) stored procedure. Currently, delayed replication is not supported on RDS for MariaDB.

9. On the Amazon RDS DB instance, issue the [mysql.rds_start_replication \(p. 964\)](#) command to start replication.

```
CALL mysql.rds_start_replication;
```

Exporting data from a MySQL DB instance by using replication

To export data from a MySQL 5.6 or later DB instance to a MySQL instance running external to Amazon RDS, you can use replication. In this scenario, the MySQL DB instance is the *source MySQL DB instance*, and the MySQL instance running external to Amazon RDS is the *external MySQL database*.

The source MySQL DB instance must be running version 5.6.13 or later. The external MySQL database can run either on-premises in your data center, or on an Amazon EC2 instance. The external MySQL database must run the same version as the source MySQL DB instance, or a later version.

Replication to an external MySQL database is only supported during the time it takes to export a database from the source MySQL DB instance. The replication should be terminated when the data has been exported and applications can start accessing the external MySQL instance.

The following list shows the steps to take. Each step is discussed in more detail in later sections.

1. Prepare an external MySQL DB instance.
2. Prepare the source MySQL DB instance for replication.
3. Use the mysqldump utility to transfer the database from the source MySQL DB instance to the external MySQL database.
4. Start replication to the external MySQL database.
5. After the export completes, stop replication.

Prepare an external MySQL database

Perform the following steps to prepare the external MySQL database.

To prepare the external MySQL database

1. Install the external MySQL database.
2. Connect to the external MySQL database as the master user. Then create the users required to support the administrators, applications, and services that access the database.
3. Follow the directions in the MySQL documentation to prepare the external MySQL database as a replica. For more information, see [the MySQL documentation](#).
4. Configure an egress rule for the external MySQL database to operate as a read replica during the export. The egress rule allows the external MySQL database to connect to the source MySQL DB instance during replication. Specify an egress rule that allows Transmission Control Protocol (TCP) connections to the port and IP address of the source MySQL DB instance.

Specify the appropriate egress rules for your environment:

- If the external MySQL database is running in an Amazon EC2 instance in a virtual private cloud (VPC) based on the Amazon VPC service, specify the egress rules in a VPC security group. For more information, see [Controlling access with security groups \(p. 1699\)](#).
 - If the external MySQL database is running in an Amazon EC2 instance that is not in a VPC, specify the egress rules in an EC2-Classic security group.
 - If the external MySQL database is installed on-premises, specify the egress rules in a firewall.
5. If the external MySQL database is running in a VPC, configure rules for the VPC access control list (ACL) rules in addition to the security group egress rule:
 - Configure an ACL ingress rule allowing TCP traffic to ports 1024–65535 from the IP address of the source MySQL DB instance.

- Configure an ACL egress rule allowing outbound TCP traffic to the port and IP address of the source MySQL DB instance.

For more information about Amazon VPC network ACLs, see [Network ACLs](#) in *Amazon VPC User Guide*.

6. (Optional) Set the `max_allowed_packet` parameter to the maximum size to avoid replication errors. We recommend this setting.

Prepare the source MySQL DB instance

Perform the following steps to prepare the source MySQL DB instance as the replication source.

To prepare the source MySQL DB instance

1. Ensure that your client computer has enough disk space available to save the binary logs while setting up replication.
2. Connect to the source MySQL DB instance, and create a replication account by following the directions in [Creating a user for replication](#) in the MySQL documentation.
3. Configure ingress rules on the system running the source MySQL DB instance to allow the external MySQL database to connect during replication. Specify an ingress rule that allows TCP connections to the port used by the source MySQL DB instance from the IP address of the external MySQL database.
4. Specify the egress rules:
 - If the source MySQL DB instance is running in a VPC, specify the ingress rules in a VPC security group. For more information, see [Controlling access with security groups \(p. 1699\)](#).
 - If the source MySQL DB instance isn't running in a VPC, specify the ingress rules in a DB security group. For more information, see [Authorizing network access to a DB security group from an IP range \(p. 1708\)](#).
5. If source MySQL DB instance is running in a VPC, configure VPC ACL rules in addition to the security group ingress rule:
 - Configure an ACL ingress rule to allow TCP connections to the port used by the Amazon RDS instance from the IP address of the external MySQL database.
 - Configure an ACL egress rule to allow TCP connections from ports 1024–65535 to the IP address of the external MySQL database.

For more information about Amazon VPC network ACLs, see [Network ACLs](#) in the *Amazon VPC User Guide*.

6. Ensure that the backup retention period is set long enough that no binary logs are purged during the export. If any of the logs are purged before the export has completed, you must restart replication from the beginning. For more information about setting the backup retention period, see [Working with backups \(p. 328\)](#).
7. Use the `mysql.rds_set_configuration` stored procedure to set the binary log retention period long enough that the binary logs aren't purged during the export. For more information, see [Accessing MySQL binary logs \(p. 525\)](#).
8. Create an Amazon RDS read replica from the source MySQL DB instance to further ensure that the binary logs of the source MySQL DB instance are not purged. For more information, see [Creating a read replica \(p. 283\)](#).
9. After the Amazon RDS read replica has been created, call the `mysql.rds_stop_replication` stored procedure to stop the replication process. The source MySQL DB instance no longer purges its binary log files, so they are available for the replication process.

10. (Optional) Set both the `max_allowed_packet` parameter and the `slave_max_allowed_packet` parameter to the maximum size to avoid replication errors. The maximum size for both parameters is 1 GB. We recommend this setting for both parameters. For information about setting parameters, see [Modifying parameters in a DB parameter group \(p. 232\)](#).

Copy the database

Perform the following steps to copy the database.

To copy the database

1. Connect to the RDS read replica of the source MySQL DB instance, and run the MySQL `SHOW SLAVE STATUS\G` statement. Note the values for the following:
 - `Master_Host`
 - `Master_Port`
 - `Master_Log_File`
 - `Exec_Master_Log_Pos`
2. Use the `mysqldump` utility to create a snapshot, which copies the data from Amazon RDS to your local client computer. Then run another utility to load the data into the external MySQL database. Ensure that your client computer has enough space to hold the `mysqldump` files from the databases to be replicated. This process can take several hours for very large databases. Follow the directions in [Creating a data snapshot using mysqldump](#) in the MySQL documentation.

The following example runs `mysqldump` on a client, and then pipes the dump into the `mysql` client utility, which loads the data into the external MySQL database.

For Linux, macOS, or Unix:

```
mysqldump -h source_MySQL_DB_instance_endpoint \
-u user \
-ppassword \
--port=3306 \
--single-transaction \
--routines \
--triggers \
--databases database database2 \
--compress \
--port 3306
```

For Windows:

```
mysqldump -h source_MySQL_DB_instance_endpoint ^
-u user ^
-ppassword ^
--port=3306 ^
--single-transaction ^
--routines ^
--triggers ^
--databases database database2 ^
--compress ^
--port 3306
```

The following example runs `mysqldump` on a client and writes the dump to a file.

For Linux, macOS, or Unix:

```
mysqldump -h source_MySQL_DB_instance_endpoint \
-u user \
-ppassword \
--port=3306 \
--single-transaction \
--routines \
--triggers \
--databases database database2 > path/rds-dump.sql
```

For Windows:

```
mysqldump -h source_MySQL_DB_instance_endpoint ^
-u user ^
-ppassword ^
--port=3306 ^
--single-transaction ^
--routines ^
--triggers ^
--databases database database2 > path\rds-dump.sql
```

Complete the export

Perform the following steps to complete the export.

To complete the export

1. Load the mysqldump files to create the databases on the external MySQL database.
2. On the Amazon RDS read replica, call the `mysql.rds_start_replication` stored procedure. Doing this starts replication from the source MySQL DB instance and exports all source changes that have occurred after you stopped replication from the Amazon RDS read replica.
3. Use the MySQL `CHANGE MASTER` statement to configure the external MySQL database. Specify the ID and password of the user granted `REPLICATION SLAVE` permissions. Specify the `Master_Host`, `Master_Port`, `Relay_Master_Log_File`, and `Exec_Master_Log_Pos` values that you got from the MySQL `SHOW SLAVE STATUS\G` statement that you ran on the RDS read replica. For more information, see [the MySQL documentation](#).
4. Use the MySQL `START SLAVE` command to initiate replication from the source MySQL DB instance to the external MySQL database.
5. Run the MySQL `SHOW SLAVE STATUS\G` command on the external MySQL database to verify that it is operating as a read replica. For more information about interpreting the results, see [the MySQL documentation](#).
6. After replication on the external MySQL database has caught up with the source MySQL DB instance, use the MySQL `STOP SLAVE` command to stop replication from the source MySQL DB instance.
7. On the Amazon RDS read replica, call the `mysql.rds_start_replication` stored procedure. Doing this allows Amazon RDS to start purging the binary log files from the source MySQL DB instance.

Options for MySQL DB instances

This appendix describes options, or additional features, that are available for Amazon RDS instances running the MySQL DB engine. To enable these options, you can add them to a custom option group, and then associate the option group with your DB instance. For more information about working with option groups, see [Working with option groups \(p. 212\)](#).

Amazon RDS supports the following options for MySQL:

Option	Option ID	Engine versions
MariaDB Audit Plugin support (p. 926)	MARIADB_AUDIT_PLUGIN	All MySQL 5.6 versions
		MySQL 5.7.16 and later 5.7 versions
MySQL memcached support (p. 929)	MEMCACHED	All MySQL 5.6, 5.7, and 8.0 versions

MariaDB Audit Plugin support

Amazon RDS supports using the MariaDB Audit Plugin on MySQL database instances. The MariaDB Audit Plugin records database activity such as users logging on to the database, queries run against the database, and more. The record of database activity is stored in a log file.

Note

Currently, the MariaDB Audit Plugin is only supported for the following RDS for MySQL versions:

- All 5.6 versions
 - MySQL 5.7.16 and later 5.7 versions

Audit Plugin option settings

Amazon RDS supports the following settings for the MariaDB Audit Plugin option.

Option setting	Valid values	Default value	Description
SERVER_AUDIT	/rdsdbdata/log/audit/	/rdsdbdata/log/audit/	The location of the log file. The log file contains the record of the activity specified in SERVER_AUDIT_EVENTS. For more information, see Viewing and listing database log files (p. 504) and Accessing MySQL database log files (p. 519) .
SERVER_AUDIT_SIZE	100000000	1000000	The size in bytes that when reached, causes the file to rotate. For more information, see Overview of MySQL database logs (p. 519) .
SERVER_AUDIT_ROTATIONS	100	9	The number of log rotations to save. For more information, see Overview of MySQL database logs (p. 519) and Downloading a database log file (p. 504) .
SERVER_AUDIT_EVENTS	CONNECT, QUERY, QUERY_DDL, QUERY_DML, QUERY_DCL	CONNECT, QUERY	<p>The types of activity to record in the log. Installing the MariaDB Audit Plugin is itself logged.</p> <ul style="list-style-type: none"> • CONNECT: Log successful and unsuccessful connections to the database, and disconnections from the database. • QUERY: Log the text of all queries run against the database. • QUERY_DDL: Similar to the QUERY event, but returns only data definition language (DDL) queries (CREATE, ALTER, and so on). • QUERY_DML: Similar to the QUERY event, but returns only data manipulation language (DML) queries (INSERT, UPDATE, and so on, and also SELECT). • QUERY_DCL: Similar to the QUERY event, but returns only data control language (DCL) queries (GRANT, REVOKE, and so on). <p>For MySQL, TABLE is not supported.</p>

Option setting	Valid values	Default value	Description
SERVER_AUDIT_USERS	Multiple comma-separated values	None	Include only activity from the specified users. By default, activity is recorded for all users. If a user is specified in both SERVER_AUDIT_EXCL_USERS and SERVER_AUDIT_INCL_USERS, then activity is recorded for the user.
SERVER_AUDIT_EXCL_USERS	Multiple comma-separated values	None	Exclude activity from the specified users. By default, activity is recorded for all users. If a user is specified in both SERVER_AUDIT_EXCL_USERS and SERVER_AUDIT_INCL_USERS, then activity is recorded for the user.
SERVER_AUDIT_LOGGING	ON	ON	The rdsadmin user queries the database every second to check the health of the database. Depending on your other settings, this activity can possibly cause the size of your log file to grow very large, very quickly. If you don't need to record this activity, add the rdsadmin user to the SERVER_AUDIT_EXCL_USERS list.
SERVER_AUDIT_QUERY_SIZE	002147483647M1024		The limit on the length of the query string in a record.

Adding the MariaDB Audit Plugin

The general process for adding the MariaDB Audit Plugin to a DB instance is the following:

- Create a new option group, or copy or modify an existing option group
- Add the option to the option group
- Associate the option group with the DB instance

After you add the MariaDB Audit Plugin, you don't need to restart your DB instance. As soon as the option group is active, auditing begins immediately.

Important

Adding the MariaDB Audit Plugin to a DB instance might cause an outage. We recommend adding the MariaDB Audit Plugin during a maintenance window or during a time of low database workload.

To add the MariaDB Audit Plugin

1. Determine the option group you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a

- custom DB option group. Choose **mysql** for **Engine**, and choose **5.6** or **5.7** for **Major engine version**. For more information, see [Creating an option group \(p. 214\)](#).
2. Add the **MARIADB_AUDIT_PLUGIN** option to the option group, and configure the option settings. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#). For more information about each setting, see [Audit Plugin option settings \(p. 926\)](#).
 3. Apply the option group to a new or existing DB instance.
 - For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
 - For an existing DB instance, you apply the option group by modifying the instance and attaching the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Viewing and downloading the MariaDB Audit Plugin log

After you enable the MariaDB Audit Plugin, you access the results in the log files the same way you access any other text-based log files. The audit log files are located at `/rdsdbdata/log/audit/`. For information about viewing the log file in the console, see [Viewing and listing database log files \(p. 504\)](#). For information about downloading the log file, see [Downloading a database log file \(p. 504\)](#).

Modifying MariaDB Audit Plugin settings

After you enable the MariaDB Audit Plugin, you can modify the settings. For more information about how to modify option settings, see [Modifying an option setting \(p. 221\)](#). For more information about each setting, see [Audit Plugin option settings \(p. 926\)](#).

Removing the MariaDB Audit Plugin

Amazon RDS doesn't support turning off logging in the MariaDB Audit Plugin. However, you can remove the plugin from a DB instance. When you remove the MariaDB Audit Plugin, the DB instance is restarted automatically to stop auditing.

To remove the MariaDB Audit Plugin from a DB instance, do one of the following:

- Remove the MariaDB Audit Plugin option from the option group it belongs to. This change affects all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).
- Modify the DB instance and specify a different option group that doesn't include the plugin. This change affects a single DB instance. You can specify the default (empty) option group, or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

MySQL memcached support

Amazon RDS supports using the memcached interface to InnoDB tables that was introduced in MySQL 5.6. The memcached API enables applications to use InnoDB tables in a manner similar to NoSQL key-value data stores.

The memcached interface is a simple, key-based cache. Applications use memcached to insert, manipulate, and retrieve key-value data pairs from the cache. MySQL 5.6 introduced a plugin that implements a daemon service that exposes data from InnoDB tables through the memcached protocol. For more information about the MySQL memcached plugin, see [InnoDB integration with memcached](#).

To enable memcached support for an RDS for MySQL 5.6 or later instance

1. Determine the security group to use for controlling access to the memcached interface. If the set of applications already using the SQL interface are the same set that will access the memcached interface, you can use the existing VPC or DB security group used by the SQL interface. If a different set of applications will access the memcached interface, define a new VPC or DB security group. For more information about managing security groups, see [Controlling access with security groups \(p. 1699\)](#)
2. Create a custom DB option group, selecting MySQL as the engine type and a 5.6 or later version. For more information about creating an option group, see [Creating an option group \(p. 214\)](#).
3. Add the MEMCACHED option to the option group. Specify the port that the memcached interface will use, and the security group to use in controlling access to the interface. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).
4. Modify the option settings to configure the memcached parameters, if necessary. For more information about how to modify option settings, see [Modifying an option setting \(p. 221\)](#).
5. Apply the option group to an instance. Amazon RDS enables memcached support for that instance when the option group is applied:
 - You enable memcached support for a new instance by specifying the custom option group when you launch the instance. For more information about launching a MySQL instance, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
 - You enable memcached support for an existing instance by specifying the custom option group when you modify the instance. For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
6. Specify which columns in your MySQL tables can be accessed through the memcached interface. The memcached plug-in creates a catalog table named `containers` in a dedicated database named `innodb_memcache`. You insert a row into the `containers` table to map an InnoDB table for access through memcached. You specify a column in the InnoDB table that is used to store the memcached key values, and one or more columns that are used to store the data values associated with the key. You also specify a name that a memcached application uses to refer to that set of columns. For details on inserting rows in the `containers` table, see [InnoDB memcached plugin internals](#). For an example of mapping an InnoDB table and accessing it through memcached, see [Writing applications for the InnoDB memcached plugin](#).
7. If the applications accessing the memcached interface are on different computers or EC2 instances than the applications using the SQL interface, add the connection information for those computers to the VPC or DB security group associated with the MySQL instance. For more information about managing security groups, see [Controlling access with security groups \(p. 1699\)](#).

You turn off the memcached support for an instance by modifying the instance and specifying the default option group for your MySQL version. For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

MySQL memcached security considerations

The memcached protocol does not support user authentication. For more information about MySQL memcached security considerations, see [memcached deployment](#) and [Using memcached as a MySQL caching layer](#).

You can take the following actions to help increase the security of the memcached interface:

- Specify a different port than the default of 11211 when adding the **MEMCACHED** option to the option group.
- Ensure that you associate the memcached interface with either a VPC or DB security group that limits access to known, trusted client addresses or EC2 instances. For more information about managing security groups, see [Controlling access with security groups \(p. 1699\)](#).

MySQL memcached connection information

To access the memcached interface, an application must specify both the DNS name of the Amazon RDS instance and the memcached port number. For example, if an instance has a DNS name of `my-cache-instance.cg034hpkmmt.region.rds.amazonaws.com` and the memcached interface is using port 11212, the connection information specified in PHP would be:

```
<?php  
  
$cache = new Memcache;  
$cache->connect('my-cache-instance.cg034hpkmmt.region.rds.amazonaws.com', 11212);  
?>
```

To find the DNS name and memcached port of a MySQL DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the top right corner of the AWS Management Console, select the region that contains the DB instance.
3. In the navigation pane, choose **Databases**.
4. Choose the MySQL DB instance name to display its details.
5. In the **Connect** section, note the value of the **Endpoint** field. The DNS name is the same as the endpoint. Also, note that the port in the **Connect** section is not used to access the memcached interface.
6. In the **Details** section, note the name listed in the **Option Group** field.
7. In the navigation pane, choose **Option groups**.
8. Choose the name of the option group used by the MySQL DB instance to show the option group details. In the **Options** section, note the value of the **Port** setting for the **MEMCACHED** option.

MySQL memcached option settings

Amazon RDS exposes the MySQL memcached parameters as option settings in the Amazon RDS **MEMCACHED** option.

MySQL memcached parameters

- **DAEMON_MEMCACHED_R_BATCH_SIZE** – an integer that specifies how many memcached read operations (get) to perform before doing a COMMIT to start a new transaction. The allowed values are 1 to 4294967295; the default is 1. The option does not take effect until the instance is restarted.

- **DAEMON_MEMCACHED_W_BATCH_SIZE** – an integer that specifies how many memcached write operations, such as add, set, or incr, to perform before doing a COMMIT to start a new transaction. The allowed values are 1 to 4294967295; the default is 1. The option does not take effect until the instance is restarted.
- **INNODB_API_BK_COMMIT_INTERVAL** – an integer that specifies how often to auto-commit idle connections that use the InnoDB memcached interface. The allowed values are 1 to 1073741824; the default is 5. The option takes effect immediately, without requiring that you restart the instance.
- **INNODB_API_DISABLE_ROWLOCK** – a Boolean that disables (1 (true)) or enables (0 (false)) the use of row locks when using the InnoDB memcached interface. The default is 0 (false). The option does not take effect until the instance is restarted.
- **INNODB_API_ENABLE_MDL** – a Boolean that when set to 0 (false) locks the table used by the InnoDB memcached plugin, so that it cannot be dropped or altered by DDL through the SQL interface. The default is 0 (false). The option does not take effect until the instance is restarted.
- **INNODB_API_TRX_LEVEL** – an integer that specifies the transaction isolation level for queries processed by the memcached interface. The allowed values are 0 to 3. The default is 0. The option does not take effect until the instance is restarted.

Amazon RDS configures these MySQL memcached parameters, and they cannot be modified: DAEMON_MEMCACHED_LIB_NAME, DAEMON_MEMCACHED_LIB_PATH, and INNODB_API_ENABLE_BINLOG. The parameters that MySQL administrators set by using `daemon_memcached_options` are available as individual MEMCACHED option settings in Amazon RDS.

MySQL `daemon_memcached_options` parameters

- **BINDING_PROTOCOL** – a string that specifies the binding protocol to use. The allowed values are auto, ascii, or binary. The default is auto, which means the server automatically negotiates the protocol with the client. The option does not take effect until the instance is restarted.
- **BACKLOG_QUEUE_LIMIT** – an integer that specifies how many network connections can be waiting to be processed by memcached. Increasing this limit may reduce errors received by a client that is not able to connect to the memcached instance, but does not improve the performance of the server. The allowed values are 1 to 2048; the default is 1024. The option does not take effect until the instance is restarted.
- **CAS_DISABLED** – a Boolean that enables (1 (true)) or disables (0 (false)) the use of compare and swap (CAS), which reduces the per-item size by 8 bytes. The default is 0 (false). The option does not take effect until the instance is restarted.
- **CHUNK_SIZE** – an integer that specifies the minimum chunk size, in bytes, to allocate for the smallest item's key, value, and flags. The allowed values are 1 to 48. The default is 48 and you can significantly improve memory efficiency with a lower value. The option does not take effect until the instance is restarted.
- **CHUNK_SIZE_GROWTH_FACTOR** – a float that controls the size of new chunks. The size of a new chunk is the size of the previous chunk times CHUNK_SIZE_GROWTH_FACTOR. The allowed values are 1 to 2; the default is 1.25. The option does not take effect until the instance is restarted.
- **ERROR_ON_MEMORY_EXHAUSTED** – a Boolean that when set to 1 (true) specifies that memcached will return an error rather than evicting items when there is no more memory to store items. If set to 0 (false), memcached will evict items if there is no more memory. The default is 0 (false). The option does not take effect until the instance is restarted.
- **MAX_SIMULTANEOUS_CONNECTIONS** – an integer that specifies the maximum number of concurrent connections. Setting this value to anything under 10 prevents MySQL from starting. The allowed values are 10 to 1024; the default is 1024. The option does not take effect until the instance is restarted.
- **VERBOSITY** – a string that specifies the level of information logged in the MySQL error log by the memcached service. The default is v. The option does not take effect until the instance is restarted. The allowed values are:

- **v** – Logs errors and warnings while running the main event loop.
- **vv** – In addition to the information logged by v, also logs each client command and the response.
- **vvv** – In addition to the information logged by vv, also logs internal state transitions.

Amazon RDS configures these MySQL DAEMON_MEMCACHED_OPTIONS parameters, they cannot be modified: DAEMON_PROCESS, LARGE_MEMORY_PAGES, MAXIMUM_CORE_FILE_LIMIT, MAX_ITEM_SIZE, LOCK_DOWN_PAGE_MEMORY, MASK, IDFILE, REQUESTS_PER_EVENT, SOCKET, and USER.

Common DBA tasks for MySQL DB instances

This section describes the Amazon RDS-specific implementations of some common DBA tasks for DB instances running the MySQL database engine. In order to deliver a managed service experience, Amazon RDS does not provide shell access to DB instances, and it restricts access to certain system procedures and tables that require advanced privileges.

For information about working with MySQL log files on Amazon RDS, see [Accessing MySQL database log files \(p. 519\)](#).

Topics

- [Ending a session or query \(p. 933\)](#)
- [Skipping the current replication error \(p. 933\)](#)
- [Working with InnoDB tablespaces to improve crash recovery times \(p. 934\)](#)
- [Managing the global status history \(p. 936\)](#)

Ending a session or query

You can end user sessions or queries on DB instances by using the `rds_kill` and `rds_kill_query` commands. First connect to your MySQL DB instance, then issue the appropriate command as shown following. For more information, see [Connecting to a DB instance running the MySQL database engine \(p. 840\)](#).

```
CALL mysql.rds_kill(thread-ID)
CALL mysql.rds_kill_query(thread-ID)
```

For example, to end the session that is running on thread 99, you would type the following:

```
CALL mysql.rds_kill(99);
```

To end the query that is running on thread 99, you would type the following:

```
CALL mysql.rds_kill_query(99);
```

Skipping the current replication error

Amazon RDS provides a mechanism for you to skip an error on your read replicas if the error is causing your read replica to stop responding and the error doesn't affect the integrity of your data. First connect to your MySQL DB instance, then issue the appropriate commands as shown following. For more information, see [Connecting to a DB instance running the MySQL database engine \(p. 840\)](#).

Note

You should first verify that the error can be safely skipped. In a MySQL utility, connect to the read replica and run the following MySQL command:

```
SHOW SLAVE STATUS\G
```

For information about the values returned, see [the MySQL documentation](#).

To skip the error, you can issue the following command:

```
CALL mysql.rds_skip_repl_error;
```

This command has no effect if you run it on the source DB instance, or on a read replica that has not encountered a replication error.

For more information, such as the versions of MySQL that support `mysql.rds_skip_repl_error`, see [mysql.rds_skip_repl_error \(p. 968\)](#).

Important

If you attempt to call `mysql.rds_skip_repl_error` and encounter the following error: `ERROR 1305 (42000): PROCEDURE mysql.rds_skip_repl_error does not exist`, then upgrade your MySQL DB instance to the latest minor version or one of the minimum minor versions listed in [mysql.rds_skip_repl_error \(p. 968\)](#).

Working with InnoDB tablespaces to improve crash recovery times

Every table in MySQL consists of a table definition, data, and indexes. The MySQL storage engine InnoDB stores table data and indexes in a *tablespace*. InnoDB creates a global shared tablespace that contains a data dictionary and other relevant metadata, and it can contain table data and indexes. InnoDB can also create separate tablespaces for each table and partition. These separate tablespaces are stored in files with a .ibd extension and the header of each tablespace contains a number that uniquely identifies it.

Amazon RDS provides a parameter in a MySQL parameter group called `innodb_file_per_table`. This parameter controls whether InnoDB adds new table data and indexes to the shared tablespace (by setting the parameter value to 0) or to individual tablespaces (by setting the parameter value to 1). Amazon RDS sets the default value for `innodb_file_per_table` parameter to 1, which allows you to drop individual InnoDB tables and reclaim storage used by those tables for the DB instance. In most use cases, setting the `innodb_file_per_table` parameter to 1 is the recommended setting.

You should set the `innodb_file_per_table` parameter to 0 when you have a large number of tables, such as over 1000 tables when you use standard (magnetic) or general purpose SSD storage or over 10,000 tables when you use Provisioned IOPS storage. When you set this parameter to 0, individual tablespaces are not created and this can improve the time it takes for database crash recovery.

MySQL processes each metadata file, which includes tablespaces, during the crash recovery cycle. The time it takes MySQL to process the metadata information in the shared tablespace is negligible compared to the time it takes to process thousands of tablespace files when there are multiple tablespaces. Because the tablespace number is stored within the header of each file, the aggregate time to read all the tablespace files can take up to several hours. For example, a million InnoDB tablespaces on standard storage can take from five to eight hours to process during a crash recovery cycle. In some cases, InnoDB can determine that it needs additional cleanup after a crash recovery cycle so it will begin another crash recovery cycle, which will extend the recovery time. Keep in mind that a crash recovery cycle also entails rolling-back transactions, fixing broken pages, and other operations in addition to the processing of tablespace information.

Since the `innodb_file_per_table` parameter resides in a parameter group, you can change the parameter value by editing the parameter group used by your DB instance without having to reboot the DB instance. After the setting is changed, for example, from 1 (create individual tables) to 0 (use shared tablespace), new InnoDB tables will be added to the shared tablespace while existing tables continue to have individual tablespaces. To move an InnoDB table to the shared tablespace, you must use the `ALTER TABLE` command.

Migrating multiple tablespaces to the shared tablespace

You can move an InnoDB table's metadata from its own tablespace to the shared tablespace, which will rebuild the table metadata according to the `innodb_file_per_table` parameter setting. First connect to your MySQL DB instance, then issue the appropriate commands as shown following. For more information, see [Connecting to a DB instance running the MySQL database engine \(p. 840\)](#).

```
ALTER TABLE table_name ENGINE = InnoDB, ALGORITHM=COPY;
```

For example, the following query returns an `ALTER TABLE` statement for every InnoDB table that is not in the shared tablespace.

For MySQL 5.6 and 5.7 DB instances:

```
SELECT CONCAT('ALTER TABLE `',
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '``', '```'), '``.`',
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '``', '```'), '``' ENGINE=InnoDB,
ALGORITHM=COPY;') AS Query
FROM INFORMATION_SCHEMA.INNODB_SYS_TABLES
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

For MySQL 8.0 DB instances:

```
SELECT CONCAT('ALTER TABLE `',
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '``', '```'), '``.`',
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '``', '```'), '``' ENGINE=InnoDB,
ALGORITHM=COPY;') AS Query
FROM INFORMATION_SCHEMA.INNODB_TABLES
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

Rebuilding a MySQL table to move the table's metadata to the shared tablespace requires additional storage space temporarily to rebuild the table, so the DB instance must have storage space available. During rebuilding, the table is locked and inaccessible to queries. For small tables or tables not frequently accessed, this might not be an issue. For large tables or tables frequently accessed in a heavily concurrent environment, you can rebuild tables on a read replica.

You can create a read replica and migrate table metadata to the shared tablespace on the read replica. While the `ALTER TABLE` statement blocks access on the read replica, the source DB instance is not affected. The source DB instance will continue to generate its binary logs while the read replica lags during the table rebuilding process. Because the rebuilding requires additional storage space and the replay log file can become large, you should create a read replica with storage allocated that is larger than the source DB instance.

To create a read replica and rebuild InnoDB tables to use the shared tablespace, take the following steps:

1. Make sure that backup retention is enabled on the source DB instance so that binary logging is enabled.
2. Use the AWS Management Console or AWS CLI to create a read replica for the source DB instance. Because the creation of a read replica involves many of the same processes as crash recovery, the creation process can take some time if there is a large number of InnoDB tablespaces. Allocate more storage space on the read replica than is currently used on the source DB instance.
3. When the read replica has been created, create a parameter group with the parameter settings `read_only = 0` and `innodb_file_per_table = 0`. Then associate the parameter group with the read replica.
4. Issue the following SQL statement for all tables that you want migrated on the replica:

```
ALTER TABLE name ENGINE = InnoDB
```

5. When all of your `ALTER TABLE` statements have completed on the read replica, verify that the read replica is connected to the source DB instance and that the two instances are in sync.
6. Use the console or CLI to promote the read replica to be the instance. Make sure that the parameter group used for the new standalone DB instance has the `innodb_file_per_table` parameter set to 0. Change the name of the new standalone DB instance, and point any applications to the new standalone DB instance.

Managing the global status history

MySQL maintains many status variables that provide information about its operation. Their value can help you detect locking or memory issues on a DB instance . The values of these status variables are cumulative since last time the DB instance was started. You can reset most status variables to 0 by using the `FLUSH STATUS` command.

To allow for monitoring of these values over time, Amazon RDS provides a set of procedures that will snapshot the values of these status variables over time and write them to a table, along with any changes since the last snapshot. This infrastructure, called Global Status History (GoSH), is installed on all MySQL DB instances starting with versions 5.5.23. GoSH is disabled by default.

To enable GoSH, you first enable the event scheduler from a DB parameter group by setting the parameter `event_scheduler` to ON. For information about creating and modifying a DB parameter group, see [Working with DB parameter groups \(p. 228\)](#).

You can then use the procedures in the following table to enable and configure GoSH. First connect to your MySQL DB instance, then issue the appropriate commands as shown following. For more information, see [Connecting to a DB instance running the MySQL database engine \(p. 840\)](#). For each procedure, type the following:

```
CALL procedure-name;
```

Where *procedure-name* is one of the procedures in the table.

Procedure	Description
<code>mysql.rds_enable_gsh_collector</code>	Enables GoSH to take default snapshots at intervals specified by <code>rds_set_gsh_collector</code> .
<code>mysql.rds_set_gsh_collector</code>	Specifies the interval, in minutes, between snapshots. Default value is 5.
<code>mysql.rds_disable_gsh_collector</code>	Disables snapshots.
<code>mysql.rds_collect_global_status</code>	Takes a snapshot on demand.
<code>mysql.rds_enable_gsh_rotation</code>	Enables rotation of the contents of the <code>mysql.rds_global_status_history</code> table to <code>mysql.rds_global_status_history_old</code> at intervals specified by <code>rds_set_gsh_rotation</code> .
<code>mysql.rds_set_gsh_rotation</code>	Specifies the interval, in days, between table rotations. Default value is 7.
<code>mysql.rds_disable_gsh_rotation</code>	Disables table rotation.
<code>mysql.rds_rotate_global_status_history</code>	Rotates the contents of the <code>mysql.rds_global_status_history</code> table to <code>mysql.rds_global_status_history_old</code> on demand.

When GoSH is running, you can query the tables that it writes to. For example, to query the hit ratio of the Innodb buffer pool, you would issue the following query:

```
select a.collection_end, a.collection_start, ((a.variable_Delta-b.variable_delta)/a.variable_delta)*100 as "HitRatio"
```

```
from mysql.rds_global_status_history as a join mysql.rds_global_status_history as b on
a.collection_end = b.collection_end
where a.variable_name = 'Innodb_buffer_pool_read_requests' and b.variable_name =
'Innodb_buffer_pool_reads'
```

Using Kerberos authentication for MySQL

You can use Kerberos authentication to authenticate users when they connect to your MySQL DB instance. The DB instance works with AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) to enable Kerberos authentication. When users authenticate with a MySQL DB instance joined to the trusting domain, authentication requests are forwarded. Forwarded requests go to the domain directory that you create with AWS Directory Service.

Keeping all of your credentials in the same directory can save you time and effort. With this approach, you have a centralized place for storing and managing credentials for multiple DB instances. Using a directory can also improve your overall security profile.

Amazon RDS supports Kerberos authentication for MySQL DB instances in the following AWS Regions:

- US East (Ohio)
- US East (N. Virginia)
- US West (N. California)
- US West (Oregon)
- Asia Pacific (Mumbai)
- Asia Pacific (Seoul)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Canada (Central)
- Europe (Frankfurt)
- Europe (Ireland)
- Europe (London)
- Europe (Stockholm)
- South America (São Paulo)
- China (Beijing)
- China (Ningxia)

To set up Kerberos authentication for a MySQL DB instance, complete the following general steps, described in more detail later:

1. Use AWS Managed Microsoft AD to create an AWS Managed Microsoft AD directory. You can use the AWS Management Console, the AWS CLI, or the AWS Directory Service to create the directory. For details about doing so, see [Create your AWS Managed Microsoft AD directory](#) in the *AWS Directory Service Administration Guide*.
2. Create an AWS Identity and Access Management (IAM) role that uses the managed IAM policy `AmazonRDSDirectoryServiceAccess`. The role allows Amazon RDS to make calls to your directory.

For the role to allow access, the AWS Security Token Service (AWS STS) endpoint must be activated in the AWS Region for your AWS account. AWS STS endpoints are active by default in all AWS Regions, and you can use them without any further actions. For more information, see [Activating and deactivating AWS STS in an AWS Region](#) in the *IAM User Guide*.

3. Create and configure users in the AWS Managed Microsoft AD directory using the Microsoft Active Directory tools. For more information about creating users in your Active Directory, see [Manage users and groups in AWS managed Microsoft AD](#) in the *AWS Directory Service Administration Guide*.
4. Create or modify a MySQL DB instance. If you use either the CLI or RDS API in the create request, specify a domain identifier with the `Domain` parameter. Use the `d-*` identifier that was generated when you created your directory and the name of the role that you created.

If you modify an existing MySQL DB instance to use Kerberos authentication, set the domain and IAM role parameters for the DB instance. Locate the DB instance in the same VPC as the domain directory.

5. Use the Amazon RDS master user credentials to connect to the MySQL DB instance. Create the user in MySQL using the `CREATE USER` clause `IDENTIFIED WITH 'auth_pam'`. Users that you create this way can log in to the MySQL DB instance using Kerberos authentication.

Setting up Kerberos authentication for MySQL DB instances

You use AWS Managed Microsoft AD to set up Kerberos authentication for a MySQL DB instance. To set up Kerberos authentication, you take the following steps.

Step 1: Create a directory using AWS Managed Microsoft AD

AWS Directory Service creates a fully managed Active Directory in the AWS Cloud. When you create an AWS Managed Microsoft AD directory, AWS Directory Service creates two domain controllers and Domain Name System (DNS) servers on your behalf. The directory servers are created in different subnets in a VPC. This redundancy helps make sure that your directory remains accessible even if a failure occurs.

When you create an AWS Managed Microsoft AD directory, AWS Directory Service performs the following tasks on your behalf:

- Sets up an Active Directory within the VPC.
- Creates a directory administrator account with the user name Admin and the specified password. You use this account to manage your directory.

Note

Be sure to save this password. AWS Directory Service doesn't store it. You can reset it, but you can't retrieve it.

- Creates a security group for the directory controllers.

When you launch an AWS Managed Microsoft AD, AWS creates an Organizational Unit (OU) that contains all of your directory's objects. This OU has the NetBIOS name that you typed when you created your directory and is located in the domain root. The domain root is owned and managed by AWS.

The Admin account that was created with your AWS Managed Microsoft AD directory has permissions for the most common administrative activities for your OU:

- Create, update, or delete users
- Add resources to your domain such as file or print servers, and then assign permissions for those resources to users in your OU
- Create additional OUs and containers
- Delegate authority
- Restore deleted objects from the Active Directory Recycle Bin
- Run AD and DNS Windows PowerShell modules on the Active Directory Web Service

The Admin account also has rights to perform the following domain-wide activities:

- Manage DNS configurations (add, remove, or update records, zones, and forwarders)
- View DNS event logs
- View security event logs

To create a directory with AWS Managed Microsoft AD

1. Sign in to the AWS Management Console and open the AWS Directory Service console at <https://console.aws.amazon.com/directoryservicev2/>.
2. In the navigation pane, choose **Directories** and choose **Set up Directory**.
3. Choose **AWS Managed Microsoft AD**. AWS Managed Microsoft AD is the only option that you can currently use with Amazon RDS.
4. Enter the following information:

Directory DNS name

The fully qualified name for the directory, such as `corp.example.com`.

Directory NetBIOS name

The short name for the directory, such as `CORP`.

Directory description

(Optional) A description for the directory.

Admin password

The password for the directory administrator. The directory creation process creates an administrator account with the user name Admin and this password.

The directory administrator password and can't include the word "admin." The password is case-sensitive and must be 8–64 characters in length. It must also contain at least one character from three of the following four categories:

- Lowercase letters (a–z)
- Uppercase letters (A–Z)
- Numbers (0–9)
- Non-alphanumeric characters (~!@#\$%^&*_+=`|\{}{};:"";<>,?./)

Confirm password

The administrator password retyped.

5. Choose **Next**.
6. Enter the following information in the **Networking** section and then choose **Next**:

VPC

The VPC for the directory. Create the MySQL DB instance in this same VPC.

Subnets

Subnets for the directory servers. The two subnets must be in different Availability Zones.

7. Review the directory information and make any necessary changes. When the information is correct, choose **Create directory**.

Review & create

Review

Directory type

Microsoft AD

VPC

vpc-8b6b78e9 ([REDACTED])

Directory DNS name

corp.example.com

Subnets

subnet-75128d10 ([REDACTED] , us-east-1a)

subnet-f51665dd ([REDACTED] , us-east-1b)

Directory NetBIOS name

CORP

Directory description

My directory

Pricing

Edition

Standard

Free trial eligible [Learn more](#)

30-day limited trial

~USD [REDACTED] *

* Includes two domain controllers, USD [REDACTED] /mo for each additional domain controller.

[Cancel](#)

[Previous](#)

[Create di...](#)

It takes several minutes for the directory to be created. When it has been successfully created, the **Status** value changes to **Active**.

To see information about your directory, choose the directory name in the directory listing. Note the **Directory ID** value because you need this value when you create or modify your MySQL DB instance.

Directory details		
Directory type	VPC	Status
Microsoft AD	vpc-6594f31c [1]	Active
Edition	Subnets	Last updated
Standard	subnet-7d36a227 [1] subnet-a2ab49c6 [1]	Tuesday, January 7, 2020
Directory ID d-90670a8d36	Availability zones	Launch time
Directory DNS name	us-east-1c, us-east-1d	Tuesday, January 7, 2020
corp.example.com	DNS address	
Directory NetBIOS name	[REDACTED]	
CORP		
Description - Edit		
My directory		

[Application management](#) | [Scale & share](#) | [Networking & security](#) | [Maintenance](#)

Step 2: Create the IAM role for use by Amazon RDS

For Amazon RDS to call AWS Directory Service for you, an IAM role that uses the managed IAM policy `AmazonRDSDirectoryServiceAccess` is required. This role allows Amazon RDS to make calls to the AWS Directory Service.

When a DB instance is created using the AWS Management Console and the console user has the `iam:CreateRole` permission, the console creates this role automatically. In this case, the role name is `rds-directoryservice-kerberos-access-role`. Otherwise, you must create the IAM role manually. When you create this IAM role, choose `Directory Service`, and attach the AWS managed policy `AmazonRDSDirectoryServiceAccess` to it.

For more information about creating IAM roles for a service, see [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.

Note

The IAM role used for Windows Authentication for RDS for Microsoft SQL Server can't be used for RDS for MySQL.

Optionally, you can create policies with the required permissions instead of using the managed IAM policy `AmazonRDSDirectoryServiceAccess`. In this case, the IAM role must have the following IAM trust policy.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "directoryservice.rds.amazonaws.com",
                    "rds.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

The role must also have the following IAM role policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ds:DescribeDirectories",
                "ds:AuthorizeApplication",
                "ds:UnauthorizeApplication",
                "ds:GetAuthorizedApplicationDetails"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Step 3: Create and configure users

You can create users with the Active Directory Users and Computers tool. This tool is part of the Active Directory Domain Services and Active Directory Lightweight Directory Services tools. Users represent individual people or entities that have access to your directory.

To create users in an AWS Directory Service directory, you must be connected to an Amazon EC2 instance based on Microsoft Windows. This instance must be a member of the AWS Directory Service directory and be logged in as a user that has privileges to create users. For more information, see [Manage users and groups in AWS Managed Microsoft AD](#) in the *AWS Directory Service Administration Guide*.

Step 4: Create or modify a MySQL DB instance

Create or modify a MySQL DB instance for use with your directory. You can use the console, CLI, or RDS API to associate a DB instance with a directory. You can do this in one of the following ways:

- Create a new MySQL DB instance using the console, the [create-db-instance](#) CLI command, or the [CreateDBInstance](#) RDS API operation.

For instructions, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

- Modify an existing MySQL DB instance using the console, the [modify-db-instance](#) CLI command, or the [ModifyDBInstance](#) RDS API operation.

For instructions, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

- Restore a MySQL DB instance from a DB snapshot using the console, the [restore-db-instance-from-db-snapshot](#) CLI command, or the [RestoreDBInstanceFromDBSnapshot](#) RDS API operation.

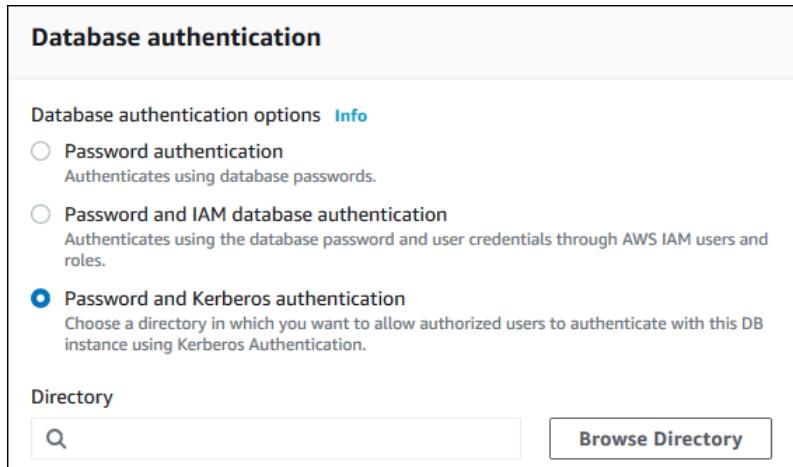
For instructions, see [Restoring from a DB snapshot \(p. 349\)](#).

- Restore a MySQL DB instance to a point-in-time using the console, the [restore-db-instance-to-point-in-time](#) CLI command, or the [RestoreDBInstanceToPointInTime](#) RDS API operation.

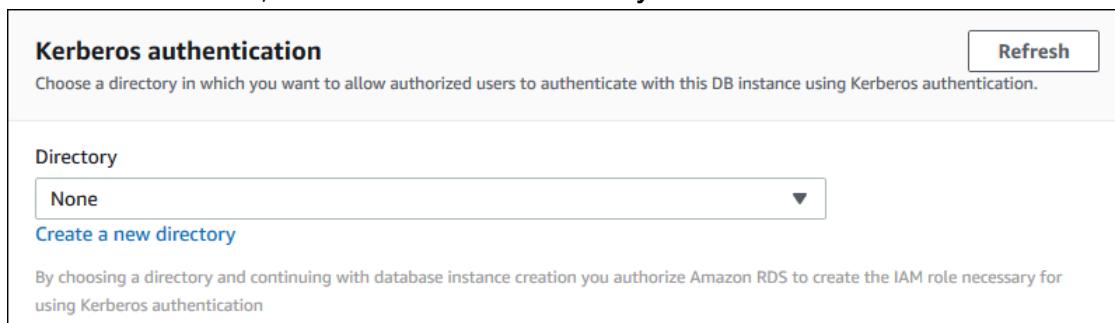
For instructions, see [Restoring a DB instance to a specified time \(p. 389\)](#).

Kerberos authentication is only supported for MySQL DB instances in a VPC. The DB instance can be in the same VPC as the directory, or in a different VPC. The DB instance must use a security group that allows egress within the directory's VPC so the DB instance can communicate with the directory.

When you use the console to create a DB instance, choose **Password and Kerberos authentication** in the **Database authentication** section. Choose **Browse Directory** and then select the directory, or choose **Create a new directory**.



When you use the console to modify or restore a DB instance, choose the directory in the **Kerberos authentication** section, or choose **Create a new directory**.



Use the CLI or RDS API to associate a DB instance with a directory. The following parameters are required for the DB instance to be able to use the domain directory you created:

- For the `--domain` parameter, use the domain identifier ("d-*" identifier) generated when you created the directory.
- For the `--domain-iam-role-name` parameter, use the role you created that uses the managed IAM policy `AmazonRDSdirectoryServiceAccess`.

For example, the following CLI command modifies a DB instance to use a directory.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--domain d-ID \
--domain-iam-role-name role-name
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--domain d-ID ^
--domain-iam-role-name role-name
```

Important

If you modify a DB instance to enable Kerberos authentication, reboot the DB instance after making the change.

Step 5: Create Kerberos authentication MySQL logins

Use the Amazon RDS master user credentials to connect to the MySQL DB instance as you do any other DB instance. The DB instance is joined to the AWS Managed Microsoft AD domain. Thus, you can provision MySQL logins and users from the Active Directory users in your domain. Database permissions are managed through standard MySQL permissions that are granted to and revoked from these logins.

You can allow an Active Directory user to authenticate with MySQL. To do this, first use the Amazon RDS master user credentials to connect to the MySQL DB instance as with any other DB instance. After you're logged in, create an externally authenticated user with PAM (Pluggable Authentication Modules) in MySQL as shown following.

```
CREATE USER 'testuser'@'%' IDENTIFIED WITH 'auth_pam';
UPDATE mysql.user SET ssl_type = 'any' WHERE ssl_type = '' AND PLUGIN = 'auth_pam' and USER = 'testuser';
FLUSH PRIVILEGES;
```

Replace *testuser* with the user name. Users (both humans and applications) from your domain can now connect to the DB instance from a domain joined client machine using Kerberos authentication.

Important

We strongly recommended that clients use SSL/TLS connections when using PAM authentication. If they don't use SSL/TLS connections, the password might be sent as clear text in some cases.

Managing a DB instance in a domain

You can use the CLI or the RDS API to manage your DB instance and its relationship with your managed Active Directory. For example, you can associate an Active Directory for Kerberos authentication and disassociate an Active Directory to disable Kerberos authentication. You can also move a DB instance to be externally authenticated by one Active Directory to another.

For example, using the Amazon RDS API, you can do the following:

- To reattempt enabling Kerberos authentication for a failed membership, use the `ModifyDBInstance` API operation and specify the current membership's directory ID.
- To update the IAM role name for membership, use the `ModifyDBInstance` API operation and specify the current membership's directory ID and the new IAM role.

- To disable Kerberos authentication on a DB instance, use the `ModifyDBInstance` API operation and specify `none` as the domain parameter.
- To move a DB instance from one domain to another, use the `ModifyDBInstance` API operation and specify the domain identifier of the new domain as the domain parameter.
- To list membership for each DB instance, use the `DescribeDBInstances` API operation.

Understanding domain membership

After you create or modify your DB instance, it becomes a member of the domain. You can view the status of the domain membership for the DB instance by running the `describe-db-instances` CLI command. The status of the DB instance can be one of the following:

- `kerberos-enabled` – The DB instance has Kerberos authentication enabled.
- `enabling-kerberos` – AWS is in the process of enabling Kerberos authentication on this DB instance.
- `pending-enable-kerberos` – The enabling of Kerberos authentication is pending on this DB instance.
- `pending-maintenance-enable-kerberos` – AWS will attempt to enable Kerberos authentication on the DB instance during the next scheduled maintenance window.
- `pending-disable-kerberos` – The disabling of Kerberos authentication is pending on this DB instance.
- `pending-maintenance-disable-kerberos` – AWS will attempt to disable Kerberos authentication on the DB instance during the next scheduled maintenance window.
- `enable-kerberos-failed` – A configuration problem has prevented AWS from enabling Kerberos authentication on the DB instance. Check and fix your configuration before reissuing the DB instance `modify` command.
- `disabling-kerberos` – AWS is in the process of disabling Kerberos authentication on this DB instance.

A request to enable Kerberos authentication can fail because of a network connectivity issue or an incorrect IAM role. For example, suppose that you create a DB instance or modify an existing DB instance and the attempt to enable Kerberos authentication fails. If this happens, re-issue the `modify` command or modify the newly created DB instance to join the domain.

Connecting to MySQL with Kerberos authentication

To connect to MySQL with Kerberos authentication, you must log in using the Kerberos authentication type.

To create a database user that you can connect to using Kerberos authentication, use an `IDENTIFIED WITH` clause on the `CREATE USER` statement. For instructions, see [Step 5: Create Kerberos authentication MySQL logins \(p. 945\)](#).

To avoid errors, use the MariaDB `mysql` client. You can download MariaDB software at <https://downloads.mariadb.org/>.

At a command prompt, connect to one of the endpoints associated with your MySQL DB instance. Follow the general procedures in [Connecting to a DB instance running the MySQL database engine \(p. 840\)](#). When you're prompted for the password, enter the Kerberos password associated with that user name.

Restoring a MySQL DB instance and adding it to a domain

You can restore a DB snapshot or complete a point-in-time restore for a MySQL DB instance and then add it to a domain. After the DB instance is restored, modify the DB instance using the process explained in [Step 4: Create or modify a MySQL DB instance \(p. 943\)](#) to add the DB instance to a domain.

Kerberos authentication MySQL limitations

The following limitations apply to Kerberos authentication for MySQL:

- A Managed Active Directory that has been shared with you isn't supported.
- Kerberos authentication is supported for the following Amazon RDS for MySQL versions:
 - Amazon RDS for MySQL version 8.0.13, 8.0.15, and 8.0.16
 - Amazon RDS for MySQL version 5.7.24, 5.7.25, and 5.7.26
- You must reboot the DB instance after enabling the feature.
- The domain name length can't be longer than 61 characters.
- You can't enable Kerberos authentication and IAM authentication at the same time. Choose one authentication method or the other for your MySQL DB instance.
- Don't modify the DB instance port after enabling the feature.
- Don't use Kerberos authentication with read replicas.
- To delete a DB instance with this feature enabled, first disable the feature. To do this, use the `modify-db-instance` CLI command for the DB instance and specify `none` for the `--domain` parameter.

If you use the CLI or RDS API to delete a DB instance with this feature enabled, expect a delay.

Known issues and limitations for MySQL on Amazon RDS

Known issues and limitations for working with MySQL on Amazon RDS are as follows.

Topics

- [Inconsistent InnoDB buffer pool size \(p. 948\)](#)
- [Index merge optimization returns wrong results \(p. 948\)](#)
- [Log file size \(p. 949\)](#)
- [MySQL parameter exceptions for Amazon RDS DB instances \(p. 949\)](#)
- [MySQL file size limits in Amazon RDS \(p. 950\)](#)
- [MySQL Keyring Plugin not supported \(p. 951\)](#)

Inconsistent InnoDB buffer pool size

For MySQL 5.7, there is currently a bug in the way that the InnoDB buffer pool size is managed. MySQL 5.7 might adjust the value of the `innodb_buffer_pool_size` parameter to a large value that can result in the InnoDB buffer pool growing too large and using up too much memory. This effect can cause the MySQL database engine to stop running or can prevent the MySQL database engine from starting. This issue is more common for DB instance classes that have less memory available.

To resolve this issue, set the value of the `innodb_buffer_pool_size` parameter to a multiple of the product of the `innodb_buffer_pool_instances` parameter value and the `innodb_buffer_pool_chunk_size` parameter value. For example, you might set the `innodb_buffer_pool_size` parameter value to a multiple of eight times the product of the `innodb_buffer_pool_instances` and `innodb_buffer_pool_chunk_size` parameter values, as shown in the following example.

```
innodb_buffer_pool_chunk_size = 536870912
innodb_buffer_pool_instances = 4
innodb_buffer_pool_size = (536870912 * 4) * 8 = 17179869184
```

For details on this MySQL 5.7 bug, go to <https://bugs.mysql.com/bug.php?id=79379> in the MySQL documentation.

Index merge optimization returns wrong results

Queries that use index merge optimization might return wrong results due to a bug in the MySQL query optimizer that was introduced in MySQL 5.5.37. When you issue a query against a table with multiple indexes the optimizer scans ranges of rows based on the multiple indexes, but does not merge the results together correctly. For more information on the query optimizer bug, go to <http://bugs.mysql.com/bug.php?id=72745> and <http://bugs.mysql.com/bug.php?id=68194> in the MySQL bug database.

For example, consider a query on a table with two indexes where the search arguments reference the indexed columns.

```
SELECT * FROM table1
WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

In this case, the search engine will search both indexes. However, due to the bug, the merged results are incorrect.

To resolve this issue, you can do one of the following:

- Set the `optimizer_switch` parameter to `index_merge=off` in the DB parameter group for your MySQL DB instance. For information on setting DB parameter group parameters, see [Working with DB parameter groups \(p. 228\)](#).
- Upgrade your MySQL DB instance to MySQL version 5.6, 5.7, or 8.0. For more information, see [Upgrading a MySQL DB snapshot \(p. 863\)](#).
- If you cannot upgrade your instance or change the `optimizer_switch` parameter, you can work around the bug by explicitly identifying an index for the query, for example:

```
SELECT * FROM table1
USE INDEX covering_index
WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

For more information, go to [Index merge optimization](#).

Log file size

For MySQL, there is a size limit on BLOBS written to the redo log. To account for this limit, ensure that the `innodb_log_file_size` parameter for your MySQL DB instance is 10 times larger than the largest BLOB data size found in your tables, plus the length of other variable length fields (`VARCHAR`, `VARBINARY`, `TEXT`) in the same tables. For information on how to set parameter values, see [Working with DB parameter groups \(p. 228\)](#). For information on the redo log BLOB size limit, go to [Changes in MySQL 5.6.20](#).

MySQL parameter exceptions for Amazon RDS DB instances

Some MySQL parameters require special considerations when used with an Amazon RDS DB instance.

`lower_case_table_names`

Because Amazon RDS uses a case-sensitive file system, setting the value of the `lower_case_table_names` server parameter to 2 ("names stored as given but compared in lowercase") is not supported. The following are the supported values for Amazon RDS for MySQL DB instances:

- 0 ("names stored as given and comparisons are case-sensitive") is supported for all Amazon RDS for MySQL versions.
- 1 ("names stored in lowercase and comparisons are not case-sensitive") is supported for Amazon RDS for MySQL version 5.5, version 5.6, version 5.7, and version 8.0.19 and higher 8.0 versions.

The `lower_case_table_names` parameter should be set as part of a custom DB parameter group before creating a DB instance. You should avoid changing the `lower_case_table_names` parameter for existing database instances because doing so could cause inconsistencies with point-in-time recovery backups and read replica DB instances.

Read replicas should always use the same `lower_case_table_names` parameter value as the source DB instance.

`long_query_time`

You can set the `long_query_time` parameter to a floating point value which allows you to log slow queries to the MySQL slow query log with microsecond resolution. You can set a value such as 0.1

seconds, which would be 100 milliseconds, to help when debugging slow transactions that take less than one second.

MySQL file size limits in Amazon RDS

For MySQL DB instances, the maximum provisioned storage limit constrains the size of a table to a maximum size of 16 TB when using InnoDB file-per-table tablespaces. This limit also constrains the system tablespace to a maximum size of 16 TB. InnoDB file-per-table tablespaces (with tables each in their own tablespace) is set by default for MySQL DB instances.

Note

Some existing DB instances have a lower limit. For example, MySQL DB instances created before April 2014 have a file and table size limit of 2 TB. This 2 TB file size limit also applies to DB instances or read replicas created from DB snapshots taken before April 2014, regardless of when the DB instance was created.

There are advantages and disadvantages to using InnoDB file-per-table tablespaces, depending on your application. To determine the best approach for your application, go to [File-per-table tablespaces](#) in the MySQL documentation.

We don't recommend allowing tables to grow to the maximum file size. In general, a better practice is to partition data into smaller tables, which can improve performance and recovery times.

One option that you can use for breaking a large table up into smaller tables is partitioning. Partitioning distributes portions of your large table into separate files based on rules that you specify. For example, if you store transactions by date, you can create partitioning rules that distribute older transactions into separate files using partitioning. Then periodically, you can archive the historical transaction data that doesn't need to be readily available to your application. For more information, go to [Partitioning](#) in the MySQL documentation.

To determine the file size of a table

- Use the following SQL command to determine if any of your tables are too large and are candidates for partitioning.

```
SELECT TABLE_SCHEMA, TABLE_NAME,
round(((DATA_LENGTH + INDEX_LENGTH) / 1024 / 1024), 2) As "Approximate size (MB)"
FROM information_schema.TABLES
WHERE TABLE_SCHEMA NOT IN ('mysql', 'information_schema', 'performance_schema');
```

To enable InnoDB file-per-table tablespaces

- To enable InnoDB file-per-table tablespaces, set the *innodb_file_per_table* parameter to 1 in the parameter group for the DB instance.

To disable InnoDB file-per-table tablespaces

- To disable InnoDB file-per-table tablespaces, set the *innodb_file_per_table* parameter to 0 in the parameter group for the DB instance.

For information on updating a parameter group, see [Working with DB parameter groups \(p. 228\)](#).

When you have enabled or disabled InnoDB file-per-table tablespaces, you can issue an `ALTER TABLE` command to move a table from the global tablespace to its own tablespace, or from its own tablespace to the global tablespace as shown in the following example:

```
ALTER TABLE table_name ENGINE=InnoDB;
```

MySQL Keyring Plugin not supported

Currently, Amazon RDS for MySQL does not support the MySQL `keyring_aws` Amazon Web Services Keyring Plugin.

MySQL on Amazon RDS SQL reference

This appendix describes system stored procedures that are available for Amazon RDS instances running the MySQL DB engine.

Overview

The following system stored procedures are supported for Amazon RDS DB instances running MySQL.

Replication

- [mysql.rds_set_master_auto_position \(p. 953\)](#)
- [mysql.rds_set_external_master \(p. 954\)](#)
- [mysql.rds_set_external_master_with_delay \(p. 956\)](#)
- [mysql.rds_set_external_master_with_auto_position \(p. 959\)](#)
- [mysql.rds_reset_external_master \(p. 961\)](#)
- [mysql.rds_import_binlog_ssl_material \(p. 962\)](#)
- [mysql.rds_remove_binlog_ssl_material \(p. 963\)](#)
- [mysql.rds_set_source_delay \(p. 964\)](#)
- [mysql.rds_start_replication \(p. 964\)](#)
- [mysql.rds_start_replication_until \(p. 965\)](#)
- [mysql.rds_start_replication_until_gtid \(p. 966\)](#)
- [mysql.rds_stop_replication \(p. 967\)](#)
- [mysql.rds_skip_transaction_with_gtid \(p. 968\)](#)
- [mysql.rds_skip_repl_error \(p. 968\)](#)
- [mysql.rds_next_master_log \(p. 969\)](#)

InnoDB cache warming

- [mysql.rds_innodb_buffer_pool_dump_now \(p. 971\)](#)
- [mysql.rds_innodb_buffer_pool_load_now \(p. 971\)](#)
- [mysql.rds_innodb_buffer_pool_load_abort \(p. 972\)](#)

Managing additional configuration (for example, binlog file retention)

- [mysql.rds_set_configuration \(p. 972\)](#)
- [mysql.rds_show_configuration \(p. 974\)](#)

Ending a session or query

- [mysql.rds_kill \(p. 974\)](#)
- [mysql.rds_kill_query \(p. 975\)](#)

Logging

- [mysql.rds_rotate_general_log \(p. 975\)](#)
- [mysql.rds_rotate_slow_log \(p. 976\)](#)

Managing the global status history

- [mysql.rds_enable_gsh_collector \(p. 976\)](#)
- [mysql.rds_set_gsh_collector \(p. 976\)](#)
- [mysql.rds_disable_gsh_collector \(p. 976\)](#)
- [mysql.rds_collect_global_status_history \(p. 977\)](#)
- [mysql.rds_enable_gsh_rotation \(p. 977\)](#)
- [mysql.rds_set_gsh_rotation \(p. 977\)](#)
- [mysql.rds_disable_gsh_rotation \(p. 977\)](#)
- [mysql.rds_rotate_global_status_history \(p. 978\)](#)

SQL reference conventions

Following, you can find explanations for the conventions that are used to describe the syntax of the system stored procedures and tables described in the SQL reference section.

Character	Description
UPPERCASE	Words in uppercase are keywords.
[]	Square brackets indicate optional arguments.
{ }	Braces indicate that you are required to choose one of the arguments inside the braces.
	Pipes separate arguments that you can choose.
<i>italics</i>	Words in italics indicate placeholders. You must insert the appropriate value in place of the word in italics.
...	An ellipsis indicates that you can repeat the preceding element.
'	Words in single quotes indicate that you must type the quotes.

mysql.rds_set_master_auto_position

Sets the replication mode to be based on either binary log file positions or on global transaction identifiers (GTIDs).

Syntax

```
CALL mysql.rds_set_master_auto_position (
    auto_position_mode
);
```

Parameters

auto_position_mode

A value that indicates whether to use log file position replication or GTID-based replication:

- 0 – Use the replication method based on binary log file position. The default is 0.

- 1 – Use the GTID-based replication method.

Usage notes

The master user must run the `mysql.rds_set_master_auto_position` procedure.

For RDS for MySQL 5.7, this procedure is supported for MySQL 5.7.23 and later MySQL 5.7 versions. This procedure is not supported for RDS for MySQL 5.5, 5.6, or 8.0.

mysql.rds_set_external_master

Configures a MySQL DB instance to be a read replica of an instance of MySQL running external to Amazon RDS.

Important

To run this procedure, `autocommit` must be enabled. To enable it, set the `autocommit` parameter to 1. For information about modifying parameters, see [Modifying parameters in a DB parameter group \(p. 232\)](#).

Note

You can use the [mysql.rds_set_external_master_with_delay \(p. 956\)](#) stored procedure to configure an external source database instance and delayed replication.

Syntax

```
CALL mysql.rds_set_external_master (
    host_name
    , host_port
    , replication_user_name
    , replication_user_password
    , mysql_binary_log_file_name
    , mysql_binary_log_file_location
    , ssl_encryption
);
```

Parameters

host_name

The host name or IP address of the MySQL instance running external to Amazon RDS to become the source database instance.

host_port

The port used by the MySQL instance running external to Amazon RDS to be configured as the source database instance. If your network configuration includes Secure Shell (SSH) port replication that converts the port number, specify the port number that is exposed by SSH.

replication_user_name

The ID of a user with `REPLICATION CLIENT` and `REPLICATION SLAVE` permissions on the MySQL instance running external to Amazon RDS. We recommend that you provide an account that is used solely for replication with the external instance.

replication_user_password

The password of the user ID specified in `replication_user_name`.

mysql_binary_log_file_name

The name of the binary log on the source database instance that contains the replication information.

mysql_binary_log_file_location

The location in the `mysql_binary_log_file_name` binary log at which replication starts reading the replication information.

ssl_encryption

A value that specifies whether Secure Socket Layer (SSL) encryption is used on the replication connection. 1 specifies to use SSL encryption, 0 specifies to not use encryption. The default is 0.

Note

The `MASTER_SSL_VERIFY_SERVER_CERT` option isn't supported. This option is set to 0, which means that the connection is encrypted, but the certificates aren't verified.

Usage notes

The master user must run the `mysql.rds_set_external_master` procedure. This procedure must be run on the MySQL DB instance to be configured as the read replica of a MySQL instance running external to Amazon RDS.

Before you run `mysql.rds_set_external_master`, you must configure the instance of MySQL running external to Amazon RDS to be a source database instance. To connect to the MySQL instance running external to Amazon RDS, you must specify `replication_user_name` and `replication_user_password` values that indicate a replication user that has `REPLICATION CLIENT` and `REPLICATION SLAVE` permissions on the external instance of MySQL.

To configure an external instance of MySQL as a source database instance

1. Using the MySQL client of your choice, connect to the external instance of MySQL and create a user account to be used for replication. The following is an example.

MySQL 5.5, 5.6, and 5.7

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

MySQL 8.0

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY 'password';
```

2. On the external instance of MySQL, grant `REPLICATION CLIENT` and `REPLICATION SLAVE` privileges to your replication user. The following example grants `REPLICATION CLIENT` and `REPLICATION SLAVE` privileges on all databases for the 'repl_user' user for your domain.

MySQL 5.5, 5.6, and 5.7

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

MySQL 8.0

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

To use encrypted replication, configure source database instance to use SSL connections. Also, import the certificate authority certificate, client certificate, and client key into the DB instance or DB cluster using the [mysql.rds_import_binlog_ssl_material \(p. 962\)](#) procedure.

Note

We recommend that you use read replicas to manage replication between two Amazon RDS DB instances when possible. When you do so, we recommend that you use only this and other replication-related stored procedures. These practices enable more complex replication topologies between Amazon RDS DB instances. We offer these stored procedures primarily to enable replication with MySQL instances running external to Amazon RDS. For information about managing replication between Amazon RDS DB instances, see [Working with read replicas \(p. 278\)](#).

After calling `mysql.rds_set_external_master` to configure an Amazon RDS DB instance as a read replica, you can call [mysql.rds_start_replication \(p. 964\)](#) on the read replica to start the replication process. You can call [mysql.rds_reset_external_master \(p. 961\)](#) to remove the read replica configuration.

When `mysql.rds_set_external_master` is called, Amazon RDS records the time, user, and an action of `set master` in the `mysql.rds_history` and `mysql.rds_replication_status` tables.

Examples

When run on a MySQL DB instance, the following example configures the DB instance to be a read replica of an instance of MySQL running external to Amazon RDS.

```
call mysql.rds_set_external_master(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.0777',
  120,
  0);
```

mysql.rds_set_external_master_with_delay

Configures an RDS for MySQL DB instance to be a read replica of an instance of MySQL running external to Amazon RDS and configures delayed replication.

Important

To run this procedure, `autocommit` must be enabled. To enable it, set the `autocommit` parameter to 1. For information about modifying parameters, see [Modifying parameters in a DB parameter group \(p. 232\)](#).

Syntax

```
CALL mysql.rds_set_external_master_with_delay (
  host_name
, host_port
, replication_user_name
, replication_user_password
, mysql_binary_log_file_name
, mysql_binary_log_file_location
, ssl_encryption
, delay
);
```

Parameters

host_name

The host name or IP address of the MySQL instance running external to Amazon RDS that will become the source database instance.

host_port

The port used by the MySQL instance running external to Amazon RDS to be configured as the source database instance. If your network configuration includes SSH port replication that converts the port number, specify the port number that is exposed by SSH.

replication_user_name

The ID of a user with `REPLICATION CLIENT` and `REPLICATION SLAVE` permissions on the MySQL instance running external to Amazon RDS. We recommend that you provide an account that is used solely for replication with the external instance.

replication_user_password

The password of the user ID specified in `replication_user_name`.

mysql_binary_log_file_name

The name of the binary log on the source database instance contains the replication information.

mysql_binary_log_file_location

The location in the `mysql_binary_log_file_name` binary log at which replication will start reading the replication information.

ssl_encryption

A value that specifies whether Secure Socket Layer (SSL) encryption is used on the replication connection. 1 specifies to use SSL encryption, 0 specifies to not use encryption. The default is 0.

Note

The `MASTER_SSL_VERIFY_SERVER_CERT` option isn't supported. This option is set to 0, which means that the connection is encrypted, but the certificates aren't verified.

delay

The minimum number of seconds to delay replication from source database instance.

The limit for this parameter is one day (86400 seconds).

Usage notes

The master user must run the `mysql.rds_set_external_master_with_delay` procedure. This procedure must be run on the MySQL DB instance to be configured as the read replica of a MySQL instance running external to Amazon RDS.

Before you run `mysql.rds_set_external_master_with_delay`, you must configure the instance of MySQL running external to Amazon RDS to be a source database instance. To connect to the MySQL instance running external to Amazon RDS, you must specify values for `replication_user_name` and `replication_user_password`. These values must indicate a replication user that has `REPLICATION CLIENT` and `REPLICATION SLAVE` permissions on the external instance of MySQL.

To configure an external instance of MySQL as a source database instance

1. Using the MySQL client of your choice, connect to the external instance of MySQL and create a user account to be used for replication. The following is an example.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
```

2. On the external instance of MySQL, grant REPLICATION CLIENT and REPLICATION SLAVE privileges to your replication user. The following example grants REPLICATION CLIENT and REPLICATION SLAVE privileges on all databases for the 'repl_user' user for your domain.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'  
IDENTIFIED BY 'SomePassWord'
```

For more information, see [Replication with a MySQL or MariaDB instance running external to Amazon RDS \(p. 914\)](#).

Note

We recommend that you use read replicas to manage replication between two Amazon RDS DB instances when possible. When you do so, we recommend that you use only this and other replication-related stored procedures. These practices enable more complex replication topologies between Amazon RDS DB instances. We offer these stored procedures primarily to enable replication with MySQL instances running external to Amazon RDS. For information about managing replication between Amazon RDS DB instances, see [Working with read replicas \(p. 278\)](#).

After calling `mysql.rds_set_external_master_with_delay` to configure an Amazon RDS DB instance as a read replica, you can call [mysql.rds_start_replication \(p. 964\)](#) on the read replica to start the replication process. You can call [mysql.rds_reset_external_master \(p. 961\)](#) to remove the read replica configuration.

When you call `mysql.rds_set_external_master_with_delay`, Amazon RDS records the time, the user, and an action of `set master` in the `mysql.rds_history` and `mysql.rds_replication_status` tables.

For disaster recovery, you can use this procedure with the [mysql.rds_start_replication_until \(p. 965\)](#) or [mysql.rds_start_replication_until_gtid \(p. 966\)](#) stored procedure. To roll forward changes to a delayed read replica to the time just before a disaster, you can run the `mysql.rds_set_external_master_with_delay` procedure. After the `mysql.rds_start_replication_until` procedure stops replication, you can promote the read replica to be the new primary DB instance by using the instructions in [Promoting a read replica to be a standalone DB instance \(p. 285\)](#).

To use the `mysql.rds_start_replication_until_gtid` procedure, GTID-based replication must be enabled. To skip a specific GTID-based transaction that is known to cause disaster, you can use the [mysql.rds_skip_transaction_with_gtid \(p. 968\)](#) stored procedure. For more information about working with GTID-based replication, see [Using GTID-based replication for RDS for MySQL \(p. 910\)](#).

The `mysql.rds_set_external_master_with_delay` procedure is available in these versions of RDS for MySQL:

- MySQL 5.6.40 and later 5.6 versions
- MySQL 5.7.22 and later 5.7 versions

Examples

When run on a MySQL DB instance, the following example configures the DB instance to be a read replica of an instance of MySQL running external to Amazon RDS. It sets the minimum replication delay to one hour (3,600 seconds) on the MySQL DB instance. A change from the MySQL source database instance running external to Amazon RDS is not applied on the MySQL DB instance read replica for at least one hour.

```
call mysql.rds_set_external_master_with_delay(
    'Externaldb.some.com',
    3306,
    'repl_user',
    'SomePassWOrd',
    'mysql-bin-changelog.000777',
    120,
    0,
    3600);
```

mysql.rds_set_external_master_with_auto_position

Configures an RDS for MySQL DB instance to be a read replica of an instance of MySQL running external to Amazon RDS. This procedure also configures delayed replication and replication based on global transaction identifiers (GTIDs).

Important

To run this procedure, `autocommit` must be enabled. To enable it, set the `autocommit` parameter to 1. For information about modifying parameters, see [Modifying parameters in a DB parameter group \(p. 232\)](#).

Syntax

```
CALL mysql.rds_set_external_master_with_auto_position (
    host_name
    , host_port
    , replication_user_name
    , replication_user_password
    , ssl_encryption
    , delay
);
```

Parameters

host_name

The host name or IP address of the MySQL instance running external to Amazon RDS to become the source database instance.

host_port

The port used by the MySQL instance running external to Amazon RDS to be configured as the source database instance. If your network configuration includes Secure Shell (SSH) port replication that converts the port number, specify the port number that is exposed by SSH.

replication_user_name

The ID of a user with `REPLICATION CLIENT` and `REPLICATION SLAVE` permissions on the MySQL instance running external to Amazon RDS. We recommend that you provide an account that is used solely for replication with the external instance.

replication_user_password

The password of the user ID specified in `replication_user_name`.

ssl_encryption

A value that specifies whether Secure Socket Layer (SSL) encryption is used on the replication connection. 1 specifies to use SSL encryption, 0 specifies to not use encryption. The default is 0.

Note

The `MASTER_SSL_VERIFY_SERVER_CERT` option isn't supported. This option is set to 0, which means that the connection is encrypted, but the certificates aren't verified.

delay

The minimum number of seconds to delay replication from source database instance.

The limit for this parameter is one day (86,400 seconds).

Usage notes

The master user must run the `mysql.rds_set_external_master_with_auto_position` procedure. This procedure must be run on the MySQL DB instance to be configured as the read replica of a MySQL instance running external to Amazon RDS.

For RDS for MySQL 5.7, this procedure is supported for MySQL 5.7.23 and later MySQL 5.7 versions. This procedure is not supported for RDS for MySQL 5.5, 5.6, or 8.0.

Before you run `mysql.rds_set_external_master_with_auto_position`, you must configure the instance of MySQL running external to Amazon RDS to be a source database instance. To connect to the MySQL instance running external to Amazon RDS, you must specify values for `replication_user_name` and `replication_user_password`. These values must indicate a replication user that has `REPLICATION CLIENT` and `REPLICATION SLAVE` permissions on the external instance of MySQL.

To configure an external instance of MySQL as a source database instance

1. Using the MySQL client of your choice, connect to the external instance of MySQL and create a user account to be used for replication. The following is an example.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
```

2. On the external instance of MySQL, grant `REPLICATION CLIENT` and `REPLICATION SLAVE` privileges to your replication user. The following example grants `REPLICATION CLIENT` and `REPLICATION SLAVE` privileges on all databases for the '`repl_user`' user for your domain.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'  
IDENTIFIED BY 'SomePassWord'
```

For more information, see [Replication with a MySQL or MariaDB instance running external to Amazon RDS \(p. 914\)](#).

Note

We recommend that you use read replicas to manage replication between two Amazon RDS DB instances when possible. When you do so, we recommend that you use only this and other replication-related stored procedures. These practices enable more complex replication topologies between Amazon RDS DB instances. We offer these stored procedures primarily to enable replication with MySQL instances running external to Amazon RDS. For information about managing replication between Amazon RDS DB instances, see [Working with read replicas \(p. 278\)](#).

After calling `mysql.rds_set_external_master_with_auto_position` to configure an Amazon RDS DB instance as a read replica, you can call [mysql.rds_start_replication \(p. 964\)](#) on the read replica to start the replication process. You can call [mysql.rds_reset_external_master \(p. 961\)](#) to remove the read replica configuration.

When you call `mysql.rds_set_external_master_with_auto_position`, Amazon RDS records the time, the user, and an action of `set master` in the `mysql.rds_history` and `mysql.rds_replication_status` tables.

For disaster recovery, you can use this procedure with the [mysql.rds_start_replication_until \(p. 965\)](#) or [mysql.rds_start_replication_until_gtid \(p. 966\)](#) stored procedure. To roll forward changes to a delayed read replica to the time just before a disaster, you can run the `mysql.rds_set_external_master_with_auto_position` procedure. After the `mysql.rds_start_replication_until_gtid` procedure stops replication, you can promote the read replica to be the new primary DB instance by using the instructions in [Promoting a read replica to be a standalone DB instance \(p. 285\)](#).

To use the `mysql.rds_start_replication_until_gtid` procedure, GTID-based replication must be enabled. To skip a specific GTID-based transaction that is known to cause disaster, you can use the [mysql.rds_skip_transaction_with_gtid \(p. 968\)](#) stored procedure. For more information about working with GTID-based replication, see [Using GTID-based replication for RDS for MySQL \(p. 910\)](#).

Examples

When run on a MySQL DB instance, the following example configures the DB instance to be a read replica of an instance of MySQL running external to Amazon RDS. It sets the minimum replication delay to one hour (3,600 seconds) on the MySQL DB instance. A change from the MySQL source database instance running external to Amazon RDS is not applied on the MySQL DB instance read replica for at least one hour.

```
call mysql.rds_set_external_master_with_auto_position(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
  0,
  3600);
```

mysql.rds_reset_external_master

Reconfigures a MySQL DB instance to no longer be a read replica of an instance of MySQL running external to Amazon RDS.

Important

To run this procedure, `autocommit` must be enabled. To enable it, set the `autocommit` parameter to 1. For information about modifying parameters, see [Modifying parameters in a DB parameter group \(p. 232\)](#).

Syntax

```
CALL mysql.rds_reset_external_master;
```

Usage notes

The master user must run the `mysql.rds_reset_external_master` procedure. This procedure must be run on the MySQL DB instance to be removed as a read replica of a MySQL instance running external to Amazon RDS.

Note

We recommend that you use read replicas to manage replication between two Amazon RDS DB instances when possible. When you do so, we recommend that you use only this and other replication-related stored procedures. These practices enable more complex replication

topologies between Amazon RDS DB instances. We offer these stored procedures primarily to enable replication with MySQL instances running external to Amazon RDS. For information about managing replication between Amazon RDS DB instances, see [Working with read replicas \(p. 278\)](#).

For more information about using replication to import data from an instance of MySQL running external to Amazon RDS, see [Restoring a backup into a MySQL DB instance \(p. 871\)](#).

mysql.rds_import_binlog_ssl_material

Imports the certificate authority certificate, client certificate, and client key into an Aurora MySQL DB cluster. The information is required for SSL communication and encrypted replication.

Note

Currently, this procedure is only supported for Aurora MySQL version 5.6.

Syntax

```
CALL mysql.rds_import_binlog_ssl_material (
    ssl_material
);
```

Parameters

ssl_material

JSON payload that contains the contents of the following .pem format files for a MySQL client:

- "ssl_ca": "*Certificate authority certificate*"
- "ssl_cert": "*Client certificate*"
- "ssl_key": "*Client key*"

Usage notes

Prepare for encrypted replication before you run this procedure:

- If you don't have SSL enabled on the external MySQL source database instance and don't have a client key and client certificate prepared, enable SSL on the MySQL database server and generate the required client key and client certificate.
- If SSL is enabled on the external source database instance, supply a client key and certificate for the Aurora MySQL DB cluster. If you don't have these, generate a new key and certificate for the Aurora MySQL DB cluster. To sign the client certificate, you must have the certificate authority key you used to configure SSL on the external MySQL source database instance.

For more information, see [Creating SSL certificates and keys using openssl](#) in the MySQL documentation.

Important

After you prepare for encrypted replication, use an SSL connection to run this procedure. The client key must not be transferred across an insecure connection.

This procedure imports SSL information from an external MySQL database into an Aurora MySQL DB cluster. The SSL information is in .pem format files that contain the SSL information for the Aurora MySQL DB cluster. During encrypted replication, the Aurora MySQL DB cluster acts a client to the MySQL database server. The certificates and keys for the Aurora MySQL client are in files in .pem format.

You can copy the information from these files into the `ssl_material` parameter in the correct JSON payload. To support encrypted replication, import this SSL information into the Aurora MySQL DB cluster.

The JSON payload must be in the following format.

```
'{"ssl_ca":"-----BEGIN CERTIFICATE-----  
ssl_ca_pem_body_code  
-----END CERTIFICATE-----\n","ssl_cert":"-----BEGIN CERTIFICATE-----  
ssl_cert_pem_body_code  
-----END CERTIFICATE-----\n","ssl_key":"-----BEGIN RSA PRIVATE KEY-----  
ssl_key_pem_body_code  
-----END RSA PRIVATE KEY-----\n"}'
```

Examples

The following example imports SSL information into an Aurora MySQL DB cluster. In .pem format files, the body code typically is longer than the body code shown in the example.

```
call mysql.rds_import_binlog_ssl_material(  
'{"ssl_ca":"-----BEGIN CERTIFICATE-----  
AAAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V  
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr  
lsLnB1Itntckij7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+OFzz  
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb  
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE  
-----END CERTIFICATE-----\n","ssl_cert":"-----BEGIN CERTIFICATE-----  
AAAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V  
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr  
lsLnB1Itntckij7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+OFzz  
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb  
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE  
-----END CERTIFICATE-----\n","ssl_key":"-----BEGIN RSA PRIVATE KEY-----  
AAAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V  
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr  
lsLnB1Itntckij7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+OFzz  
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb  
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE  
-----END RSA PRIVATE KEY-----\n"}');
```

Note

For information about using Amazon Aurora, see the [Amazon Aurora User Guide](#).

mysql.rds_remove_binlog_ssl_material

Removes the certificate authority certificate, client certificate, and client key for SSL communication and encrypted replication. This information is imported by using [mysql.rds_import_binlog_ssl_material \(p. 962\)](#).

Note

Currently, this procedure is only supported for Aurora MySQL version 5.6.

Syntax

```
CALL mysql.rds_remove_binlog_ssl_material;
```

mysql.rds_set_source_delay

Sets the minimum number of seconds to delay replication from source database instance to the current read replica. Use this procedure when you are connected to a read replica to delay replication from its source database instance.

Syntax

```
CALL mysql.rds_set_source_delay(  
    delay  
) ;
```

Parameters

delay

The minimum number of seconds to delay replication from the source database instance.

The limit for this parameter is one day (86400 seconds).

Usage notes

The master user must run the `mysql.rds_set_source_delay` procedure.

For disaster recovery, you can use this procedure with the [mysql.rds_start_replication_until \(p. 965\)](#) stored procedure or the [mysql.rds_start_replication_until_gtid \(p. 966\)](#) stored procedure. To roll forward changes to a delayed read replica to the time just before a disaster, you can run the `mysql.rds_set_source_delay` procedure. After the `mysql.rds_start_replication_until` or `mysql.rds_start_replication_until_gtid` procedure stops replication, you can promote the read replica to be the new primary DB instance by using the instructions in [Promoting a read replica to be a standalone DB instance \(p. 285\)](#).

To use the `mysql.rds_start_replication_until_gtid` procedure, GTID-based replication must be enabled. To skip a specific GTID-based transaction that is known to cause disaster, you can use the [mysql.rds_skip_transaction_with_gtid \(p. 968\)](#) stored procedure. For more information on GTID-based replication, see [Using GTID-based replication for RDS for MySQL \(p. 910\)](#).

The `mysql.rds_set_source_delay` procedure is available in these versions of RDS for MySQL:

- MySQL 5.6.40 and later 5.6 versions
- MySQL 5.7.22 and later 5.7 versions

Examples

To delay replication from source database instance to the current read replica for at least one hour (3,600 seconds), you can call `mysql.rds_set_source_delay` with the following parameter:

```
CALL mysql.rds_set_source_delay(3600);
```

mysql.rds_start_replication

Initiates replication from a MySQL DB instance.

Note

You can use the [mysql.rds_start_replication_until \(p. 965\)](#) or [mysql.rds_start_replication_until_gtid \(p. 966\)](#) stored procedure to initiate replication from an RDS for MySQL DB instance and stop replication at the specified binary log file location.

Syntax

```
CALL mysql.rds_start_replication;
```

Usage notes

The master user must run the `mysql.rds_start_replication` procedure.

If you are configuring replication to import data from an instance of MySQL running external to Amazon RDS, you call `mysql.rds_start_replication` on the read replica to start the replication process after you have called [mysql.rds_set_external_master \(p. 954\)](#) to build the replication configuration. For more information, see [Restoring a backup into a MySQL DB instance \(p. 871\)](#).

If you are configuring replication to export data to an instance of MySQL external to Amazon RDS, you call `mysql.rds_start_replication` and `mysql.rds_stop_replication` on the read replica to control some replication actions, such as purging binary logs. For more information, see [Exporting data from a MySQL DB instance by using replication \(p. 921\)](#).

You can also call `mysql.rds_start_replication` on the read replica to restart any replication process that you previously stopped by calling [mysql.rds_stop_replication \(p. 967\)](#). For more information, see [Working with read replicas \(p. 278\)](#).

mysql.rds_start_replication_until

Initiates replication from an RDS for MySQL DB instance and stops replication at the specified binary log file location.

Syntax

```
CALL mysql.rds_start_replication_until (
  replication_log_file
  , replication_stop_point
);
```

Parameters

replication_log_file

The name of the binary log on the source database instance contains the replication information.

replication_stop_point

The location in the `replication_log_file` binary log at which replication will stop.

Usage notes

The master user must run the `mysql.rds_start_replication_until` procedure.

You can use this procedure with delayed replication for disaster recovery. If you have delayed replication configured, you can use this procedure to roll forward changes to a delayed read replica to the time

just before a disaster. After this procedure stops replication, you can promote the read replica to be the new primary DB instance by using the instructions in [Promoting a read replica to be a standalone DB instance \(p. 285\)](#).

You can configure delayed replication using the following stored procedures:

- [mysql.rds_set_configuration \(p. 972\)](#)
- [mysql.rds_set_external_master_with_delay \(p. 956\)](#)
- [mysql.rds_set_source_delay \(p. 964\)](#)

The file name specified for the `replication_log_file` parameter must match the source database instance binlog file name.

When the `replication_stop_point` parameter specifies a stop location that is in the past, replication is stopped immediately.

The `mysql.rds_start_replication_until` procedure is available in these versions of RDS for MySQL:

- MySQL 5.6.40 and later 5.6 versions
- MySQL 5.7.22 and later 5.7 versions

Examples

The following example initiates replication and replicates changes until it reaches location 120 in the `mysql-bin-changelog.000777` binary log file.

```
call mysql.rds_start_replication_until(
    'mysql-bin-changelog.000777',
    120);
```

mysql.rds_start_replication_until_gtid

Initiates replication from an RDS for MySQL DB instance and stops replication immediately after the specified global transaction identifier (GTID).

Syntax

```
CALL mysql.rds_start_replication_until_gtid (
  gtid
);
```

Parameters

gtid

The GTID after which replication is to stop.

Usage notes

The master user must run the `mysql.rds_start_replication_until_gtid` procedure.

For RDS for MySQL 5.7, this procedure is supported for MySQL 5.7.23 and later MySQL 5.7 versions. This procedure is not supported for RDS for MySQL 5.5, 5.6, or 8.0.

You can use this procedure with delayed replication for disaster recovery. If you have delayed replication configured, you can use this procedure to roll forward changes to a delayed read replica to the time just before a disaster. After this procedure stops replication, you can promote the read replica to be the new primary DB instance by using the instructions in [Promoting a read replica to be a standalone DB instance \(p. 285\)](#).

You can configure delayed replication using the following stored procedures:

- [mysql.rds_set_configuration \(p. 972\)](#)
- [mysql.rds_set_external_master_with_auto_position \(p. 959\)](#)
- [mysql.rds_set_source_delay \(p. 964\)](#)

When the gtid parameter specifies a transaction that has already been run by the replica, replication is stopped immediately.

Examples

The following example initiates replication and replicates changes until it reaches GTID 3E11FA47-71CA-11E1-9E33-C80AA9429562:23.

```
call mysql.rds_start_replication_until_gtid('3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

mysql.rds_stop_replication

Stops replication from a MySQL DB instance.

Syntax

```
CALL mysql.rds_stop_replication;
```

Usage notes

The master user must run the `mysql.rds_stop_replication` procedure.

If you are configuring replication to import data from an instance of MySQL running external to Amazon RDS, you call `mysql.rds_stop_replication` on the read replica to stop the replication process after the import has completed. For more information, see [Restoring a backup into a MySQL DB instance \(p. 871\)](#).

If you are configuring replication to export data to an instance of MySQL external to Amazon RDS, you call `mysql.rds_start_replication` and `mysql.rds_stop_replication` on the read replica to control some replication actions, such as purging binary logs. For more information, see [Exporting data from a MySQL DB instance by using replication \(p. 921\)](#).

You can also use `mysql.rds_stop_replication` to stop replication between two Amazon RDS DB instances. You typically stop replication to perform a long running operation on the read replica, such as creating a large index on the read replica. You can restart any replication process that you stopped by calling [mysql.rds_start_replication \(p. 964\)](#) on the read replica. For more information, see [Working with read replicas \(p. 278\)](#).

mysql.rds_skip_transaction_with_gtid

Skips replication of a transaction with the specified global transaction identifier (GTID) on a MySQL DB instance.

You can use this procedure for disaster recovery when a specific GTID transaction is known to cause a problem. Use this stored procedure to skip the problematic transaction. Examples of problematic transactions include transactions that disable replication, delete important data, or cause the DB instance to become unavailable.

Syntax

```
CALL mysql.rds_skip_transaction_with_gtid (
  gtid_to_skip
);
```

Parameters

gtid_to_skip

The GTID of the replication transaction to skip.

Usage notes

The master user must run the `mysql.rds_skip_transaction_with_gtid` procedure.

For RDS for MySQL 5.7, this procedure is supported for MySQL 5.7.23 and later MySQL 5.7 versions. This procedure is not supported for RDS for MySQL 5.5, 5.6, or 8.0.

Examples

The following example skips replication of the transaction with the GTID 3E11FA47-71CA-11E1-9E33-C80AA9429562:23.

```
call mysql.rds_skip_transaction_with_gtid('3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

mysql.rds_skip_repl_error

Skips and deletes a replication error on a MySQL DB instance.

Syntax

```
CALL mysql.rds_skip_repl_error;
```

Usage notes

The master user must run the `mysql.rds_skip_repl_error` procedure.

To determine if there are errors, run the MySQL `show slave status\G` command. If a replication error isn't critical, you can run `mysql.rds_skip_repl_error` to skip the error. If there are multiple

errors, `mysql.rds_skip_repl_error` deletes the first error, then warns that others are present. You can then use `show slave status\G` to determine the correct course of action for the next error. For information about the values returned, see [the MySQL documentation](#).

For more information about addressing replication errors with Amazon RDS, see [Troubleshooting a MySQL read replica problem \(p. 908\)](#).

Important

If you try to call `mysql.rds_skip_repl_error`, you might encounter the following error:
`ERROR 1305 (42000): PROCEDURE mysql.rds_skip_repl_error does not exist.`
If you do, upgrade your MySQL DB instance to the latest minor version or one of the minimum minor versions listed in this topic.

Replication stopped error

When you call the `mysql.rds_skip_repl_error` command, you might receive the following error message: `Slave is down or disabled`.

This error message appears because replication has stopped and could not be restarted.

If you need to skip a large number of errors, the replication lag can increase beyond the default retention period for binary log (binlog) files. In this case, you might encounter a fatal error due to binlog files being purged before they have been replayed on the read replica. This purge causes replication to stop, and you can no longer call the `mysql.rds_skip_repl_error` command to skip replication errors.

You can mitigate this issue by increasing the number of hours that binlog files are retained on your source database instance. After you have increased the binlog retention time, you can restart replication and call the `mysql.rds_skip_repl_error` command as needed.

To set the binlog retention time, use the [mysql.rds_set_configuration \(p. 972\)](#) procedure and specify a configuration parameter of '`binlog retention hours`' along with the number of hours to retain binlog files on the DB cluster. The following example sets the retention period for binlog files to 48 hours.

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

mysql.rds_next_master_log

Changes the source database instance log position to the start of the next binary log on the source database instance. Use this procedure only if you are receiving replication I/O error 1236 on a read replica.

Syntax

```
CALL mysql.rds_next_master_log(  
curr_master_log  
);
```

Parameters

curr_master_log

The index of the current master log file. For example, if the current file is named `mysql-bin-changelog.012345`, then the index is 12345. To determine the current master log file name, run the `SHOW SLAVE STATUS` command and view the `Master_Log_File` field.

Usage notes

The master user must run the `mysql.rds_next_master_log` procedure.

Warning

Call `mysql.rds_next_master_log` only if replication fails after a failover of a Multi-AZ DB instance that is the replication source, and the `Last_IO_Errno` field of `SHOW SLAVE STATUS` reports I/O error 1236.

Calling `mysql.rds_next_master_log` may result in data loss in the read replica if transactions in the source instance were not written to the binary log on disk before the failover event occurred. You can reduce the chance of this happening by configuring the source instance parameters `sync_binlog = 1` and `innodb_support_xa = 1`, although this may reduce performance. For more information, see [Working with read replicas \(p. 278\)](#).

Examples

Assume replication fails on an Amazon RDS read replica. Running `SHOW SLAVE STATUS\G` on the read replica returns the following result:

```
***** 1. row *****
Slave_IO_State:
    Master_Host: myhostXXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
    Master_User: MasterUser
    Master_Port: 3306
    Connect_Retry: 10
    Master_Log_File: mysql-bin-changelog.012345
    Read_Master_Log_Pos: 1219393
    Relay_Log_File: relaylog.012340
    Relay_Log_Pos: 30223388
    Relay_Master_Log_File: mysql-bin-changelog.012345
    Slave_IO_Running: No
    Slave_SQL_Running: Yes
    Replicate_Do_DB:
    Replicate_Ignore_DB:
    Replicate_Do_Table:
    Replicate_Ignore_Table:
    Replicate_Wild_Do_Table:
    Replicate_Wild_Ignore_Table:
        Last_Error:
        Skip_Counter: 0
        Exec_Master_Log_Pos: 30223232
        Relay_Log_Space: 5248928866
        Until_Condition: None
        Until_Log_File:
        Until_Log_Pos: 0
    Master_SSL_Allowed: No
    Master_SSL_CA_File:
    Master_SSL_CA_Path:
        Master_SSL_Cert:
        Master_SSL_Cipher:
        Master_SSL_Key:
    Seconds_Behind_Master: NULL
Master_SSL_Verify_Server_Cert: No
    Last_IO_Error: Got fatal error 1236 from master when reading data from
binary log: 'Client requested master to start replication from impossible position; the
first event 'mysql-bin-changelog.013406' at 1219393, the last event read from '/rdsdbdata/
log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/rdsdbdata/log/
binlog/mysql-bin-changelog.012345' at 4.'
    Last_SQL_Error:
```

```
Replicate_Ignore_Server_Ids:  
Master_Server_Id: 67285976
```

The `Last_IO_Error` field shows that the instance is receiving I/O error 1236. The `Master_Log_File` field shows that the file name is `mysql-bin-changelog.012345`, which means that the log file index is 12345. To resolve the error, you can call `mysql.rds_next_master_log` with the following parameter:

```
CALL mysql.rds_next_master_log(12345);
```

mysql.rds_innodb_buffer_pool_dump_now

Dumps the current state of the buffer pool to disk. For more information, see [InnoDB cache warming \(p. 837\)](#).

Syntax

```
CALL mysql.rds_innodb_buffer_pool_dump_now();
```

Usage notes

The master user must run the `mysql.rds_innodb_buffer_pool_dump_now` procedure.

The `mysql.rds_innodb_buffer_pool_dump_now` procedure is available in these versions of RDS for MySQL:

- MySQL 5.6
- MySQL 5.7
- MySQL 8.0

mysql.rds_innodb_buffer_pool_load_now

Loads the saved state of the buffer pool from disk. For more information, see [InnoDB cache warming \(p. 837\)](#).

Syntax

```
CALL mysql.rds_innodb_buffer_pool_load_now();
```

Usage notes

The master user must run the `mysql.rds_innodb_buffer_pool_load_now` procedure.

The `mysql.rds_innodb_buffer_pool_load_now` procedure is available in these versions of RDS for MySQL:

- MySQL 5.6
- MySQL 5.7

- MySQL 8.0

mysql.rds_innodb_buffer_pool_load_abort

Cancels a load of the saved buffer pool state while in progress. For more information, see [InnoDB cache warming \(p. 837\)](#).

Syntax

```
CALL mysql.rds_innodb_buffer_pool_load_abort();
```

Usage notes

The master user must run the `mysql.rds_innodb_buffer_pool_load_abort` procedure.

The `mysql.rds_innodb_buffer_pool_load_abort` procedure is available in these versions of RDS for MySQL:

- MySQL 5.6
- MySQL 5.7
- MySQL 8.0

mysql.rds_set_configuration

Specifies the number of hours to retain binary logs or the number of seconds to delay replication.

Syntax

```
CALL mysql.rds_set_configuration(name,value);
```

Parameters

name

The name of the configuration parameter to set.

value

The value of the configuration parameter.

Usage notes

The `mysql.rds_set_configuration` stored procedure is available in these versions of RDS for MySQL:

- MySQL 5.6
- MySQL 5.7
- MySQL 8.0

The `mysql.rds_set_configuration` procedure supports the following configuration parameters:

- [Binlog retention hours \(p. 973\)](#)
- [Target delay \(p. 973\)](#)

Binlog retention hours

The `binlog retention hours` parameter is used to specify the number of hours to retain binary log files. Amazon RDS normally purges a binary log as soon as possible, but the binary log might still be required for replication with a MySQL database external to Amazon RDS. For RDS for MySQL, the default value of `binlog retention hours` is `NULL` (do not retain binary logs). The default value for Aurora MySQL is `24` (retain binary logs for 1 day).

To specify the number of hours for Amazon RDS to retain binary logs on a DB instance, use the `mysql.rds_set_configuration` stored procedure and specify a period with enough time for replication to occur, as shown in the following example.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

For MySQL DB instances, the maximum `binlog retention hours` value is `168` (7 days).

After you set the retention period, monitor storage usage for the DB instance to make sure that the retained binary logs don't take up too much storage.

Target delay

Use the `target delay` parameter to specify the number of seconds to delay replication from source database instance to the read replica. The specified delay applies to new replicas created from the current DB instance. Amazon RDS normally replicates changes as soon as possible, but some environments might want to delay replication. For example, when replication is delayed, you can roll forward a delayed read replica to the time just before a disaster. If a table is dropped accidentally, you can use delayed replication to recover it quickly. The default value of `target delay` is `0` (don't delay replication).

For disaster recovery, you can use this configuration parameter with the [mysql.rds_start_replication_until \(p. 965\)](#) stored procedure or the [mysql.rds_start_replication_until_gtid \(p. 966\)](#) stored procedure. To roll forward changes to a delayed read replica to the time just before a disaster, you can run the `mysql.rds_set_configuration` procedure with this parameter set. After the `mysql.rds_start_replication_until` or `mysql.rds_start_replication_until_gtid` procedure stops replication, you can promote the read replica to be the new primary DB instance by using the instructions in [Promoting a read replica to be a standalone DB instance \(p. 285\)](#).

To use the `mysql.rds_start_replication_until_gtid` procedure, GTID-based replication must be enabled. To skip a specific GTID-based transaction that is known to cause disaster, you can use the [mysql.rds_skip_transaction_with_gtid \(p. 968\)](#) stored procedure. For more information about working with GTID-based replication, see [Using GTID-based replication for RDS for MySQL \(p. 910\)](#).

To specify the number of seconds for Amazon RDS to delay replication to a read replica, use the `mysql.rds_set_configuration` stored procedure and specify the number of seconds to delay replication. The following example specifies that replication is delayed by at least one hour (3,600 seconds).

```
call mysql.rds_set_configuration('target delay', 3600);
```

The limit for the `target delay` parameter is one day (86400 seconds).

Note

The `target delay` parameter is only supported for RDS for MySQL.

The target `delay` parameter is not supported for RDS for MySQL version 8.0.

mysql.rds_show_configuration

The number of hours that binary logs are retained.

Syntax

```
CALL mysql.rds_show_configuration;
```

Usage notes

To verify the number of hours that Amazon RDS retains binary logs, use the `mysql.rds_show_configuration` stored procedure.

The `mysql.rds_show_configuration` procedure is available in these versions of RDS for MySQL:

- MySQL 5.6
- MySQL 5.7
- MySQL 8.0

Examples

The following example displays the retention period:

```
call mysql.rds_show_configuration;
      name          value      description
      binlog retention hours    24      binlog retention hours specifies the
duration in hours before binary logs are automatically deleted.
```

mysql.rds_kill

Ends a connection to the MySQL server.

Syntax

```
CALL mysql.rds_kill(processID);
```

Parameters

processID

The identity of the connection thread to be ended.

Usage notes

Each connection to the MySQL server runs in a separate thread. To end a connection, use the `mysql.rds_kill` procedure and pass in the thread ID of that connection. To obtain the thread ID, use the MySQL `SHOW PROCESSLIST` command.

Examples

The following example ends a connection with a thread ID of 4243:

```
call mysql.rds_kill(4243);
```

mysql.rds_kill_query

Ends a query running against the MySQL server.

Syntax

```
CALL mysql.rds_kill_query(queryID);
```

Parameters

queryID

The identity of the query to be ended.

Usage notes

To stop a query running against the MySQL server, use the `mysql_rds_kill_query` procedure and pass in the ID of that query. To obtain the query ID, query the MySQL [INFORMATION_SCHEMA PROCESSLIST table](#). The connection to the MySQL server is retained.

Examples

The following example stops a query with a thread ID of 230040:

```
call mysql.rds_kill_query(230040);
```

mysql.rds_rotate_general_log

Rotates the `mysql.general_log` table to a backup table. For more information, see [Accessing MySQL database log files \(p. 519\)](#).

Syntax

```
CALL mysql.rds_rotate_general_log;
```

Usage notes

You can rotate the `mysql.general_log` table to a backup table by calling the `mysql.rds_rotate_general_log` procedure. When log tables are rotated, the current log table is copied to a backup log table and the entries in the current log table are removed. If a backup log table already exists, then it is deleted before the current log table is copied to the backup. You can query the backup log table if needed. The backup log table for the `mysql.general_log` table is named `mysql.general_log_backup`.

mysql.rds_rotate_slow_log

Rotates the `mysql.slow_log` table to a backup table. For more information, see [Accessing MySQL database log files \(p. 519\)](#).

Syntax

```
CALL mysql.rds_rotate_slow_log;
```

Usage notes

You can rotate the `mysql.slow_log` table to a backup table by calling the `mysql.rds_rotate_slow_log` procedure. When log tables are rotated, the current log table is copied to a backup log table and the entries in the current log table are removed. If a backup log table already exists, then it is deleted before the current log table is copied to the backup.

You can query the backup log table if needed. The backup log table for the `mysql.slow_log` table is named `mysql.slow_log_backup`.

mysql.rds_enable_gsh_collector

Enables the Global Status History (GoSH) to take default snapshots at intervals specified by `rds_set_gsh_collector`. For more information, see [Managing the global status history \(p. 936\)](#).

Syntax

```
CALL mysql.rds_enable_gsh_collector;
```

mysql.rds_set_gsh_collector

Specifies the interval, in minutes, between snapshots taken by the Global Status History (GoSH). Default value is For more information, see [Managing the global status history \(p. 936\)](#).

Syntax

```
CALL mysql.rds_set_gsh_collector(intervalPeriod);
```

Parameters

intervalPeriod

The interval, in minutes, between snapshots. Default value is

mysql.rds_disable_gsh_collector

Disables snapshots taken by the Global Status History (GoSH). For more information, see [Managing the global status history \(p. 936\)](#).

Syntax

```
CALL mysql.rds_disable_gsh_collector;
```

mysql.rds_collect_global_status_history

Takes a snapshot on demand for the Global Status History (GoSH). For more information, see [Managing the global status history \(p. 936\)](#).

Syntax

```
CALL mysql.rds_collect_global_status_history;
```

mysql.rds_enable_gsh_rotation

Enables rotation of the contents of the `mysql.global_status_history` table to `mysql.global_status_history_old` at intervals specified by `rds_set_gsh_rotation`. For more information, see [Managing the global status history \(p. 936\)](#).

Syntax

```
CALL mysql.rds_enable_gsh_rotation;
```

mysql.rds_set_gsh_rotation

Specifies the interval, in days, between rotations of the `mysql.global_status_history` table. Default value is 7. For more information, see [Managing the global status history \(p. 936\)](#).

Syntax

```
CALL mysql.rds_set_gsh_rotation(intervalPeriod);
```

Parameters

intervalPeriod

The interval, in days, between table rotations. Default value is 7.

mysql.rds_disable_gsh_rotation

Disables rotation of the `mysql.global_status_history` table. For more information, see [Managing the global status history \(p. 936\)](#).

Syntax

```
CALL mysql.rds_disable_gsh_rotation;
```

mysql.rds_rotate_global_status_history

Rotates the contents of the `mysql.global_status_history` table to `mysql.global_status_history_old` on demand. For more information, see [Managing the global status history \(p. 936\)](#).

Syntax

```
CALL mysql.rds_rotate_global_status_history;
```

Oracle on Amazon RDS

Amazon RDS supports DB instances that run the following versions and editions of Oracle Database:

- Oracle Database 19c (19.0.0.0)
- Oracle Database 18c (18.0.0.0)
- Oracle Database 12c Release 2 (12.2.0.1)
- Oracle Database 12c Release 1 (12.1.0.2)

Note

Oracle Database 18c (18.0.0.0) is on a deprecation path. Oracle Corporation will no longer provide patches for Oracle Database 18c after the end-of-support date. For more information, see [Preparing for the automatic upgrade of Oracle Database 18c \(p. 1214\)](#).
RDS for Oracle Database 11g is no longer supported.

You can create DB instances and DB snapshots, point-in-time restores, and automated or manual backups. You can use DB instances running Oracle inside a VPC. You can also add features to your Oracle DB instance by enabling various options. Amazon RDS supports Multi-AZ deployments for Oracle as a high-availability, failover solution.

To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances. It also restricts access to certain system procedures and tables that require advanced privileges. You can access databases on a DB instance using any standard SQL client application such as Oracle SQL*Plus. However, you can't access the host directly by using Telnet or Secure Shell (SSH).

When you create a DB instance using your master account, the account gets DBA privileges, with some limitations. Use this account for administrative tasks such as creating additional database accounts. You can't use SYS, SYSTEM, or other Oracle-supplied administrative accounts.

Before creating a DB instance, complete the steps in the [Setting up for Amazon RDS \(p. 67\)](#) section of this guide.

Oracle versions on Amazon RDS

Amazon RDS for Oracle supports the following major database releases.

Topics

- [Oracle Database 19c with Amazon RDS \(p. 979\)](#)
- [Oracle Database 18c on Amazon RDS \(p. 980\)](#)
- [Oracle Database 12c with Amazon RDS \(p. 981\)](#)

Oracle Database 19c with Amazon RDS

Amazon RDS supports Oracle Database 19c, which includes Oracle Enterprise Edition and Oracle Standard Edition Two.

Oracle Database 19c (19.0.0.0) includes many new features and updates from the previous version. In this section, you can find the features and changes important to using Oracle Database 19c (19.0.0.0) on Amazon RDS. For a complete list of the changes, see the [Oracle database 19c](#) documentation. For

a complete list of features supported by each Oracle Database 19c edition, see [Permitted features, options, and management packs by Oracle database offering](#) in the Oracle documentation.

Amazon RDS parameter changes for Oracle Database 19c (19.0.0.0)

Oracle Database 19c (19.0.0.0) includes several new parameters and parameters with new ranges and new default values.

The following table shows the new Amazon RDS parameters for Oracle Database 19c (19.0.0.0).

Name	Values	Modified in	Description
lob_signature_enable	TRUE, FALSE (default)	Y	Enables or disables the LOB locator signature feature.
max_datapump_parallel_per_job	1 to 1024, or AUTO	Y	Specifies the maximum number of parallel processes allowed for each Oracle Data Pump job.

The `compatible` parameter has a new maximum value for Oracle Database 19c (19.0.0.0) on Amazon RDS. The following table shows the new default value.

Parameter name	Oracle Database 19c (19.0.0.0) maximum value	Oracle Database 18c (18.0.0.0) maximum value
compatible	19.0.0	18.0.0

The following parameters were removed in Oracle Database 19c (19.0.0.0):

- `exafusion_enabled`
- `max_connections`
- `o7_dictionary_access`

Oracle Database 18c on Amazon RDS

Oracle Corporation intends to deprecate support for Oracle Database 18c (18.0.0.0) on July 1, 2021. On this date, Amazon RDS plans to do the following:

- Deprecate support for Oracle Database 18c for both BYOL and LI
- Begin upgrading all Oracle Database 18c instances automatically

The following schedule includes upgrade recommendations. For more information, see [Preparing for the automatic upgrade of Oracle Database 18c \(p. 1214\)](#).

Action or recommendation	Oracle Database 18c
We recommend that you upgrade Oracle Database 18c DB instances manually to Oracle Database 19c and validate your applications.	Now–June 30, 2021

Action or recommendation	Oracle Database 18c
We recommend that you upgrade Oracle Database 18c snapshots manually to Oracle Database 19c.	May 1, 2021
You can no longer create new Oracle Database 18c instances with Amazon RDS. You can continue to restore 18c DB snapshots without being automatically upgraded until June 30, 2021.	May 1, 2021
Amazon RDS plans to start automatic upgrades of your Oracle Database 18c instances to Oracle Database 19c.	July 1, 2021
Amazon RDS plans to start automatic upgrades to Oracle Database 19c for any Oracle Database 18c DB instances restored from snapshots.	July 1, 2021

Oracle Database 12c with Amazon RDS

Amazon RDS supports Oracle Database 12c, which includes Oracle Enterprise Edition and Oracle Standard Edition Two. Oracle Database 12c includes two major versions:

- [Oracle Database 12c Release 2 \(12.2.0.1\) with Amazon RDS \(p. 981\)](#)
- [Oracle Database 12c Release 1 \(12.1.0.2\) with Amazon RDS \(p. 984\)](#)

Oracle Database 12c Release 2 (12.2.0.1) with Amazon RDS

Oracle Database 12c Release 2 (12.2.0.1) includes many new features and updates from the previous version. In this section, you can find the features and changes important to using Oracle Database 12c Release 2 (12.2.0.1) on Amazon RDS. For a complete list of the changes, see the [Oracle Database 12c Release 2 \(12.2\) documentation](#). For a complete list of features supported by each Oracle Database 12c edition, see [Permitted features, options, and management packs by Oracle database offering](#) in the Oracle documentation.

Amazon RDS parameter changes for Oracle Database 12c Release 2 (12.2.0.1)

Oracle Database 12c Release 2 (12.2.0.1) includes 20 new parameters in addition to several parameters with new ranges and new default values.

The following table shows the new Amazon RDS parameters for Oracle Database 12c Release 2 (12.2.0.1).

Name	Values	Modified	Description
allow_global_dblinks	TRUE, FALSE (default)	Y	Specifies whether LDAP lookup for database links is allowed for the database.
approx_for_aggregation	TRUE, FALSE (default)	Y	Replaces exact query processing for aggregation queries with approximate query processing.
approx_for_count_distinct	TRUE, FALSE (default)	Y	Automatically replaces COUNT (DISTINCT <i>expr</i>) queries with APPROX_COUNT_DISTINCT queries.

Name	Values	Modifiable	Description
approx_for_percentile	NONE (default), PERCENTILE_CONT, PERCENTILE_CONT DETERMINISTIC, PERCENTILE_DISC, PERCENTILE_DISC DETERMINISTIC, ALL, ALL DETERMINISTIC	Y	Converts exact percentile functions to their approximate percentile function counterparts.
cursor_invalidation	DEFERRED, IMMEDIATE (default)	Y	Controls whether deferred cursor invalidation or immediate cursor invalidation is used for DDL statements by default.
data_guard_sync_latency	0 (default) to the number of seconds specified by the NET_TIMEOUT attribute for the LOG_ARCHIVE_DEST_n parameter	Y	Controls how many seconds the Log Writer (LGWR) process waits beyond the response of the first in a series of Oracle Data Guard SYNC redo transport mode connections.
data_transfer_cache_size	0 – 512M, rounded up to the next granule size	Y	Sets the size of the data transfer cache (in bytes) used to receive data blocks (typically from a primary database in an Oracle Data Guard environment) for consumption by an instance when an RMAN RECOVER ... NONLOGGED BLOCK command is running.
inmemory_adg_enabled	TRUE (default), FALSE	Y	Indicates whether in-memory for Active Data Guard is enabled in addition to the in-memory cache size.
inmemory_expressions_usage	STATIC_ONLY, DYNAMIC_ONLY, ENABLE (default), DISABLE	Y	Controls which In-Memory Expressions (IM expressions) are populated into the In-Memory Column Store (IM column store) and are available for queries.
inmemory_virtual_columns	ENABLE, MANUAL (default), DISABLE	Y	Controls which In-Memory Expressions (IM expressions) are populated into the In-Memory Column Store (IM column store) and are available for queries.
instance_abort_delay_time	0 (default) and higher	Y	Specifies how much time to delay an internally initiated instance shutdown (in seconds), such as when a fatal process dies or an unrecoverable instance error occurs.
instance_mode	READ-WRITE (default), READ-ONLY, READ-MOSTLY	N	Indicates whether the instance is read-write, read-only, or read-mostly.

Name	Values	Modifiable	Description
long_module_action	TRUE (default), FALSE	Y	Enables the use of longer lengths for modules and actions.
max_idle_time	0 (default) to the maximum integer. The value of 0 indicates that there is no limit.	Y	Specifies the maximum number of minutes that a session can be idle. After that point, the session is automatically terminated.
optimizer_adaptive_plans	TRUE (default), FALSE	Y	Controls adaptive plans. Adaptive plans are execution plans built with alternative choices that are decided at run time based on statistics collected as the query executes.
optimizer_adaptive_statistics	TRUE, FALSE (default)	Y	Controls adaptive statistics. Some query shapes are too complex to rely on base table statistics alone, so the optimizer augments these statistics with adaptive statistics.
outbound_dblink_protocols	ALL (default), NONE, TCP, TCPS, IPC	N	Specifies the network protocols allowed for communicating for outbound database links in the database.
resource_manage_goldengate	TRUE, FALSE (default)	Y	Determines whether Oracle GoldenGate apply processes in the database are resource managed.
standby_db_preserve_states	NONE (default), SESSION, ALL	N	Controls whether user sessions and other internal states of the instance are retained when a readable physical standby database is converted to a primary database.
uniform_log_timestamp_format	TRUE (default), FALSE	Y	Specifies that a uniform timestamp format be used in Oracle Database trace (.trc) files and log files (such as the alert log).

The `compatible` parameter has a new default value for Oracle Database 12c Release 2 (12.2.0.1) on Amazon RDS. The following table shows the new default value.

Parameter name	Oracle Database 12c Release 2 (12.2.0.1) default value	Oracle Database 12c Release 1 (12.1.0.2) default value
compatible	12.2.0	12.0.0

The `optimizer_features_enable` parameter has a new value range for Oracle Database 12c Release 2 (12.2.0.1) on Amazon RDS. For the old and new value ranges, see the following table.

Parameter name	Oracle Database 12c Release 2 (12.2.0.1) range	Oracle Database 12c Release 1 (12.1.0.2) range
optimizer_features e8.0.0 to 12.2.0.1		8.0.0 to 12.1.0.2

The following parameters were removed in Oracle Database 12c Release 2 (12.2.0.1):

- `global_context_pool_size`
- `max_enabled_roles`
- `optimizer_adaptive_features`
- `parallel_automatic_tuning`
- `parallel_degree_level`
- `use_indirect_data_buffers`

The following parameter is not supported in Oracle Database 12c Release 2 (12.2.0.1) and later:

- `sec_case_sensitive_logon`

Amazon RDS security changes for Oracle Database 12c Release 2 (12.2.0.1)

In Oracle Database 12c Release 2 (12.2.0.1), direct grant of the privilege `ADMINISTER DATABASE TRIGGER` is required for the owners of database-level triggers. During a major version upgrade to Oracle Database 12c Release 2 (12.2.0.1), Amazon RDS grants this privilege to any user that owns a trigger so that the trigger owner has the required privileges. For more information, see the My Oracle Support document [2275535.1](#).

Oracle Database 12c Release 1 (12.1.0.2) with Amazon RDS

Oracle Database 12c Release 1 (12.1.0.2) brings over 500 new features and updates from the previous version. In this section, you can find the features and changes important to using Oracle Database 12c Release 1 (12.1.0.2) on Amazon RDS. For a complete list of the changes, see the [Oracle Database 12c Release 1 \(12.1\) documentation](#). For a complete list of features supported by each Oracle Database 12c edition, see [Permitted features, options, and management packs by Oracle database edition](#) in the Oracle documentation.

Oracle Database 12c Release 1 (12.1.0.2) includes 16 new parameters that impact your Amazon RDS DB instance, and also 18 new system privileges, several no longer supported packages, and several new option group settings. For more information on these changes, see the following sections.

Amazon RDS parameter changes for Oracle Database 12c Release 1 (12.1.0.2)

Oracle Database 12c Release 1 (12.1.0.2) includes 16 new parameters in addition to several parameters with new ranges and new default values.

The following table shows the new Amazon RDS parameters for Oracle Database 12c Release 1 (12.1.0.2).

Name	Values	Modifi	Description
connection_brokers	CONNECTION_BROKERSN = broker_description[...]		Specifies connection broker types, the number of connection brokers of each type, and the maximum number of connections per broker.

Name	Values	Modifiable	Description
db_index_compression_inheritance	TABLESPACE, TABL, ALL, NONE	Y	Displays the options that are set for table or tablespace level compression inheritance.
db_big_table_cache_percent_target	0-90	Y	Specifies the cache section target size for automatic big table caching, as a percentage of the buffer cache.
heat_map	ON, OFF	Y	Enables the database to track read and write access of all segments and modification of database blocks that are due to data manipulation language (DML) and data definition language (DDL) statements.
inmemory_clause_default	INMEMORY, NO INMEMORY	Y	INMEMORY_CLAUSE_DEFAULT enables you to specify a default In-Memory Column Store (IM column store) clause for new tables and materialized views.
inmemory_clause_default_memory_compression	NO MEMCOMPRESS, MEMCOMPRESS FOR DML, MEMCOMPRESS FOR QUERY, MEMCOMPRESS FOR QUERY LOW, MEMCOMPRESS FOR QUERY HIGH, MEMCOMPRESS FOR CAPACITY, MEMCOMPRESS FOR CAPACITY LOW, MEMCOMPRESS FOR CAPACITY HIGH	Y	See INMEMORY_CLAUSE_DEFAULT.
inmemory_clause_default_priority	PRIORITY LOW, PRIORITY MEDIUM, PRIORITY HIGH, PRIORITY CRITICAL, PRIORITY NONE	Y	See INMEMORY_CLAUSE_DEFAULT.
inmemory_force	DEFAULT, OFF	Y	INMEMORY_FORCE allows you to specify whether tables and materialized view that are specified as INMEMORY are populated into the In-Memory Column Store (IM column store) or not.
inmemory_max_populate_servers	Null	N	INMEMORY_MAX_POPULATE_SERVERS specifies the maximum number of background populate servers to use for In-Memory Column Store (IM column store) population, so that these servers don't overload the rest of the system.

Name	Values	Modifiable	Description
inmemory_query	ENABLE (default), DISABLE	Y	INMEMORY_QUERY is used to enable or disable in-memory queries for the entire database at the session or system level.
inmemory_size	0, 104857600-274877906944	Y	INMEMORY_SIZE sets the size of the In-Memory Column Store (IM column store) on a database instance.
inmemory_trickle_repopulate_servers_percent	0 to 50 percent	Y	INMEMORY_TRICKLE_REPOPULATE_SERVERS_PERCENT limits the maximum number of background populate servers used for In-Memory Column Store (IM column store) repopulation. This limit is applied because trickle repopulation is designed to use only a small percentage of the populate servers.
max_string_size	STANDARD (default), EXTENDED	N	Controls the maximum size of VARCHAR2, NVARCHAR2, and RAW. For more information, see Enabling extended data types (p. 1103) .
optimizer_adaptive_features	TRUE (default), FALSE	Y	Enables or disables all of the adaptive optimizer features.
optimizer_adaptive_reporting_only	TRUE, FALSE (default)	Y	Controls reporting-only mode for adaptive optimizations.
pdb_file_name_convert	There is no default value.	N	Maps names of existing files to new file names.
pga_aggregate_limit	0-max of memory	Y	Specifies a limit on the aggregate PGA memory consumed by the instance.
processor_group_name	There is no default value.	N	Instructs the database instance to run itself within the specified operating system processor group.
spatial_vector_acceleration	TRUE, FALSE	N	Enables or disables the spatial vector acceleration, part of spatial option.
temp_undo_enabled	TRUE, FALSE (default)	Y	Determines whether transactions within a particular session can have a temporary undo log.
threaded_execution	TRUE, FALSE	N	Enables the multithreaded Oracle model, but prevents OS authentication.
unified_audit_sga_queue_size	1 MB - 30 MB	Y	Specifies the size of the system global area (SGA) queue for unified auditing.
use_dedicated_broker	TRUE, FALSE	N	Determines how dedicated servers are spawned.

Several parameters have new value ranges for Oracle Database 12c Release 1 (12.1.0.2) on Amazon RDS. For the old and new value ranges, see the following table.

Parameter name	Oracle Database 12c Release 1 (12.1.0.2) range
audit_trail	os db [, extended] xml [, extended]
compatible	If you upgrade to 12.2.0.1, 18c, or 19c, COMPATIBLE must be 11.2.0 or higher. We recommend that you use the default settings for COMPATIBLE for your version of Oracle Database unless you have a reason to change it. If COMPATIBLE is not explicitly set, Amazon RDS automatically sets this parameter to 12.0.0.
db_securefile	PERMITTED PREFERRED ALWAYS IGNORE FORCE
db_writer_processes	1-100
optimizer_features_enable	8.0.0 to 12.1.0.2
parallel_degree_policy	MANUAL,LIMITED,AUTO,ADAPTIVE
parallel_min_server	0 to parallel_max_servers

One parameter has a new default value for Oracle Database 12c on Amazon RDS. The following table shows the new default value.

Parameter name	Oracle Database 12c default value
job_queue_processes	50

Amazon RDS system privileges for Oracle Database 12c Release 1 (12.1.0.2)

Several new system privileges have been granted to the system account for Oracle Database 12c Release 1 (12.1.0.2). These new system privileges include the following:

- ALTER ANY CUBE BUILD PROCESS
- ALTER ANY MEASURE FOLDER
- ALTER ANY SQL TRANSLATION PROFILE
- CREATE ANY SQL TRANSLATION PROFILE
- CREATE SQL TRANSLATION PROFILE
- DROP ANY SQL TRANSLATION PROFILE
- EM EXPRESS CONNECT
- EXEMPT DDL REDACTION POLICY
- EXEMPT DML REDACTION POLICY
- EXEMPT REDACTION POLICY
- LOGMINING
- REDEFINE ANY TABLE
- SELECT ANY CUBE BUILD PROCESS
- SELECT ANY MEASURE FOLDER
- USE ANY SQL TRANSLATION PROFILE

Amazon RDS options for Oracle Database 12c Release 1 (12.1.0.2)

Several Oracle options changed between Oracle Database 11g and Oracle Database 12c Release 1 (12.1.0.2), though most of the options remain the same between the two versions. The Oracle Database 12c Release 1 (12.1.0.2) changes include the following:

- Oracle Enterprise Manager Database Express 12c replaced Oracle Enterprise Manager 11g Database Control. For more information, see [Oracle Enterprise Manager Database Express \(p. 1150\)](#).
- The option XMLDB is installed by default in Oracle Database 12c Release 1 (12.1.0.2). You no longer need to install this option yourself.

Amazon RDS PL/SQL packages for Oracle Database 12c Release 1 (12.1.0.2)

Oracle Database 12c Release 1 (12.1.0.2) includes a number of new built-in PL/SQL packages. The packages included with Amazon RDS for Oracle Database 12c Release 1 (12.1.0.2) include the following.

Package name	Description
CTX_ANL	The CTX_ANL package is used with AUTO_LEXER and provides procedures for adding and dropping a custom dictionary from the lexer.
DBMS_APP_CONT	The DBMS_APP_CONT package provides an interface to determine if the in-flight transaction on a now unavailable session committed or not, and if the last call on that session completed or not.
DBMS_AUTO_REPORT	The DBMS_AUTO_REPORT package provides an interface to view SQL Monitoring and Real-time Automatic Database Diagnostic Monitor (ADDM) data that has been captured into Automatic Workload Repository (AWR).
DBMS_GOLDENGATE_AUTH	The DBMS_GOLDENGATE_AUTH package provides subprograms for granting privileges to and revoking privileges from GoldenGate administrators.
DBMS_HEAT_MAP	The DBMS_HEAT_MAP package provides an interface to externalize heatmaps at various levels of storage including block, extent, segment, object, and tablespace.
DBMS_ILM	The DBMS_ILM package provides an interface for implementing Information Lifecycle Management (ILM) strategies using Automatic Data Optimization (ADO) policies.
DBMS_ILM_ADMIN	The DBMS_ILM_ADMIN package provides an interface to customize Automatic Data Optimization (ADO) policy execution.
DBMS_PART	The DBMS_PART package provides an interface for maintenance and management operations on partitioned objects.
DBMS_PRIVILEGE_CAPTURE	The DBMS_PRIVILEGE_CAPTURE package provides an interface to database privilege analysis.
DBMS_QOPATCH	The DBMS_QOPATCH package provides an interface to view the installed database patches.
DBMS_REDACT	The DBMS_REDACT package provides an interface to Oracle Data Redaction, which enables you to mask (redact) data that is returned from queries issued by low-privileged users or an application.

Package name	Description
DBMS_SPD	The DBMS_SPD package provides subprograms for managing SQL plan directives (SPD).
DBMS_SQL_TRANSLATOR	The DBMS_SQL_TRANSLATOR package provides an interface for creating, configuring, and using SQL translation profiles.
DBMS_SQL_MONITOR	The DBMS_SQL_MONITOR package provides information about real-time SQL Monitoring and real-time Database Operation Monitoring.
DBMS_SYNC_REFRESH	The DBMS_SYNC_REFRESH package provides an interface to perform a synchronous refresh of materialized views.
DBMS_TSDP_MANAGE	The DBMS_TSDP_MANAGE package provides an interface to import and manage sensitive columns and sensitive column types in the database. DBMS_TSDP_MANAGE is used with the DBMS_TSDP_PROTECT package for transparent sensitive data protection (TSDP) policies. DBMS_TSDP_MANAGE is available with the Enterprise Edition only.
DBMS_TSDP_PROTECT	The DBMS_TSDP_PROTECT package provides an interface to configure transparent sensitive data protection (TSDP) policies in conjunction with the DBMS_TSDP_MANAGE package. DBMS_TSDP_PROTECT is available with the Enterprise Edition only.
DBMS_XDB_CONFIG	The DBMS_XDB_CONFIG package provides an interface for configuring Oracle XML DB and its repository.
DBMS_XDB_CONSTANTS	The DBMS_XDB_CONSTANTS package provides an interface to commonly used constants. Oracle recommends using constants instead of dynamic strings to avoid typographical errors.
DBMS_XDB_REPOS	The DBMS_XDB_REPOS package provides an interface to operate on the Oracle XML database Repository.
DBMS_XMLSCHEMA_ANNOTATE	The DBMS_XMLSCHEMA_ANNOTATE package provides an interface to manage and configure the structured storage model, mainly through the use of pre-registration schema annotations.
DBMS_XMLSTORAGE_MANAGE	The DBMS_XMLSTORAGE_MANAGE package provides an interface to manage and modify XML storage after schema registration has been completed.
DBMS_XSTREAM_ADMIN	The DBMS_XSTREAM_ADMIN package provides interfaces for streaming database changes between an Oracle database and other systems. XStream enables applications to stream out or stream in database changes.
DBMS_XSTREAM_AUTH	The DBMS_XSTREAM_AUTH package provides subprograms for granting privileges to and revoking privileges from XStream administrators.
UTL_CALL_STACK	The UTL_CALL_STACK package provides an interface to provide information about currently executing subprograms.

Oracle Database 12c Release 1 (12.1.0.2) packages not supported

Several Oracle Database 11g PL/SQL packages are not supported in Oracle Database 12c Release 1 (12.1.0.2). These packages include the following:

- DBMS_AUTO_TASK_IMMEDIATE
- DBMS_CDC_PUBLISH
- DBMS_CDC_SUBSCRIBE
- DBMS_EXPFIL
- DBMS_OBFUSCATION_TOOLKIT
- DBMS_RLMGR
- SDO_NET_MEM

Oracle licensing options

Amazon RDS for Oracle has two licensing options: License Included (LI) and Bring Your Own License (BYOL). After you create an Oracle DB instance on Amazon RDS, you can change the licensing model by modifying the DB instance. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

License Included

In the License Included model, you don't need to purchase Oracle licenses separately. AWS holds the license for the Oracle database software. In this model, if you have an AWS Support account with case support, contact AWS Support for both Amazon RDS and Oracle Database service requests. The License Included model is only supported on Amazon RDS for Oracle Database Standard Edition Two (SE2).

Bring Your Own License (BYOL)

In the BYOL model, you can use your existing Oracle Database licenses to run Oracle deployments on Amazon RDS. You must have the appropriate Oracle Database license (with Software Update License and Support) for the DB instance class and Oracle Database edition you wish to run. You must also follow Oracle's policies for licensing Oracle Database software in the cloud computing environment. For more information on Oracle's licensing policy for Amazon EC2, see [Licensing Oracle software in the cloud computing environment](#).

In this model, you continue to use your active Oracle support account, and you contact Oracle directly for Oracle Database service requests. If you have an AWS Support account with case support, you can contact AWS Support for Amazon RDS issues. Amazon Web Services and Oracle have a multi-vendor support process for cases that require assistance from both organizations.

Amazon RDS supports the BYOL model only for Oracle Database Enterprise Edition (EE) and Oracle Database Standard Edition Two (SE2).

Integrating with AWS License Manager

To make it easier to monitor Oracle license usage in the BYOL model, [AWS License Manager](#) integrates with Amazon RDS for Oracle. License Manager supports tracking of RDS for Oracle engine editions and licensing packs based on virtual cores (vCPUs). You can also use License Manager with AWS Organizations to manage all of your organizational accounts centrally.

The following table shows the product information filters for RDS for Oracle.

Filter	Name	Description
Engine Edition	oracle-ee	Oracle Database Enterprise Edition (EE)
	oracle-se2	Oracle Database Standard Edition Two (SE2)
License Pack	data_guard	See Working with Oracle replicas for Amazon RDS (p. 1119) (Oracle Active Data Guard)
	olap	See Oracle OLAP (p. 1180)
	ols	See Oracle Label Security (p. 1167)
	diagnostic pack sqlt	See Oracle SQLT (p. 1193)
	tuning pack sqlt	See Oracle SQLT (p. 1193)

To track license usage of your Oracle DB instances, you can create a license configuration. In this case, RDS for Oracle resources that match the product information filter are automatically associated with the license configuration. Discovery of Oracle DB instances can take up to 24 hours.

Console

To create a license configuration to track the license usage of your Oracle DB instances

1. Go to <https://console.aws.amazon.com/license-manager/>.
2. Create a license configuration.

For instructions, see [Create a license configuration](#) in the *AWS License Manager User Guide*.

Add a rule for an **RDS Product Information Filter** in the **Product Information** panel.

For more information, see [ProductInformation](#) in the *AWS License Manager API Reference*.

AWS CLI

To create a license configuration by using the AWS CLI, call the `create-license-configuration` command. Use the `--cli-input-json` or `--cli-input-yaml` parameters to pass the parameters to the command.

Example

The following code creates a license configuration for Oracle Enterprise Edition.

```
aws license-manager create-license-configuration --cli-input-json file://rds-oracle-ee.json
```

The following is the sample `rds-oracle-ee.json` file used in the example.

```
{
    "Name": "rds-oracle-ee",
    "Description": "RDS Oracle Enterprise Edition",
    "LicenseCountingType": "vCPU",
    "LicenseCountHardLimit": false,
    "ProductInformationList": [
        {
            "Name": "rds-oracle-ee"
        }
    ]
}
```

```
"ResourceType": "RDS",
"ProductInformationFilterList": [
    {
        "ProductInformationFilterName": "Engine Edition",
        "ProductInformationFilterValue": ["oracle-ee"],
        "ProductInformationFilterComparator": "EQUALS"
    }
]
```

For more information about product information, see [Automated discovery of resource inventory](#) in the *AWS License Manager User Guide*.

For more information about the --cli-input parameter, see [Generating AWS CLI skeleton and input parameters from a JSON or YAML input file](#) in the *AWS CLI User Guide*.

Migrating between Oracle editions

If you have an unused BYOL Oracle license appropriate for the edition and class of DB instance that you plan to run, you can migrate from Standard Edition 2 (SE2) to Enterprise Edition (EE). You can't migrate from Enterprise Edition to other editions.

To change the edition and retain your data

1. Create a snapshot of the DB instance.
For more information, see [Creating a DB snapshot \(p. 346\)](#).
2. Restore the snapshot to a new DB instance, and select the Oracle database edition you want to use.
For more information, see [Restoring from a DB snapshot \(p. 349\)](#).
3. (Optional) Delete the old DB instance, unless you want to keep it running and have the appropriate Oracle Database licenses for it.
For more information, see [Deleting a DB instance \(p. 324\)](#).

Licensing Oracle Multi-AZ deployments

Amazon RDS supports Multi-AZ deployments for Oracle as a high-availability, failover solution. We recommend Multi-AZ for production workloads. For more information, see [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).

If you use the Bring Your Own License model, you must have a license for both the primary DB instance and the standby DB instance in a Multi-AZ deployment.

RDS for Oracle instance classes

The computation and memory capacity of a DB instance is determined by its DB instance class. The DB instance class you need depends on your processing power and memory requirements. For more information, see [DB instance classes \(p. 7\)](#).

The following are the DB instance classes supported for Oracle.

Oracle edition	Oracle Database 19c, Oracle Database 18c, and Oracle Database 12c Release 2 (12.2.0.1) support	Oracle Database 12c Release 1 (12.1.0.2) support
Enterprise Edition (EE) Bring Your Own License (BYOL)	db.m5.large–db.m5.24xlarge db.m4.large–db.m4.16xlarge db.z1d.large–db.z1d.12xlarge db.x1e.xlarge–db.x1e.32xlarge db.x1.16xlarge–db.x1.32xlarge db.r5.large–db.r5.24xlarge db.r5b.large–db.r5b.24xlarge db.r4.large–db.r4.16xlarge db.t3.small–db.t3.2xlarge	db.m5.large–db.m5.24xlarge db.m4.large–db.m4.16xlarge db.z1d.large–db.z1d.12xlarge db.x1e.xlarge–db.x1e.32xlarge db.x1.16xlarge–db.x1.32xlarge db.r5.large–db.r5.24xlarge db.r5b.large–db.r5b.24xlarge db.r4.large–db.r4.16xlarge db.t3.micro–db.t3.2xlarge
Standard Edition 2 (SE2) Bring Your Own License (BYOL)	db.m5.large–db.m5.4xlarge db.m4.large–db.m4.4xlarge db.z1d.large–db.z1d.3xlarge db.r5.large–db.r5.4xlarge db.r5b.large–db.r5b.4xlarge db.r4.large–db.r4.4xlarge db.t3.small–db.t3.2xlarge	db.m5.large–db.m5.4xlarge db.m4.large–db.m4.4xlarge db.z1d.large–db.z1d.3xlarge db.r5.large–db.r5.4xlarge db.r5b.large–db.r5b.4xlarge db.r4.large–db.r4.4xlarge db.t3.micro–db.t3.2xlarge
Standard Edition 2 (SE2) License Included	db.m5.large–db.m5.4xlarge db.m4.large–db.m4.4xlarge db.r5.large–db.r5.4xlarge db.r4.large–db.r4.4xlarge db.t3.small–db.t3.2xlarge	db.m5.large–db.m5.4xlarge db.m4.large–db.m4.4xlarge db.r5.large–db.r5.4xlarge db.r4.large–db.r4.4xlarge db.t3.micro–db.t3.2xlarge

Note

We encourage all BYOL customers to consult their licensing agreement to assess the impact of Amazon RDS for Oracle deprecations. For more information on the compute capacity of DB instance classes supported by Amazon RDS for Oracle, see [DB instance classes \(p. 7\)](#) and [Configuring the processor for a DB instance class \(p. 20\)](#).

Note

If you have DB snapshots of DB instances that were using deprecated DB instance classes, you can choose a DB instance class that is not deprecated when you restore the DB snapshots. For more information, see [Restoring from a DB snapshot \(p. 349\)](#).

Deprecated DB instance classes for Oracle

Following are the DB instance classes deprecated for Amazon RDS for Oracle:

- db.m1, db.m2, db.m3
- db.t1, db.t2
- db.r1, db.r2, db.r3

The preceding DB instance classes have been replaced by better performing DB instance classes that are generally available at a lower cost. Amazon RDS for Oracle automatically scales DB instances to DB instance classes that are not deprecated.

If you have DB instances that use deprecated DB instance classes, Amazon RDS will modify each one automatically to use a comparable DB instance class that is not deprecated. You can change the DB instance class for a DB instance yourself by modifying the DB instance. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

If you have DB snapshots of DB instances that were using deprecated DB instance classes, you can choose a DB instance class that is not deprecated when you restore the DB snapshots. For more information, see [Restoring from a DB snapshot \(p. 349\)](#).

RDS for Oracle features

Amazon RDS for Oracle supports most of the features and capabilities of Oracle Database. Some features might have limited support or restricted privileges. Some features are only available in Enterprise Edition, and some require additional licenses. For more information about Oracle Database features for specific Oracle Database versions, see the *Oracle Database Licensing Information User Manual* for the version you're using.

Note

These lists are not exhaustive.

Topics

- [Supported features for RDS for Oracle \(p. 994\)](#)
- [Unsupported features for RDS for Oracle \(p. 996\)](#)

Supported features for RDS for Oracle

Amazon RDS Oracle supports the following Oracle Database features:

- Advanced Compression
- Application Express (APEX)

For more information, see [Oracle Application Express \(APEX\) \(p. 1140\)](#).

- Automatic Memory Management
- Automatic Undo Management
- Automatic Workload Repository (AWR)

For more information, see [Generating performance reports with Automatic Workload Repository \(AWR\) \(p. 1053\)](#).

- Active Data Guard with Maximum Performance in the same AWS Region or across AWS Regions

For more information, see [Working with Oracle replicas for Amazon RDS \(p. 1119\)](#).

- Continuous Query Notification (version 12.1.0.2.v7 and later)

For more information, see [Using Continuous Query Notification \(CQN\) in the Oracle documentation](#).

- Data Redaction
- Database Change Notification

For more information, see [Database Change Notification](#) in the Oracle documentation.

Note

This feature changes to Continuous Query Notification in Oracle Database 12c Release 1 (12.1) and later.

- Database In-Memory (Oracle Database 12c and later)
- Distributed Queries and Transactions
- Edition-Based Redefinition

For more information, see [Setting the default edition for a DB instance \(p. 1057\)](#).

- EM Express (12c and later)

For more information, see [Oracle Enterprise Manager \(p. 1149\)](#).

- Fine-Grained Auditing
- Flashback Table, Flashback Query, Flashback Transaction Query
- Import/export (legacy and Data Pump) and SQL*Loader

For more information, see [Importing data into Oracle on Amazon RDS \(p. 1106\)](#).

- Java Virtual Machine (JVM)

For more information, see [Oracle Java virtual machine \(p. 1164\)](#).

- Label Security (Oracle Database 12c and later)

For more information, see [Oracle Label Security \(p. 1167\)](#).

- Locator

For more information, see [Oracle Locator \(p. 1170\)](#).

- Materialized Views
- Multimedia

For more information, see [Oracle Multimedia \(p. 1173\)](#).

- Network encryption

For more information, see [Oracle native network encryption \(p. 1176\)](#) and [Oracle Secure Sockets Layer \(p. 1182\)](#).

- Partitioning
- Spatial and Graph

For more information, see [Oracle Spatial \(p. 1190\)](#).

- Star Query Optimization
- Streams and Advanced Queuing
- Summary Management – Materialized View Query Rewrite
- Text (File and URL data store types are not supported)
- Total Recall
- Transparent Data Encryption (TDE)

For more information, see [Oracle Transparent Data Encryption \(p. 1204\)](#).

- Unified Auditing, Mixed Mode (Oracle Database 12c and later)

For more information, see [Mixed mode auditing](#) in the Oracle documentation.

- XML DB (without the XML DB Protocol Server)

For more information, see [Oracle XML DB \(p. 1208\)](#).

- Virtual Private Database

Unsupported features for RDS for Oracle

Amazon RDS Oracle doesn't support the following Oracle Database features:

- Automatic Storage Management (ASM)
- Database Vault
- Flashback Database
- Messaging Gateway
- Multitenant
- Oracle Enterprise Manager Cloud Control Management Repository
- Real Application Clusters (Oracle RAC)
- Real Application Testing
- Unified Auditing, Pure Mode
- Workspace Manager (WMSYS) schema

Warning

In general, Amazon RDS doesn't prevent you from creating schemas for unsupported features. However, if you create schemas for Oracle features and components that require SYS privileges, you can damage the data dictionary and affect the availability of your instance. Use only supported features and schemas that are available in [Adding options to Oracle DB instances \(p. 1126\)](#).

RDS for Oracle parameters

In Amazon RDS, you manage parameters using parameter groups. For more information, see [Working with DB parameter groups \(p. 228\)](#). To view the supported parameters for a specific Oracle Database edition and version, run the AWS CLI `describe-engine-default-parameters` command.

For example, to view the supported parameters for the Enterprise Edition of Oracle Database 12c Release 2 (12.2), run the following command.

```
aws rds describe-engine-default-parameters \
--db-parameter-group-family oracle-ee-12.2
```

RDS for Oracle character sets

Amazon RDS for Oracle supports two types of character sets: the DB character set and national character set.

DB character set

The Oracle database character set is used in the CHAR, VARCHAR2, and CLOB data types. The database also uses this character set for metadata such as table names, column names, and SQL statements. The Oracle database character set is typically referred to as the *DB character set*.

You set the character set when you create a DB instance. You can't change the DB character set after you create the database.

Supported DB character sets

The following table lists the Oracle DB character sets that are supported in Amazon RDS. You can use a value from this table with the --character-set-name parameter of the AWS CLI [create-db-instance](#) command or with the CharacterSetName parameter of the Amazon RDS API [CreateDBInstance](#) operation.

Value	Description
AL32UTF8	Unicode 5.0 UTF-8 Universal character set (default)
AR8ISO8859P6	ISO 8859-6 Latin/Arabic
AR8MSWIN1256	Microsoft Windows Code Page 1256 8-bit Latin/Arabic
BLT8ISO8859P13	ISO 8859-13 Baltic
BLT8MSWIN1257	Microsoft Windows Code Page 1257 8-bit Baltic
CL8ISO8859P5	ISO 8859-5 Latin/Cyrillic
CL8MSWIN1251	Microsoft Windows Code Page 1251 8-bit Latin/Cyrillic
EE8ISO8859P2	ISO 8859-2 East European
EL8ISO8859P7	ISO 8859-7 Latin/Greek
EE8MSWIN1250	Microsoft Windows Code Page 1250 8-bit East European
EL8MSWIN1253	Microsoft Windows Code Page 1253 8-bit Latin/Greek
IW8ISO8859P8	ISO 8859-8 Latin/Hebrew
IW8MSWIN1255	Microsoft Windows Code Page 1255 8-bit Latin/Hebrew
JA16EUC	EUC 24-bit Japanese
JA16EUCTILDE	Same as JA16EUC except for mapping of wave dash and tilde to and from Unicode
JA16SJIS	Shift-JIS 16-bit Japanese
JA16SJISTILDE	Same as JA16SJIS except for mapping of wave dash and tilde to and from Unicode

Value	Description
KO16MSWIN949	Microsoft Windows Code Page 949 Korean
NE8ISO8859P10	ISO 8859-10 North European
NEE8ISO8859P4	ISO 8859-4 North and Northeast European
TH8TISASCII	Thai Industrial Standard 620-2533-ASCII 8-bit
TR8MSWIN1254	Microsoft Windows Code Page 1254 8-bit Turkish
US7ASCII	ASCII 7-bit American
UTF8	Unicode 3.0 UTF-8 Universal character set, CESU-8 compliant
VN8MSWIN1258	Microsoft Windows Code Page 1258 8-bit Vietnamese
WE8ISO8859P1	Western European 8-bit ISO 8859 Part 1
WE8ISO8859P15	ISO 8859-15 West European
WE8ISO8859P9	ISO 8859-9 West European and Turkish
WE8MSWIN1252	Microsoft Windows Code Page 1252 8-bit West European
ZHS16GBK	GBK 16-bit Simplified Chinese
ZHT16HKSCS	Microsoft Windows Code Page 950 with Hong Kong Supplementary Character Set HKSCS-2001. Character set conversion is based on Unicode 3.0.
ZHT16MSWIN950	Microsoft Windows Code Page 950 Traditional Chinese
ZHT32EUC	EUC 32-bit Traditional Chinese

NLS_LANG environment variable

A *locale* is a set of information addressing linguistic and cultural requirements that corresponds to a given language and country. Setting the NLS_LANG environment variable in your client's environment is the simplest way to specify locale behavior for Oracle. This variable sets the language and territory used by the client application and the database server. It also indicates the client's character set, which corresponds to the character set for data entered or displayed by a client application. For more information on NLS_LANG and character sets, see [What is a character set or code page?](#) in the Oracle documentation.

NLS initialization parameters

You can also set the following National Language Support (NLS) initialization parameters at the instance level for an Oracle DB instance in Amazon RDS:

- NLS_DATE_FORMAT
- NLS_LENGTH_SEMANTICS
- NLS_NCHAR_CONV_EXCP

- NLS_TIME_FORMAT
- NLS_TIME_TZ_FORMAT
- NLS_TIMESTAMP_FORMAT
- NLS_TIMESTAMP_TZ_FORMAT

For information about modifying instance parameters, see [Working with DB parameter groups \(p. 228\)](#).

You can set other NLS initialization parameters in your SQL client. For example, the following statement sets the NLS_LANGUAGE initialization parameter to GERMAN in a SQL client that is connected to an Oracle DB instance:

```
ALTER SESSION SET NLS_LANGUAGE=GERMAN;
```

For information about connecting to an Oracle DB instance with a SQL client, see [Connecting to your Oracle DB instance \(p. 1001\)](#).

National character set

The national character set is used in the NCHAR, NVARCHAR2, and NCLOB data types. The national character set is typically referred to as the *NCHAR character set*. Unlike the DB character set, the NCHAR character set doesn't affect database metadata.

The NCHAR character set supports the following character sets:

- AL16UTF16 (default)
- UTF8

You can specify either value with the --nchar-character-set-name parameter of the [create-db-instance](#) command (AWS CLI version 2 only). If you use the Amazon RDS API, specify the NcharCharacterSetName parameter of [CreateDBInstance](#) operation. You can't change the national character set after you create the database.

For more information about Unicode in Oracle databases, see [Supporting multilingual databases with unicode](#) in the Oracle documentation.

RDS for Oracle limitations

Following are important limitations of using Amazon RDS for Oracle.

Note

This list is not exhaustive.

Topics

- [Oracle file size limits in Amazon RDS \(p. 999\)](#)
- [Public synonyms for Oracle-supplied schemas \(p. 1000\)](#)
- [Schemas for unsupported features \(p. 1000\)](#)
- [Limitations for Oracle DBA privileges \(p. 1000\)](#)

Oracle file size limits in Amazon RDS

The maximum file size on Amazon RDS Oracle DB instances is 16 TiB (tebibytes). If you try to resize a data file in a bigfile tablespace to a value over the limit, you receive an error such as the following.

```
ORA-01237: cannot extend datafile 6
ORA-01110: data file 6: '/rdsdbdata/db/mydir/datafile/myfile.dbf'
ORA-27059: could not reduce file size
Linux-x86_64 Error: 27: File too large
Additional information: 2
```

Public synonyms for Oracle-supplied schemas

Don't create or modify public synonyms for Oracle-supplied schemas, including `SYS`, `SYSTEM`, and `RDSADMIN`. Such actions might result in invalidation of core database components and affect the availability of your DB instance.

You can create public synonyms referencing objects in your own schemas.

Schemas for unsupported features

In general, Amazon RDS doesn't prevent you from creating schemas for unsupported features. However, if you create schemas for Oracle features and components that require `SYS` privileges, you can damage the data dictionary and affect your instance availability. Use only supported features and schemas that are available in [Adding options to Oracle DB instances \(p. 1126\)](#).

Limitations for Oracle DBA privileges

In the database, a *role* is a collection of privileges that you can grant to or revoke from a user. An Oracle database uses roles to provide security.

The predefined role `DBA` normally allows all administrative privileges on an Oracle database. When you create a DB instance, your master user account gets `DBA` privileges (with some limitations). To deliver a managed experience, an RDS for Oracle database doesn't provide the following privileges for the `DBA` role:

- `ALTER DATABASE`
- `ALTER SYSTEM`
- `CREATE ANY DIRECTORY`
- `DROP ANY DIRECTORY`
- `GRANT ANY PRIVILEGE`
- `GRANT ANY ROLE`

Use the master user account for administrative tasks such as creating additional user accounts in the database. You can't use `SYS`, `SYSTEM`, and other Oracle-supplied administrative accounts.

Connecting to your Oracle DB instance

After Amazon RDS provisions your Oracle DB instance, you can use any standard SQL client application to connect to the DB instance. In this topic, you connect to a DB instance that is running the Oracle database engine by using Oracle SQL Developer or SQL*Plus.

For an example that walks you through the process of creating and connecting to a sample DB instance, see [Creating an Oracle DB instance and connecting to a database on an Oracle DB instance \(p. 93\)](#).

Topics

- [Finding the endpoint of your Oracle DB instance \(p. 1001\)](#)
- [Connecting to your DB instance using Oracle SQL developer \(p. 1003\)](#)
- [Connecting to your DB instance using SQL*Plus \(p. 1005\)](#)
- [Considerations for security groups \(p. 1006\)](#)
- [Considerations for process architecture \(p. 1006\)](#)
- [Troubleshooting connections to your Oracle DB instance \(p. 1006\)](#)
- [Modifying connection properties using sqlnet.ora parameters \(p. 1007\)](#)

Finding the endpoint of your Oracle DB instance

Each Amazon RDS DB instance has an endpoint, and each endpoint has the DNS name and port number for the DB instance. To connect to your DB instance using a SQL client application, you need the DNS name and port number for your DB instance.

You can find the endpoint for a DB instance using the Amazon RDS console or the AWS CLI.

Note

If you are using Kerberos authentication, see [Connecting to Oracle with Kerberos authentication \(p. 1025\)](#).

Console

To find the endpoint using the console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the console, choose the AWS Region of your DB instance.
3. Find the DNS name and port number for your DB Instance.
 - a. Choose **Databases** to display a list of your DB instances.
 - b. Choose the Oracle DB instance name to display the instance details.
 - c. On the **Connectivity & security** tab, copy the endpoint. Also, note the port number. You need both the endpoint and the port number to connect to the DB instance.

database-1

[Modify](#)

Summary

DB identifier database-1	CPU <div style="width: 2.3%; background-color: #ff9999; height: 10px;"></div> 2.30%	Status Available	Class db.m4.large
Role Instance	Current activity <div style="width: 0.01%; background-color: #0070C0; height: 10px;"></div> 0.01 Sessions	Engine Oracle Standard Edition Two	Region & AZ us-east-1f

[Connectivity & security](#) [Monitoring](#) [Logs & events](#) [Configuration](#) [Maintenance & backups](#)

Connectivity & security

Endpoint & port	Networking	Security
Endpoint database-1.abcdefghijkl.us-east-1.rds.amazonaws.com	Availability zone us-east-1f	VPC security groups default (sg-0a5cba2b) (active)
Port 1521	VPC vpc-1234567f	Public accessibility No
	Subnet group	

AWS CLI

To find the endpoint of an Oracle DB instance by using the AWS CLI, call the [describe-db-instances](#) command.

Example To find the endpoint using the AWS CLI

```
aws rds describe-db-instances
```

Search for `Endpoint` in the output to find the DNS name and port number for your DB instance. The `Address` line in the output contains the DNS name. The following is an example of the JSON endpoint output.

```
{"Endpoint": {  
    "HostedZoneId": "Z1PVIF0B656C1W",  
    "Port": 3306,  
    "Address": "myinstance.123456789012.us-west-2.rds.amazonaws.com"  
},
```

Note

The output might contain information for multiple DB instances.

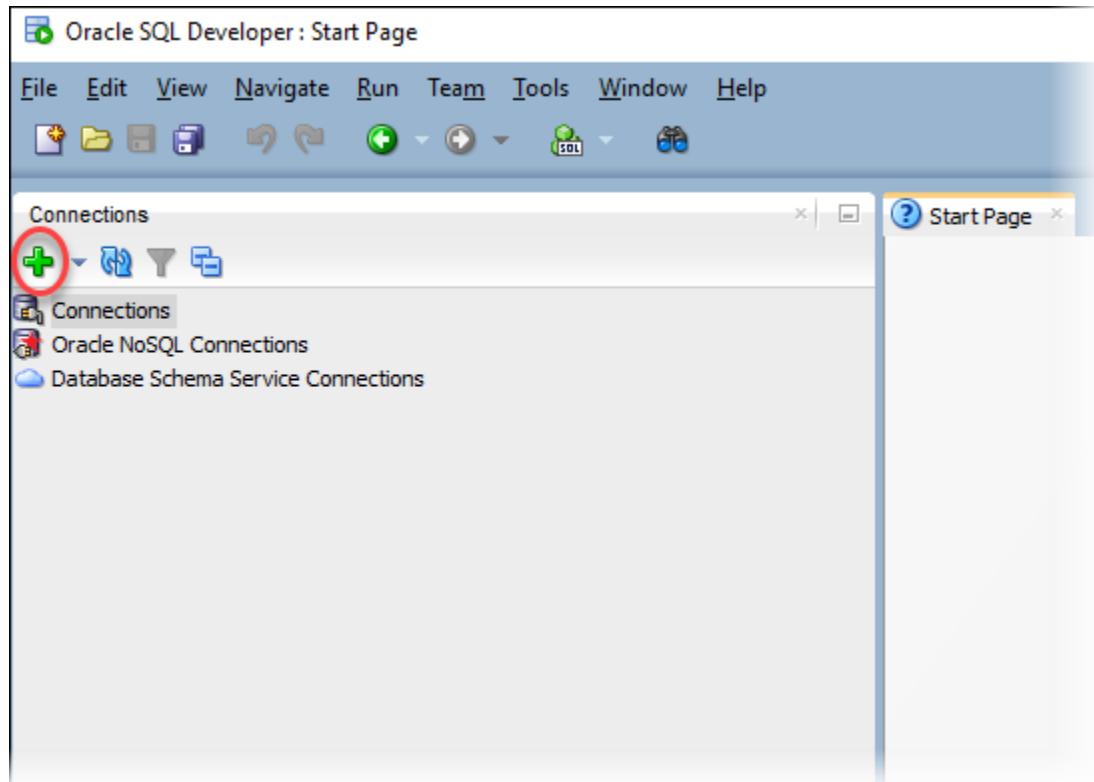
Connecting to your DB instance using Oracle SQL developer

In this procedure, you connect to your DB instance by using Oracle SQL Developer. To download a standalone version of this utility, see the [Oracle SQL developer downloads page](#).

To connect to your DB instance, you need its DNS name and port number. For information about finding the DNS name and port number for a DB instance, see [Finding the endpoint of your Oracle DB instance \(p. 1001\)](#).

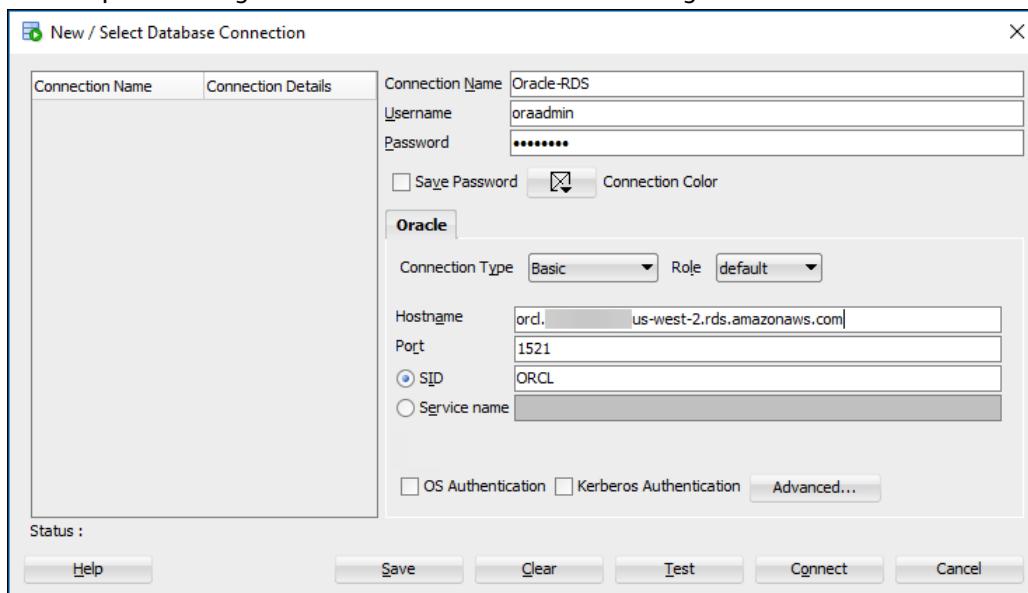
To connect to a DB instance using SQL developer

1. Start Oracle SQL Developer.
2. On the **Connections** tab, choose the **add (+)** icon.



3. In the **New>Select Database Connection** dialog box, provide the information for your DB instance:
 - For **Connection Name**, enter a name that describes the connection, such as Oracle-RDS.
 - For **Username**, enter the name of the database administrator for the DB instance.
 - For **Password**, enter the password for the database administrator.
 - For **Hostname**, enter the DNS name of the DB instance.
 - For **Port**, enter the port number.
 - For **SID**, enter the Oracle database SID.

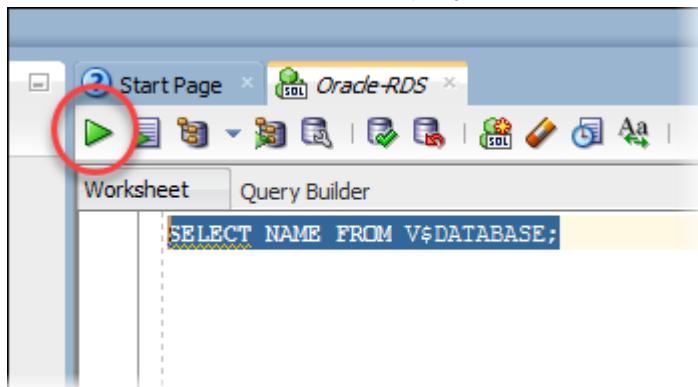
The completed dialog box should look similar to the following.



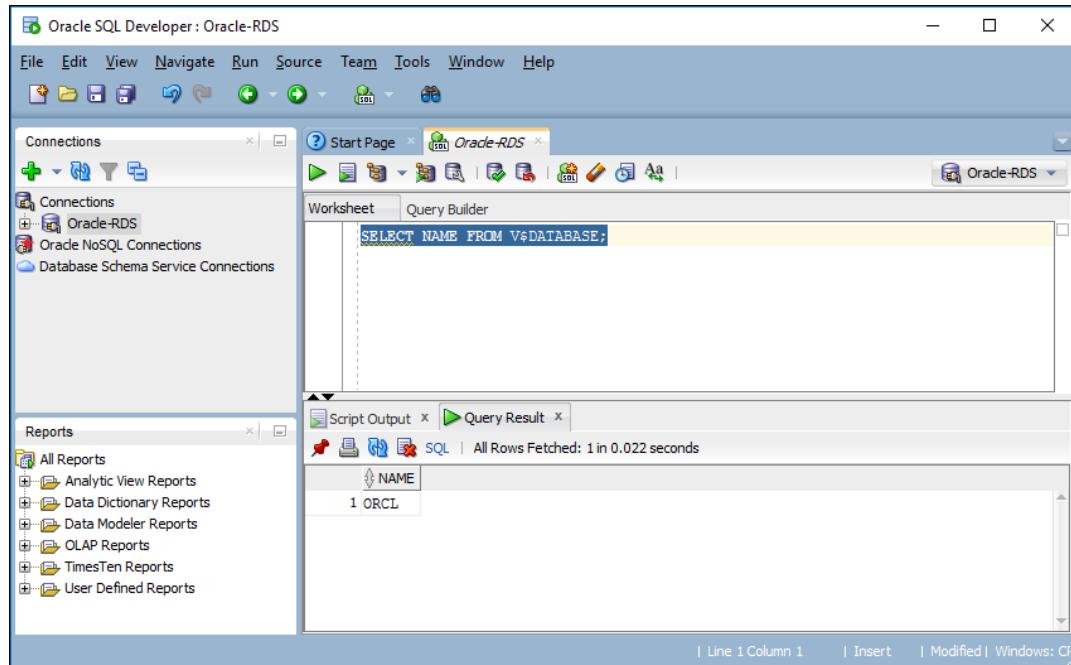
4. Choose **Connect**.
5. You can now start creating your own databases and running queries against your DB instance and databases as usual. To run a test query against your DB instance, do the following:
 - a. In the **Worksheet** tab for your connection, enter the following SQL query.

```
SELECT NAME FROM V$DATABASE;
```

- b. Choose the **execute** icon to run the query.



SQL Developer returns the database name.



Connecting to your DB instance using SQL*Plus

You can use a utility like SQL*Plus to connect to an Amazon RDS DB instance running Oracle. To download Oracle Instant Client, which includes a standalone version of SQL*Plus, see [Oracle Instant Client Downloads](#).

To connect to your DB instance, you need its DNS name and port number. For information about finding the DNS name and port number for a DB instance, see [Finding the endpoint of your Oracle DB instance \(p. 1001\)](#).

Example To connect to an Oracle DB instance using SQL*Plus

In the following examples, substitute the user name of your DB instance administrator. Also, substitute the DNS name for your DB instance, and then include the port number and the Oracle SID. The SID value is the name of the DB instance's database that you specified when you created the DB instance, and not the name of the DB instance.

For Linux, macOS, or Unix:

```
sqlplus 'user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))(CONNECT_DATA=(SID=database_name)))'
```

For Windows:

```
sqlplus user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))(CONNECT_DATA=(SID=database_name)))
```

You should see output similar to the following.

```
SQL*Plus: Release 12.1.0.2.0 Production on Mon Aug 21 09:42:20 2017
```

After you enter the password for the user, the SQL prompt appears.

```
SQL>
```

Note

The shorter format connection string (Easy connect or EZCONNECT), such as `sqlplus USER/PASSWORD@LONGER-THAN-63-CHARS-RDS-ENDPOINT-HERE:1521/DATABASE_IDENTIFIER`, might encounter a maximum character limit and should not be used to connect.

Considerations for security groups

For you to connect to your DB instance, it must be associated with a security group that contains the necessary IP addresses and network configuration. Your DB instance might use the default security group. If you assigned a default, nonconfigured security group when you created the DB instance, the firewall prevents connections.

To create a new security group, security group that you create depends on the Amazon EC2 platform for your DB instance. To determine your platform, see [Determining whether you are using the EC2-VPC or EC2-Classic platform \(p. 1718\)](#). In general, if your DB instance is on the *EC2-Classic* platform, you create a DB security group; if your DB instance is on the VPC platform, you create a VPC security group. For information about creating a new security group, see [Controlling access with security groups \(p. 1699\)](#).

After you create the new security group, you modify your DB instance to associate it with the security group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

You can enhance security by using SSL to encrypt connections to your DB instance. For more information, see [Oracle Secure Sockets Layer \(p. 1182\)](#).

Considerations for process architecture

Server processes handle user connections to an Oracle DB instance. By default, the Oracle DB instance uses dedicated server processes. With dedicated server processes, each server process services only one user process. You can optionally configure shared server processes. With shared server processes, each server process can service multiple user processes.

You might consider using shared server processes when a high number of user sessions are using too much memory on the server. You might also consider shared server processes when sessions connect and disconnect very often, resulting in performance issues. There are also disadvantages to using shared server processes. For example, they can strain CPU resources, and they are more complicated to configure and administer.

For more information about dedicated and shared server processes, see [About dedicated and shared server processes](#) in the Oracle documentation. For more information about configuring shared server processes on an RDS for Oracle DB instance, see [How do I configure Amazon RDS for Oracle database to work with shared servers?](#) in the Knowledge Center.

Troubleshooting connections to your Oracle DB instance

The following are issues you might encounter when you try to connect to your Oracle DB instance.

Issue	Troubleshooting suggestions
Unable to connect to your DB instance.	For a newly created DB instance, the DB instance has a status of creating until it is ready to use. When the state changes to available , you can connect to the DB instance. Depending on the DB instance class and the amount of storage, it can take up to 20 minutes before the new DB instance is available.
Unable to connect to your DB instance.	If you can't send or receive communications over the port that you specified when you created the DB instance, you can't connect to the DB instance. Check with your network administrator to verify that the port you specified for your DB instance allows inbound and outbound communication.
Unable to connect to your DB instance.	<p>The access rules enforced by your local firewall and the IP addresses you authorized to access your DB instance in the security group for the DB instance might not match. The problem is most likely the inbound or outbound rules on your firewall.</p> <p>You can add or edit an inbound rule in the security group. For Source, choose My IP. This allows access to the DB instance from the IP address detected in your browser. For more information, see Amazon Virtual Private Cloud VPCs and Amazon RDS (p. 1718).</p> <p>For more information about security groups, see Controlling access with security groups (p. 1699).</p> <p>To walk through the process of setting up rules for your security group, see Tutorial: Create an Amazon VPC for use with a DB instance (p. 1737).</p>
Connect failed because target host or object does not exist – Oracle, Error: ORA-12545	<p>Make sure that you specified the server name and port number correctly. For Server name, enter the DNS name from the console.</p> <p>For information about finding the DNS name and port number for a DB instance, see Finding the endpoint of your Oracle DB instance (p. 1001).</p>
Invalid username/password; logon denied – Oracle, Error: ORA-01017	You were able to reach the DB instance, but the connection was refused. This is usually caused by providing an incorrect user name or password. Verify the user name and password, and then retry.

For more information on connection issues, see [Can't connect to Amazon RDS DB instance \(p. 1746\)](#).

Modifying connection properties using sqlnet.ora parameters

The sqlnet.ora file includes parameters that configure Oracle Net features on Oracle database servers and clients. Using the parameters in the sqlnet.ora file, you can modify properties for connections in and out of the database.

For more information about why you might set sqlnet.ora parameters, see [Configuring profile parameters in the Oracle documentation](#).

Setting sqlnet.ora parameters

Amazon RDS for Oracle parameter groups include a subset of sqlnet.ora parameters. You set them in the same way that you set other Oracle parameters. The `sqlnetora.` prefix identifies which parameters are sqlnet.ora parameters. For example, in an Oracle parameter group in Amazon RDS, the `default_sdu_size` sqlnet.ora parameter is `sqlnetora.default_sdu_size`.

For information about managing parameter groups and setting parameter values, see [Working with DB parameter groups \(p. 228\)](#).

Supported sqlnet.ora parameters

Amazon RDS supports the following sqlnet.ora parameters. Changes to dynamic sqlnet.ora parameters take effect immediately.

Parameter	Valid values	Static/ Dynamic	Description
<code>sqlnetora.default_sdu_size</code>	Oracle 12c – 512 to 2097152	Dynamic	The session data unit (SDU) size, in bytes. The SDU is the amount of data that is put in a buffer and sent across the network at one time.
<code>sqlnetora.diag_adr_enabled</code>	ON, OFF	Dynamic	A value that enables or disables Automatic Diagnostic Repository (ADR) tracing. ON specifies that ADR file tracing is used. OFF specifies that non-ADR file tracing is used.
<code>sqlnetora.recv_buf_size</code>	8192 to 268435456	Dynamic	The buffer space limit for receive operations of sessions, supported by the TCP/IP, TCP/IP with SSL, and SDP protocols.
<code>sqlnetora.send_buf_size</code>	8192 to 268435456	Dynamic	The buffer space limit for send operations of sessions, supported by the TCP/IP, TCP/IP with SSL, and SDP protocols.
<code>sqlnetora.sqlnet.allowed_logon_version_client</code>	8, 9, 10, 11, 12	Dynamic	Minimum authentication protocol version allowed for clients, and servers acting as clients, to establish a connection to Oracle DB instances.
<code>sqlnetora.sqlnet.allowed_logon_version_server</code>	8, 9, 10, 11, 12, 12a	Dynamic	Minimum authentication protocol version allowed to establish a connection to Oracle DB instances.
<code>sqlnetora.sqlnet.expire_time</code>	0 to 1440	Dynamic	Time interval, in minutes, to send a check to verify that client-server connections are active.

Parameter	Valid values	Static/ Dynamic	Description
sqlnetora.sqlnet.inbound_connect_timeout	0 or 10 to 7200	Dynamic	Time, in seconds, for a client to connect with the database server and provide the necessary authentication information.
sqlnetora.sqlnet.outbound_connect_timeout	0 or 10 to 7200	Dynamic	Time, in seconds, for a client to establish an Oracle Net connection to the DB instance.
sqlnetora.sqlnet.recv_timeout	0 or 10 to 7200	Dynamic	Time, in seconds, for a database server to wait for client data after establishing a connection.
sqlnetora.sqlnet.send_timeout	0 or 10 to 7200	Dynamic	Time, in seconds, for a database server to complete a send operation to clients after establishing a connection.
sqlnetora.tcp.connect_timeout	0 or 10 to 7200	Dynamic	Time, in seconds, for a client to establish a TCP connection to the database server.
sqlnetora.trace_level_server	0, 4, 10, 16, OFF, USER, ADMIN, SUPPORT	Dynamic	For non-ADR tracing, turns server tracing on at a specified level or turns it off.

The default value for each supported sqlnet.ora parameter is the Oracle default for the release. For information about default values for Oracle Database 12c, see [Parameters for the sqlnet.ora file](#) in the Oracle Database 12c documentation.

Viewing sqlnet.ora parameters

You can view sqlnet.ora parameters and their settings using the AWS Management Console, the AWS CLI, or a SQL client.

Viewing sqlnet.ora parameters using the console

For information about viewing parameters in a parameter group, see [Working with DB parameter groups \(p. 228\)](#).

In Oracle parameter groups, the `sqlnetora.` prefix identifies which parameters are sqlnet.ora parameters.

Viewing sqlnet.ora parameters using the AWS CLI

To view the sqlnet.ora parameters that were configured in an Oracle parameter group, use the AWS CLI `describe-db-parameters` command.

To view all of the sqlnet.ora parameters for an Oracle DB instance, call the AWS CLI `download-db-log-file-portion` command. Specify the DB instance identifier, the log file name, and the type of output.

Example

The following code lists all of the sqlnet.ora parameters for `mydbinstance`.

For Linux, macOS, or Unix:

```
aws rds download-db-log-file-portion \
--db-instance-identifier mydbinstance \
--log-file-name trace/sqlnet-parameters \
--output text
```

For Windows:

```
aws rds download-db-log-file-portion ^
--db-instance-identifier mydbinstance ^
--log-file-name trace/sqlnet-parameters ^
--output text
```

Viewing sqlnet.ora parameters using a SQL client

After you connect to the Oracle DB instance in a SQL client, the following query lists the sqlnet.ora parameters.

```
SELECT * FROM TABLE
(rdsadmin.rds_file_util.read_text_file(
    p_directory => 'BDUMP',
    p_filename   => 'sqlnet-parameters'));
```

For information about connecting to an Oracle DB instance in a SQL client, see [Connecting to your Oracle DB instance \(p. 1001\)](#).

Securing Oracle DB instance connections

Amazon RDS Oracle supports SSL/TLS encrypted connections and also the Oracle Native Network Encryption (NNE) option to encrypt connections between your application and your Oracle DB instance. For more information about the Oracle Native Network Encryption option, see [Oracle native network encryption \(p. 1176\)](#).

Encrypting client connections with SSL

Secure Sockets Layer (SSL) is an industry-standard protocol for securing network connections between client and server. After SSL version 3.0, the name was changed to Transport Layer Security (TLS), but we still often refer to the protocol as SSL. Amazon RDS supports SSL encryption for Oracle DB instances. Using SSL, you can encrypt a connection between your application client and your Oracle DB instance. SSL support is available in all AWS regions for Oracle.

To enable SSL encryption for an Oracle DB instance, add the Oracle SSL option to the option group associated with the DB instance. Amazon RDS uses a second port, as required by Oracle, for SSL connections. Doing this allows both clear text and SSL-encrypted communication to occur at the same time between a DB instance and an Oracle client. For example, you can use the port with clear text communication to communicate with other resources inside a VPC while using the port with SSL-encrypted communication to communicate with resources outside the VPC.

For more information, see [Oracle Secure Sockets Layer \(p. 1182\)](#).

Note

You can't use both SSL and Oracle native network encryption (NNE) on the same DB instance. Before you can use SSL encryption, you must disable any other connection encryption.

Updating applications to use new SSL/TLS certificates

As of September 19, 2019, Amazon RDS has published new Certificate Authority (CA) certificates for connecting to your RDS DB instances using Secure Socket Layer or Transport Layer Security (SSL/TLS). Following, you can find information about updating your applications to use the new certificates.

This topic can help you to determine whether any client applications use SSL/TLS to connect to your DB instances.

Important

When you change the certificate for an Amazon RDS for Oracle DB instance, only the database listener is restarted. The DB instance isn't restarted. Existing database connections are unaffected, but new connections will encounter errors for a brief period while the listener is restarted.

Note

For client applications that use SSL/TLS to connect to your DB instances, you must update your client application trust stores to include the new CA certificates.

After you update your CA certificates in the client application trust stores, you can rotate the certificates on your DB instances. We strongly recommend testing these procedures in a development or staging environment before implementing them in your production environments.

For more information about certificate rotation, see [Rotating your SSL/TLS certificate \(p. 1636\)](#). For more information about downloading certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#). For information about using SSL/TLS with Oracle DB instances, see [Oracle Secure Sockets Layer \(p. 1182\)](#).

Topics

- [Finding out whether applications connect using SSL \(p. 1011\)](#)
- [Updating your application trust store \(p. 1012\)](#)
- [Example Java code for establishing SSL connections \(p. 1013\)](#)

Finding out whether applications connect using SSL

If your Oracle DB instance uses an option group with the `SSL` option added, you might be using SSL. Check this by following the instructions in [Listing the options and option settings for an option group \(p. 220\)](#). For information about the `SSL` option, see [Oracle Secure Sockets Layer \(p. 1182\)](#).

Check the listener log to determine whether there are SSL connections. The following is sample output in a listener log.

```
date time * (CONNECT_DATA=(CID=(PROGRAM=program)
(HOST=host)(USER=user))(SID=sid)) *
(ADDRESS=(PROTOCOL=tcp)(HOST=host)(PORT=port)) * establish * ORCL * 0
```

When `PROTOCOL` has the value `tcp` for an entry, it shows an SSL connection. However, when `HOST` is `127.0.0.1`, you can ignore the entry. Connections from `127.0.0.1` are a local management agent on the DB instance. These connections aren't external SSL connections. Therefore, you have applications connecting using SSL if you see listener log entries where `PROTOCOL` is `tcp` and `HOST` is *not* `127.0.0.1`.

To check the listener log, you can publish the log to Amazon CloudWatch Logs. For more information, see [Publishing Oracle logs to Amazon CloudWatch Logs \(p. 529\)](#).

Updating your application trust store

You can update the trust store for applications that use SQL*Plus or JDBC for SSL/TLS connections.

Updating your application trust store for SQL*Plus

You can update the trust store for applications that use SQL*Plus for SSL/TLS connections.

Note

When you update the trust store, you can retain older certificates in addition to adding the new certificates.

To update the trust store for SQL*Plus applications

1. Download the 2019 root certificate that works for all AWS Regions and put the file in the `ssl_wallet` directory.

For information about downloading the root certificate, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

2. Run the following command to update the Oracle wallet.

```
prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert  
$ORACLE_HOME/ssl_wallet/rds-ca-2019-root.pem -auto_login_only
```

Replace the file name with the one that you downloaded.

3. Run the following command to confirm that the wallet was updated successfully.

```
prompt>orapki wallet display -wallet $ORACLE_HOME/ssl_wallet
```

Your output should contain the following.

```
Trusted Certificates:  
Subject: CN=Amazon RDS Root 2019 CA,OU=Amazon RDS,O=Amazon Web Services\,  
Inc.,L=Seattle,ST=Washington,C=US
```

Updating your application trust store for JDBC

You can update the trust store for applications that use JDBC for SSL/TLS connections.

To update the trust store for JDBC applications

1. Download the 2019 root certificate that works for all AWS Regions and put the file in the `ssl_wallet` directory.

For information about downloading the root certificate, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

2. Convert the certificate to .der format using the following command.

```
openssl x509 -outform der -in rds-ca-2019-root.pem -out rds-ca-2019-root.der
```

Replace the file name with the one that you downloaded.

3. Import the certificate into the key store using the following command.

```
keytool -import -alias rds-root -keystore clientkeystore -file rds-ca-2019-root.der
```

4. Confirm that the key store was updated successfully.

```
keytool -list -v -keystore clientkeystore.jks
```

Enter the key store password when you are prompted for it.

Your output should contain the following.

```
rds-root,date, trustedCertEntry,  
Certificate fingerprint (SHA1):  
D4:0D:DB:29:E3:75:0D:FF:A6:71:C3:14:0B:BF:5F:47:8D:1C:80:96  
# This fingerprint should match the output from the below command  
openssl x509 -fingerprint -in rds-ca-2019-root.pem -noout
```

Example Java code for establishing SSL connections

The following code example shows how to set up the SSL connection using JDBC.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class OracleSslConnectionTest {
    private static final String DB_SERVER_NAME = "<dns-name-provided-by-amazon-rds>";
    private static final Integer SSL_PORT = "<ssl-option-port-configured-in-option-group>";
    private static final String DB_SID = "<oracle-sid>";
    private static final String DB_USER = "<user name>";
    private static final String DB_PASSWORD = "<password>";
    // This key store has only the prod root ca.
    private static final String KEY_STORE_FILE_PATH = "<file-path-to-keystore>";
    private static final String KEY_STORE_PASS = "<keystore-password>";

    public static void main(String[] args) throws SQLException {
        final Properties properties = new Properties();
        final String connectionString = String.format(
            "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=%d))" +
            "(CONNECT_DATA=(SID=%s)))",
            DB_SERVER_NAME, SSL_PORT, DB_SID);
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);
        properties.put("oracle.jdbc.J2EE13Compliant", "true");
        properties.put("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        properties.put("javax.net.ssl.trustStoreType", "JKS");
        properties.put("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);
        final Connection connection = DriverManager.getConnection(connectionString,
            properties);
        // If no exception, that means handshake has passed, and an SSL connection can be
        opened
    }
}
```

Important

After you have determined that your database connections use SSL/TLS and have updated your application trust store, you can update your database to use the rds-ca-2019 certificates. For instructions, see step 3 in [Updating your CA certificate by modifying your DB instance \(p. 1636\)](#).

Configuring Kerberos authentication for Amazon RDS for Oracle

You can use Kerberos authentication to authenticate users when they connect to your Amazon RDS DB instance running Oracle. In this configuration, your DB instance works with AWS Directory Service for Microsoft Active Directory, also called AWS Managed Microsoft AD. When users authenticate with an Oracle DB instance joined to the trusting domain, authentication requests are forwarded to the directory that you create with AWS Directory Service.

Keeping all of your credentials in the same directory can save you time and effort. You have a centralized place for storing and managing credentials for multiple database instances. A directory can also improve your overall security profile.

Amazon RDS supports Kerberos authentication for Oracle DB instances in the following AWS Regions:

- US East (Ohio)
- US East (N. Virginia)
- US West (N. California)
- US West (Oregon)
- Asia Pacific (Mumbai)
- Asia Pacific (Seoul)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Canada (Central)
- Europe (Frankfurt)
- Europe (Ireland)
- Europe (London)
- Europe (Stockholm)
- South America (São Paulo)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Note

Kerberos authentication isn't supported for DB instance classes that are deprecated for Oracle DB instances. For more information, see [RDS for Oracle instance classes \(p. 992\)](#).

Topics

- [Setting up Kerberos authentication for Oracle DB instances \(p. 1015\)](#)
- [Managing a DB instance in a domain \(p. 1023\)](#)
- [Connecting to Oracle with Kerberos authentication \(p. 1025\)](#)

Setting up Kerberos authentication for Oracle DB instances

Use AWS Directory Service for Microsoft Active Directory, also called AWS Managed Microsoft AD, to set up Kerberos authentication for an Oracle DB instance. To set up Kerberos authentication, complete the following steps:

- [Step 1: Create a directory using the AWS Managed Microsoft AD \(p. 1015\)](#)
- [Step 2: Create a trust \(p. 1018\)](#)
- [Step 3: Create an IAM role for use by Amazon RDS \(p. 1018\)](#)
- [Step 4: Create and configure users \(p. 1019\)](#)
- [Step 5: Enable cross-VPC traffic between the directory and the DB instance \(p. 1020\)](#)
- [Step 6: Create or modify an Oracle DB instance \(p. 1020\)](#)
- [Step 7: Create Kerberos authentication Oracle logins \(p. 1022\)](#)
- [Step 8: Configure an Oracle client \(p. 1022\)](#)

Note

During the setup, RDS creates an Oracle database user named `managed_service_user@example.com` with the `CREATE SESSION` privilege, where `example.com` is your domain name. This user corresponds to the user that Directory Service creates inside your Managed Active Directory. Periodically, RDS uses the credentials provided by the Directory Service to log in to your Oracle database. Afterwards, RDS immediately destroys the ticket cache.

Step 1: Create a directory using the AWS Managed Microsoft AD

AWS Directory Service creates a fully managed Active Directory in the AWS Cloud. When you create an AWS Managed Microsoft AD directory, AWS Directory Service creates two domain controllers and Domain Name System (DNS) servers on your behalf. The directory servers are created in different subnets in a VPC. This redundancy helps make sure that your directory remains accessible even if a failure occurs.

When you create an AWS Managed Microsoft AD directory, AWS Directory Service performs the following tasks on your behalf:

- Sets up an Active Directory within the VPC.
- Creates a directory administrator account with the user name Admin and the specified password. You use this account to manage your directory.

Note

Be sure to save this password. AWS Directory Service doesn't store it. You can reset it, but you can't retrieve it.

- Creates a security group for the directory controllers.

When you launch an AWS Managed Microsoft AD, AWS creates an Organizational Unit (OU) that contains all of your directory's objects. This OU has the NetBIOS name that you typed when you created your directory and is located in the domain root. The domain root is owned and managed by AWS.

The Admin account that was created with your AWS Managed Microsoft AD directory has permissions for the most common administrative activities for your OU:

- Create, update, or delete users
- Add resources to your domain such as file or print servers, and then assign permissions for those resources to users in your OU
- Create additional OUs and containers

- Delegate authority
- Restore deleted objects from the Active Directory Recycle Bin
- Run AD and DNS Windows PowerShell modules on the Active Directory Web Service

The Admin account also has rights to perform the following domain-wide activities:

- Manage DNS configurations (add, remove, or update records, zones, and forwarders)
- View DNS event logs
- View security event logs

To create the directory, use the AWS Management Console, the AWS CLI, or the AWS Directory Service API. Make sure to open the relevant outbound ports on the directory security group so that the directory can communicate with the Oracle DB instance.

To create a directory with AWS Managed Microsoft AD

1. Sign in to the AWS Management Console and open the AWS Directory Service console at <https://console.aws.amazon.com/directoryservicev2/>.
2. In the navigation pane, choose **Directories** and choose **Set up Directory**.
3. Choose **AWS Managed Microsoft AD**. AWS Managed Microsoft AD is the only option that you can currently use with Amazon RDS.
4. Enter the following information:

Directory DNS name

The fully qualified name for the directory, such as **corp.example.com**.

Directory NetBIOS name

The short name for the directory, such as **CORP**.

Directory description

(Optional) A description for the directory.

Admin password

The password for the directory administrator. The directory creation process creates an administrator account with the user name Admin and this password.

The directory administrator password and can't include the word "admin." The password is case-sensitive and must be 8–64 characters in length. It must also contain at least one character from three of the following four categories:

- Lowercase letters (a–z)
- Uppercase letters (A–Z)
- Numbers (0–9)
- Non-alphanumeric characters (~!@#\$%^&*_+=`|\{}{};:"'<>,.?/)

Confirm password

The administrator password retyped.

5. Choose **Next**.
6. Enter the following information in the **Networking** section and then choose **Next**:

VPC

The VPC for the directory. Create the Oracle DB instance in this same VPC.

Subnets

Subnets for the directory servers. The two subnets must be in different Availability Zones.

7. Review the directory information and make any necessary changes. When the information is correct, choose **Create directory**.

Review & create

Review	
Directory type	VPC
Microsoft AD	vpc-8b6b78e9 ([REDACTED])
Directory DNS name	Subnets
corp.example.com	subnet-75128d10 ([REDACTED], us-east-1a) subnet-f51665dd ([REDACTED], us-east-1b)
Directory NetBIOS name	
CORP	
Directory description	
My directory	
Pricing	
Edition	Free trial eligible Learn more
Standard	30-day limited trial
~USD [REDACTED] *	
* Includes two domain controllers, USD [REDACTED] /mo for each additional domain controller.	
Cancel Previous Create directory	

It takes several minutes for the directory to be created. When it has been successfully created, the **Status** value changes to **Active**.

To see information about your directory, choose the directory name in the directory listing. Note the **Directory ID** value because you need this value when you create or modify your Oracle DB instance.

The screenshot shows the 'Directory details' page for a Microsoft AD directory. The 'Directory ID' field, which contains the value 'd-90670a8d36', is circled in red. Other visible fields include 'Directory type' (Microsoft AD), 'VPC' (vpc-6594f31c), 'Status' (Active), 'Edition' (Standard), 'Subnets' (subnet-7d36a227, subnet-a2ab49c6), 'Last updated' (Tuesday, January 7, 2020), 'Availability zones' (us-east-1c, us-east-1d), 'Launch time' (Tuesday, January 7, 2020), 'DNS address' (redacted), and a 'Description' field containing 'My directory'. Below the table, there are tabs for 'Application management' (which is selected), 'Scale & share', 'Networking & security', and 'Maintenance'.

Step 2: Create a trust

If you plan to use AWS Managed Microsoft AD only, move on to [Step 3: Create an IAM role for use by Amazon RDS \(p. 1018\)](#).

To get Kerberos authentication using an on-premises or self-hosted Microsoft Active Directory, create a forest trust or external trust. The trust can be one-way or two-way. For more information about setting up forest trusts using AWS Directory Service, see [When to create a trust relationship](#) in the *AWS Directory Service Administration Guide*.

Step 3: Create an IAM role for use by Amazon RDS

For Amazon RDS to call AWS Directory Service for you, an IAM role that uses the managed IAM policy `AmazonRDSDirectoryServiceAccess` is required. This role allows Amazon RDS to make calls to the AWS Directory Service.

Note

For the role to allow access, the AWS Security Token Service (AWS STS) endpoint must be activated in the correct AWS Region for your AWS account. AWS STS endpoints are active by default in all AWS Regions, and you can use them without any further actions. For more information, see [Activating and deactivating AWS STS in an AWS Region](#) in the *IAM User Guide*.

When a DB instance is created using the AWS Management Console and the console user has the `iam:CreateRole` permission, the console creates this role automatically. In this case, the role name is `rds-directoryservice-kerberos-access-role`. Otherwise, you must create the IAM role

manually. When you create this IAM role, choose **Directory Service**, and attach the AWS managed policy `AmazonRDSDirectoryServiceAccess` to it.

For more information about creating IAM roles for a service, see [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.

Note

The IAM role used for Windows Authentication for RDS for Microsoft SQL Server can't be used for RDS for Oracle.

Optionally, you can create policies with the required permissions instead of using the managed IAM policy `AmazonRDSDirectoryServiceAccess`. In this case, the IAM role must have the following IAM trust policy.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "",  
            "Effect": "Allow",  
            "Principal": {  
                "Service": [  
                    "directoryservice.rds.amazonaws.com",  
                    "rds.amazonaws.com"  
                ]  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

The role must also have the following IAM role policy.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "ds:DescribeDirectories",  
                "ds:AuthorizeApplication",  
                "ds:UnauthorizeApplication",  
                "ds:GetAuthorizedApplicationDetails"  
            ],  
            "Effect": "Allow",  
            "Resource": "*"  
        }  
    ]  
}
```

Step 4: Create and configure users

You can create users with the Active Directory Users and Computers tool, which is one of the Active Directory Domain Services and Active Directory Lightweight Directory Services tools. In this case, *users* are individual people or entities that have access to your directory.

To create users in an AWS Directory Service directory, you must be connected to a Windows-based Amazon EC2 instance that is a member of the AWS Directory Service directory. At the same time, you must be logged in as a user that has privileges to create users. For more information about creating users in your Microsoft Active Directory, see [Manage users and groups in AWS Managed Microsoft AD](#) in the *AWS Directory Service Administration Guide*.

Step 5: Enable cross-VPC traffic between the directory and the DB instance

If you plan to locate the directory and the DB instance in the same VPC, skip this step and move on to [Step 6: Create or modify an Oracle DB instance \(p. 1020\)](#).

If you plan to locate the directory and the DB instance in different AWS accounts or VPCs, configure cross-VPC traffic using VPC peering or [AWS Transit Gateway](#). The following procedure enables traffic between VPCs using VPC peering. Follow the instructions in [What is VPC peering?](#) in the *Amazon Virtual Private Cloud Peering Guide*.

To enable cross-VPC traffic using VPC peering

1. Set up appropriate VPC routing rules to ensure that network traffic can flow both ways.
2. Ensure that the DB instance's security group can receive inbound traffic from the directory's security group. For more information, see [Best practices for AWS Managed Microsoft AD](#) in the *AWS Directory Service Administration Guide*.
3. Ensure that there is no network access control list (ACL) rule to block traffic.

If a different AWS account owns the directory, you must share the directory.

To share the directory between AWS accounts

1. Start sharing the directory with the AWS account that the DB instance will be created in by following the instructions in [Tutorial: Sharing your AWS Managed Microsoft AD directory for seamless EC2 Domain-join](#) in the *AWS Directory Service Administration Guide*.
2. Sign in to the AWS Directory Service console using the account for the DB instance, and ensure that the domain has the SHARED status before proceeding.
3. While signed into the AWS Directory Service console using the account for the DB instance, note the **Directory ID** value. You use this directory ID to join the DB instance to the domain.

Step 6: Create or modify an Oracle DB instance

Create or modify an Oracle DB instance for use with your directory. You can use the console, CLI, or RDS API to associate a DB instance with a directory. You can do this in one of the following ways:

- Create a new Oracle DB instance using the console, the [create-db-instance](#) CLI command, or the [CreateDBInstance](#) RDS API operation.

For instructions, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- Modify an existing Oracle DB instance using the console, the [modify-db-instance](#) CLI command, or the [ModifyDBInstance](#) RDS API operation.

For instructions, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
- Restore an Oracle DB instance from a DB snapshot using the console, the [restore-db-instance-from-db-snapshot](#) CLI command, or the [RestoreDBInstanceFromDBSnapshot](#) RDS API operation.

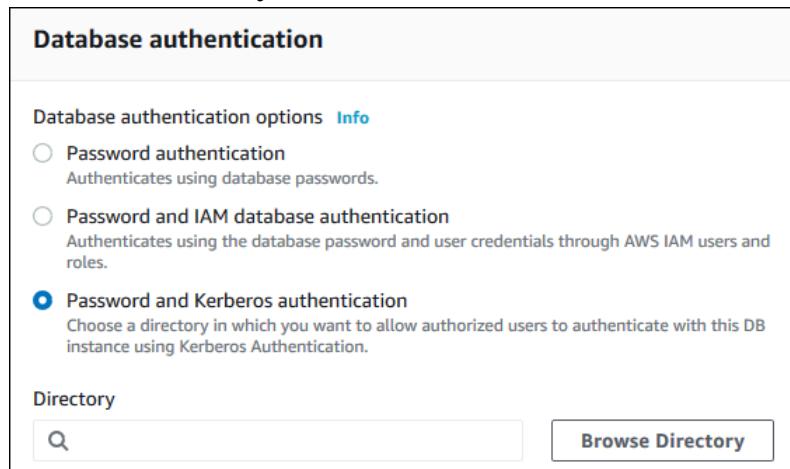
For instructions, see [Restoring from a DB snapshot \(p. 349\)](#).
- Restore an Oracle DB instance to a point-in-time using the console, the [restore-db-instance-to-point-in-time](#) CLI command, or the [RestoreDBInstanceToPointInTime](#) RDS API operation.

For instructions, see [Restoring a DB instance to a specified time \(p. 389\)](#).

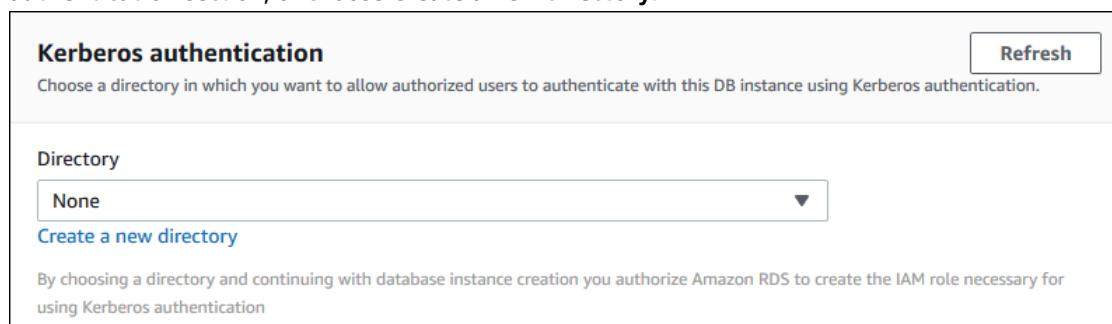
Kerberos authentication is only supported for Oracle DB instances in a VPC. The DB instance can be in the same VPC as the directory, or in a different VPC. When you create or modify the DB instance, do the following:

- Provide the domain identifier (d-* identifier) that was generated when you created your directory.
- Provide the name of the IAM role that you created.
- Ensure that the DB instance security group can receive inbound traffic from the directory security group and send outbound traffic to the directory.

When you use the console to create a DB instance, choose **Password and Kerberos authentication** in the **Database authentication** section. Choose **Browse Directory** and then select the directory, or choose **Create a new directory**.



When you use the console to modify or restore a DB instance, choose the directory in the **Kerberos authentication** section, or choose **Create a new directory**.



When you use the AWS CLI, the following parameters are required for the DB instance to be able to use the directory that you created:

- For the --domain parameter, use the domain identifier ("d-*" identifier) generated when you created the directory.
- For the --domain-iam-role-name parameter, use the role you created that uses the managed IAM policy `AmazonRDSDirectoryServiceAccess`.

For example, the following CLI command modifies a DB instance to use a directory.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--domain d-ID \
--domain-iam-role-name role-name
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--domain d-ID ^
--domain-iam-role-name role-name
```

Important

If you modify a DB instance to enable Kerberos authentication, reboot the DB instance after making the change.

Note

MANAGED_SERVICE_USER is a service account whose name is randomly generated by Directory Service for RDS. During the Kerberos authentication setup, RDS for Oracle creates a user with the same name and assigns it the CREATE SESSION privilege. The Oracle DB user is identified externally as *MANAGED_SERVICE_USER@EXAMPLE.COM*, where *EXAMPLE.COM* is the name of your domain. Periodically, RDS uses the credentials provided by the Directory Service to log in to your Oracle database. Afterward, RDS immediately destroys the ticket cache.

Step 7: Create Kerberos authentication Oracle logins

Use the Amazon RDS master user credentials to connect to the Oracle DB instance as you do any other DB instance. The DB instance is joined to the AWS Managed Microsoft AD domain. Thus, you can provision Oracle logins and users from the Microsoft Active Directory users and groups in your domain. To manage database permissions, you grant and revoke standard Oracle permissions to these logins.

To allow a Microsoft Active Directory user to authenticate with Oracle

1. Connect to the Oracle DB instance using your Amazon RDS master user credentials.
2. Create an externally authenticated user in Oracle database.

In the following example, replace *KRBUSER@CORP.EXAMPLE.COM* with the user name and domain name.

```
CREATE USER "KRBUSER@CORP.EXAMPLE.COM" IDENTIFIED EXTERNALLY;
GRANT CREATE SESSION TO "KRBUSER@CORP.EXAMPLE.COM";
```

Users (both humans and applications) from your domain can now connect to the Oracle DB instance from a domain joined client machine using Kerberos authentication.

Step 8: Configure an Oracle client

To configure an Oracle client, meet the following requirements:

- Create a configuration file named krb5.conf (Linux) or krb5.ini (Windows) to point to the domain. Configure the Oracle client to use this configuration file.
- Verify that traffic can flow between the client host and AWS Directory Service over DNS port 53 and Kerberos ports (88 and 464 for managed AWS Directory Service) over TCP/UDP.
- Verify that traffic can flow between the client host and the DB instance over the database port.

Following is sample content for AWS Managed Microsoft AD.

```
[libdefaults]
default_realm = EXAMPLE.COM
[realms]
EXAMPLE.COM = {
    kdc = example.com
```

```
    admin_server = example.com
}
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

Following is sample content for on-premise Microsoft AD. In your krb5.conf or krb5.ini file, replace *on-prem-ad-server-name* with the name of your on-premises AD server.

```
[libdefaults]
default_realm = ONPREM.COM
[realms]
AWSAD.COM = {
    kdc = awasd.com
    admin_server = awasd.com
}
ONPREM.COM = {
    kdc = on-prem-ad-server-name
    admin_server = on-prem-ad-server-name
}
[domain_realm]
.awasd.com = AWSAD.COM
awsad.com= AWSAD.COM
.onprem.com = ONPREM.COM
onprem.com= ONPREM.COM
```

Note

After you configure your krb5.ini or krb5.conf file, we recommend that you reboot the server.

The following is sample sqlnet.ora content for a SQL*Plus configuration:

```
SQLNET.AUTHENTICATION_SERVICES=(KERBEROS5PRE,KERBEROS5)
SQLNET.KERBEROS5_CONF=path_to_krb5.conf_file
```

For an example of a SQL Developer configuration, see [Document 1609359.1](#) from Oracle Support.

Managing a DB instance in a domain

You can use the console, the CLI, or the RDS API to manage your DB instance and its relationship with your Microsoft Active Directory. For example, you can associate a Microsoft Active Directory to enable Kerberos authentication. You can also disassociate a Microsoft Active Directory to disable Kerberos authentication. You can also move a DB instance to be externally authenticated by one Microsoft Active Directory to another.

For example, using the CLI, you can do the following:

- To reattempt enabling Kerberos authentication for a failed membership, use the [modify-db-instance](#) CLI command and specify the current membership's directory ID for the --domain option.
- To disable Kerberos authentication on a DB instance, use the [modify-db-instance](#) CLI command and specify none for the --domain option.
- To move a DB instance from one domain to another, use the [modify-db-instance](#) CLI command and specify the domain identifier of the new domain for the --domain option.

Viewing the status of domain membership

After you create or modify your DB instance, the DB instance becomes a member of the domain. You can view the status of the domain membership for the DB instance in the console or by running the [describe-db-instances](#) CLI command. The status of the DB instance can be one of the following:

- **kerberos-enabled** – The DB instance has Kerberos authentication enabled.
- **enabling-kerberos** – AWS is in the process of enabling Kerberos authentication on this DB instance.
- **pending-enable-kerberos** – Enabling Kerberos authentication is pending on this DB instance.
- **pending-maintenance-enable-kerberos** – AWS will attempt to enable Kerberos authentication on the DB instance during the next scheduled maintenance window.
- **pending-disable-kerberos** – Disabling Kerberos authentication is pending on this DB instance.
- **pending-maintenance-disable-kerberos** – AWS will attempt to disable Kerberos authentication on the DB instance during the next scheduled maintenance window.
- **enable-kerberos-failed** – A configuration problem has prevented AWS from enabling Kerberos authentication on the DB instance. Correct the configuration problem before reissuing the command to modify the DB instance.
- **disabling-kerberos** – AWS is in the process of disabling Kerberos authentication on this DB instance.

A request to enable Kerberos authentication can fail because of a network connectivity issue or an incorrect IAM role. If the attempt to enable Kerberos authentication fails when you create or modify a DB instance, make sure that you're using the correct IAM role. Then modify the DB instance to join the domain.

Note

Only Kerberos authentication with Amazon RDS for Oracle sends traffic to the domain's DNS servers. All other DNS requests are treated as outbound network access on your DB instances running Oracle. For more information about outbound network access with Amazon RDS for Oracle, see [Setting up a custom DNS server \(p. 1045\)](#).

Force-rotating Kerberos keys

A secret key is shared between AWS Managed Microsoft AD and Amazon RDS for Oracle DB instance. This key is rotated automatically every 45 days. You can use the following Amazon RDS procedure to force the rotation of this key.

```
SELECT rdsadmin.rdsadmin_kerberos_auth_tasks.rotate_kerberos_keytab AS TASK_ID FROM DUAL;
```

Note

In a read replica configuration, this procedure is available only on the source DB instance and not on the read replica.

The `SELECT` statement returns the ID of the task in a `VARCHAR2` data type. You can view the status of an ongoing task in a bddump file. The bddump files are located in the `/rdsdbdata/log/trace` directory. Each bddump file name is in the following format.

```
dbtask-task-id.log
```

You can view the result by displaying the task's output file.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP', 'dbtask-task-id.log'));
```

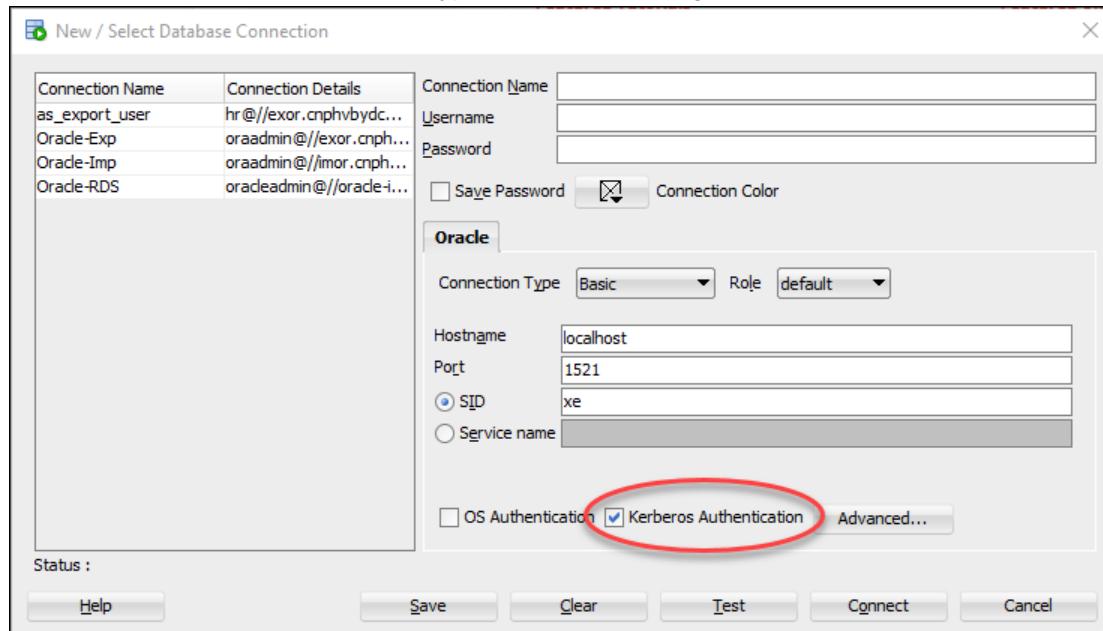
Replace `task-id` with the task ID returned by the procedure.

Note

Tasks are executed asynchronously.

Connecting to Oracle with Kerberos authentication

This section assumes that you have set up your Oracle client as described in [Step 8: Configure an Oracle client \(p. 1022\)](#). To connect to the Oracle DB with Kerberos authentication, log in using the Kerberos authentication type. For example, after launching Oracle SQL Developer, choose **Kerberos Authentication** as the authentication type, as shown following.



To connect to Oracle with Kerberos authentication with SQL*Plus:

1. At a command prompt, run the following command:

```
kinit username
```

Replace *username* with the user name and, at the prompt, enter the password stored in the Microsoft Active Directory for the user.

2. Open SQL*Plus and connect using the DNS name and port number for the Oracle DB instance.

For more information about connecting to an Oracle DB instance in SQL*Plus, see [Connecting to your DB instance using SQL*Plus \(p. 1005\)](#).

Configuring outbound network access on your Oracle DB instance

Amazon RDS supports outbound network access on your Oracle DB instances. To connect your instance to the network, you can use the following PL/SQL packages:

- **UTL_HTTP** – makes HTTP callouts from SQL and PL/SQL. You can use it to access data on the Internet over HTTP. For more information, see [UTL_HTTP](#) in the Oracle documentation.
- **UTL_TCP** – provides TCP/IP client-side access functionality in PL/SQL. This package is useful to PL/SQL applications that use Internet protocols and email. For more information, see [UTL_TCP](#) in the Oracle documentation.

- **UTL_SMTP** – provides interfaces to the SMTP commands that enable a client to dispatch emails to an SMTP server. For more information, see [UTL_SMTP](#) in the Oracle documentation.

Note the following about working with outbound network access:

- Outbound network access with **UTL_HTTP**, **UTL_TCP**, and **UTL_SMTP** is supported only for Oracle DB instances in a VPC. To determine whether or not your DB instance is in a VPC, see [Determining whether you are using the EC2-VPC or EC2-Classic platform \(p. 1718\)](#). To move a DB instance not in a VPC into a VPC, see [Moving a DB instance not in a VPC into a VPC \(p. 1735\)](#).
- To use SMTP with the **UTL_MAIL** option, see [Oracle UTL_MAIL \(p. 1206\)](#).
- The Domain Name Server (DNS) name of the remote host can be any of the following:
 - Publicly resolvable.
 - The endpoint of an Amazon RDS DB instance.
 - Resolvable through a custom DNS server. For more information, see [Setting up a custom DNS server \(p. 1045\)](#).
 - The private DNS name of an Amazon EC2 instance in the same VPC or a peered VPC. In this case, make sure that the name is resolvable through a custom DNS server. Alternatively, to use the DNS provided by Amazon, you can enable the `enableDnsSupport` attribute in the VPC settings and enable DNS resolution support for the VPC peering connection. For more information, see [DNS support in your VPC](#) and [Modifying your VPC peering connection](#).

To connect securely to remote SSL/TLS resources, you can create and upload customized Oracle wallets. By using the Amazon S3 integration with Amazon RDS for Oracle feature, you can download a wallet from Amazon S3 into Oracle DB instances. For information about Amazon S3 integration for Oracle, see [Amazon S3 integration \(p. 1127\)](#).

To create a wallet for accessing an HTTP address over UTL_HTTP

- 1. Obtain the certificate for `Amazon Root CA 1` from the [Amazon trust services repository](#).
 2. Create a new wallet and add the following certificate:

```
orapki wallet create -wallet . -auto_login_only
orapki wallet add -wallet . -trusted_cert -cert AmazonRootCA1.pem -auto_login_only
orapki wallet display -wallet .
```
 3. Upload the wallet to your Amazon S3 bucket.
 4. Complete the prerequisites for Amazon S3 integration with Oracle, and add the `S3_INTEGRATION` option to your Oracle DB instance. Ensure that the IAM role for the option has access to the Amazon S3 bucket you are using.

For more information, see [Amazon S3 integration \(p. 1127\)](#).

5. Connect to the DB instance, and create a directory in the database to hold the wallet. The following example creates a directory called `SSL_WALLET`:

```
EXEC rdsadmin.rdsadmin_util.create_directory('SSL_WALLET');
```

6. Download the wallet from your Amazon S3 bucket to the Oracle DB instance.

The following example downloads a wallet to the DB instance directory `SSL_WALLET`:

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
    p_bucket_name      => 'bucket_name',
```

```
p_s3_prefix      => 'wallet_name',
p_directory_name => 'SSL_WALLET')
AS TASK_ID FROM DUAL;
```

Replace `bucket_name` with the name of the bucket you are using, and replace `wallet_name` with the name of the wallet.

7. Set this wallet for utl_http transactions by running the following procedure:

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
BEGIN
  select directory_path into l_wallet_path from all_directories
  where upper(directory_name)='SSL_WALLET';

  utl_http.set_wallet('file:' || l_wallet_path);
END;
```

8. Access the URL from above over SSL/TLS. The following example accesses `https://status.aws.amazon.com/robots.txt`.

```
SELECT utl_http.request('https://status.aws.amazon.com/robots.txt') AS ROBOTS_TXT FROM
DUAL;

ROBOTS_TXT
-----
User-agent: *
Allow: /
```

Note

The specific certificates that are required for your wallet vary by service. For AWS services, the certificates can typically be found in the [Amazon trust services repository](#).

You can use a similar setup to send emails through UTL_SMTP over SSL/TLS (including [Amazon Simple Email Service](#)).

You can establish database links between Oracle DB instances over an SSL/TLS endpoint if the Oracle SSL option is configured for each instance. No further configuration is required. For more information, see [Oracle Secure Sockets Layer \(p. 1182\)](#).

Administering your Oracle DB instance

Following are the common management tasks you perform with an Amazon RDS DB instance. Some tasks are the same for all RDS DB instances. Other tasks are specific to RDS for Oracle.

The following tasks are common to all RDS databases, but Oracle has special considerations. For example, connect to an Oracle Database using the Oracle clients SQL*Plus and SQL Developer.

Task area	Relevant documentation
Instance Classes, Storage, and PIOPS If you are creating a production instance, learn how instance classes, storage types, and Provisioned IOPS work in Amazon RDS.	RDS for Oracle instance classes (p. 992) Amazon RDS storage types (p. 40)
Multi-AZ Deployments A production DB instance should use Multi-AZ deployments. Multi-AZ deployments provide increased availability, data durability, and fault tolerance for DB instances.	High availability (Multi-AZ) for Amazon RDS (p. 53)
Amazon Virtual Private Cloud (VPC) If your AWS account has a default VPC, then your DB instance is automatically created inside the default VPC. If your account doesn't have a default VPC, and you want the DB instance in a VPC, create the VPC and subnet groups before you create the instance.	Determining whether you are using the EC2-VPC or EC2-Classic platform (p. 1718) Working with a DB instance in a VPC (p. 1727)
Security Groups By default, DB instances use a firewall that prevents access. Make sure you create a security group with the correct IP addresses and network configuration to access the DB instance. The security group you create depends on which Amazon EC2 platform your DB instance is on, and whether you will access your DB instance from an Amazon EC2 instance. In general, if your DB instance is on the EC2-Classic platform, you should create a DB security group. Also, if your instance is on the EC2-VPC platform, you should create a VPC security group.	Determining whether you are using the EC2-VPC or EC2-Classic platform (p. 1718) Controlling access with security groups (p. 1699)
Parameter Groups If your DB instance is going to require specific database parameters, you should create a parameter group before you create the DB instance.	Working with DB parameter groups (p. 228)
Option Groups If your DB instance will require specific database options, you should create an option group before you create the DB instance.	Adding options to Oracle DB instances (p. 1126)
Connecting to Your DB Instance After creating a security group and associating it to a DB instance, you can connect to the DB instance using any standard SQL client application such as Oracle SQL*Plus.	Connecting to your Oracle DB instance (p. 1001)

Task area	Relevant documentation
Backup and Restore You can configure your DB instance to take automated backups, or take manual snapshots, and then restore instances from the backups or snapshots.	Backing up and restoring an Amazon RDS DB instance (p. 327)
Monitoring You can monitor an Oracle DB instance by using CloudWatch Amazon RDS metrics, events, and enhanced monitoring.	Viewing DB instance metrics (p. 548) Viewing Amazon RDS events (p. 503)
Log Files You can access the log files for your Oracle DB instance.	Accessing Amazon RDS database log files (p. 504)

Following, you can find a description for Amazon RDS–specific implementations of common DBA tasks for RDS Oracle. To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances, and restricts access to certain system procedures and tables that require advanced privileges. In many of the tasks, you run the `rdsadmin` package, which is an Amazon RDS–specific tool that enables you to administer your database.

The following are common DBA tasks for DB instances running Oracle:

- [System tasks \(p. 1036\)](#)

Disconnecting a session (p. 1036)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.disconnect</code> Oracle method: <code>alter system disconnect session</code>
Terminating a session (p. 1037)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.kill</code> Oracle method: <code>alter system kill session</code>
Canceling a SQL statement in a session (p. 1038)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.cancel</code> Oracle method: <code>alter system cancel sql</code>
Enabling and disabling restricted sessions (p. 1038)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.restricted_session</code> Oracle method: <code>alter system enable restricted session</code>
Flushing the shared pool (p. 1039)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.flush_shared_pool</code> Oracle method: <code>alter system flush shared_pool</code>
Flushing the buffer cache (p. 1039)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.flush_buffer_cache</code> Oracle method: <code>alter system flush buffer_cache</code>
Granting SELECT or EXECUTE privileges to SYS objects (p. 1039)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.grant_sys_object</code> Oracle method: <code>grant</code>

Revoking SELECT or EXECUTE privileges on SYS objects (p. 1041)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.revoke_sys_object</code> Oracle method: <code>REVOKE</code>
Granting privileges to non-master users (p. 1042)	Amazon RDS method: <code>GRANT</code> Oracle method: <code>GRANT</code>
Creating custom functions to verify passwords (p. 1042)	Amazon RDS method: <code>rdsadmin.rdsadmin_password_verify.create_verify_function</code> Amazon RDS method: <code>rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn</code>
Setting up a custom DNS server (p. 1045)	—
Listing allowed system diagnostic events (p. 1047)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.list_allowed_system_events</code> Oracle method: —
Setting system diagnostic events (p. 1047)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.set_allowed_system_events</code> Oracle method: <code>ALTER SYSTEM SET EVENTS '<i>set_event_clause</i>'</code>
Listing system diagnostic events that are set (p. 1048)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.list_set_system_events</code> Oracle method: <code>ALTER SESSION SET EVENTS 'IMMEDIATE EVENTDUMP(SYSTEM)'</code>
Unsetting system diagnostic events (p. 1049)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.unset_system_event</code> Oracle method: <code>ALTER SYSTEM SET EVENTS '<i>unset_event_clause</i>'</code>

- [Database tasks \(p. 1049\)](#)

Changing the global name of a database (p. 1050)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.rename_global_name</code> Oracle method: <code>ALTER DATABASE RENAME</code>
Creating and sizing tablespaces (p. 1050)	Amazon RDS method: <code>CREATE TABLESPACE</code> Oracle method: <code>ALTER DATABASE</code>
Setting the default tablespace (p. 1051)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.alter_default_tablespace</code> Oracle method: <code>ALTER DATABASE DEFAULT TABLESPACE</code>

Setting the default temporary tablespace (p. 1051)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.alter_default_temp_tablespace</code> Oracle method: <code>alter database default temporary tablespace</code>
Checkpointing a database (p. 1051)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.checkpoint</code> Oracle method: <code>alter system checkpoint</code>
Setting distributed recovery (p. 1051)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.enable_distr_recovery</code> Oracle method: <code>alter system enable distributed recovery</code>
Setting the database time zone (p. 1052)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.alter_db_time_zone</code> Oracle method: <code>alter database set time_zone</code>
Working with Oracle external tables (p. 1052)	—
Generating performance reports with Automatic Workload Repository (AWR) (p. 1053)	Amazon RDS method: <code>rdsadmin.rdsadmin_diagnostic_util procedures</code> Oracle method: <code>dbms_workload_repository package</code>
Adjusting database links for use with DB instances in a VPC (p. 1057)	—
Setting the default edition for a DB instance (p. 1057)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.alter_default_edition</code> Oracle method: <code>alter database default edition</code>
Enabling auditing for the SYS.AUD\$ table (p. 1057)	Amazon RDS method: <code>rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table</code> Oracle method: <code>audit</code>
Disabling auditing for the SYS.AUD\$ table (p. 1058)	Amazon RDS method: <code>rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table</code> Oracle method: <code>noaudit</code>
Cleaning up interrupted online index builds (p. 1058)	Amazon RDS method: <code>rdsadmin.rdsadmin_dbms_repair.online_index_clean</code> Oracle method: <code>dbms_repair.online_index_clean</code>
Skipping corrupt blocks (p. 1059)	Amazon RDS method: Several <code>rdsadmin.rdsadmin_dbms_repair</code> procedures Oracle method: <code>dbms_repair package</code>

Resizing the temporary tablespace in a read replica (p. 1061)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.resize_temp_tablespace</code> or <code>rdsadmin.rdsadmin_util.resize_tempfile</code> procedure Oracle method: —
Purging the recycle bin (p. 1062)	Amazon RDS method: <code>exec rdsadmin.rdsadmin_util.purge_dba_recyclebin</code> Oracle method: <code>purge dba_recyclebin</code>

- [Log tasks \(p. 1062\)](#)

Setting force logging (p. 1063)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.force_logging</code> Oracle method: <code>alter database force logging</code>
Setting supplemental logging (p. 1063)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.alter_supple</code> Oracle method: <code>alter database add supplemental log</code>
Switching online log files (p. 1064)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.switch_logfi</code> Oracle method: <code>alter system switch logfile</code>
Adding online redo logs (p. 1064)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.add_logfile</code>
Dropping online redo logs (p. 1065)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.drop_logfile</code>
Resizing online redo logs (p. 1065)	—
Retaining archived redo logs (p. 1067)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.set_configur</code>
Accessing transaction logs (p. 1068)	Amazon RDS method: <code>rdsadmin.rdsadmin_master_util.creat</code> Amazon RDS method: <code>rdsadmin.rdsadmin_master_util.creat</code>

- [RMAN tasks \(p. 1069\)](#)

Validating DB instance files (p. 1072)	Amazon RDS method: <code>rdsadmin_rman_util.procedure</code> Oracle method: <code>RMAN VALIDATE</code>
Enabling and disabling block change tracking (p. 1075)	Amazon RDS method: <code>rdsadmin_rman_util.procedure</code> Oracle method: <code>ALTER DATABASE</code>
Crosschecking archived redo logs (p. 1076)	Amazon RDS method: <code>rdsadmin_rman_util.crosscheck_archi</code> Oracle method: <code>RMAN BACKUP</code>
Backing up archived redo logs (p. 1077)	Amazon RDS method: <code>rdsadmin_rman_util.procedure</code> Oracle method: <code>RMAN BACKUP</code>
Performing a full database backup (p. 1082)	Amazon RDS method: <code>rdsadmin_rman_util.backup_database</code> Oracle method: <code>RMAN BACKUP</code>
Performing an incremental database backup (p. 1083)	Amazon RDS method: <code>rdsadmin_rman_util.backup_database</code> Oracle method: <code>RMAN BACKUP</code>
Performing a tablespace backup (p. 1084)	Amazon RDS method: <code>rdsadmin_rman_util.backup_database</code> Oracle method: <code>RMAN BACKUP</code>

- [Oracle Scheduler tasks \(p. 1085\)](#)

Modifying DBMS_SCHEDULER jobs (p. 1086)	Amazon RDS method: <code>dbms_scheduler.set_attribute</code> Oracle method: <code>dbms_scheduler.set_attribute</code>
Setting the time zone for Oracle Scheduler jobs (p. 1086)	Amazon RDS method: <code>dbms_scheduler.set_attribute</code> Oracle method: <code>dbms_scheduler.set_attribute</code>

Disabling SYS-owned Oracle Scheduler jobs (p. 1087)	Amazon RDS method: <code>rdsadmin.rdsadmin_dbms_scheduler.disable</code> Oracle method: <code>dbms_scheduler.disable</code>
Enabling SYS-owned Oracle Scheduler jobs (p. 1087)	Amazon RDS method: <code>rdsadmin.rdsadmin_dbms_scheduler.enable</code> Oracle method: <code>dbms_scheduler.enable</code>
Modifying the repeat interval for jobs of CALENDAR type (p. 1087)	Amazon RDS method: <code>rdsadmin.rdsadmin_dbms_scheduler.set_attribute</code> Oracle method: <code>dbms_scheduler.set_attribute</code>
Modifying the repeat interval for jobs of NAMED type (p. 1088)	Amazon RDS method: <code>rdsadmin.rdsadmin_dbms_scheduler.set_attribute</code> Oracle method: <code>dbms_scheduler.set_attribute</code>

- [Diagnostic tasks \(p. 1089\)](#)

Listing incidents (p. 1090)	Amazon RDS method: <code>rdsadmin.rdsadmin_adrci_util.list_incident</code> Oracle method: ADRCI command <code>show incident</code>
Listing problems (p. 1091)	Amazon RDS method: <code>rdsadmin.rdsadmin_adrci_util.list_problem</code> Oracle method: ADRCI command <code>show problem</code>
Creating incident packages (p. 1093)	Amazon RDS method: <code>rdsadmin.rdsadmin_adrci_util.create_ipspack</code> Oracle method: ADRCI command <code>ips create package</code>
Showing trace files (p. 1094)	Amazon RDS method: <code>rdsadmin.rdsadmin_adrci_util.show_tracefile</code> Oracle method: ADRCI command <code>show tracefile</code>

- [Other tasks \(p. 1095\)](#)

Creating and dropping directories in the main data storage space (p. 1095)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.create_directory</code> Oracle method: <code>CREATE DIRECTORY</code> Amazon RDS method: <code>rdsadmin.rdsadmin_util.drop_directory</code> Oracle method: <code>DROP DIRECTORY</code>
Listing files in a DB instance directory (p. 1096)	Amazon RDS method: <code>rdsadmin.rds_file_util.listdir</code> Oracle method: —
Reading files in a DB instance directory (p. 1096)	Amazon RDS method: <code>rdsadmin.rds_file_util.read_text_file</code> Oracle method: —
Accessing Opatch files (p. 1097)	Amazon RDS method: <code>rdsadmin.rds_file_util.read_text_file</code> or <code>rdsadmin.tracefile_listing</code> Oracle method: <code>opatch</code>
Setting parameters for advisor tasks (p. 1099)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.advisor_task</code> Oracle method: Various stored package procedures
Disabling AUTO_STATS_ADVISOR_TASK (p. 1100)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.advisor_task</code> Oracle method: —
Re-enabling AUTO_STATS_ADVISOR_TASK (p. 1101)	Amazon RDS method: <code>rdsadmin.rdsadmin_util.dbms_stats_initialize</code> Oracle method: —
Enabling HugePages for an Oracle DB instance (p. 1101)	Amazon RDS method: <code>use_large_pages</code> RDS parameter Oracle method: <code>use_large_pages</code> initialization parameter

[Enabling extended data types \(p. 1103\)](#)

Amazon RDS method:
`max_string_size` RDS parameter

Oracle method:
`max_string_size` initialization parameter

You can also use Amazon RDS procedures for Amazon S3 integration with Oracle and for running OEM Management Agent database tasks. For more information, see [Amazon S3 integration \(p. 1127\)](#) and [Performing database tasks with the Management Agent \(p. 1160\)](#).

Performing common system tasks for Oracle DB instances

Following, you can find how to perform certain common DBA tasks related to the system on your Amazon RDS DB instances running Oracle. To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances, and restricts access to certain system procedures and tables that require advanced privileges.

Topics

- [Disconnecting a session \(p. 1036\)](#)
- [Terminating a session \(p. 1037\)](#)
- [Canceling a SQL statement in a session \(p. 1038\)](#)
- [Enabling and disabling restricted sessions \(p. 1038\)](#)
- [Flushing the shared pool \(p. 1039\)](#)
- [Flushing the buffer cache \(p. 1039\)](#)
- [Granting SELECT or EXECUTE privileges to SYS objects \(p. 1039\)](#)
- [Revoking SELECT or EXECUTE privileges on SYS objects \(p. 1041\)](#)
- [Granting privileges to non-master users \(p. 1042\)](#)
- [Creating custom functions to verify passwords \(p. 1042\)](#)
- [Setting up a custom DNS server \(p. 1045\)](#)
- [Setting and unsetting system diagnostic events \(p. 1046\)](#)

Disconnecting a session

To disconnect the current session by ending the dedicated server process, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.disconnect`. The `disconnect` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>sid</code>	number	—	Yes	The session identifier.
<code>serial</code>	number	—	Yes	The serial number of the session.

Parameter name	Data type	Default	Required	Description
method	varchar	'IMMEDIATE'	No	Valid values are 'IMMEDIATE' or 'POST_TRANSACTION'.

The following example disconnects a session.

```
begin
    rdsadmin.rdsadmin_util.disconnect(
        sid    => sid,
        serial => serial_number);
end;
/
```

To get the session identifier and the session serial number, query the `V$SESSION` view. The following example gets all sessions for the user `AWSUSER`.

```
select SID, SERIAL#, STATUS from V$SESSION where USERNAME = 'AWSUSER';
```

The database must be open to use this method. For more information about disconnecting a session, see [ALTER SYSTEM](#) in the Oracle documentation.

Terminating a session

To terminate a session, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.kill`. The `kill` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
sid	number	—	Yes	The session identifier.
serial	number	—	Yes	The serial number of the session.
method	varchar	null	No	Valid values are 'IMMEDIATE' or 'PROCESS'.

The following example terminates a session.

```
begin
    rdsadmin.rdsadmin_util.kill(
        sid    => sid,
        serial => serial_number);
end;
/
```

To get the session identifier and the session serial number, query the `V$SESSION` view. The following example gets all sessions for the user `AWSUSER`.

```
SELECT SID, SERIAL#, STATUS FROM V$SESSION WHERE USERNAME = 'AWSUSER';
```

You can specify either `IMMEDIATE` or `PROCESS` as a value for the `method` parameter. By specifying `PROCESS` as the `method` value, you can terminate the processes associated with a session. Do this only if terminating the session using `IMMEDIATE` as the `method` value was unsuccessful.

Canceling a SQL statement in a session

To cancel a SQL statement in a session, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.cancel`.

Note

This procedure is supported for Oracle Database 18c (18.0.0.0) and later.

The `cancel` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>sid</code>	number	—	Yes	The session identifier.
<code>serial</code>	number	—	Yes	The serial number of the session.
<code>sql_id</code>	varchar2	null	No	The SQL identifier of the SQL statement.

The following example cancels a SQL statement in a session.

```
begin
    rdsadmin.rdsadmin_util.cancel(
        sid      => sid,
        serial   => serial_number,
        sql_id   => sql_id);
end;
/
```

To get the session identifier, the session serial number, and the SQL identifier of a SQL statement, query the `V$SESSION` view. The following example gets all sessions and SQL identifiers for the user `AWSUSER`.

```
select SID, SERIAL#, SQL_ID, STATUS from V$SESSION where USERNAME = 'AWSUSER';
```

Enabling and disabling restricted sessions

To enable and disable restricted sessions, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.restricted_session`. The `restricted_session` procedure has the following parameters.

Parameter name	Data type	Default	Yes	Description
<code>p_enable</code>	boolean	true	No	Set to <code>true</code> to enable restricted sessions, <code>false</code> to disable restricted sessions.

The following example shows how to enable and disable restricted sessions.

```
/* Verify that the database is currently unrestricted. */
```

```
SELECT LOGINS FROM V$INSTANCE;

LOGINS
-----
ALLOWED

/* Enable restricted sessions */

exec rdsadmin.rdsadmin_util.restricted_session(p_enable => true);

/* Verify that the database is now restricted. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
-----
RESTRICTED

/* Disable restricted sessions */

exec rdsadmin.rdsadmin_util.restricted_session(p_enable => false);

/* Verify that the database is now unrestricted again. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
-----
ALLOWED
```

Flushing the shared pool

To flush the shared pool, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.flush_shared_pool`. The `flush_shared_pool` procedure has no parameters.

The following example flushes the shared pool.

```
exec rdsadmin.rdsadmin_util.flush_shared_pool;
```

Flushing the buffer cache

To flush the buffer cache, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.flush_buffer_cache`. The `flush_buffer_cache` procedure has no parameters.

The following example flushes the buffer cache.

```
exec rdsadmin.rdsadmin_util.flush_buffer_cache;
```

Granting SELECT or EXECUTE privileges to SYS objects

Usually you transfer privileges by using roles, which can contain many objects. To grant privileges to a single object, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.grant_sys_object`. The procedure grants only privileges that the master user has already been granted through a role or direct grant.

The `grant_sys_object` procedure has the following parameters.

Important

For all parameter values, use uppercase unless you created the user with a case-sensitive identifier. For example, if you run `CREATE USER myuser` or `CREATE USER MYUSER`, the data dictionary stores `MYUSER`. However, if you use double quotes in `CREATE USER "MyUser"`, the data dictionary stores `MyUser`.

Parameter name	Data type	Default	Required	Description
<code>p_obj_name</code>	<code>varchar2</code>	—	Yes	The name of the object to grant privileges for. The object can be a directory, function, package, procedure, sequence, table, or view. Object names must be spelled exactly as they appear in <code>DBA_OBJECTS</code> . Most system objects are defined in uppercase, so we recommend that you try that first.
<code>p_grantee</code>	<code>varchar2</code>	—	Yes	The name of the object to grant privileges to. The object can be a schema or a role.
<code>p_privilege</code>	<code>varchar2</code>	<code>null</code>	Yes	—
<code>p_grant_option</code>	<code>boolean</code>	<code>false</code>	No	Set to <code>true</code> to use the <code>with grant option</code> . The <code>p_grant_option</code> parameter is supported for 12.1.0.2.v4 and later, all 12.2.0.1 versions, all 18.0.0.0 versions, and all 19.0.0 versions.

The following example grants select privileges on an object named `V_$SESSION` to a user named `USER1`.

```
begin
    rdsadmin.rdsadmin_util.grant_sys_object(
        p_obj_name  => 'V_$SESSION',
        p_grantee   => 'USER1',
        p_privilege => 'SELECT');
end;
/
```

The following example grants select privileges on an object named `V_$SESSION` to a user named `USER1` with the grant option.

```
begin
    rdsadmin.rdsadmin_util.grant_sys_object(
        p_obj_name      => 'V_$SESSION',
        p_grantee       => 'USER1',
        p_privilege     => 'SELECT',
        p_grant_option => true);
```

```
end;
/
```

To be able to grant privileges on an object, your account must have those privileges granted to it directly with the grant option, or via a role granted using with admin option. In the most common case, you may want to grant SELECT on a DBA view that has been granted to the SELECT_CATALOG_ROLE role. If that role isn't already directly granted to your user using with admin option, then you can't transfer the privilege. If you have the DBA privilege, then you can grant the role directly to another user.

The following example grants the SELECT_CATALOG_ROLE and EXECUTE_CATALOG_ROLE to USER1. Since the with admin option is used, USER1 can now grant access to SYS objects that have been granted to SELECT_CATALOG_ROLE.

```
GRANT SELECT_CATALOG_ROLE TO USER1 WITH ADMIN OPTION;
GRANT EXECUTE_CATALOG_ROLE to USER1 WITH ADMIN OPTION;
```

Objects already granted to PUBLIC do not need to be re-granted. If you use the grant_sys_object procedure to re-grant access, the procedure call succeeds.

Revoking SELECT or EXECUTE privileges on SYS objects

To revoke privileges on a single object, use the Amazon RDS procedure rdsadmin.rdsadmin_util.revoke_sys_object. The procedure only revokes privileges that the master account has already been granted through a role or direct grant.

The revoke_sys_object procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
p_obj_name	varchar2	—	Yes	The name of the object to revoke privileges for. The object can be a directory, function, package, procedure, sequence, table, or view. Object names must be spelled exactly as they appear in DBA_OBJECTS. Most system objects are defined in upper case, so we recommend you try that first.
p_revoker	varchar2	—	Yes	The name of the object to revoke privileges for. The object can be a schema or a role.
p_privilege	varchar2	null	Yes	—

The following example revokes select privileges on an object named V_\$SESSION from a user named USER1.

```
begin
    rdsadmin.rdsadmin_util.revoke_sys_object(
        p_obj_name  => 'V_$SESSION',
        p_revoker   => 'USER1',
        p_privilege => 'SELECT');
end;
```

```
end;  
/
```

Granting privileges to non-master users

You can grant select privileges for many objects in the `SYS` schema by using the `SELECT_CATALOG_ROLE` role. The `SELECT_CATALOG_ROLE` role gives users `SELECT` privileges on data dictionary views. The following example grants the role `SELECT_CATALOG_ROLE` to a user named `user1`.

```
GRANT SELECT_CATALOG_ROLE TO user1;
```

You can grant `EXECUTE` privileges for many objects in the `SYS` schema by using the `EXECUTE_CATALOG_ROLE` role. The `EXECUTE_CATALOG_ROLE` role gives users `EXECUTE` privileges for packages and procedures in the data dictionary. The following example grants the role `EXECUTE_CATALOG_ROLE` to a user named `user1`.

```
GRANT EXECUTE_CATALOG_ROLE TO user1;
```

The following example gets the permissions that the roles `SELECT_CATALOG_ROLE` and `EXECUTE_CATALOG_ROLE` allow.

```
SELECT *  
  FROM ROLE_TAB_PRIVS  
 WHERE ROLE IN ('SELECT_CATALOG_ROLE', 'EXECUTE_CATALOG_ROLE')  
 ORDER BY ROLE, TABLE_NAME ASC;
```

The following example creates a non-master user named `user1`, grants the `CREATE SESSION` privilege, and grants the `SELECT` privilege on a database named `sh.sales`.

```
CREATE USER user1 IDENTIFIED BY PASSWORD;  
GRANT CREATE SESSION TO user1;  
GRANT SELECT ON sh.sales TO user1;
```

Creating custom functions to verify passwords

You can create a custom password verification function in two ways. If you want to use standard verification logic, and to store your function in the `SYS` schema, use the `create_verify_function` procedure. If you want to use custom verification logic, or you don't want to store your function in the `SYS` schema, use the `create_passthrough_verify_fcn` procedure.

The `create_verify_function` procedure

The `create_verify_function` procedure is supported for version 12.1.0.2.v5 and later of Oracle Database 12c Release 1 (12.1), all Oracle Database 12c Release 2 (12.2.0.1) versions, all Oracle Database 18c versions, and all Oracle Database 19c versions.

You can create a custom function to verify passwords by using the Amazon RDS procedure `rdsadmin.rdsadmin_password_verify.create_verify_function`. The `create_verify_function` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>p_verify_function_name</code>	varchar2	—	Yes	The name for your custom function. This function is

Parameter name	Data type	Default	Required	Description
				created for you in the SYS schema. You assign this function to user profiles.
p_min_length	number	8	No	The minimum number of characters required.
p_max_length	number	256	No	The maximum number of characters allowed.
p_min_letters	number	1	No	The minimum number of letters required.
p_min_uppercase	number	0	No	The minimum number of uppercase letters required.
p_min_lowercase	number	0	No	The minimum number of lowercase letters required.
p_min_digits	number	1	No	The minimum number of digits required.
p_min_special	number	0	No	The minimum number of special characters required.
p_min_different_chars	number	3	No	The minimum number of different characters required between the old and new password.
p_disallow_username	boolean	true	No	Set to true to disallow the user name in the password.
p_disallow_reverse	boolean	true	No	Set to true to disallow the reverse of the user name in the password.
p_disallow_db_name	boolean	true	No	Set to true to disallow the database or server name in the password.
p_disallow_simple_string	boolean	true	No	Set to true to disallow simple strings as the password.
p_disallow_whitespace	boolean	false	No	Set to true to disallow white space characters in the password.
p_disallow_at_sign	boolean	false	No	Set to true to disallow the @ character in the password.

You can create multiple password verification functions.

There are restrictions on the name of your custom function. Your custom function can't have the same name as an existing system object. The name can be no more than 30 characters long. Also, the name must include one of the following strings: PASSWORD, VERIFY, COMPLEXITY, ENFORCE, or STRENGTH.

The following example creates a function named `CUSTOM_PASSWORD_FUNCTION`. The function requires that a password has at least 12 characters, 2 uppercase characters, 1 digit, and 1 special character, and that the password disallows the @ character.

```
begin
    rdsadmin.rdsadmin_password_verify.create_verify_function(
        p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
        p_min_length          => 12,
        p_min_uppercase       => 2,
        p_min_digits          => 1,
        p_min_special         => 1,
        p_disallow_at_sign   => true);
end;
/
```

To see the text of your verification function, query `DBA_SOURCE`. The following example gets the text of a custom password function named `CUSTOM_PASSWORD_FUNCTION`.

```
COL TEXT FORMAT a150

SELECT TEXT
  FROM DBA_SOURCE
 WHERE OWNER = 'SYS'
   AND NAME = 'CUSTOM_PASSWORD_FUNCTION'
 ORDER BY LINE;
```

To associate your verification function with a user profile, use `alter profile`. The following example associates a verification function with the `DEFAULT` user profile.

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

To see what user profiles are associated with what verification functions, query `DBA_PROFILES`. The following example gets the profiles that are associated with the custom verification function named `CUSTOM_PASSWORD_FUNCTION`.

```
SELECT * FROM DBA_PROFILES WHERE RESOURCE_NAME = 'PASSWORD' AND LIMIT =
  'CUSTOM_PASSWORD_FUNCTION';

PROFILE          RESOURCE_NAME          RESOURCE  LIMIT
-----          -----
-----          -----          -----
DEFAULT          PASSWORD_VERIFY_FUNCTION      PASSWORD
CUSTOM_PASSWORD_FUNCTION
```

The following example gets all profiles and the password verification functions that they are associated with.

```
SELECT * FROM DBA_PROFILES WHERE RESOURCE_NAME = 'PASSWORD_VERIFY_FUNCTION';

PROFILE          RESOURCE_NAME          RESOURCE  LIMIT
-----          -----
-----          -----          -----
DEFAULT          PASSWORD_VERIFY_FUNCTION      PASSWORD
CUSTOM_PASSWORD_FUNCTION
RDSADMIN          PASSWORD_VERIFY_FUNCTION      PASSWORD  NULL
```

The `create_passthrough_verify_fcn` procedure

The `create_passthrough_verify_fcn` procedure is supported for version 12.1.0.2.v7 and later of Oracle Database 12c Release 1 (12.1), all Oracle Database 12c Release 2 (12.2) versions, all Oracle Database 18c versions, and all Oracle Database 19c versions.

You can create a custom function to verify passwords by using the Amazon RDS procedure `rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn`. The `create_passthrough_verify_fcn` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>p_verify_function_name</code>	varchar2	—	Yes	The name for your custom verification function. This is a wrapper function that is created for you in the SYS schema, and it doesn't contain any verification logic. You assign this function to user profiles.
<code>p_target_owner</code>	varchar2	—	Yes	The schema owner for your custom verification function.
<code>p_target_function_name</code>	varchar2	—	Yes	The name of your existing custom function that contains the verification logic. Your custom function must return a boolean. Your function should return <code>true</code> if the password is valid and <code>false</code> if the password is invalid.

The following example creates a password verification function that uses the logic from the function named `PASSWORD_LOGIC_EXTRA_STRONG`.

```
begin
    rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn(
        p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
        p_target_owner         => 'TEST_USER',
        p_target_function_name => 'PASSWORD_LOGIC_EXTRA_STRONG');
end;
/
```

To associate the verification function with a user profile, use `alter profile`. The following example associates the verification function with the `DEFAULT` user profile.

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

Setting up a custom DNS server

Amazon RDS supports outbound network access on your DB instances running Oracle. For more information about outbound network access, including prerequisites, see [Configuring outbound network access on your Oracle DB instance \(p. 1025\)](#).

Amazon RDS Oracle allows Domain Name Service (DNS) resolution from a custom DNS server owned by the customer. You can resolve only fully qualified domain names from your Amazon RDS DB instance through your custom DNS server.

After you set up your custom DNS name server, it takes up to 30 minutes to propagate the changes to your DB instance. After the changes are propagated to your DB instance, all outbound network traffic requiring a DNS lookup queries your DNS server over port 53.

To set up a custom DNS server for your Amazon RDS for Oracle DB instance, do the following:

- From the DHCP options set attached to your virtual private cloud (VPC), set the `domain-name-servers` option to the IP address of your DNS name server. For more information, see [DHCP options sets](#).

Note

The `domain-name-servers` option accepts up to four values, but your Amazon RDS DB instance uses only the first value.

- Ensure that your DNS server can resolve all lookup queries, including public DNS names, Amazon EC2 private DNS names, and customer-specific DNS names. If the outbound network traffic contains any DNS lookups that your DNS server can't handle, your DNS server must have appropriate upstream DNS providers configured.
- Configure your DNS server to produce User Datagram Protocol (UDP) responses of 512 bytes or less.
- Configure your DNS server to produce Transmission Control Protocol (TCP) responses of 1024 bytes or less.
- Configure your DNS server to allow inbound traffic from your Amazon RDS DB instances over port 53. If your DNS server is in an Amazon VPC, the VPC must have a security group that contains inbound rules that permit UDP and TCP traffic on port 53. If your DNS server is not in an Amazon VPC, it must have appropriate firewall allow-listing to permit UDP and TCP inbound traffic on port 53.

For more information, see [Security groups for your VPC](#) and [Adding and removing rules](#).

- Configure the VPC of your Amazon RDS DB instance to allow outbound traffic over port 53. Your VPC must have a security group that contains outbound rules that allow UDP and TCP traffic on port 53.

For more information, see [Security groups for your VPC](#) and [Adding and removing rules](#).

- The routing path between the Amazon RDS DB instance and the DNS server has to be configured correctly to allow DNS traffic.
 - If the Amazon RDS DB instance and the DNS server are not in the same VPC, a peering connection has to be set up between them. For more information, see [What is VPC peering?](#)

Setting and unsetting system diagnostic events

To set and unset diagnostic events at the session level, you can use the Oracle SQL statement `ALTER SESSION SET EVENTS`. However, to set events at the system level you can't use Oracle SQL. Instead, use the system event procedures in the `rdsadmin.rdsadmin_util` package. The system event procedures are available in the following engine versions:

- [19.0.0.0.ru-2020-10.rur-2020-10.r1 \(p. 1266\)](#) or higher 19c versions
- [18.0.0.0.ru-2020-10.rur-2020-10.r1 \(p. 1298\)](#) or higher 18c versions
- [12.2.0.1.ru-2020-10.rur-2020-10.r1 \(p. 1326\)](#) or higher 12.2.0.1 versions
- [12.1.0.2.V22 \(p. 1364\)](#) or higher 12.1 versions

Important

Internally, the `rdsadmin.rdsadmin_util` package sets events by using the `ALTER SYSTEM SET EVENTS` statement. This `ALTER SYSTEM` statement isn't documented in the Oracle Database documentation. Some system diagnostic events can generate large amounts of

tracing information, cause contention, or affect database availability. We recommend that you test specific diagnostic events in your nonproduction database, and only set events in your production database under guidance of Oracle Support.

Listing allowed system diagnostic events

To list the system events that you can set, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.list_allowed_system_events`. This procedure accepts no parameters.

The following example lists all system events that you can set.

```
SET SERVEROUTPUT ON
EXEC rdsadmin.rdsadmin_util.list_allowed_system_events;
```

The following sample output lists event numbers and their descriptions. Use the Amazon RDS procedures `set_system_event` to set these events and `unset_system_event` to unset them.

```
604 - error occurred at recursive SQL level
942 - table or view does not exist
1401 - inserted value too large for column
1403 - no data found
1410 - invalid ROWID
1422 - exact fetch returns more than requested number of rows
1426 - numeric overflow
1427 - single-row subquery returns more than one row
1476 - divisor is equal to zero
1483 - invalid length for DATE or NUMBER bind variable
1489 - result of string concatenation is too long
1652 - unable to extend temp segment by  in tablespace
1858 - a non-numeric character was found where a numeric was expected
4031 - unable to allocate  bytes of shared memory ("","","","","")
6502 - PL/SQL: numeric or value error
10027 - Specify Deadlock Trace Information to be Dumped
10046 - enable SQL statement timing
10053 - CBO Enable optimizer trace
10173 - Dynamic Sampling time-out error
10442 - enable trace of kst for ORA-01555 diagnostics
12008 - error in materialized view refresh path
12012 - error on auto execute of job
12504 - TNS:listener was not given the SERVICE_NAME in CONNECT_DATA
14400 - inserted partition key does not map to any partition
31693 - Table data object failed to load/unload and is being skipped due to error:
```

Note

The list of the allowed system events can change over time. To make sure that you have the most recent list of eligible events, use `rdsadmin.rdsadmin_util.list_allowed_system_events`.

Setting system diagnostic events

To set a system event, use the Amazon RDS procedure

`rdsadmin.rdsadmin_util.set_system_event`. You can only set events listed in the output of `rdsadmin.rdsadmin_util.list_allowed_system_events`. The `set_system_event` procedure accepts the following parameters.

Parameter name	Data type	Default	Required	Description
<code>p_event</code>	number	—	Yes	The system event number. The value must

Parameter name	Data type	Default	Required	Description
				be one of the event numbers reported by <code>list_allowed_system_events</code> .
p_level	number	—	Yes	The event level. See the Oracle Database documentation or Oracle Support for descriptions of different level values.

The procedure `set_system_event` constructs and runs the required `ALTER SYSTEM SET EVENTS` statements according to the following principles:

- The event type (`context` or `errorstack`) is determined automatically.
- A statement in the form `ALTER SYSTEM SET EVENTS 'event LEVEL event_level'` sets the context events. This notation is equivalent to `ALTER SYSTEM SET EVENTS 'event TRACE NAME CONTEXT FOREVER, LEVEL event_level'`.
- A statement in the form `ALTER SYSTEM SET EVENTS 'event ERRORSTACK (event_level)'` sets the error stack events. This notation is equivalent to `ALTER SYSTEM SET EVENTS 'event TRACE NAME ERRORSTACK LEVEL event_level'`.

The following example sets event 942 at level 3, and event 10442 at level 10. Sample output is included.

```

SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.set_system_event(942,3);
Setting system event 942 with: alter system set events '942 errorstack (3)'

PL/SQL procedure successfully completed.

SQL> EXEC rdsadmin.rdsadmin_util.set_system_event(10442,10);
Setting system event 10442 with: alter system set events '10442 level 10'

PL/SQL procedure successfully completed.

```

Listing system diagnostic events that are set

To list the system events that are currently set, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.list_set_system_events`. This procedure reports only events set at system level by `set_system_event`.

The following example lists the active system events.

```

SET SERVEROUTPUT ON
EXEC rdsadmin.rdsadmin_util.list_set_system_events;

```

The following sample output shows the list of events, the event type, the level at which the events are currently set, and the time when the event was set.

```

942 errorstack (3) - set at 2020-11-03 11:42:27
10442 level 10 - set at 2020-11-03 11:42:41

PL/SQL procedure successfully completed.

```

Unsetting system diagnostic events

To unset a system event, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.unset_system_event`. You can only unset events listed in the output of `rdsadmin.rdsadmin_util.list_allowed_system_events`. The `unset_system_event` procedure accepts the following parameter.

Parameter name	Data type	Default	Required	Description
<code>p_event</code>	number	—	Yes	The system event number. The value must be one of the event numbers reported by <code>list_allowed_system_events</code> .

The following example unsets events 942 and 10442. Sample output is included.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.unset_system_event(942);
Unsetting system event 942 with: alter system set events '942 off'

PL/SQL procedure successfully completed.

SQL> EXEC rdsadmin.rdsadmin_util.unset_system_event(10442);
Unsetting system event 10442 with: alter system set events '10442 off'

PL/SQL procedure successfully completed.
```

Performing common database tasks for Oracle DB instances

Following, you can find how to perform certain common DBA tasks related to databases on your Amazon RDS DB instances running Oracle. To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances. Amazon RDS also restricts access to some system procedures and tables that require advanced privileges.

Topics

- [Changing the global name of a database \(p. 1050\)](#)
- [Creating and sizing tablespaces \(p. 1050\)](#)
- [Setting the default tablespace \(p. 1051\)](#)
- [Setting the default temporary tablespace \(p. 1051\)](#)
- [Checkpointing a database \(p. 1051\)](#)
- [Setting distributed recovery \(p. 1051\)](#)
- [Setting the database time zone \(p. 1052\)](#)
- [Working with Oracle external tables \(p. 1052\)](#)
- [Generating performance reports with Automatic Workload Repository \(AWR\) \(p. 1053\)](#)
- [Adjusting database links for use with DB instances in a VPC \(p. 1057\)](#)
- [Setting the default edition for a DB instance \(p. 1057\)](#)
- [Enabling auditing for the SYS.AUD\\$ table \(p. 1057\)](#)
- [Disabling auditing for the SYS.AUD\\$ table \(p. 1058\)](#)
- [Cleaning up interrupted online index builds \(p. 1058\)](#)

- [Skipping corrupt blocks \(p. 1059\)](#)
- [Resizing the temporary tablespace in a read replica \(p. 1061\)](#)
- [Purging the recycle bin \(p. 1062\)](#)

Changing the global name of a database

To change the global name of a database, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.rename_global_name`. The `rename_global_name` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>p_new_global_name</code>	varchar2	—	Yes	The new global name for the database.

The database must be open for the name change to occur. For more information about changing the global name of a database, see [ALTER DATABASE](#) in the Oracle documentation.

The following example changes the global name of a database to `new_global_name`.

```
exec rdsadmin.rdsadmin_util.rename_global_name(p_new_global_name => 'new_global_name');
```

Creating and sizing tablespaces

Amazon RDS only supports Oracle Managed Files (OMF) for data files, log files, and control files. When you create data files and log files, you can't specify the physical file names.

By default, tablespaces are created with auto-extend enabled, and no maximum size. Because of these default settings, tablespaces can grow to consume all allocated storage. We recommend that you specify an appropriate maximum size on permanent and temporary tablespaces, and that you carefully monitor space usage.

The following example creates a tablespace named `users2` with a starting size of 1 gigabyte and a maximum size of 10 gigabytes:

```
CREATE TABLESPACE users2 DATAFILE SIZE 1G AUTOEXTEND ON MAXSIZE 10G;
```

The following example creates temporary tablespace named `temp01`:

```
CREATE TEMPORARY TABLESPACE temp01;
```

We recommend that you don't use smallfile tablespaces because you can't resize smallfile tablespaces with Amazon RDS for Oracle. However, you can add a datafile to a smallfile tablespace.

You can resize a bigfile tablespace by using `ALTER TABLESPACE`. You can specify the size in kilobytes (K), megabytes (M), gigabytes (G), or terabytes (T).

The following example resizes a bigfile tablespace named `users2` to 200 MB.

```
ALTER TABLESPACE users2 RESIZE 200M;
```

The following example adds an additional datafile to a smallfile tablespace named `users2`.

```
ALTER TABLESPACE users2 ADD DATAFILE SIZE 100000M AUTOEXTEND ON NEXT 250m
MAXSIZE UNLIMITED;
```

Setting the default tablespace

To set the default tablespace, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.alter_default_tablespace`. The `alter_default_tablespace` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>tablespace_name</code>	<code>varchar</code>	—	Yes	The name of the default tablespace.

The following example sets the default tablespace to `users2`:

```
EXEC rdsadmin.rdsadmin_util.alter_default_tablespace(tablespace_name => 'users2');
```

Setting the default temporary tablespace

To set the default temporary tablespace, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.alter_default_temp_tablespace`. The `alter_default_temp_tablespace` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>tablespace_name</code>	<code>varchar</code>	—	Yes	The name of the default temporary tablespace.

The following example sets the default temporary tablespace to `temp01`.

```
EXEC rdsadmin.rdsadmin_util.alter_default_temp_tablespace(tablespace_name => 'temp01');
```

Checkpointing a database

To checkpoint the database, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.checkpoint`. The `checkpoint` procedure has no parameters.

The following example checkpoints the database.

```
EXEC rdsadmin.rdsadmin_util.checkpoint;
```

Setting distributed recovery

To set distributed recovery, use the Amazon RDS procedures `rdsadmin.rdsadmin_util.enable_distr_recovery` and `disable_distr_recovery`. The procedures have no parameters.

The following example enables distributed recovery.

```
EXEC rdsadmin.rdsadmin_util.enable_distr_recovery;
```

The following example disables distributed recovery.

```
EXEC rdsadmin.rdsadmin_util.disable_distr_recovery;
```

Setting the database time zone

You can set the time zone of your Amazon RDS Oracle database in the following ways:

- The `Timezone` option

The `Timezone` option changes the time zone at the host level and affects all date columns and values such as `SYSDATE`. For more information, see [Oracle time zone \(p. 1201\)](#).

- The Amazon RDS procedure `rdsadmin.rdsadmin_util.alter_db_time_zone`

The `alter_db_time_zone` procedure changes the time zone for only certain data types, and doesn't change `SYSDATE`. There are additional restrictions on setting the time zone listed in the [Oracle documentation](#).

Note

You can also set the default time zone for Oracle Scheduler. For more information, see [Setting the time zone for Oracle Scheduler jobs \(p. 1086\)](#).

The `alter_db_time_zone` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>p_new_tz</code>	<code>varchar2</code>	—	Yes	The new time zone as a named region or an absolute offset from Coordinated Universal Time (UTC). Valid offsets range from -12:00 to +14:00.

The following example changes the time zone to UTC plus three hours.

```
EXEC rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => '+3:00');
```

The following example changes the time zone to the Africa/Algiers time zone.

```
EXEC rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => 'Africa/Algiers');
```

After you alter the time zone by using the `alter_db_time_zone` procedure, reboot your DB instance for the change to take effect. For more information, see [Rebooting a DB instance \(p. 276\)](#). For information about upgrading time zones, see [Time zone considerations \(p. 1213\)](#).

Working with Oracle external tables

Oracle external tables are tables with data that is not in the database. Instead, the data is in external files that the database can access. By using external tables, you can access data without loading it into the database. For more information about external tables, see [Managing external tables](#) in the Oracle documentation.

With Amazon RDS, you can store external table files in directory objects. You can create a directory object, or you can use one that is predefined in the Oracle database, such as the `DATA_PUMP_DIR`

directory. For information about creating directory objects, see [Creating and dropping directories in the main data storage space \(p. 1095\)](#). You can query the ALL_DIRECTORIES view to list the directory objects for your Amazon RDS Oracle DB instance.

Note

Directory objects point to the main data storage space (Amazon EBS volume) used by your instance. The space used—along with data files, redo logs, audit, trace, and other files—counts against allocated storage.

You can move an external data file from one Oracle database to another by using the [DBMS_FILE_TRANSFER](#) package or the [UTL_FILE](#) package. The external data file is moved from a directory on the source database to the specified directory on the destination database. For information about using DBMS_FILE_TRANSFER, see [Importing using Oracle Data Pump \(p. 1106\)](#).

After you move the external data file, you can create an external table with it. The following example creates an external table that uses the emp_xt_file1.txt file in the USER_DIR1 directory.

```
CREATE TABLE emp_xt (
    emp_id      NUMBER,
    first_name  VARCHAR2(50),
    last_name   VARCHAR2(50),
    user_name   VARCHAR2(20)
)
ORGANIZATION EXTERNAL (
    TYPE ORACLE_LOADER
    DEFAULT DIRECTORY USER_DIR1
    ACCESS PARAMETERS (
        RECORDS DELIMITED BY NEWLINE
        FIELDS TERMINATED BY ','
        MISSING FIELD VALUES ARE NULL
        (emp_id,first_name,last_name,user_name)
    )
    LOCATION ('emp_xt_file1.txt')
)
PARALLEL
REJECT LIMIT UNLIMITED;
```

Suppose that you want to move data that is in an Amazon RDS Oracle DB instance into an external data file. In this case, you can populate the external data file by creating an external table and selecting the data from the table in the database. For example, the following SQL statement creates the orders_xt external table by querying the orders table in the database.

```
CREATE TABLE orders_xt
    ORGANIZATION EXTERNAL
    (
        TYPE ORACLE_DATAPUMP
        DEFAULT DIRECTORY DATA_PUMP_DIR
        LOCATION ('orders_xt.dmp')
    )
    AS SELECT * FROM orders;
```

In this example, the data is populated in the orders_xt.dmp file in the DATA_PUMP_DIR directory.

Generating performance reports with Automatic Workload Repository (AWR)

To gather performance data and generate reports, Oracle recommends Automatic Workload Repository (AWR). AWR requires Oracle Database Enterprise Edition and a license for the Diagnostics and Tuning packs. To enable AWR, set the CONTROL_MANAGEMENT_PACK_ACCESS initialization parameter to either DIAGNOSTIC or DIAGNOSTIC+TUNING.

Working with AWR reports in RDS

To generate AWR reports, you can run scripts such as `awrrpt.sql`. These scripts are installed on the database host server. In Amazon RDS, you don't have direct access to the host. However, you can get copies of SQL scripts from another installation of Oracle Database.

You can also use AWR by running procedures in the `SYS.DBMS_WORKLOAD_REPOSITORY` PL/SQL package. You can use this package to manage baselines and snapshots, and also to display ASH and AWR reports. For example, to generate an AWR report in text format run the `DBMS_WORKLOAD_REPOSITORY.AWR_REPORT_TEXT` procedure. However, you can't reach these AWR reports from the AWS Management Console.

When working with AWR, we recommend using the `rdsadmin.rdsadmin_diagnostic_util` procedures. You can use these procedures to generate the following:

- AWR reports
- Active Session History (ASH) reports
- Automatic Database Diagnostic Monitor (ADDM) reports
- Oracle Data Pump Export dump files of AWR data

The `rdsadmin_diagnostic_util` procedures save the reports to the DB instance file system. You can access these reports from the console. You can also access reports using the `rdsadmin.rds_file_util` procedures, and you can access reports that are copied to Amazon S3 using the S3 Integration option. For more information, see [Reading files in a DB instance directory \(p. 1096\)](#) and [Amazon S3 integration \(p. 1127\)](#).

You can use the `rdsadmin_diagnostic_util` procedures in the following Amazon RDS for Oracle DB engine versions:

- 12.1.0.2.v20 or higher 12.1 versions
- 12.2.0.1.ru-2020-04.rur-2020-04.r1 or higher 12.2 versions
- 18.0.0.0.ru-2020-04.rur-2020-04.r1 or higher 18c versions
- 19.0.0.0.ru-2020-04.rur-2020-04.r1 or higher 19c versions

Common parameters for the diagnostic utility package

You typically use the following parameters when managing AWR and ADDM with the `rdsadmin_diagnostic_util` package.

Parameter	Data type	Default	Requires	Description
<code>begin_snap_id</code>	NUMBER	—	Yes	The ID of the beginning snapshot.
<code>end_snap_id</code>	NUMBER	—	Yes	The ID of the ending snapshot.
<code>dump_directory</code>	VARCHAR	BDUMP	No	The directory to write the report or export file to. If you specify a nondefault directory, the user that runs the <code>rdsadmin_diagnostic_util</code> procedures must have write permissions for the directory.
<code>report_type</code>	VARCHAR	HTML	No	The format of the report. Valid values are <code>TEXT</code> and <code>HTML</code> .
<code>dbid</code>	NUMBER	—	No	A valid database identifier (DBID) shown in the <code>DBA_HIST_DATABASE_INSTANCE</code> view for Oracle. If this

Parameter	Data type	Default	Requires	Description
				parameter is not specified, RDS uses the current DBID, which is shown in the V\$DATABASE.DBID view.

You typically use the following parameters when managing ASH with the rdsadmin_diagnostic_util package.

Parameter	Data type	Default	Requires	Description
begin_time	DATE	—	Yes	The beginning time of the ASH analysis.
end_time	DATE	—	Yes	The ending time of the ASH analysis.
slot_width	NUMBER	0	No	The duration of the slots (in seconds) used in the "Top Activity" section of the ASH report. If this parameter isn't specified, the time interval between begin_time and end_time uses no more than 10 slots.
sid	NUMBER	Null	No	The session ID.
sql_id	VARCHAR2	Null	No	The SQL ID.
wait_class	VARCHAR2	Null	No	The wait class name.
service_hash	NUMBER	Null	No	The service name hash.
module_name	VARCHAR2	Null	No	The module name.
action_name	VARCHAR2	Null	No	The action name.
client_id	VARCHAR2	Null	No	The application-specific ID of the database session.
plsql_entry	VARCHAR2	Null	No	The PL/SQL entry point.

Generating an AWR report

To generate an AWR report, use the `rdsadmin.rdsadmin_diagnostic_util.awr_report` procedure.

The following example generates a AWR report for the snapshot range 101–106. The output text file is named `awrrpt_101_106.txt`. You can access this report from the AWS Management Console.

```
exec rdsadmin.rdsadmin_diagnostic_util.awr_report(101,106,'TEXT');
```

The following example generates an HTML report for the snapshot range 63–65. The output HTML file is named `awrrpt_63_65.html`. The procedure writes the report to the nondefault database directory named `AWR_RPT_DUMP`.

```
exec rdsadmin.rdsadmin_diagnostic_util.awr_report(63,65,'HTML','AWR_RPT_DUMP');
```

Extracting AWR data into a dump file

To extract AWR data into a dump file, use the `rdsadmin.rdsadmin_diagnostic_util.awr_extract` procedure.

The following example extracts the snapshot range 101–106. The output dump file is named `awrextract_101_106.dmp`. You can access this file through the console.

```
exec rdsadmin.rdsadmin_diagnostic_util.awr_extract(101,106);
```

The following example extracts the snapshot range 63–65. The output dump file is named `awrextract_63_65.dmp`. The file is stored in the nondefault database directory named `AWR_RPT_DUMP`.

```
exec rdsadmin.rdsadmin_diagnostic_util.awr_extract(63,65,'AWR_RPT_DUMP');
```

Generating an ADDM report

To generate an ADDM report, use the `rdsadmin.rdsadmin_diagnostic_util.addm_report` procedure.

The following example generates an ADDM report for the snapshot range 101–106. The output text file is named `addmrpt_101_106.txt`. You can access the report through the console.

```
exec rdsadmin.rdsadmin_diagnostic_util.addm_report(101,106);
```

The following example generates an ADDM report for the snapshot range 63–65. The output text file is named `addmrpt_63_65.txt`. The file is stored in the nondefault database directory named `ADDM_RPT_DUMP`.

```
exec rdsadmin.rdsadmin_diagnostic_util.addm_report(63,65,'ADDM_RPT_DUMP');
```

Generating an ASH report

To generate an ASH report, use the `rdsadmin.rdsadmin_diagnostic_util.ash_report` procedure.

The following example generates an ASH report that includes the data from 14 minutes ago until the current time. The name of the output file uses the format `ashrptend_time.txt`, where `begin_time` and `end_time` use the format `YYYYMMDDHH24MISS`. You can access the file through the console.

```
BEGIN
    rdsadmin.rdsadmin_diagnostic_util.ash_report(
        begin_time      =>      SYSDATE-14/1440,
        end_time        =>      SYSDATE,
        report_type    =>      'TEXT');
END;
/
```

The following example generates an ASH report that includes the data from November 18, 2019, at 6:07 PM through November 18, 2019, at 6:15 PM. The name of the output HTML report is `ashrpt_20190918180700_20190918181500.html`. The report is stored in the nondefault database directory named `AWR_RPT_DUMP`.

```
BEGIN
    rdsadmin.rdsadmin_diagnostic_util.ash_report(
        begin_time      =>      TO_DATE('2019-09-18 18:07:00', 'YYYY-MM-DD HH24:MI:SS'),
        end_time        =>      TO_DATE('2019-09-18 18:15:00', 'YYYY-MM-DD HH24:MI:SS'),
        report_type    =>      'html',
        dump_directory =>      'AWR_RPT_DUMP');
END;
```

/

Accessing AWR reports from the console or CLI

To access AWR reports or export dump files, you can use the AWS Management Console or AWS CLI. For more information, see [Downloading a database log file \(p. 504\)](#).

Adjusting database links for use with DB instances in a VPC

To use Oracle database links with Amazon RDS DB instances inside the same virtual private cloud (VPC) or peered VPCs, the two DB instances should have a valid route between them. Verify the valid route between the DB instances by using your VPC routing tables and network access control list (ACL).

The security group of each DB instance must allow ingress to and egress from the other DB instance. The inbound and outbound rules can refer to security groups from the same VPC or a peered VPC. For more information, see [Updating your security groups to reference peered VPC security groups](#).

If you have configured a custom DNS server using the DHCP Option Sets in your VPC, your custom DNS server must be able to resolve the name of the database link target. For more information, see [Setting up a custom DNS server \(p. 1045\)](#).

For more information about using database links with Oracle Data Pump, see [Importing using Oracle Data Pump \(p. 1106\)](#).

Setting the default edition for a DB instance

You can redefine database objects in a private environment called an edition. You can use edition-based redefinition to upgrade an application's database objects with minimal downtime.

You can set the default edition of an Amazon RDS Oracle DB instance using the Amazon RDS procedure `rdsadmin.rdsadmin_util.alter_default_edition`.

The following example sets the default edition for the Amazon RDS Oracle DB instance to `RELEASE_V1`.

```
exec rdsadmin.rdsadmin_util.alter_default_edition('RELEASE_V1');
```

The following example sets the default edition for the Amazon RDS Oracle DB instance back to the Oracle default.

```
exec rdsadmin.rdsadmin_util.alter_default_edition('ORACLEDB');
```

For more information about Oracle edition-based redefinition, see [About editions and edition-based redefinition](#) in the Oracle documentation.

Enabling auditing for the SYS.AUD\$ table

To enable auditing on the database audit trail table `SYS.AUD$`, use the Amazon RDS procedure `rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table`. The only supported audit property is `ALL`. You can't audit or not audit individual statements or operations.

Enabling auditing is supported for Oracle DB instances running the following versions:

- 12.1.0.2.v14 and later 12.1 versions
- All 12.2.0.1 versions
- All 18.0.0.0 versions
- All 19.0.0.0 versions

The `audit_all_sys_aud_table` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>p_by_access</code>	boolean	true	No	Set to <code>true</code> to audit BY ACCESS. Set to <code>false</code> to audit BY SESSION.

The following query returns the current audit configuration for `SYS.AUD$` for a database.

```
SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER='SYS' AND OBJECT_NAME='AUD$';
```

The following commands enable audit of ALL on `SYS.AUD$` BY ACCESS.

```
exec rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table;
exec rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => true);
```

The following command enables audit of ALL on `SYS.AUD$` BY SESSION.

```
exec rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => false);
```

For more information, see [AUDIT \(traditional auditing\)](#) in the Oracle documentation.

Disabling auditing for the `SYS.AUD$` table

To disable auditing on the database audit trail table `SYS.AUD$`, use the Amazon RDS procedure `rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table`. This procedure takes no parameters.

The following query returns the current audit configuration for `SYS.AUD$` for a database:

```
SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER='SYS' AND OBJECT_NAME='AUD$';
```

The following command disables audit of ALL on `SYS.AUD$`.

```
exec rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table;
```

For more information, see [NOAUDIT \(traditional auditing\)](#) in the Oracle documentation.

Cleaning up interrupted online index builds

To clean up failed online index builds, use the Amazon RDS procedure `rdsadmin.rdsadmin_dbms_repair.online_index_clean`.

The `online_index_clean` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>object_id</code>	binary_integer	ALL_INDEX_ID	No	The object ID of the index. Typically, you can use the object ID from the ORA-08104 error text.

Parameter name	Data type	Default	Required	Description
wait_for_lock	binary_integer	rdsadmin.rdsadmin_dbms_repair.lock_nowait	\$admin_dbms	<p>Specify <code>rdsadmin.rdsadmin_dbms_repair.lock_nowait</code> to try to get a lock on the underlying object and retry until an internal limit is reached if the lock fails.</p> <p>Specify <code>rdsadmin.rdsadmin_dbms_repair.lock_noretry</code> to try to get a lock on the underlying object but not retry if the lock fails.</p>

The following example cleans up a failed online index build:

```

declare
    is_clean boolean;
begin
    is_clean := rdsadmin.rdsadmin_dbms_repair.online_index_clean(
        object_id      => 1234567890,
        wait_for_lock => rdsadmin.rdsadmin_dbms_repair.lock_nowait
    );
end;
/

```

For more information, see [ONLINE_INDEX_CLEAN function](#) in the Oracle documentation.

Skipping corrupt blocks

To skip corrupt blocks during index and table scans, use the `rdsadmin.rdsadmin_dbms_repair` package.

The following procedures wrap the functionality of the `sys.dbms_repair.admin_table` procedure and take no parameters:

- `rdsadmin.rdsadmin_dbms_repair.create_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.create_orphan_keys_table`
- `rdsadmin.rdsadmin_dbms_repair.drop_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.drop_orphan_keys_table`
- `rdsadmin.rdsadmin_dbms_repair.purge_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.purge_orphan_keys_table`

The following procedures take the same parameters as their counterparts in the `DBMS_REPAIR` package for Oracle databases:

- `rdsadmin.rdsadmin_dbms_repair.check_object`
- `rdsadmin.rdsadmin_dbms_repair.dump_orphan_keys`
- `rdsadmin.rdsadmin_dbms_repair.fix_corrupt_blocks`
- `rdsadmin.rdsadmin_dbms_repair.rebuild_freelists`

- `rdsadmin.rdsadmin_dbms_repair.segment_fix_status`
- `rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks`

For more information about handling database corruption, see [DBMS_REPAIR](#) in the Oracle documentation.

Example Responding to corrupt blocks

This example shows the basic workflow for responding to corrupt blocks. Your steps will depend on the location and nature of your block corruption.

Important

Before attempting to repair corrupt blocks, review the [DBMS_REPAIR](#) documentation carefully.

To skip corrupt blocks during index and table scans

1. Run the following procedures to create repair tables if they don't already exist.

```
exec rdsadmin.rdsadmin_dbms_repair.create_repair_table;
exec rdsadmin.rdsadmin_dbms_repair.create_orphan_keys_table;
```

2. Run the following procedures to check for existing records and purge them if appropriate.

```
SELECT COUNT(*) FROM SYS.REPAIR_TABLE;
SELECT COUNT(*) FROM SYS.ORPHAN_KEY_TABLE;
SELECT COUNT(*) FROM SYS.DBA_REPAIR_TABLE;
SELECT COUNT(*) FROM SYS.DBA_ORPHAN_KEY_TABLE;

exec rdsadmin.rdsadmin_dbms_repair.purge_repair_table;
exec rdsadmin.rdsadmin_dbms_repair.purge_orphan_keys_table;
```

3. Run the following procedure to check for corrupt blocks.

```
SET SERVEROUTPUT ON
DECLARE v_num_corrupt INT;
BEGIN
    v_num_corrupt := 0;
    rdsadmin.rdsadmin_dbms_repair.check_object (
        schema_name => '&corruptionOwner',
        object_name => '&corruptionTable',
        corrupt_count => v_num_corrupt
    );
    dbms_output.put_line('number corrupt: '||to_char(v_num_corrupt));
END;
/

COL CORRUPT_DESCRIPTION FORMAT a30
COL REPAIR_DESCRIPTION FORMAT a30

SELECT OBJECT_NAME, BLOCK_ID, CORRUPT_TYPE, MARKED_CORRUPT,
       CORRUPT_DESCRIPTION, REPAIR_DESCRIPTION
  FROM   SYS.REPAIR_TABLE;

SELECT SKIP_CORRUPT
  FROM   DBA_TABLES
 WHERE  OWNER = '&corruptionOwner'
 AND    TABLE_NAME = '&corruptionTable';
```

4. Use the `skip_corrupt_blocks` procedure to enable or disable corruption skipping for affected tables. Depending on the situation, you may also need to extract data to a new table, and then drop the table containing the corrupt block.

Run the following procedure to enable corruption skipping for affected tables.

```
begin
    rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks (
        schema_name => '&corruptionOwner',
        object_name => '&corruptionTable',
        object_type => rdsadmin.rdsadmin_dbms_repair.table_object,
        flags => rdsadmin.rdsadmin_dbms_repair.skip_flag);
end;
/
select skip_corrupt from dba_tables where owner = '&corruptionOwner' and table_name =
    '&corruptionTable';
```

Run the following procedure to disable corruption skipping.

```
begin
    rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks (
        schema_name => '&corruptionOwner',
        object_name => '&corruptionTable',
        object_type => rdsadmin.rdsadmin_dbms_repair.table_object,
        flags => rdsadmin.rdsadmin_dbms_repair.noskip_flag);
end;
/
select skip_corrupt from dba_tables where owner = '&corruptionOwner' and table_name =
    '&corruptionTable';
```

5. When you have completed all repair work, run the following procedures to drop the repair tables.

```
exec rdsadmin.rdsadmin_dbms_repair.drop_repair_table;
exec rdsadmin.rdsadmin_dbms_repair.drop_orphan_keys_table;
```

Resizing the temporary tablespace in a read replica

By default, Oracle tablespaces are created with auto-extend enabled and no maximum size. Because of these default settings, tablespaces can grow too large in some cases. We recommend that you specify an appropriate maximum size on permanent and temporary tablespaces, and that you carefully monitor space usage.

To resize the temporary space in a read replica for an Oracle DB instance, use either the `rdsadmin.rdsadmin_util.resize_temp_tablespace` or the `rdsadmin.rdsadmin_util.resize_tempfile` Amazon RDS procedure.

The `resize_temp_tablespace` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>temp_tbs</code>	varchar2	—	Yes	The name of the temporary tablespace to resize.
<code>size</code>	varchar2	—	Yes	You can specify the size in bytes (the default), kilobytes (K), megabytes (M), or gigabytes (G).

The `resize_tempfile` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
file_id	binary_integer	—	Yes	The file identifier of the temporary tablespace to resize.
size	varchar2	—	Yes	You can specify the size in bytes (the default), kilobytes (K), megabytes (M), or gigabytes (G).

The following examples resize a temporary tablespace named `TEMP` to the size of 4 gigabytes on a read replica.

```
exec rdsadmin.rdsadmin_util.resize_temp_tablespace('TEMP','4G');
```

```
exec rdsadmin.rdsadmin_util.resize_temp_tablespace('TEMP','4096000000');
```

The following example resizes a temporary tablespace based on the tempfile with the file identifier 1 to the size of 2 megabytes on a read replica.

```
exec rdsadmin.rdsadmin_util.resize_tempfile(1,'2M');
```

For more information about read replicas for Oracle DB instances, see [Working with Oracle replicas for Amazon RDS \(p. 1119\)](#).

Purging the recycle bin

When you drop a table, your Oracle database doesn't immediately remove its storage space. The database renames the table and places it and any associated objects in a recycle bin. Purging the recycle bin removes these items and releases their storage space.

To purge the entire recycle bin, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.purge_dba_recyclebin`. However, this procedure can't purge the recycle bin of `SYS` and `RDSADMIN` objects. If you need to purge these objects, contact AWS Support.

The following example purges the entire recycle bin.

```
exec rdsadmin.rdsadmin_util.purge_dba_recyclebin;
```

Performing common log-related tasks for Oracle DB instances

Following, you can find how to perform certain common DBA tasks related to logging on your Amazon RDS DB instances running Oracle. To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances, and restricts access to certain system procedures and tables that require advanced privileges.

For more information, see [Oracle database log files \(p. 527\)](#).

Topics

- [Setting force logging \(p. 1063\)](#)
- [Setting supplemental logging \(p. 1063\)](#)

- [Switching online log files \(p. 1064\)](#)
- [Adding online redo logs \(p. 1064\)](#)
- [Dropping online redo logs \(p. 1065\)](#)
- [Resizing online redo logs \(p. 1065\)](#)
- [Retaining archived redo logs \(p. 1067\)](#)
- [Accessing transaction logs \(p. 1068\)](#)

Setting force logging

In force logging mode, Oracle logs all changes to the database except changes in temporary tablespaces and temporary segments (NOLOGGING clauses are ignored). For more information, see [Specifying FORCE LOGGING mode](#) in the Oracle documentation.

To set force logging, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.force_logging`. The `force_logging` procedure has the following parameters.

Parameter name	Data type	Default	Yes	Description
<code>p_enable</code>	boolean	true	No	Set to <code>true</code> to put the database in force logging mode, <code>false</code> to remove the database from force logging mode.

The following example puts the database in force logging mode.

```
exec rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

Setting supplemental logging

If you enable supplemental logging, LogMiner has the necessary information to support chained rows and clustered tables. For more information, see [Supplemental logging](#) in the Oracle documentation.

Oracle Database doesn't enable supplemental logging by default. To enable and disable supplemental logging, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.alter_supplemental_logging`. For more information about how Amazon RDS manages the retention of archived redo logs for Oracle DB instances, see [Retaining archived redo logs \(p. 1067\)](#).

The `alter_supplemental_logging` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>p_action</code>	varchar2	—	Yes	'ADD' to add supplemental logging, 'DROP' to drop supplemental logging.
<code>p_type</code>	varchar2	null	No	The type of supplemental logging. Valid values are 'ALL', 'FOREIGN KEY', 'PRIMARY KEY', 'UNIQUE', or PROCEDURAL.

The following example enables supplemental logging.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD');
end;
/
```

The following example enables supplemental logging for all fixed-length maximum size columns.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD',
        p_type    => 'ALL');
end;
/
```

The following example enables supplemental logging for primary key columns.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD',
        p_type    => 'PRIMARY KEY');
end;
/
```

Switching online log files

To switch log files, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.switch_logfile`. The `switch_logfile` procedure has no parameters.

The following example switches log files.

```
exec rdsadmin.rdsadmin_util.switch_logfile;
```

Adding online redo logs

An Amazon RDS DB instance running Oracle starts with four online redo logs, 128 MB each. To add additional redo logs, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.add_logfile`.

The `add_logfile` procedure has the following parameters.

Note

The parameters are mutually exclusive.

Parameter name	Data type	Default	Required	Description
bytes	positive	null	No	The size of the log file in bytes.
p_size	varchar2	—	Yes	The size of the log file. You can specify the size in kilobytes (K), megabytes (M), or gigabytes (G).

The following command adds a 100 MB log file.

```
exec rdsadmin.rdsadmin_util.add_logfile(p_size => '100M');
```

Dropping online redo logs

To drop redo logs, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.drop_logfile`. The `drop_logfile` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
grp	positive	—	Yes	The group number of the log.

The following example drops the log with group number 3.

```
exec rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
```

You can only drop logs that have a status of unused or inactive. The following example gets the statuses of the logs.

```
SELECT GROUP#, STATUS FROM V$LOG;

GROUP#      STATUS
-----
1           CURRENT
2           INACTIVE
3           INACTIVE
4           UNUSED
```

Resizing online redo logs

An Amazon RDS DB instance running Oracle starts with four online redo logs, 128 MB each. The following example shows how you can use Amazon RDS procedures to resize your logs from 128 MB each to 512 MB each.

```
/* Query V$LOG to see the logs.          */
/* You start with 4 logs of 128 MB each. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#      BYTES      STATUS
-----
1           134217728  INACTIVE
2           134217728  CURRENT
3           134217728  INACTIVE
4           134217728  INACTIVE

/* Add four new logs that are each 512 MB */

exec rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
exec rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
exec rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
exec rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);

/* Query V$LOG to see the logs. */
/* Now there are 8 logs.        */
```

```

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#      BYTES      STATUS
----- -----
1          134217728  INACTIVE
2          134217728  CURRENT
3          134217728  INACTIVE
4          134217728  INACTIVE
5          536870912  UNUSED
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED

/* Drop each inactive log using the group number. */

exec rdsadmin.rdsadmin_util.drop_logfile(grp => 1);
exec rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
exec rdsadmin.rdsadmin_util.drop_logfile(grp => 4);

/* Query V$LOG to see the logs. */
/* Now there are 5 logs.           */

select GROUP#, BYTES, STATUS from V$LOG;

GROUP#      BYTES      STATUS
----- -----
2          134217728  CURRENT
5          536870912  UNUSED
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED

/* Switch logs so that group 2 is no longer current. */

exec rdsadmin.rdsadmin_util.switch_logfile;

/* Query V$LOG to see the logs.           */
/* Now one of the new logs is current. */

SQL>SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#      BYTES      STATUS
----- -----
2          134217728  ACTIVE
5          536870912  CURRENT
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED

/* If the status of log 2 is still "ACTIVE", issue a checkpoint to clear it to "INACTIVE".
 */

exec rdsadmin.rdsadmin_util.checkpoint;

/* Query V$LOG to see the logs.           */
/* Now the final original log is inactive. */

select GROUP#, BYTES, STATUS from V$LOG;

```

```

GROUP#    BYTES    STATUS
-----
2        134217728  INACTIVE
5        536870912  CURRENT
6        536870912  UNUSED
7        536870912  UNUSED
8        536870912  UNUSED

# Drop the final inactive log.

exec rdsadmin.rdsadmin_util.drop_logfile(grp => 2);

/* Query V$LOG to see the logs.      */
/* Now there are four 512 MB logs. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#    BYTES    STATUS
-----
5        536870912  CURRENT
6        536870912  UNUSED
7        536870912  UNUSED
8        536870912  UNUSED

```

Retaining archived redo logs

You can retain archived redo logs locally on your DB instance for use with products like Oracle LogMiner (DBMS_LOGMNR). After you have retained the redo logs, you can use LogMiner to analyze the logs. For more information, see [Using LogMiner to analyze redo log files](#) in the Oracle documentation.

To retain archived redo logs, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.set_configuration`. The `set_configuration` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>name</code>	<code>varchar</code>	—	Yes	The name of the configuration to update.
<code>value</code>	<code>varchar</code>	—	Yes	The value for the configuration.

The following example retains 24 hours of redo logs.

```

begin
    rdsadmin.rdsadmin_util.set_configuration(
        name  => 'archivelog retention hours',
        value => '24');
end;
/
commit;

```

Note

The commit is required for the change to take effect.

To view how long archived redo logs are kept for your DB instance, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.show_configuration`.

The following example shows the log retention time.

```
set serveroutput on
exec rdsadmin.rdsadmin_util.show_configuration;
```

The output shows the current setting for `archivelog retention hours`. The following output shows that archived redo logs are kept for 48 hours.

```
NAME:archivelog retention hours
VALUE:48
DESCRIPTION:ArchiveLog expiration specifies the duration in hours before archive/redo log
files are automatically deleted.
```

Because the archived redo logs are retained on your DB instance, ensure that your DB instance has enough allocated storage for the retained logs. To determine how much space your DB instance has used in the last X hours, you can run the following query, replacing X with the number of hours.

```
select sum(BLOCKS * BLOCK_SIZE) bytes
  from V$ARCHIVED_LOG
 where FIRST_TIME >= SYSDATE-(X/24) and DEST_ID=1;
```

Archived redo logs are only generated if the backup retention period of your DB instance is greater than zero. By default the backup retention period is greater than zero, so unless you explicitly set yours to zero, archived redo logs are generated for your DB instance.

After the archived redo logs are removed from your DB instance, you can't download them again to your DB instance. Amazon RDS retains the archived redo logs outside of your DB instance to support restoring your DB instance to a point in time. Amazon RDS retains the archived redo logs outside of your DB instance based on the backup retention period configured for your DB instance. To modify the backup retention period for your DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Note

In some cases, you might be using JDBC on Linux to download archived redo logs and experience long latency times and connection resets. In such cases, the issues might be caused by the default random number generator setting on your Java client. We recommend setting your JDBC drivers to use a nonblocking random number generator.

Accessing transaction logs

Accessing transaction logs is supported for version 12.1.0.2.v7 and later of Oracle Database 12c Release 1 (12.1), all Oracle Database 12c Release 2 (12.2.0.1) versions, all Oracle Database 18c versions, and all Oracle Database 19c versions.

You might want to access your online and archived redo log files for mining with external tools such as GoldenGate, Attunity, Informatica, and others. If you want to access your online and archived redo log files, you must first create directory objects that provide read-only access to the physical file paths.

The following code creates directories that provide read-only access to your online and archived redo log files:

Important

This code also revokes the `DROP ANY DIRECTORY` privilege.

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

After you create directory objects for your online and archived redo log files, you can read the files by using PL/SQL. For more information about reading files from directory objects, see [Listing files in a DB instance directory \(p. 1096\)](#) and [Reading files in a DB instance directory \(p. 1096\)](#).

The following code drops the directories for your online and archived redo log files.

```
exec rdsadmin.rdsadmin_master_util.drop_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.drop_onlinelog_dir;
```

The following code grants and revokes the `DROP ANY DIRECTORY` privilege.

```
exec rdsadmin.rdsadmin_master_util.revoke_drop_any_directory;
exec rdsadmin.rdsadmin_master_util.grant_drop_any_directory;
```

Performing common RMAN tasks for Oracle DB instances

In the following section, you can find how you can perform Oracle Recovery Manager (RMAN) DBA tasks on your Amazon RDS DB instances running Oracle. To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances. It also restricts access to certain system procedures and tables that require advanced privileges.

You can use the Amazon RDS package `rdsadmin.rdsadmin_rman_util` to perform RMAN backups of your Amazon RDS for Oracle database to disk. The `rdsadmin.rdsadmin_rman_util` package supports full and incremental database file backups, tablespace backups, and archive log backups.

RMAN backups consume storage space on the Amazon RDS DB instance host. When you perform a backup, you specify an Oracle directory object as a parameter in the procedure call. The backup files are placed in the specified directory. You can use default directories, such as `DATA_PUMP_DIR`, or create a new directory. For more information, see [Creating and dropping directories in the main data storage space \(p. 1095\)](#).

After an RMAN backup has finished, you can copy the backup files off the Amazon RDS for Oracle DB instance host. You might do this for the purpose of restoring to a non-RDS host or for long-term storage of backups. For example, you can copy the backup files to an Amazon S3 bucket. For more information, see using [Amazon S3 integration \(p. 1127\)](#).

The backup files for RMAN backups remain on the Amazon RDS DB instance host until you remove them manually. You can use the `UTL_FILE.FREMOVE` Oracle procedure to remove files from a directory. For more information, see [FREMOVE procedure](#) in the Oracle documentation.

When backing up archived redo logs or performing a full or incremental backup that includes archived redo logs, redo log retention must be set to a nonzero value. For more information, see [Retaining archived redo logs \(p. 1067\)](#).

Note

For backing up and restoring to another Amazon RDS for Oracle DB instance, you can continue to use the Amazon RDS backup and restore features. For more information, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#)

Currently, RMAN restore isn't supported for Amazon RDS for Oracle DB instances.

Topics

- [Common parameters for RMAN procedures \(p. 1070\)](#)
- [Validating DB instance files \(p. 1072\)](#)
- [Enabling and disabling block change tracking \(p. 1075\)](#)
- [Crosschecking archived redo logs \(p. 1076\)](#)
- [Backing up archived redo logs \(p. 1077\)](#)

- [Performing a full database backup \(p. 1082\)](#)
- [Performing an incremental database backup \(p. 1083\)](#)
- [Performing a tablespace backup \(p. 1084\)](#)

Common parameters for RMAN procedures

You can use procedures in the Amazon RDS package `rdsadmin.rdsadmin_rman_util` to perform tasks with RMAN. Several parameters are common to the procedures in the package. The package has the following common parameters.

Parameter name	Data type	Valid values	Default	Required	Description
<code>p_owner</code>	<code>varchar2</code>	A valid owner of the directory specified in <code>p_directory_name</code> .	—	Yes	The owner of the directory to contain the backup files.
<code>p_directory_name</code>	<code>varchar2</code>	A valid database directory name.	—	Yes	The name of the directory to contain the backup files.
<code>p_label</code>	<code>varchar2</code>	a-z, A-Z, 0-9, '_', '-', '.'	—	No	A unique string that is included in the backup file names. Note The limit is 30 characters.
<code>p_compress</code>	<code>boolean</code>	<code>TRUE</code> , <code>FALSE</code>	<code>FALSE</code>	No	Specify <code>TRUE</code> to enable BASIC backup compression. Specify <code>FALSE</code> to disable BASIC backup compression.
<code>p_include_archive_log</code>	<code>boolean</code>	<code>TRUE</code> , <code>FALSE</code>	<code>FALSE</code>	No	Specify <code>TRUE</code> to include archived redo logs in the backup. Specify <code>FALSE</code> to exclude archived redo logs from the backup. If you include archived redo logs in the backup, set retention to one hour or greater using the <code>rdsadmin.rdsadmin_util.set_co</code> procedure. Also, call the

Parameter name	Data type	Valid values	Default	Required	Description
					<code>rdsadmin.rdsadmin_rman_util.c</code> procedure immediately before running the backup. Otherwise, the backup might fail due to missing archived redo log files that have been deleted by Amazon RDS management procedures.
<code>p_include_controlfile</code>	boolean	TRUE, FALSE	FALSE	No	Specify TRUE to include the control file in the backup. Specify FALSE to exclude the control file from the backup.
<code>p_optimize</code>	boolean	TRUE, FALSE	TRUE	No	Specify TRUE to enable backup optimization, if archived redo logs are included, to reduce backup size. Specify FALSE to disable backup optimization.
<code>p_parallel</code>	number	A valid integer between 1 and 254 for Oracle Database Enterprise Edition (EE) 1 for other Oracle Database editions	1	No	Number of channels.

Parameter name	Data type	Valid values	Default	Required	Description
p_rman_to_dbms_output	boolean	TRUE, FALSE	FALSE	No	<p>When TRUE, the RMAN output is sent to the DBMS_OUTPUT package in addition to a file in the BDUMP directory. In SQL*Plus, use SET SERVEROUTPUT ON to see the output.</p> <p>When FALSE, the RMAN output is only sent to a file in the BDUMP directory.</p>
p_section_size_mb	number	A valid integer	NULL	No	<p>The section size in megabytes (MB).</p> <p>Validates in parallel by dividing each file into the specified section size.</p> <p>When NULL, the parameter is ignored.</p>
p_validation_type	varchar2	'PHYSICAL' 'PHYSICAL+LOGICAL'	'PHYSICAL'	'No'	<p>The level of corruption detection.</p> <p>Specify 'PHYSICAL' to check for physical corruption. An example of physical corruption is a block with a mismatch in the header and footer.</p> <p>Specify 'PHYSICAL+LOGICAL' to check for logical inconsistencies in addition to physical corruption. An example of logical corruption is a corrupt block.</p>

Validating DB instance files

You can use the Amazon RDS package `rdsadmin.rdsadmin_rman_util` to validate Amazon RDS for Oracle DB instance files, such as data files, tablespaces, control files, or server parameter files (SPFILEs).

For more information about RMAN validation, see [Validating database files and backups](#) and [VALIDATE](#) in the Oracle documentation.

Topics

- [Validating a DB instance \(p. 1073\)](#)
- [Validating a tablespace \(p. 1074\)](#)
- [Validating a control file \(p. 1074\)](#)
- [Validating an SPFILE \(p. 1074\)](#)
- [Validating a data file \(p. 1074\)](#)

Validating a DB instance

To validate all of the relevant files used by an Amazon RDS Oracle DB instance, use the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.validate_database`.

This procedure uses the following common parameters for RMAN tasks:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

The following example validates the DB instance using the default values for the parameters.

```
exec rdsadmin.rdsadmin_rman_util.validate_database;
```

The following example validates the DB instance using the specified values for the parameters.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.validate_database(
        p_validation_type      => 'PHYSICAL+LOGICAL',
        p_parallel             => 4,
        p_section_size_mb     => 10,
        p_rman_to_dbms_output => FALSE);
END;
/
```

When the `p_rman_to_dbms_output` parameter is set to `FALSE`, the RMAN output is written to a file in the `BDUMP` directory.

To view the files in the `BDUMP` directory, run the following `SELECT` statement.

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime;
```

To view the contents of a file in the `BDUMP` directory, run the following `SELECT` statement.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP', 'rds-rman-
validate-nnn.txt'));
```

Replace the file name with the name of the file you want to view.

Validating a tablespace

To validate the files associated with a tablespace, use the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.validate_tablespace`.

This procedure uses the following common parameters for RMAN tasks:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

This procedure also uses the following additional parameter.

Parameter name	Data type	Valid values	Default	Required	Description
<code>p_tablespace_name</code>	varchar2	A valid tablespace name	—	Yes	The name of the tablespace.

Validating a control file

To validate only the control file used by an Amazon RDS Oracle DB instance, use the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.validate_current_controlfile`.

This procedure uses the following common parameter for RMAN tasks:

- `p_validation_type`
- `p_rman_to_dbms_output`

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

Validating an SPFILE

To validate only the server parameter file (SPFILE) used by an Amazon RDS Oracle DB instance, use the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.validate_spfile`.

This procedure uses the following common parameter for RMAN tasks:

- `p_validation_type`
- `p_rman_to_dbms_output`

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

Validating a data file

To validate a data file, use the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.validate_datafile`.

This procedure uses the following common parameters for RMAN tasks:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

This procedure also uses the following additional parameters.

Parameter name	Data type	Valid values	Default	Required	Description
<code>p_datafile</code>	varchar2	A valid datafile ID number or a valid datafile name including complete path	—	Yes	The datafile ID number (from <code>v\$datafile.file#</code>) or the full datafile name including the path (from <code>v\$datafile.name</code>).
<code>p_from_block</code>	number	A valid integer	NULL	No	The number of the block where the validation starts within the data file. When this is NULL, 1 is used.
<code>p_to_block</code>	number	A valid integer	NULL	No	The number of the block where the validation ends within the data file. When this is NULL, the maximum block in the data file is used.

Enabling and disabling block change tracking

Block changing tracking records changed blocks in a tracking file. This technique can improve the performance of incremental backups. For more information, see [Using Block Change Tracking to Improve Incremental Backup Performance](#) in the Oracle Database documentation.

To enable block change tracking for a DB instance, use the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.enable_block_change_tracking`. To disable block change tracking, use `disable_block_change_tracking`. These procedures take no parameters.

Read-only replicas support block change tracking. If you create a read-only replica from a source DB that uses block change tracking, the replica uses block change tracking. You can't enable block change tracking on a mounted replica. If you place a mounted replica in read-only mode, block change tracking isn't enabled, but you can enable it using `enable_block_change_tracking`. If you promote an Oracle replica to a source DB, you can use block change tracking just as for any other Oracle DB instance.

Block change tracking procedures are supported for the following DB engine versions:

- 12.1.0.2.v15 or higher 12.1 versions

- 12.2.0.1.ru-2019-01.rur-2019-01.r1 or higher 12.2 versions
- All 18.0.0.0 versions
- All 19.0.0.0 versions

To determine whether block change tracking is enabled for your DB instance, run the following query.

```
SELECT STATUS, FILENAME FROM V$BLOCK_CHANGE_TRACKING;
```

The following example enables block change tracking for a DB instance.

```
EXEC rdsadmin.rdsadmin_rman_util.enable_block_change_tracking;
```

The following example disables block change tracking for a DB instance.

```
EXEC rdsadmin.rdsadmin_rman_util.disable_block_change_tracking;
```

Crosschecking archived redo logs

You can crosscheck archived redo logs using the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.crosscheck_archivelog`.

You can use this procedure to crosscheck the archived redo logs registered in the control file and optionally delete the expired logs records. When RMAN makes a backup, it creates a record in the control file. Over time, these records increase the size of the control file. We recommend that you remove expired records periodically.

Note

Standard Amazon RDS backups don't use RMAN and therefore don't create records in the control file.

This procedure uses the common parameter `p_rman_to_dbms_output` for RMAN tasks.

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

This procedure also uses the following additional parameter.

Parameter name	Data type	Valid values	Default	Required	Description
<code>p_delete_expired</code>	boolean	TRUE, FALSE	TRUE	No	<p>When TRUE, delete expired archived redo log records from the control file.</p> <p>When FALSE, retain the expired archived redo log records in the control file.</p>

This procedure is supported for the following Amazon RDS for Oracle DB engine versions:

- 12.1.0.2.v15 or higher 12.1 versions
- 12.2.0.1.ru-2019-01.rur-2019-01.r1 or higher 12.2 versions

- All 18.0.0.0 versions
- All 19.0.0.0 versions

The following example marks archived redo log records in the control file as expired, but does not delete the records.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
        p_delete_expired      => FALSE,
        p_rman_to_dbms_output => FALSE);
END;
/
```

The following example deletes expired archived redo log records from the control file.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
        p_delete_expired      => TRUE,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Backing up archived redo logs

You can use the Amazon RDS package `rdsadmin.rdsadmin_rman_util` to back up archived redo logs for an Amazon RDS Oracle DB instance.

The procedures for backing up archived redo logs are supported for the following Amazon RDS for Oracle DB engine versions:

- 12.1.0.2.v15 or higher 12.1 versions
- 12.2.0.1.ru-2019-01.rur-2019-01.r1 or higher 12.2 versions
- All 18.0.0.0 versions
- All 19.0.0.0 versions

Topics

- [Backing up all archived redo logs \(p. 1077\)](#)
- [Backing up an archived redo log from a date range \(p. 1078\)](#)
- [Backing up an archived redo log from an SCN range \(p. 1079\)](#)
- [Backing up an archived redo log from a sequence number range \(p. 1081\)](#)

Backing up all archived redo logs

To back up all of the archived redo logs for an Amazon RDS Oracle DB instance, use the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.backup_archivelog_all`.

This procedure uses the following common parameters for RMAN tasks:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`

- `p_compress`
- `p_rman_to_dbms_output`

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

The following example backs up all archived redo logs for the DB instance.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_all(
        p_owner      => 'SYS',
        p_directory_name  => 'MYDIRECTORY',
        p_parallel    => 4,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Backing up an archived redo log from a date range

To back up specific archived redo logs for an Amazon RDS Oracle DB instance by specifying a date range, use the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.backup_archivelog_date`. The date range specifies which archived redo logs to back up.

This procedure uses the following common parameters for RMAN tasks:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

This procedure also uses the following additional parameters.

Parameter name	Data type	Valid values	Default	Required	Description
<code>p_from_date</code>	date	A date that is between the <code>start_date</code> and <code>next_date</code> of an archived redo log that exists on disk. The value must be less than	—	Yes	The starting date for the archived log backups.

Parameter name	Data type	Valid values	Default	Required	Description
		or equal to the value specified for <code>p_to_date</code> .			
<code>p_to_date</code>	date	A date that is between the <code>start_date</code> and <code>next_date</code> of an archived redo log that exists on disk. The value must be greater than or equal to the value specified for <code>p_from_date</code> .	—	Yes	The ending date for the archived log backups.

The following example backs up archived redo logs in the date range for the DB instance.

```

BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_date(
        p_owner          => 'SYS',
        p_directory_name => 'MYDIRECTORY',
        p_from_date      => '03/01/2019 00:00:00',
        p_to_date        => '03/02/2019 00:00:00',
        p_parallel       => 4,
        p_rman_to_dbms_output => FALSE);
END;
/

```

Backing up an archived redo log from an SCN range

To back up specific archived redo logs for an Amazon RDS Oracle DB instance by specifying a system change number (SCN) range, use the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.backup_archivelog_scn`. The SCN range specifies which archived redo logs to back up.

This procedure uses the following common parameters for RMAN tasks:

- `p_owner`
- `p_directory_name`

- p_label
- p_parallel
- p_compress
- p_rman_to_dbms_output

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

This procedure also uses the following additional parameters.

Parameter name	Data type	Valid values	Default	Required	Description
p_from_scn	number	An SCN of an archived redo log that exists on disk. The value must be less than or equal to the value specified for p_to_scn.	—	Yes	The starting SCN for the archived log backups.
p_to_scn	number	An SCN of an archived redo log that exists on disk. The value must be greater than or equal to the value specified for p_from_scn.	—	Yes	The ending SCN for the archived log backups.

The following example backs up archived redo logs in the SCN range for the DB instance.

```

BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_scn(
        p_owner          => 'SYS',
        p_directory_name => 'MYDIRECTORY',
        p_from_scn       => 1533835,
        p_to_scn         => 1892447,
        p_parallel       => 4,
        p_rman_to_dbms_output => FALSE);
END;
/

```

Backing up an archived redo log from a sequence number range

To back up specific archived redo logs for an Amazon RDS Oracle DB instance by specifying a sequence number range, use the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.backup_archive_log_sequence`. The sequence number range specifies which archived redo logs to back up.

This procedure uses the following common parameters for RMAN tasks:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

This procedure also uses the following additional parameters.

Parameter name	Data type	Valid values	Default	Required	Description
<code>p_from_sequence</code>	number	A sequence number of an archived redo log that exists on disk. The value must be less than or equal to the value specified for <code>p_to_sequence</code> .	—	Yes	The starting sequence number for the archived log backups.
<code>p_to_sequence</code>	number	A sequence number of an archived redo log that exists on disk. The value must be greater than or equal to the value	—	Yes	The ending sequence number for the archived log backups.

Parameter name	Data type	Valid values	Default	Required	Description
		specified for p_from_sequence.			

The following example backs up archived redo logs in the sequence number range for the DB instance.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_sequence(
        p_owner          => 'SYS',
        p_directory_name => 'MYDIRECTORY',
        p_from_sequence   => 11160,
        p_to_sequence     => 11160,
        p_parallel        => 4,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Performing a full database backup

You can perform a backup of all blocks of data files included in the backup using Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.backup_database_full`.

This procedure uses the following common parameters for RMAN tasks:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

This procedure is supported for the following Amazon RDS for Oracle DB engine versions:

- 12.1.0.2.v15 or higher 12.1 versions
- 12.2.0.1.ru-2019-01.rur-2019-01.r1 or higher 12.2 versions
- All 18.0.0.0 versions
- All 19.0.0.0 versions

The following example performs a full backup of the DB instance using the specified values for the parameters.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_database_full(
        p_owner          => 'SYS',
        p_directory_name => 'MYDIRECTORY',
```

```

    p_parallel      => 4,
    p_section_size_mb  => 10,
    p_rman_to_dbms_output => FALSE);
END;
/

```

Performing an incremental database backup

You can perform an incremental backup of your DB instance using the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.backup_database_incremental`.

For more information about incremental backups, see [Incremental backups](#) in the Oracle documentation.

This procedure uses the following common parameters for RMAN tasks:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_include_controlfile`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

This procedure is supported for the following Amazon RDS for Oracle DB engine versions:

- 12.1.0.2.v15 or higher 12.1 versions
- 12.2.0.1.ru-2019-01.rur-2019-01.r1 or higher 12.2 versions
- All 18.0.0.0 versions
- All 19.0.0.0 versions

This procedure also uses the following additional parameter.

Parameter name	Data type	Valid values	Default	Required	Description
<code>p_level</code>	number	0, 1	0	No	Specify 0 to enable a full incremental backup. Specify 1 to enable a non-cumulative incremental backup.

The following example performs an incremental backup of the DB instance using the specified values for the parameters.

```

BEGIN
  rdsadmin.rdsadmin_rman_util.backup_database_incremental(

```

```

    p_owner          => 'SYS',
    p_directory_name => 'MYDIRECTORY',
    p_level          => 1,
    p_parallel        => 4,
    p_section_size_mb => 10,
    p_rman_to_dbms_output => FALSE);
END;
/

```

Performing a tablespace backup

You can perform a DB instance tablespace using the Amazon RDS procedure `rdsadmin.rdsadmin_rman_util.backup_tablespace`.

This procedure uses the following common parameters for RMAN tasks:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_include_controlfile`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`

For more information, see [Common parameters for RMAN procedures \(p. 1070\)](#).

This procedure also uses the following additional parameter.

Parameter name	Data type	Valid values	Default	Required	Description
<code>p_tablespace_name</code>	varchar2	A valid tablespace name.	—	Yes	The name of the tablespace to back up.

This procedure is supported for the following Amazon RDS for Oracle DB engine versions:

- 12.1.0.2.v15 or higher 12.1 versions
- 12.2.0.1.ru-2019-01.rur-2019-01.r1 or higher 12.2 versions
- All 18.0.0.0 versions
- All 19.0.0.0 versions

The following example performs a tablespace backup using the specified values for the parameters.

```

BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tablespace(
        p_owner          => 'SYS',
        p_directory_name => 'MYDIRECTORY',
        p_tablespace_name => MYTABLESPACE,

```

```

    p_parallel      => 4,
    p_section_size_mb  => 10,
    p_rman_to_dbms_output => FALSE);
END;
/

```

Performing common scheduling tasks for Oracle DB instances

Some SYS-owned scheduler jobs can interfere with normal database operations, and Oracle Support recommends they be disabled or the job schedule be modified. You can use the Amazon RDS package `rdsadmin.rdsadmin_dbms_scheduler` to perform tasks for SYS-owned Oracle Scheduler jobs.

These procedures are supported for the following Amazon RDS for Oracle DB engine versions:

- 19c
- 18c
- 12.2.0.2.ru-2019-07.rur-2019-07.r1 or higher 12.2 versions
- 12.1.0.2.v17 or higher 12.1 versions

Common parameters for Oracle Scheduler procedures

To perform tasks with Oracle Scheduler, use procedures in the Amazon RDS package `rdsadmin.rdsadmin_dbms_scheduler`. Several parameters are common to the procedures in the package. The package has the following common parameters.

Parameter name	Data type	Valid values	Default	Required	Description
name	varchar2	'SYS.BSLN_MAINTAIN_STATS_JOB'	Yes	No	The name of the job to modify. Note Currently, you can only modify SYS.CLEANUP_ONLINE_IND_BUILD and SYS.BSLN_MAINTAIN_STATS_JOB jobs.
attribute	varchar2	'REPEAT_INTERVAL','SCHEDULE_NAME'	Yes	No	Attribute to modify. To modify the repeat interval for the job, specify 'REPEAT_INTERVAL'. To modify the schedule name for the job, specify 'SCHEDULE_NAME'.
value	varchar2	A valid schedule	-	Yes	The new value of the attribute.

Parameter name	Data type	Valid values	Default	Required	Description
		interval or schedule name, depending on attribute used.			

Modifying DBMS_SCHEDULER jobs

You can use the Oracle procedure `dbms_scheduler.set_attribute` to modify certain components of Oracle Scheduler. For more information, see [DBMS_SCHEDULER](#) and [SET_ATTRIBUTE procedure](#) in the Oracle documentation.

When working with Amazon RDS DB instances, prepend the schema name `SYS` to the object name. The following example sets the resource plan attribute for the Monday window object.

```
begin
    dbms_scheduler.set_attribute(
        name      => 'SYS.MONDAY_WINDOW',
        attribute => 'RESOURCE_PLAN',
        value     => 'resource_plan_1');
end;
/
```

Note

Some SYS-owned Oracle Scheduler jobs can interfere with normal database operations. Oracle Support recommends that they be disabled or that the job schedule be modified. Amazon RDS for Oracle doesn't provide the required privileges to modify SYS-owned Oracle Scheduler jobs using the `DBMS_SCHEDULER` package. Instead, you can use the procedures in the Amazon RDS package `rdsadmin.rdsadmin_dbms_scheduler` to perform tasks for SYS-owned Oracle Scheduler jobs. For information about using these procedures, see [Performing common scheduling tasks for Oracle DB instances \(p. 1085\)](#).

Setting the time zone for Oracle Scheduler jobs

To modify the time zone for Oracle Scheduler, you can use the Oracle procedure `dbms_scheduler.set_scheduler_attribute`. For more information about the `dbms_scheduler` package, see [DBMS_SCHEDULER](#) and [SET_SCHEDULER_ATTRIBUTE](#) in the Oracle documentation.

To modify the current time zone setting

1. Connect to the database using a client such as SQL Developer. For more information, see [Connecting to your DB instance using Oracle SQL developer \(p. 1003\)](#).
2. Set the default time zone as following, substituting your time zone for `time_zone_name`.

```
begin
    dbms_scheduler.set_scheduler_attribute(
        attribute => 'default_timezone',
        value     => 'time_zone_name'
    );
end;
/
```

In the following example, you change the time zone to Asia/Shanghai.

Start by querying the current time zone, as shown following.

```
SELECT VALUE FROM DBA_SCHEDULER_GLOBAL_ATTRIBUTE WHERE ATTRIBUTE_NAME='DEFAULT_TIMEZONE';
```

The output shows that the current time zone is ETC/UTC.

```
VALUE
-----
Etc/UTC
```

Then you set the time zone to Asia/Shanghai.

```
begin
    dbms_scheduler.set_scheduler_attribute(
        attribute => 'default_timezone',
        value => 'Asia/Shanghai'
    );
end;
/
```

For more information about changing the system time zone, see [Oracle time zone \(p. 1201\)](#).

Disabling SYS-owned Oracle Scheduler jobs

To disable a SYS-owned Oracle Scheduler job, use the `rdsadmin.rdsadmin_dbms_scheduler.disable` procedure.

This procedure uses the `name` common parameter for Oracle Scheduler tasks. For more information, see [Common parameters for Oracle Scheduler procedures \(p. 1085\)](#).

The following example disables the `SYS.CLEANUP_ONLINE_IND_BUILD` Oracle Scheduler job.

```
BEGIN
    rdsadmin.rdsadmin_dbms_scheduler.disable('SYS.CLEANUP_ONLINE_IND_BUILD');
END;
/
```

Enabling SYS-owned Oracle Scheduler jobs

To enable a SYS-owned Oracle Scheduler job, use the `rdsadmin.rdsadmin_dbms_scheduler.enable` procedure.

This procedure uses the `name` common parameter for Oracle Scheduler tasks. For more information, see [Common parameters for Oracle Scheduler procedures \(p. 1085\)](#).

The following example enables the `SYS.CLEANUP_ONLINE_IND_BUILD` Oracle Scheduler job.

```
BEGIN
    rdsadmin.rdsadmin_dbms_scheduler.enable('SYS.CLEANUP_ONLINE_IND_BUILD');
END;
/
```

Modifying the repeat interval for jobs of CALENDAR type

To modify the repeat interval to modify a SYS-owned Oracle Scheduler job of `CALENDAR` type, use the `rdsadmin.rdsadmin_dbms_scheduler.disable` procedure.

This procedure uses the following common parameters for Oracle Scheduler tasks:

- **name**
- **attribute**
- **value**

For more information, see [Common parameters for Oracle Scheduler procedures \(p. 1085\)](#).

The following example modifies the repeat interval of the `SYS.CLEANUP_ONLINE_IND_BUILD` Oracle Scheduler job.

```
BEGIN
    rdsadmin.rdsadmin_dbms_scheduler.set_attribute(
        name      => 'SYS.CLEANUP_ONLINE_IND_BUILD',
        attribute => 'repeat_interval',
        value     => 'freq=daily;byday=FRI,SAT;byhour=20;byminute=0;bysecond=0');
END;
/
```

Modifying the repeat interval for jobs of NAMED type

Some Oracle Scheduler jobs use a schedule name instead of an interval. For this type of jobs, you must create a new named schedule in the master user schema. Use the standard Oracle `sys.dbms_scheduler.create_schedule` procedure to do this. Also, use the `rdsadmin.rdsadmin_dbms_scheduler.set_attribute` procedure to assign the new named schedule to the job.

This procedure uses the following common parameter for Oracle Scheduler tasks:

- **name**
- **attribute**
- **value**

For more information, see [Common parameters for Oracle Scheduler procedures \(p. 1085\)](#).

The following example modifies the repeat interval of the `SYS.BSLN_MAINTAIN_STATS_JOB` Oracle Scheduler job.

```
BEGIN
    dbms_scheduler.create_schedule (
        schedule_name  => 'rds_master_user.new_schedule',
        start_date     => SYSTIMESTAMP,
        repeat_interval =>
    'freq=daily;byday=MON,TUE,WED,THU,FRI;byhour=0;byminute=0;bysecond=0',
        end_date       => NULL,
        comments       => 'Repeats daily forever');
END;
/

BEGIN
    rdsadmin.rdsadmin_dbms_scheduler.set_attribute (
        name      => 'SYS.BSLN_MAINTAIN_STATS_JOB',
        attribute => 'schedule_name',
        value     => 'rds_master_user.new_schedule');
END;
```

/

Performing common diagnostic tasks for Oracle DB instances

Oracle Database includes a fault diagnosability infrastructure that you can use to investigate database problems. In Oracle terminology, a *problem* is a critical error such as a code bug or data corruption. An *incident* is the occurrence of a problem. If the same error occurs three times, then the infrastructure shows three incidents of this problem. For more information, see [Diagnosing and resolving problems](#) in the Oracle Database documentation.

The Automatic Diagnostic Repository Command Interpreter (ADRCI) utility is an Oracle command-line tool that you use to manage diagnostic data. For example, you can use this tool to investigate problems and package diagnostic data. An *incident package* includes diagnostic data for an incident or all incidents that reference a specific problem. You can upload an incident package, which is implemented as a .zip file, to Oracle Support.

To deliver a managed service experience, Amazon RDS doesn't provide shell access to ADRCI.

To perform diagnostic tasks for your Oracle instance, instead use the Amazon RDS package `rdsadmin.rdsadmin_adrci_util`.

By using the functions in `rdsadmin_adrci_util`, you can list and package problems and incidents, and also show trace files. All functions return a task ID. This ID forms part of the name of log file that contains the ADRCI output, as in `dbtask-task_id.log`. The log file resides in the BDUMP directory.

Common parameters for diagnostic procedures

To perform diagnostic tasks, use functions in the Amazon RDS package `rdsadmin.rdsadmin_adrci_util`. The package has the following common parameters.

Parameter name	Data type	Valid values	Default	Required	Description
<code>incident_id</code>	number	A valid incident ID or null	Null	No	If the value is null, the function shows all incidents. If the value isn't null and represents a valid incident ID, the function shows the specified incident.
<code>problem_id</code>	number	A valid problem ID or null	Null	No	If the value is null, the function shows all problems. If the value isn't null and represents a valid problem ID, the function shows the specified problem.
<code>last</code>	number	A valid integer greater than 0 or null	Null	No	If the value is null, then the function displays at most 50 items. If the value isn't null, the function displays the specified number.

Listing incidents

To list diagnostic incidents for Oracle, use the Amazon RDS function `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents`. You can list incidents in either basic or detailed mode. By default, the function lists the 50 most recent incidents.

This function uses the following common parameters:

- `incident_id`
- `problem_id`

If you specify both of the preceding parameters, `incident_id` overrides `problem_id`. For more information, see [Common parameters for diagnostic procedures \(p. 1089\)](#).

This function uses the following additional parameter.

Parameter name	Data type	Valid values	Default	Required	Description
<code>detail</code>	<code>boolean</code>	TRUE or FALSE	<code>FALSE</code>	No	If TRUE, the function lists incidents in detail mode. If FALSE, the function lists incidents in basic mode.

To list all incidents, call the `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` function without any arguments. You can store the output in a SQL client variable.

```
SQL> VAR task_id VARCHAR2(80);
SQL> exec :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_incidents;
PL/SQL procedure successfully completed.
```

To get the task ID, specify the variable in a query of the dual table.

```
SQL> SELECT :task_id FROM DUAL;
:TASK_ID
-----
1590786706158-3126
```

To read the log file, call the Amazon RDS procedure `rdsadmin.rds_file_util.read_text_file`. Supply the task ID as part of the file name. The following output shows three incidents: 53523, 53522, and 53521.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
'dbtask-'||:task_id||'.log'));
TEXT
-----
2020-05-29 21:11:46.193 UTC [INFO ] Listing ADRCI incidents.
2020-05-29 21:11:46.256 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*****
INCIDENT_ID PROBLEM_KEY                                         CREATE_TIME
```

```

-----
53523      ORA 700 [EVENT_CREATED INCIDENT] [942] [SIMULATED_ERROR_003 2020-05-29
20:15:20.928000 +00:00
53522      ORA 700 [EVENT_CREATED INCIDENT] [942] [SIMULATED_ERROR_002 2020-05-29
20:15:15.247000 +00:00
53521      ORA 700 [EVENT_CREATED INCIDENT] [942] [SIMULATED_ERROR_001 2020-05-29
20:15:06.047000 +00:00
3 rows fetched

2020-05-29 21:11:46.256 UTC [INFO ] The ADRCI incidents were successfully listed.
2020-05-29 21:11:46.256 UTC [INFO ] The task finished successfully.

14 rows selected.

```

To list a particular incident, specify its ID using the `incident_id` parameter. In the following example, you query the log file for incident 53523 only.

```

SQL> exec :task_id :=
  rdsadmin.rdsadmin_adrci_util.list_adrci_incidents(incident_id=>53523);

PL/SQL procedure successfully completed.

SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
  'dbtask-'||:task_id||'.log'));

TEXT
-----
2020-05-29 21:15:25.358 UTC [INFO ] Listing ADRCI incidents.
2020-05-29 21:15:25.426 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*****
INCIDENT_ID          PROBLEM_KEY
CREATE_TIME
-----
53523      ORA 700 [EVENT_CREATED INCIDENT] [942] [SIMULATED_ERROR_003 2020-05-29
20:15:20.928000 +00:00
1 rows fetched

2020-05-29 21:15:25.427 UTC [INFO ] The ADRCI incidents were successfully listed.
2020-05-29 21:15:25.427 UTC [INFO ] The task finished successfully.

12 rows selected.

```

Listing problems

To list diagnostic problems for Oracle, use the Amazon RDS function `rdsadmin.rdsadmin_adrci_util.list_adrci_problems`.

By default, the function lists the 50 most recent problems.

This function uses the common parameter `problem_id`. For more information, see [Common parameters for diagnostic procedures \(p. 1089\)](#).

To get the task ID for all problems, call the `rdsadmin.rdsadmin_adrci_util.list_adrci_problems` function without any arguments, and store the output in a SQL client variable.

```
SQL> exec :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_problems;
```

```
PL/SQL procedure successfully completed.
```

To read the log file, call the `rdsadmin.rds_file_util.read_text_file` function, supplying the task ID as part of the file name. In the following output, the log file shows three problems: 1, 2, and 3.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
  'dbtask-'||:task_id||'.log'));

TEXT
-----
2020-05-29 21:18:50.764 UTC [INFO ] Listing ADRCI problems.
2020-05-29 21:18:50.829 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*****
PROBLEM_ID      PROBLEM_KEY                               LAST INCIDENT
LASTINC_TIME
-----
2          ORA 700 [EVENT_CREATED INCIDENT] [942] [SIMULATED_ERROR_003 53523
2020-05-29 20:15:20.928000 +00:00
3          ORA 700 [EVENT_CREATED INCIDENT] [942] [SIMULATED_ERROR_002 53522
2020-05-29 20:15:15.247000 +00:00
1          ORA 700 [EVENT_CREATED INCIDENT] [942] [SIMULATED_ERROR_001 53521
2020-05-29 20:15:06.047000 +00:00
3 rows fetched

2020-05-29 21:18:50.829 UTC [INFO ] The ADRCI problems were successfully listed.
2020-05-29 21:18:50.829 UTC [INFO ] The task finished successfully.

14 rows selected.
```

In the following example, you list problem 3 only.

```
SQL> exec :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_problems(problem_id=>3);

PL/SQL procedure successfully completed.
```

To read the log file for problem 3, call `rdsadmin.rds_file_util.read_text_file`. Supply the task ID as part of the file name.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
  'dbtask-'||:task_id||'.log'));

TEXT
-----
2020-05-29 21:19:42.533 UTC [INFO ] Listing ADRCI problems.
2020-05-29 21:19:42.599 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*****
PROBLEM_ID PROBLEM_KEY                               LAST INCIDENT
LASTINC_TIME
-----
3          ORA 700 [EVENT_CREATED INCIDENT] [942] [SIMULATED_ERROR_002 53522
2020-05-29 20:15:15.247000 +00:00
1 rows fetched

2020-05-29 21:19:42.599 UTC [INFO ] The ADRCI problems were successfully listed.
2020-05-29 21:19:42.599 UTC [INFO ] The task finished successfully.
```

```
12 rows selected.
```

Creating incident packages

You can create incident packages using the Amazon RDS function `rdsadmin.rdsadmin_adrci_util.create_adrci_package`. The output is a .zip file that you can supply to Oracle Support.

This function uses the following common parameters:

- `problem_id`
- `incident_id`

Make sure to specify one of the preceding parameters. If you specify both parameters, `incident_id` overrides `problem_id`. For more information, see [Common parameters for diagnostic procedures \(p. 1089\)](#).

To create a package for a specific incident, call the Amazon RDS function `rdsadmin.rdsadmin_adrci_util.create_adrci_package` with the `incident_id` parameter. The following example creates a package for incident 53523.

```
SQL> exec :task_id :=
  rdsadmin.rdsadmin_adrci_util.create_adrci_package(incident_id=>53523);

PL/SQL procedure successfully completed.
```

To read the log file, call the `rdsadmin.rds_file_util.read_text_file`. You can supply the task ID as part of the file name. The output shows that you generated incident package `ORA700EVE_20200529212043_COM_1.zip`.

```
SSQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
  'dbtask-'||:task_id||'.log'));

TEXT
-----
2020-05-29 21:20:43.031 UTC [INFO ] The ADRCI package is being created.
2020-05-29 21:20:47.641 UTC [INFO ] Generated package 1 in file /rdsdbdata/log/trace/
ORA700EVE_20200529212043_COM_1.zip, mode complete
2020-05-29 21:20:47.642 UTC [INFO ] The ADRCI package was successfully created.
2020-05-29 21:20:47.642 UTC [INFO ] The task finished successfully.
```

To package diagnostic data for a particular problem, specify its ID using the `problem_id` parameter. In the following example, you package data for problem 3 only.

```
SQL> exec :task_id := rdsadmin.rdsadmin_adrci_util.create_adrci_package(problem_id=>3);

PL/SQL procedure successfully completed.
```

To read the task output, call `rdsadmin.rds_file_util.read_text_file`, supplying the task ID as part of the file name. The output shows that you generated incident package `ORA700EVE_20200529212111_COM_1.zip`.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
  'dbtask-'||:task_id||'.log'));
```

TEXT

```
2020-05-29 21:21:11.050 UTC [INFO ] The ADRCI package is being created.
2020-05-29 21:21:15.646 UTC [INFO ] Generated package 2 in file /rdsdbdata/log/trace/
ORA700EVE_20200529212111_COM_1.zip, mode complete
2020-05-29 21:21:15.646 UTC [INFO ] The ADRCI package was successfully created.
2020-05-29 21:21:15.646 UTC [INFO ] The task finished successfully.
```

Showing trace files

You can show trace files using the Amazon RDS function `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile`.

This function uses the following parameter.

Parameter name	Data type	Valid values	Default	Required	Description
filename	varchar2	A valid trace file name	Null	No	If the value is null, the function shows all trace files. If it isn't null, the function shows the specified file.

To show the trace file, call the Amazon RDS function `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile` with the `incident_id` parameter.

```
SQL> exec :task_id := rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile;
PL/SQL procedure successfully completed.
```

To list the trace file names, call the Amazon RDS procedure `rdsadmin.rds_file_util.read_text_file`, supplying the task ID as part of the file name.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
'dbtask-'||:task_id||'.log')) WHERE TEXT LIKE '%/alert_%';
```

TEXT

```
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-28
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-27
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-26
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-25
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-24
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-23
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-22
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-21
diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log
```

9 rows selected.

In the following example, you generate output for `alert_ORCL.log`.

```
SQL> exec :task_id := rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile('diag/rdbms/orcl_a/
ORCL/trace/alert_ORCL.log');
PL/SQL procedure successfully completed.
```

To read the log file, call `rdsadmin.rds_file_util.read_text_file`. Supply the task ID as part of the file name. The output shows the first 10 lines of `alert_ORCL.log`.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
  'dbtask-'||:task_id||'.log')) WHERE ROWNUM <= 10;

TEXT
-----
2020-05-29 21:24:02.083 UTC [INFO ] The trace files are being displayed.
2020-05-29 21:24:02.128 UTC [INFO ] Thu May 28 23:59:10 2020
Thread 1 advanced to log sequence 2048 (LGWR switch)
  Current log# 3 seq# 2048 mem# 0: /rdsdbdata/db/ORCL_A/onlinelog/o1_mf_3_hbl2p8xs_.log
Thu May 28 23:59:10 2020
Archived Log entry 2037 added for thread 1 sequence 2047 ID 0x5d62ce43 dest 1:
Fri May 29 00:04:10 2020
Thread 1 advanced to log sequence 2049 (LGWR switch)
  Current log# 4 seq# 2049 mem# 0: /rdsdbdata/db/ORCL_A/onlinelog/o1_mf_4_hbl2qgmh_.log
Fri May 29 00:04:10 2020

10 rows selected.
```

Performing miscellaneous tasks for Oracle DB instances

Following, you can find how to perform miscellaneous DBA tasks on your Amazon RDS DB instances running Oracle. To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances, and restricts access to certain system procedures and tables that require advanced privileges.

Topics

- [Creating and dropping directories in the main data storage space \(p. 1095\)](#)
- [Listing files in a DB instance directory \(p. 1096\)](#)
- [Reading files in a DB instance directory \(p. 1096\)](#)
- [Accessing Opatch files \(p. 1097\)](#)
- [Managing advisor tasks \(p. 1099\)](#)
- [Enabling HugePages for an Oracle DB instance \(p. 1101\)](#)
- [Enabling extended data types \(p. 1103\)](#)

Creating and dropping directories in the main data storage space

To create directories, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.create_directory`. You can create up to 10,000 directories, all located in your main data storage space. To drop directories, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.drop_directory`.

The `create_directory` and `drop_directory` procedures have the following required parameter.

Parameter name	Data type	Default	Required	Description
<code>p_directory_name</code>	varchar2	—	Yes	The name of the directory.

The following example creates a new directory named `PRODUCT_DESCRIPTIONS`.

```
exec rdsadmin.rdsadmin_util.create_directory(p_directory_name => 'product_descriptions');
```

The data dictionary stores the directory name in uppercase. You can list the directories by querying DBA_DIRECTORIES. The system chooses the actual host pathname automatically. The following example gets the directory path for the directory named PRODUCT_DESCRIPTIONS:

```
SELECT DIRECTORY_PATH
  FROM DBA_DIRECTORIES
 WHERE DIRECTORY_NAME='PRODUCT_DESCRIPTIONS';

DIRECTORY_PATH
-----
/rdsdbdata/userdirs/01
```

The master user name for the DB instance has read and write privileges in the new directory, and can grant access to other users. EXECUTE privileges are not available for directories on a DB instance. Directories are created in your main data storage space and will consume space and I/O bandwidth.

The following example drops the directory named PRODUCT_DESCRIPTIONS.

```
exec rdsadmin.rdsadmin_util.drop_directory(p_directory_name => 'product_descriptions');
```

Note

You can also drop a directory by using the Oracle SQL command DROP DIRECTORY.

Dropping a directory doesn't remove its contents. Because the rdsadmin.rdsadmin_util.create_directory procedure can reuse pathnames, files in dropped directories can appear in a newly created directory. Before you drop a directory, we recommend that you use UTL_FILE.FREMOVE to remove files from the directory. For more information, see [FREMOVE procedure](#) in the Oracle documentation.

Listing files in a DB instance directory

To list the files in a directory, use the Amazon RDS procedure rdsadmin.rds_file_util.listdir. The listdir procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
p_directory	varchar2	—	Yes	The name of the directory to list.

The following example lists the files in the directory named PRODUCT_DESCRIPTIONS.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'PRODUCT_DESCRIPTIONS'));
```

Reading files in a DB instance directory

To read a text file, use the Amazon RDS procedure rdsadmin.rds_file_util.read_text_file. The read_text_file procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
p_directory	varchar2	—	Yes	The name of the directory that contains the file.

Parameter name	Data type	Default	Required	Description
p_filename	varchar2	—	Yes	The name of the file to read.

The following example creates the file `rice.txt` in the directory `PRODUCT_DESCRIPTIONS`.

```
declare
    fh sys.utl_file.file_type;
begin
    fh := utl_file=fopen(location=>'PRODUCT_DESCRIPTIONS', filename=>'rice.txt',
    open_mode=>'w');
    utl_file.put(file=>fh, buffer=>'AnyCompany brown rice, 15 lbs');
    utl_file.fclose(file=>fh);
end;
/
```

The following example reads the file `rice.txt` from the directory `PRODUCT_DESCRIPTIONS`.

```
SELECT * FROM TABLE
(rdsadmin.rds_file_util.read_text_file(
    p_directory => 'PRODUCT_DESCRIPTIONS',
    p_filename  => 'rice.txt'));
```

Accessing Opatch files

Opatch is an Oracle utility that enables the application and rollback of patches to Oracle software. The Oracle mechanism for determining which patches have been applied to a database is the `opatch lsinventory` command. To open service requests for Bring Your Own Licence (BYOL) customers, Oracle Support requests the `lsinventory` file and sometimes the `lsinventory_detail` file generated by Opatch.

To deliver a managed service experience, Amazon RDS doesn't provide shell access to Opatch. Instead, the Oracle DB instance automatically creates the inventory files every hour in the `BDUMP` directory. You have read and write access on this directory. If you don't see your files in `BDUMP`, or the files are out of date, wait an hour and then try again.

Note

The examples in this section assume that the `BDUMP` directory is named `BDUMP`. On a read replica, the `BDUMP` directory name is different. To learn how to get the `BDUMP` name by querying `V$DATABASE.DB_UNIQUE_NAME` on a read replica, see [Listing files \(p. 527\)](#).

The inventory files use the Amazon RDS naming convention `lsinventory-dbv.txt` and `lsinventory_detail-dbv.txt`, where `dbv` is the full name of your DB version. The `lsinventory-dbv.txt` file is available on all DB versions. The corresponding detail file is available on the following DB versions:

- 19.0.0.0, ru-2020-01.rur-2020-01.r1 or later
- 18.0.0.0, ru-2020-01.rur-2020-01.r1 or later
- 12.2.0.1, ru-2020-01.rur-2020-01.r1 or later
- 12.1.0.2, v19 or later

For example, if your DB version is 19.0.0.0.ru-2020-04.rur-2020-04.r1, then your inventory files have the following names.

lsinventory-19.0.0.0.ru-2020-04.rur-2020-04.r1.txt

```
lsinventory_detail-19.0.0.0.ru-2020-04.rur-2020-04.r1.txt
```

Ensure that you download the files that match the current version of your DB engine.

Console

To download an inventory file using the console

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the name of the DB instance that has the log file that you want to view.
4. Choose the **Logs & events** tab.
5. Scroll down to the **Logs** section.
6. In the **Logs** section, search for `lsinventory`.
7. Select the file that you want to access, and then choose **Download**.

SQL

To read the `lsinventory-dbv.txt` in a SQL client, you can use a `SELECT` statement. For this technique, use either of the following `rdsadmin` functions: `rdsadmin.rds_file_util.read_text_file` or `rdsadmin.tracefile_listing`.

In the following sample query, replace `dbv` with your Oracle DB version. For example, your DB version might be `19.0.0.0.ru-2020-04.rur-2020-04.r1`.

```
SELECT text
FROM   TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP', 'lsinventory-dbv.txt'));
```

PL/SQL

To read the `lsinventory-dbv.txt` in a SQL client, you can write a PL/SQL program. This program uses `utl_file` to read the file, and `dbms_output` to print it. These are Oracle-supplied packages.

In the following sample program, replace `dbv` with your Oracle DB version. For example, your DB version might be `19.0.0.0.ru-2020-04.rur-2020-04.r1`.

```
SET SERVEROUTPUT ON
DECLARE
  v_file          SYS.UTL_FILE.FILE_TYPE;
  v_line          VARCHAR2(1000);
  v_oracle_home_type  VARCHAR2(1000);
  c_directory    VARCHAR2(30) := 'BDUMP';
  c_output_file   VARCHAR2(30) := 'lsinventory-dbv.txt';
BEGIN
  v_file := SYS.UTL_FILE.FOPEN(c_directory, c_output_file, 'r');
  LOOP
    BEGIN
      SYS.UTL_FILE.GET_LINE(v_file, v_line,1000);
      DBMS_OUTPUT.PUT_LINE(v_line);
    EXCEPTION
      WHEN no_data_found THEN
        EXIT;
    END;
  END LOOP;
END;
/
```

Or query `rdsadmin.tracefile_listing`, and spool the output to a file. The following example spools the output to `/tmp/tracefile.txt`.

```
SPOOL /tmp/tracefile.txt
SELECT *
FROM   rdsadmin.tracefile_listing
WHERE  FILENAME LIKE 'lsinventory%';
SPOOL OFF;
```

Managing advisor tasks

Oracle Database includes a number of advisors. Each advisor supports automated and manual tasks. You can use procedures in the `rdsadmin.rdsadmin_util` package to manage some advisor tasks.

The advisor task procedures are available in the following engine versions:

- Version 19.0.0.0.ru-2021-01.rur-2021-01.r1 (p. 1256) or higher 19c versions
- Version 18.0.0.0.ru-2021-01.rur-2021-01.r1 (p. 1293) or higher 18c versions
- Version 12.2.0.1.ru-2021-01.rur-2021-01.r1 (p. 1321) or higher 12.2.0.1 versions

Topics

- [Setting parameters for advisor tasks \(p. 1099\)](#)
- [Disabling AUTO_STATS_ADVISOR_TASK \(p. 1100\)](#)
- [Re-enabling AUTO_STATS_ADVISOR_TASK \(p. 1101\)](#)

Setting parameters for advisor tasks

To set parameters for some advisor tasks, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.advisor_task_set_parameter`. The `advisor_task_set_parameter` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>p_task_name</code>	varchar2	—	Yes	<p>The name of the advisor task whose parameters you want to change. The following values are valid:</p> <ul style="list-style-type: none"> • <code>AUTO_STATS_ADVISOR_TASK</code> • <code>INDIVIDUAL_STATS_ADVISOR_TASK</code> • <code>SYS_AUTO_SPM_EVOLVE_TASK</code> • <code>SYS_AUTO_SQL_TUNING_TASK</code>
<code>p_parameter</code>	varchar2	—	Yes	<p>The name of the task parameter. To find valid parameters for an advisor task, run the following query. Substitute <code>p_task_name</code> with a valid value for <code>p_task_name</code>:</p> <pre>COL PARAMETER_NAME FORMAT a30 COL PARAMETER_VALUE FORMAT a30 SELECT PARAMETER_NAME, PARAMETER_VALUE FROM DBA_ADVISOR_PARAMETERS WHERE TASK_NAME='p_task_name' AND PARAMETER_VALUE != 'UNUSED'</pre>

Parameter name	Data type	Default	Required	Description
				ORDER BY PARAMETER_NAME;
p_value	varchar2	—	Yes	<p>The value for a task parameter. To find valid values for task parameters, run the following query. Substitute <i>p_task_name</i> with a valid value for p_task_name:</p> <pre>COL PARAMETER_NAME FORMAT a30 COL PARAMETER_VALUE FORMAT a30 SELECT PARAMETER_NAME, PARAMETER_VALUE FROM DBA_ADVISOR_PARAMETERS WHERE TASK_NAME='<i>p_task_name</i>' AND PARAMETER_VALUE != 'UNUSED' ORDER BY PARAMETER_NAME;</pre>

The following PL/SQL program sets ACCEPT_PLANS to FALSE for SYS_AUTO_SPM_EVOLVE_TASK. The SQL Plan Management automated task verifies the plans and generates a report of its findings, but does not evolve the plans automatically. You can use a report to identify new SQL plan baselines and accept them manually.

```
BEGIN
  rdsadmin.rdsadmin_util.advisor_task_set_parameter(
    p_task_name => 'SYS_AUTO_SPM_EVOLVE_TASK',
    p_parameter => 'ACCEPT_PLANS',
    p_value      => 'FALSE');
END;
```

The following PL/SQL program sets EXECUTION_DAYS_TO_EXPIRE to 10 for AUTO_STATS_ADVISOR_TASK. The predefined task AUTO_STATS_ADVISOR_TASK runs automatically in the maintenance window once per day. The example sets the retention period for the task execution to 10 days.

```
BEGIN
  rdsadmin.rdsadmin_util.advisor_task_set_parameter(
    p_task_name => 'AUTO_STATS_ADVISOR_TASK',
    p_parameter => 'EXECUTION_DAYS_TO_EXPIRE',
    p_value      => '10');
END;
```

Disabling AUTO_STATS_ADVISOR_TASK

To disable AUTO_STATS_ADVISOR_TASK, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.advisor_task_drop`. The `advisor_task_drop` procedure accepts the following parameter.

Note

This procedure is available in Oracle Database 12c Release 2 (12.2.0.1) and later.

Parameter name	Data type	Default	Required	Description
p_task_name	varchar2	—	Yes	The name of the advisor task to be disabled. The only valid value is AUTO_STATS_ADVISOR_TASK.

The following command drops AUTO_STATS_ADVISOR_TASK.

```
EXEC rdsadmin.rdsadmin_util.advisor_task_drop('AUTO_STATS_ADVISOR_TASK')
```

You can re-enabling AUTO_STATS_ADVISOR_TASK using
`rdsadmin.rdsadmin_util.dbms_stats_init`.

Re-enabling AUTO_STATS_ADVISOR_TASK

To re-enable AUTO_STATS_ADVISOR_TASK, use the Amazon RDS procedure `rdsadmin.rdsadmin_util.dbms_stats_init`. The `dbms_stats_init` procedure takes no parameters.

The following command re-enables AUTO_STATS_ADVISOR_TASK.

```
EXEC rdsadmin.rdsadmin_util.dbms_stats_init()
```

Enabling HugePages for an Oracle DB instance

Amazon RDS for Oracle supports Linux kernel HugePages for increased database scalability. HugePages results in smaller page tables and less CPU time spent on memory management, increasing the performance of large database instances. For more information, see [Overview of HugePages](#) in the Oracle documentation.

You can use HugePages with the following versions and editions of Oracle Database:

- 19.0.0.0, all editions
- 18.0.0.0, all editions
- 12.2.0.1, all editions
- 12.1.0.2, all editions

The `use_large_pages` parameter controls whether HugePages are enabled for a DB instance. The possible settings for this parameter are `ONLY`, `FALSE`, and `{DBInstanceClassHugePagesDefault}`. The `use_large_pages` parameter is set to `{DBInstanceClassHugePagesDefault}` in the default DB parameter group for Oracle.

To control whether HugePages are enabled for a DB instance automatically, you can use the `DBInstanceClassHugePagesDefault` formula variable in parameter groups. The value is determined as follows:

- For the DB instance classes mentioned in the table following, `DBInstanceClassHugePagesDefault` always evaluates to `FALSE` by default, and `use_large_pages` evaluates to `FALSE`. You can enable HugePages manually for these DB instance classes if the DB instance class has at least 14 GiB of memory.
- For DB instance classes not mentioned in the table following, if the DB instance class has less than 14 GiB of memory, `DBInstanceClassHugePagesDefault` always evaluates to `FALSE`. Also, `use_large_pages` evaluates to `FALSE`.
- For DB instance classes not mentioned in the table following, if the instance class has at least 14 GiB of memory and less than 100 GiB of memory, `DBInstanceClassHugePagesDefault` evaluates to `TRUE` by default. Also, `use_large_pages` evaluates to `ONLY`. You can disable HugePages manually by setting `use_large_pages` to `FALSE`.
- For DB instance classes not mentioned in the table following, if the instance class has at least 100 GiB of memory, `DBInstanceClassHugePagesDefault` always evaluates to `TRUE`. Also, `use_large_pages` evaluates to `ONLY` and HugePages can't be disabled.

HugePages are not enabled by default for the following DB instance classes.

DB instance class family	DB instance classes with HugePages not enabled by default
db.m5	db.m5.large
db.m4	db.m4.large, db.m4.xlarge, db.m4.2xlarge, db.m4.4xlarge, db.m4.10xlarge
db.t3	db.t3.micro, db.t3.small, db.t3.medium, db.t3.large

For more information about DB instance classes, see [Hardware specifications for DB instance classes \(p. 33\)](#).

To enable HugePages for new or existing DB instances manually, set the `use_large_pages` parameter to `ONLY`. You can't use HugePages with Oracle Automatic Memory Management (AMM). If you set the parameter `use_large_pages` to `ONLY`, then you must also set both `memory_target` and `memory_max_target` to 0. For more information about setting DB parameters for your DB instance, see [Working with DB parameter groups \(p. 228\)](#).

You can also set the `sga_target`, `sga_max_size`, and `pga_aggregate_target` parameters. When you set system global area (SGA) and program global area (PGA) memory parameters, add the values together. Subtract this total from your available instance memory (`DBInstanceClassMemory`) to determine the free memory beyond the HugePages allocation. You must leave free memory of at least 2 GiB, or 10 percent of the total available instance memory, whichever is smaller.

After you configure your parameters, you must reboot your DB instance for the changes to take effect. For more information, see [Rebooting a DB instance \(p. 276\)](#).

Note

The Oracle DB instance defers changes to SGA-related initialization parameters until you reboot the instance without failover. In the Amazon RDS console, choose **Reboot** but *do not* choose **Reboot with failover**. In the AWS CLI, call the `reboot-db-instance` command with the `--no-force-failover` parameter. The DB instance does not process the SGA-related parameters during failover or during other maintenance operations that cause the instance to restart.

The following is a sample parameter configuration for HugePages that enables HugePages manually. You should set the values to meet your needs.

```
memory_target          = 0
memory_max_target     = 0
pga_aggregate_target = {DBInstanceClassMemory*1/8}
sga_target            = {DBInstanceClassMemory*3/4}
sga_max_size          = {DBInstanceClassMemory*3/4}
use_large_pages       = ONLY
```

Assume the following parameters values are set in a parameter group.

```
memory_target          = IF({DBInstanceClassHugePagesDefault}, 0,
{DBInstanceClassMemory*3/4})
memory_max_target      = IF({DBInstanceClassHugePagesDefault}, 0,
{DBInstanceClassMemory*3/4})
pga_aggregate_target  = IF({DBInstanceClassHugePagesDefault},
{DBInstanceClassMemory*1/8}, 0)
sga_target             = IF({DBInstanceClassHugePagesDefault},
{DBInstanceClassMemory*3/4}, 0)
```

```

sga_max_size      = IF({DBInstanceClassHugePagesDefault},
{DBInstanceClassMemory*3/4}, 0)
use_large_pages   = {DBInstanceClassHugePagesDefault}

```

The parameter group is used by a db.r4 DB instance class with less than 100 GiB of memory. With these parameter settings and `use_large_pages` set to `{DBInstanceClassHugePagesDefault}`, HugePages are enabled on the db.r4 instance.

Consider another example with following parameters values set in a parameter group.

```

memory_target      = IF({DBInstanceClassHugePagesDefault}, 0,
{DBInstanceClassMemory*3/4})
memory_max_target  = IF({DBInstanceClassHugePagesDefault}, 0,
{DBInstanceClassMemory*3/4})
pga_aggregate_target = IF({DBInstanceClassHugePagesDefault},
{DBInstanceClassMemory*1/8}, 0)
sga_target         = IF({DBInstanceClassHugePagesDefault},
{DBInstanceClassMemory*3/4}, 0)
sga_max_size       = IF({DBInstanceClassHugePagesDefault},
{DBInstanceClassMemory*3/4}, 0)
use_large_pages    = FALSE

```

The parameter group is used by a db.r4 DB instance class and a db.r5 DB instance class, both with less than 100 GiB of memory. With these parameter settings, HugePages are disabled on the db.r4 and db.r5 instance.

Note

If this parameter group is used by a db.r4 DB instance class or db.r5 DB instance class with at least 100 GiB of memory, the `FALSE` setting for `use_large_pages` is overridden and set to `ONLY`. In this case, a customer notification regarding the override is sent.

After HugePages are active on your DB instance, you can view HugePages information by enabling enhanced monitoring. For more information, see [Using Enhanced Monitoring \(p. 471\)](#).

Enabling extended data types

Amazon RDS Oracle Database 12c supports extended data types. With extended data types, the maximum size is 32,767 bytes for the `VARCHAR2`, `NVARCHAR2`, and `RAW` data types. To use extended data types, set the `MAX_STRING_SIZE` parameter to `EXTENDED`. For more information, see [Extended data types](#) in the Oracle documentation.

If you don't want to use extended data types, keep the `MAX_STRING_SIZE` parameter set to `STANDARD` (the default). When this parameter is set to `STANDARD`, the size limits are 4,000 bytes for the `VARCHAR2` and `NVARCHAR2` data types, and 2,000 bytes for the `RAW` data type.

You can enable extended data types on a new or existing DB instance. For new DB instances, DB instance creation time is typically longer when you enable extended data types. For existing DB instances, the DB instance is unavailable during the conversion process.

The following are considerations for a DB instance with extended data types enabled:

- When you enable extended data types for a DB instance, you can't change the DB instance back to use the standard size for data types. After a DB instance is converted to use extended data types, if you set the `MAX_STRING_SIZE` parameter back to `STANDARD` it results in the `incompatible-parameters` status.
- When you restore a DB instance that uses extended data types, you must specify a parameter group with the `MAX_STRING_SIZE` parameter set to `EXTENDED`. During restore, if you specify the default

parameter group or any other parameter group with `MAX_STRING_SIZE` set to `STANDARD` it results in the `incompatible-parameters` status.

- We recommend that you don't enable extended data types for Oracle DB instances running on the t2.micro DB instance class.

When the DB instance status is `incompatible-parameters` because of the `MAX_STRING_SIZE` setting, the DB instance remains unavailable until you set the `MAX_STRING_SIZE` parameter to `EXTENDED` and reboot the DB instance.

Enabling extended data types for a new DB instance

To enable extended data types for a new DB instance

1. Set the `MAX_STRING_SIZE` parameter to `EXTENDED` in a parameter group.

To set the parameter, you can either create a new parameter group or modify an existing parameter group.

For more information, see [Working with DB parameter groups \(p. 228\)](#).

2. Create a new Amazon RDS Oracle DB instance, and associate the parameter group with `MAX_STRING_SIZE` set to `EXTENDED` with the DB instance.

For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

Enabling extended data types for an existing DB instance

When you modify a DB instance to enable extended data types, the data in the database is converted to use the extended sizes. The DB instance is unavailable during the conversion. The amount of time it takes to convert the data depends on the DB instance class used by the DB instance and the size of the database.

Note

After you enable extended data types, you can't perform a point-in-time restore to a time during the conversion. You can restore to the time immediately before the conversion or after the conversion.

To enable extended data types for an existing DB instance

1. Take a snapshot of the database.

If there are invalid objects in the database, Amazon RDS tries to recompile them. The conversion to extended data types can fail if Amazon RDS can't recompile an invalid object. The snapshot enables you to restore the database if there is a problem with the conversion. Always check for invalid objects before conversion and fix or drop those invalid objects. For production databases, we recommend testing the conversion process on a copy of your DB instance first.

For more information, see [Creating a DB snapshot \(p. 346\)](#).

2. Set the `MAX_STRING_SIZE` parameter to `EXTENDED` in a parameter group.

To set the parameter, you can either create a new parameter group or modify an existing parameter group.

For more information, see [Working with DB parameter groups \(p. 228\)](#).

3. Modify the DB instance to associate it with the parameter group with `MAX_STRING_SIZE` set to `EXTENDED`.

For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

4. Reboot the DB instance for the parameter change to take effect.

For more information, see [Rebooting a DB instance \(p. 276\)](#).

Importing data into Oracle on Amazon RDS

How you import data into an Amazon RDS DB instance depends on the following:

- The amount of data you have
- The number of database objects in your database
- The variety of database objects in your database

For example, you can use Oracle SQL Developer to import a simple, 20 MB database. You can use Oracle Data Pump to import complex databases, or databases that are several hundred megabytes or several terabytes in size.

You can also use AWS Database Migration Service (AWS DMS) to import data into an Amazon RDS DB instance. AWS DMS can migrate databases without downtime. For many database engines, ongoing replication can continue until you are ready to switch over to the target database. You can migrate to Oracle from either the same database engine or a different database engine using AWS DMS. If you are migrating from a different database engine, you can use the AWS Schema Conversion Tool to migrate schema objects that are not migrated by AWS DMS. For more information about AWS DMS, see [What is AWS Database Migration Service](#).

Before you use any of these migration techniques, we recommend that you back up your database. After you import the data, you can back up your Amazon RDS DB instances by creating snapshots. Later, you can restore the snapshots. For more information, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#).

Note

You can also import data into Oracle using files from Amazon S3. For example, you can download Data Pump files from Amazon S3 to the DB instance host. For more information, see [Amazon S3 integration \(p. 1127\)](#).

Importing using Oracle SQL Developer

For small databases, you can use Oracle SQL Developer, a graphical Java tool distributed without cost by Oracle. You can install this tool on your desktop computer (Windows, Linux, or Mac) or on one of your servers. Oracle SQL Developer provides options for migrating data between two Oracle databases, or for migrating data from other databases, such as MySQL, to Oracle. Oracle SQL Developer is best suited for migrating small databases. We recommend that you read the Oracle SQL Developer product documentation before you begin migrating your data.

After you install SQL Developer, you can use it to connect to your source and target databases. Use the **Database Copy** command on the Tools menu to copy your data to your Amazon RDS instance.

To download Oracle SQL Developer, go to <http://www.oracle.com/technetwork/developer-tools/sql-developer>.

Oracle also has documentation on how to migrate from other databases, including MySQL and SQL Server. For more information, see <http://www.oracle.com/technetwork/database/migration> in the Oracle documentation.

Importing using Oracle Data Pump

Oracle Data Pump is a long-term replacement for the Oracle Export/Import utilities. Oracle Data Pump is the preferred way to move large amounts of data from an Oracle installation to an Amazon RDS DB instance. You can use Oracle Data Pump for several scenarios:

- Import data from an Oracle database (either on-premises or Amazon EC2 instance) to an Amazon RDS for Oracle DB instance.
- Import data from an RDS for Oracle DB instance to an Oracle database (either on-premises or Amazon EC2 instance).
- Import data between RDS for Oracle DB instances (for example, to migrate data from EC2-Classic to VPC).

To download Oracle Data Pump utilities, see [Oracle database software downloads](#) on the Oracle Technology Network website.

For compatibility considerations when migrating between versions of Oracle Database, see [the Oracle documentation](#).

When you import data with Oracle Data Pump, you must transfer the dump file that contains the data from the source database to the target database. You can transfer the dump file using an Amazon S3 bucket or by using a database link between the two databases.

When you use Oracle Data Pump to import data into an Oracle DB instance, we recommend the following best practices:

- Perform imports in schema or table mode to import specific schemas and objects.
- Limit the schemas you import to those required by your application.
- Don't import in full mode.

Because Amazon RDS for Oracle does not allow access to SYS or SYSDBA administrative users, importing in full mode, or importing schemas for Oracle-maintained components, might damage the Oracle data dictionary and affect the stability of your database.

- When loading large amounts of data, transfer the dump file to the target Amazon RDS for Oracle DB instance, take a DB snapshot of your instance, and then test the import to verify that it succeeds. If database components are invalidated, you can delete the DB instance and re-create it from the DB snapshot. The restored DB instance includes any dump files staged on the DB instance when you took the DB snapshot.
- Don't import dump files that were created using the Oracle Data Pump export parameters TRANSPORT_TABLESPACES, TRANSPORTABLE, or TRANSPORT_FULL_CHECK. Amazon RDS for Oracle DB instances don't support importing these dump files.
- Don't import dump files that contain Oracle Scheduler objects (jobs, programs, and schedules) in the schemas SYS, SYSTEM, RDSADMIN, RDSSEC, and RDS_DATAGUARD. Amazon RDS for Oracle DB instances do not support importing these dump files.

Note

To exclude unsupported Scheduler objects, use additional directives during the Data Pump export. If you use DBMS_DATAPUMP, add an additional METADATA_FILTER before the DBMS_METADATA.START_JOB:

```
DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'EXCLUDE_NAME_EXPR',
q'[IN (SELECT NAME FROM SYS.OBJ$
      WHERE TYPE# IN (66,67,74)
      AND OWNER# IN
          (SELECT USER# FROM SYS.USER$
            WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
          )
      )]', 'PROCOBJ');
```

If you use expdp, create a parameter file that contains the exclude directive shown in the following example. Then use PARFILE=*parameter_file* with your expdp command.

```
exclude=procobj:"IN
  (SELECT NAME FROM sys.OBJ$
  WHERE TYPE# IN (66,67,74)
  AND OWNER# IN
    (SELECT USER# FROM SYS.USER$
    WHERE NAME IN ('RDSADMIN', 'SYS', 'SYSTEM', 'RDS_DATAGUARD', 'RDSSEC')
  )
 )"
```

The examples in this section show one way to import data into an Oracle database. However, Oracle Data Pump permits many ways to import data. To learn more about Oracle Data Pump, see [the Oracle documentation](#).

The examples in this section use the `DBMS_DATAPUMP` package. The same tasks can be accomplished by using the Oracle Data Pump command line utilities `impdp` and `expdp`. You can install these utilities on a remote host as part of an Oracle Client installation, including Oracle Instant Client.

Topics

- [Importing data with Oracle Data Pump and an Amazon S3 bucket \(p. 1108\)](#)
- [Importing data with Oracle Data Pump and a database link \(p. 1111\)](#)

Importing data with Oracle Data Pump and an Amazon S3 bucket

The following import process uses Oracle Data Pump and an Amazon S3 bucket. The process exports data on the source database using the Oracle `DBMS_DATAPUMP` package and puts the dump file in an Amazon S3 bucket. It then downloads the dump file from the Amazon S3 bucket to the `DATA_PUMP_DIR` directory on the target Amazon RDS for Oracle DB instance. The final step imports the data from the copied dump file into the Amazon RDS for Oracle DB instance using the package `DBMS_DATAPUMP`.

The process has the following requirements:

- You must have an Amazon S3 bucket available for file transfers, and the Amazon S3 bucket must be in the same AWS Region as the DB instance. For instructions, see [Create a bucket](#) in the *Amazon Simple Storage Service Getting Started Guide*.
- The object that you upload into the Amazon S3 bucket must be 5 TB or less. For more information about working with objects in Amazon S3, see [Amazon Simple Storage Service Developer Guide](#).

Note

If you dump file exceeds 5 TB, you can run the Oracle Data Pump export with the parallel option. This operation spreads the data into multiple dump files so that you do not exceed the 5 TB limit for individual files.

- You must prepare the Amazon S3 bucket for Amazon RDS integration by following the instructions in [Prerequisites for Amazon RDS for Oracle integration with Amazon S3 \(p. 1127\)](#).
- You must ensure that you have enough storage space to store the dump file on the source instance and the target DB instance.

Note

This process imports a dump file into the `DATA_PUMP_DIR` directory, a preconfigured directory on all Oracle DB instances. This directory is located on the same storage volume as your data files. When you import the dump file, the existing Oracle data files use more space. Thus, you should make sure that your DB instance can accommodate that additional use of space. The imported dump file is not automatically deleted or purged from the `DATA_PUMP_DIR` directory. To remove the imported dump file, use `UTL_FILE.FREMOVE`, found on the Oracle website.

The import process using Oracle Data Pump and an Amazon S3 bucket has the following steps.

Topics

- [Step 1: Grant privileges to the user on the Amazon RDS target instance \(p. 1109\)](#)
- [Step 2: Use DBMS_DATAPUMP to create a dump file \(p. 1109\)](#)
- [Step 3: Upload the dump file to your Amazon S3 bucket \(p. 1110\)](#)
- [Step 4: Copy the exported dump file from the Amazon S3 bucket to the target DB instance \(p. 1110\)](#)
- [Step 5: Use DBMS_DATAPUMP to import the data file on the target DB instance \(p. 1110\)](#)
- [Step 6: Clean up \(p. 1111\)](#)

Step 1: Grant privileges to the user on the Amazon RDS target instance

To grant privileges to the user on the RDS target instance, take the following steps:

1. Use SQL Plus or Oracle SQL Developer to connect to the Amazon RDS target Oracle DB instance into which the data will be imported. Connect as the Amazon RDS master user. For information about connecting to the DB instance, see [Connecting to your Oracle DB instance \(p. 1001\)](#).
2. Create the required tablespaces before you import the data. For more information, see [Creating and sizing tablespaces \(p. 1050\)](#).
3. If the user account into which the data is imported doesn't exist, create the user account and grant the necessary permissions and roles. If you plan to import data into multiple user schemas, create each user account and grant the necessary privileges and roles to it.

For example, the following commands create a new user and grant the necessary permissions and roles to import the data into the user's schema.

```
CREATE USER schema_1 IDENTIFIED BY <password>;
GRANT CREATE SESSION, RESOURCE TO schema_1;
ALTER USER schema_1 QUOTA 100M ON users;
```

This example grants the new user the CREATE SESSION privilege and the RESOURCE role. Additional privileges and roles might be required depending on the database objects that you import.

Note

Replace *schema_1* with the name of your schema in this step and in the following steps.

Step 2: Use DBMS_DATAPUMP to create a dump file

Use SQL Plus or Oracle SQL Developer to connect to the source Oracle instance with an administrative user. If the source database is an Amazon RDS for Oracle DB instance, connect with the Amazon RDS master user. Next, use the Oracle Data Pump utility to create a dump file.

The following script creates a dump file named *sample.dmp* in the DATA_PUMP_DIR directory that contains the *SCHEMA_1* schema. Replace *SCHEMA_1* with the name of the schema that you want to export.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(operation => 'EXPORT', job_mode => 'SCHEMA',
  job_name=>null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample.dmp',
    directory => 'DATA_PUMP_DIR',
```

```

filetype  => dbms_datapump.ku$_file_type_dump_file);
DBMS_DATAPUMP.ADD_FILE(
  handle    => v_hdnl,
  filename  => 'sample_exp.log',
  directory => 'DATA_PUMP_DIR',
  filetype  => dbms_datapump.ku$_file_type_log_file);
DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1''));
DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'EXCLUDE_NAME_EXPR',
  q'[IN (SELECT NAME FROM sys.OBJ$ WHERE TYPE# IN (66,67,74) AND OWNER# IN
  (SELECT USER# FROM SYS.USER$ WHERE NAME IN
  ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')))]','PROCOBJ');
DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/

```

Note

Data Pump jobs are started asynchronously. For information about monitoring a Data Pump job, see [Monitoring job status](#) in the Oracle documentation. You can view the contents of the export log by using the `rdsadmin.rds_file_util.read_text_file` procedure. For more information, see [Reading files in a DB instance directory \(p. 1096\)](#).

Step 3: Upload the dump file to your Amazon S3 bucket

Upload the dump file to the Amazon S3 bucket.

Use the Amazon RDS procedure `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` to copy the dump file to the Amazon S3 bucket. The following example uploads all of the files from the `DATA_PUMP_DIR` directory to an Amazon S3 bucket named `mys3bucket`.

```

SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
  p_bucket_name    => 'mys3bucket',
  p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;

```

The SELECT statement returns the ID of the task in a VARCHAR2 data type.

For more information, see [Uploading files from an Oracle DB instance to an Amazon S3 bucket \(p. 1134\)](#).

Step 4: Copy the exported dump file from the Amazon S3 bucket to the target DB instance

Use SQL Plus or Oracle SQL Developer to connect to the Amazon RDS target Oracle DB instance. Next, use the Amazon RDS procedure `rdsadmin.rdsadmin_s3_tasks.download_from_s3` to copy the dump file from the Amazon S3 bucket to the target DB instance. The following example downloads all of the files from an Amazon S3 bucket named `mys3bucket` to the `DATA_PUMP_DIR` directory.

```

SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
  p_bucket_name    => 'mys3bucket',
  p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;

```

The SELECT statement returns the ID of the task in a VARCHAR2 data type.

For more information, see [Downloading files from an Amazon S3 bucket to an Oracle DB instance \(p. 1136\)](#).

Step 5: Use DBMS_DATAPUMP to import the data file on the target DB instance

Use Oracle Data Pump to import the schema in the DB instance. Additional options such as `METADATA_REMAP` might be required.

Connect to the DB instance with the Amazon RDS master user account to perform the import.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'SCHEMA',
    job_name   => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_copied.dmp',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_imp.log',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1''));
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

Note

Data Pump jobs are started asynchronously. For information about monitoring a Data Pump job, see [Monitoring job status](#) in the Oracle documentation. You can view the contents of the import log by using the `rdsadmin.rds_file_util.read_text_file` procedure. For more information, see [Reading files in a DB instance directory \(p. 1096\)](#).

You can verify the data import by viewing the user's tables on the DB instance. For example, the following query returns the number of tables for `SCHEMA_1`.

```
SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER='SCHEMA_1';
```

Step 6: Clean up

After the data has been imported, you can delete the files that you don't want to keep. You can list the files in the `DATA_PUMP_DIR` using the following command.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('DATA_PUMP_DIR')) ORDER BY MTIME;
```

To delete files in the `DATA_PUMP_DIR` that you no longer require, use the following command.

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','<file name>');
```

For example, the following command deletes the file named "sample_copied.dmp".

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','sample_copied.dmp');
```

Importing data with Oracle Data Pump and a database link

The following import process uses Oracle Data Pump and the Oracle `DBMS_FILE_TRANSFER` package. The process connects to a source Oracle instance, which can be an on-premises or Amazon EC2 instance, or an Amazon RDS for Oracle DB instance. The process then exports data using the `DBMS_DATAPUMP` package. Next, it uses the `DBMS_FILE_TRANSFER.PUT_FILE` method to copy the dump file from the Oracle instance to the `DATA_PUMP_DIR` directory on the target Amazon RDS for Oracle DB instance that

is connected using a database link. The final step imports the data from the copied dump file into the Amazon RDS for Oracle DB instance using the `DBMS_DATAPUMP` package.

The process has the following requirements:

- You must have execute privileges on the `DBMS_FILE_TRANSFER` and `DBMS_DATAPUMP` packages.
- You must have write privileges to the `DATA_PUMP_DIR` directory on the source DB instance.
- You must ensure that you have enough storage space to store the dump file on the source instance and the target DB instance.

Note

This process imports a dump file into the `DATA_PUMP_DIR` directory, a preconfigured directory on all Oracle DB instances. This directory is located on the same storage volume as your data files. When you import the dump file, the existing Oracle data files use more space. Thus, you should make sure that your DB instance can accommodate that additional use of space. The imported dump file is not automatically deleted or purged from the `DATA_PUMP_DIR` directory. To remove the imported dump file, use [UTL_FILE.FREMOVE](#), found on the Oracle website.

The import process using Oracle Data Pump and the `DBMS_FILE_TRANSFER` package has the following steps.

Topics

- [Step 1: Grant privileges to the user on the Amazon RDS target instance \(p. 1112\)](#)
- [Step 2: Grant privileges to the user on the source database \(p. 1113\)](#)
- [Step 3: Use DBMS_DATAPUMP to create a dump file \(p. 1113\)](#)
- [Step 4: Create a database link to the target DB instance \(p. 1114\)](#)
- [Step 5: Use DBMS_FILE_TRANSFER to copy the exported dump file to the target DB instance \(p. 1114\)](#)
- [Step 6: Use DBMS_DATAPUMP to import the data file to the target DB instance \(p. 1114\)](#)
- [Step 7: Clean up \(p. 1115\)](#)

Step 1: Grant privileges to the user on the Amazon RDS target instance

To grant privileges to the user on the RDS target instance, take the following steps:

1. Use SQL Plus or Oracle SQL Developer to connect to the Amazon RDS target Oracle DB instance into which the data will be imported. Connect as the Amazon RDS master user. For information about connecting to the DB instance, see [Connecting to your Oracle DB instance \(p. 1001\)](#).
2. Create the required tablespaces before you import the data. For more information, see [Creating and sizing tablespaces \(p. 1050\)](#).
3. If the user account into which the data is imported doesn't exist, create the user account and grant the necessary permissions and roles. If you plan to import data into multiple user schemas, create each user account and grant the necessary privileges and roles to it.

For example, the following commands create a new user and grant the necessary permissions and roles to import the data into the user's schema.

```
CREATE USER schema_1 IDENTIFIED BY <password>;
GRANT CREATE SESSION, RESOURCE TO schema_1;
ALTER USER schema_1 QUOTA 100M ON users;
```

This example grants the new user the `CREATE SESSION` privilege and the `RESOURCE` role. Additional privileges and roles might be required depending on the database objects that you import.

Note

Replace *schema_1* with the name of your schema in this step and in the following steps.

Step 2: Grant privileges to the user on the source database

Use SQL*Plus or Oracle SQL Developer to connect to the Oracle instance that contains the data to be imported. If necessary, create a user account and grant the necessary permissions.

Note

If the source database is an Amazon RDS instance, you can skip this step. You use your Amazon RDS master user account to perform the export.

The following commands create a new user and grant the necessary permissions.

```
CREATE USER export_user IDENTIFIED BY <password>;
GRANT CREATE SESSION, CREATE TABLE, CREATE DATABASE LINK TO export_user;
ALTER USER export_user QUOTA 100M ON users;
GRANT READ, WRITE ON DIRECTORY data_pump_dir TO export_user;
GRANT SELECT_CATALOG_ROLE TO export_user;
GRANT EXECUTE ON DBMS_DATAPUMP TO export_user;
GRANT EXECUTE ON DBMS_FILE_TRANSFER TO export_user;
```

Step 3: Use DBMS_DATAPUMP to create a dump file

Use SQL Plus or Oracle SQL Developer to connect to the source Oracle instance with an administrative user or with the user you created in step 2. If the source database is an Amazon RDS for Oracle DB instance, connect with the Amazon RDS master user. Next, use the Oracle Data Pump utility to create a dump file.

The following script creates a dump file named *sample.dmp* in the DATA_PUMP_DIR directory.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'EXPORT',
    job_mode  => 'SCHEMA',
    job_name   => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample.dmp',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_exp.log',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1''));
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'EXCLUDE_NAME_EXPR',
    q'[IN (SELECT NAME FROM sys.OBJ$ WHERE TYPE# IN (66,67,74) AND OWNER# IN
      (SELECT USER# FROM SYS.USER$ WHERE NAME IN
        ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')))]','PROCOBJ');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

Note

Data Pump jobs are started asynchronously. For information about monitoring a Data Pump job, see [Monitoring job status](#) in the Oracle documentation. You can view the contents of the

export log by using the `rdsadmin.rds_file_util.read_text_file` procedure. For more information, see [Reading files in a DB instance directory \(p. 1096\)](#).

Step 4: Create a database link to the target DB instance

Create a database link between your source instance and your target DB instance. Your local Oracle instance must have network connectivity to the DB instance in order to create a database link and to transfer your export dump file.

Perform this step connected with the same user account as the previous step.

If you are creating a database link between two DB instances inside the same VPC or peered VPCs, the two DB instances should have a valid route between them. The security group of each DB instance must allow ingress to and egress from the other DB instance. The security group inbound and outbound rules can refer to security groups from the same VPC or a peered VPC. For more information, see [Adjusting database links for use with DB instances in a VPC \(p. 1057\)](#).

The following command creates a database link named `to_rds` that connects to the Amazon RDS master user at the target DB instance.

```
CREATE DATABASE LINK to_rds
  CONNECT TO <master_user_account> IDENTIFIED BY <password>
  USING '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<dns or ip address of remote db>)
  (PORT=<listener port>))(CONNECT_DATA=(SID=<remote SID>)))';
```

Step 5: Use DBMS_FILE_TRANSFER to copy the exported dump file to the target DB instance

Use `DBMS_FILE_TRANSFER` to copy the dump file from the source database instance to the target DB instance. The following script copies a dump file named `sample.dmp` from the source instance to a target database link named `to_rds` (created in the previous step).

```
BEGIN
  DBMS_FILE_TRANSFER.PUT_FILE(
    source_directory_object      => 'DATA_PUMP_DIR',
    source_file_name             => 'sample.dmp',
    destination_directory_object => 'DATA_PUMP_DIR',
    destination_file_name        => 'sample_copied.dmp',
    destination_database         => 'to_rds' );
END;
/
```

Step 6: Use DBMS_DATAPUMP to import the data file to the target DB instance

Use Oracle Data Pump to import the schema in the DB instance. Additional options such as `METADATA_REMAP` might be required.

Connect to the DB instance with the Amazon RDS master user account to perform the import.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'SCHEMA',
    job_name   => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle     => v_hdnl,
    filename   => 'sample_copied.dmp',
```

```

        directory => 'DATA_PUMP_DIR',
        filetype  => dbms_datapump.ku$_file_type_dump_file );
DBMS_DATAPUMP.ADD_FILE(
    handle   => v_hdnl,
    filename => 'sample_imp.log',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_log_file);
DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1''));
DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/

```

Note

Data Pump jobs are started asynchronously. For information about monitoring a Data Pump job, see [Monitoring job status](#) in the Oracle documentation. You can view the contents of the import log by using the `rdsadmin.rds_file_util.read_text_file` procedure. For more information, see [Reading files in a DB instance directory \(p. 1096\)](#).

You can verify the data import by viewing the user's tables on the DB instance. For example, the following query returns the number of tables for `SCHEMA_1`.

```
SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER='SCHEMA_1';
```

Step 7: Clean up

After the data has been imported, you can delete the files that you don't want to keep. You can list the files in `DATA_PUMP_DIR` using the following command.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('DATA_PUMP_DIR')) ORDER BY MTIME;
```

To delete files in `DATA_PUMP_DIR` that you no longer require, use the following command.

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','<file name>');
```

For example, the following command deletes the file named "sample_copied.dmp".

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','sample_copied.dmp');
```

Oracle Export/Import utilities

The Oracle Export/Import utilities are best suited for migrations where the data size is small and data types such as binary float and double are not required. The import process creates the schema objects so you do not need to run a script to create them beforehand, making this process well suited for databases with small tables. The following example demonstrates how these utilities can be used to export and import specific tables.

To download Oracle export and import utilities, go to <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>.

Export the tables from the source database using the command below. Substitute `username/password` as appropriate.

```
exp cust_dba@ORCL FILE=exp_file.dmp TABLES=(tab1,tab2,tab3) LOG=exp_file.log
```

The export process creates a binary dump file that contains both the schema and data for the specified tables. Now this schema and data can be imported into a target database using the command:

```
imp cust_dba@targetdb FROMUSER=cust_schema TOUSER=cust_schema \
TABLES=(tab1,tab2,tab3) FILE=exp_file.dmp LOG=imp_file.log
```

There are other variations of the Export and Import commands that might be better suited to your needs. See Oracle's documentation for full details.

Oracle SQL*Loader

Oracle SQL*Loader is well suited for large databases that have a limited number of objects in them. Since the process involved in exporting from a source database and loading to a target database is very specific to the schema, the following example creates the sample schema objects, exports from a source, and then loads it into a target database.

To download Oracle SQL*Loader, go to <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>.

1. Create a sample source table using the command below.

```
CREATE TABLE customer_0 TABLESPACE users
AS (SELECT ROWNUM id, o.*
    FROM   ALL_OBJECTS o, ALL_OBJECTS x
    WHERE  ROWNUM <= 1000000);
```

2. On the target Amazon RDS instance, create a destination table that is used to load the data. The clause WHERE 1=2 ensures that you copy the structure of ALL_OBJECTS, but don't copy any of the rows.

```
CREATE TABLE customer_1 TABLESPACE users
AS (SELECT 0 AS ID, OWNER, OBJECT_NAME, CREATED
    FROM   ALL_OBJECTS
    WHERE  1=2);
```

3. The data is exported from the source database to a flat file with delimiters. This example uses SQL*Plus for this purpose. For your data, you will likely need to generate a script that does the export for all the objects in the database.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY/MM/DD HH24:MI:SS'

SET LINESIZE 800 HEADING OFF FEEDBACK OFF ARRAY 5000 PAGESIZE 0
SPOOL customer_0.out
SET MARKUP HTML PREFORMAT ON
SET COLSEP ','

SELECT id, owner, object_name, created
FROM   customer_0;

SPOOL OFF
```

4. You need to create a control file to describe the data. Again, depending on your data, you need to build a script that does this step.

```
cat << EOF > sqlldr_1.ctl
load data
infile customer_0.out
into table customer_1
APPEND
fields terminated by "," optionally enclosed by ''
(
    id          POSITION(01:10)      INTEGER EXTERNAL,
```

```
    owner      POSITION(12:41)    CHAR,
    object_name POSITION(43:72)    CHAR,
    created     POSITION(74:92)    date "YYYY/MM/DD HH24:MI:SS"
)
```

If needed, copy the files generated by the preceding code to a staging area, such as an Amazon EC2 instance.

5. Finally, import the data using SQL*Loader with the appropriate username and password for the target database.

```
sqlldr cust_dba@targetdb CONTROL=sqlldr_1.ctl BINDSIZE=10485760 READSIZE=10485760
ROWS=1000
```

Oracle materialized views

You can also make use of Oracle materialized view replication to migrate large datasets efficiently. Replication allows you to keep the target tables in sync with the source on an ongoing basis, so the actual cutover to Amazon RDS can be done later, if needed. The replication is set up using a database link from the Amazon RDS instance to the source database.

One requirement for materialized views is to allow access from the target database to the source database. In the following example, access rules were enabled on the source database to allow the Amazon RDS target database to connect to the source over SQLNet.

1. Create a user account on both source and Amazon RDS target instances that can authenticate with the same password.

```
CREATE USER dblink_user IDENTIFIED BY <password>
  DEFAULT TABLESPACE users
  TEMPORARY TABLESPACE temp;

GRANT CREATE SESSION TO dblink_user;

GRANT SELECT ANY TABLE TO dblink_user;

GRANT SELECT ANY DICTIONARY TO dblink_user;
```

2. Create a database link from the Amazon RDS target instance to the source instance using the newly created dblink_user.

```
CREATE DATABASE LINK remote_site
  CONNECT TO dblink_user IDENTIFIED BY <password>
  USING '(description=(address=(protocol=tcp) (host=<myhost>)
  (port=<listener port>)) (connect_data=(sid=<sourcedb sid>)))';
```

3. Test the link:

```
SELECT * FROM V$INSTANCE@remote_site;
```

4. Create a sample table with primary key and materialized view log on the source instance.

```
CREATE TABLE customer_0 TABLESPACE users
  AS (SELECT ROWNUM id, o./*
       FROM   ALL_OBJECTS o, ALL_OBJECTS x
      WHERE  ROWNUM <= 1000000);

ALTER TABLE customer_0 ADD CONSTRAINT pk_customer_0 PRIMARY KEY (id) USING INDEX;
```

```
CREATE MATERIALIZED VIEW LOG ON customer_0;
```

5. On the target Amazon RDS instance, create a materialized view.

```
CREATE MATERIALIZED VIEW customer_0
  BUILD IMMEDIATE REFRESH FAST
  AS (SELECT *
      FROM    cust_dba.customer_0@remote_site);
```

6. On the target Amazon RDS instance, refresh the materialized view.

```
EXEC DBMS_MV.REFRESH('CUSTOMER_0', 'f');
```

7. Drop the materialized view and include the PRESERVE TABLE clause to retain the materialized view container table and its contents.

```
DROP MATERIALIZED VIEW customer_0 PRESERVE TABLE;
```

The retained table has the same name as the dropped materialized view.

Working with Oracle replicas for Amazon RDS

To configure replication between Oracle DB instances, you can create replica databases.

Topics

- [Overview of Oracle replicas \(p. 1119\)](#)
- [Replica requirements for Oracle \(p. 1119\)](#)
- [Preparing to create an Oracle replica \(p. 1121\)](#)
- [Creating an Oracle replica in mounted mode \(p. 1122\)](#)
- [Modifying the Oracle replica mode \(p. 1123\)](#)
- [Troubleshooting Oracle replicas \(p. 1124\)](#)

Overview of Oracle replicas

An Oracle *replica* database is either mounted or read-only. An Oracle replica in read-only mode is called a *read replica*. An Oracle replica in mounted mode is called a *mounted replica*.

Read-only and mounted replicas

When creating or modifying an Oracle replica, you can place it in either of the following modes:

- Read-only. This is the default. Active Data Guard transmits and applies changes from the source database to all read replica databases.

You can create up to five read replicas from one source DB instance. For general information about read replicas that applies to all DB engines, see [Working with read replicas \(p. 278\)](#). For information about Oracle Data Guard, see [Oracle data guard concepts and administration](#) in the Oracle documentation.

- Mounted. In this case, replication uses Oracle Data Guard, but the replica database doesn't accept user connections. The primary use for mounted replicas is cross-Region disaster recovery.

A mounted replica can't serve a read-only workload. The mounted replica deletes archived redo log files after it applies them, regardless of the archived log retention policy.

You can create a combination of mounted and read-only DB replicas for the same source DB instance. You can change a read-only replica to mounted mode, or change a mounted replica to read-only mode. In either case, the Oracle database preserves the archived log retention setting.

Outages during replication

When you create an Oracle replica, no outage occurs for the source DB instance. Amazon RDS takes a snapshot of the source DB instance. This snapshot becomes the replica. Amazon RDS sets the necessary parameters and permissions for the source DB and replica without service interruption. Similarly, if you delete a replica, no outage occurs.

Replica requirements for Oracle

Before creating an Oracle replica, check the following requirements.

Version and licensing requirements for Oracle replicas

Before creating an Oracle replica, check the version and licensing requirements:

- If the replica is in read-only mode, make sure that you have an Active Data Guard license. If you place the replica in mounted mode, you don't need an Active Data Guard license. Only the Oracle DB engine supports mounted replicas.
- Oracle replicas are only available on the Oracle Enterprise Edition (EE) engine.
- Oracle replicas are available for Oracle version 12.1.0.2.v10 and higher 12.1 versions, for all 12.2 versions, for all 18c versions, and for all 19c versions.
- Oracle replicas are available for DB instances only on the EC2-VPC platform.
- Oracle replicas are available for DB instances running only on DB instance classes with two or more vCPUs. A source DB instance can't use the db.t3.micro instance class.
- The Oracle DB engine version of the source DB instance and all of its replicas must be the same. Amazon RDS upgrades the replicas immediately after upgrading the source DB instance, regardless of a replica's maintenance window. For major version upgrades of cross-Region replicas, Amazon RDS automatically does the following:
 - Generates an option group for the target version.
 - Copies all options and option settings from the original option group to the new option group.
 - Associates the upgraded cross-Region replica with the new option group.

For more information about upgrading the DB engine version, see [Upgrading the Oracle DB engine \(p. 1209\)](#).

Option requirements for Oracle replicas

Before creating a replica for Oracle, check the requirements for option groups:

- If your Oracle replica is in the same AWS Region as its source DB instance, make sure that it belongs to the same option group as the source DB instance. Modifications to the source option group or source option group membership propagate to replicas. These changes are applied to the replicas immediately after they are applied to the source DB instance, regardless of the replica's maintenance window.

For more information about option groups, see [Working with option groups \(p. 212\)](#).

- When you create an Oracle cross-Region replica, Amazon RDS creates a dedicated option group for it.

You can't remove an Oracle cross-Region replica from its dedicated option group. No other DB instances can use the dedicated option group for an Oracle cross-Region replica.

You can only add or remove the following nonreplicated options from a dedicated option group:

- NATIVE_NETWORK_ENCRYPTION
- OEM
- OEM_AGENT
- SSL

To add other options to an Oracle cross-Region replica, add them to the source DB instance's option group. The option is also installed on all of the source DB instance's replicas. For licensed options, make sure that there are sufficient licenses for the replicas.

When you promote an Oracle cross-Region replica, the promoted replica behaves the same as other Oracle DB instances, including the management of its options. You can promote a replica explicitly or implicitly by deleting its source DB instance.

For more information about option groups, see [Working with option groups \(p. 212\)](#).

Miscellaneous requirements for Oracle replicas

Before creating an Oracle replica, check the following miscellaneous requirements:

- If a DB instance is a source for one or more cross-Region replicas, the source DB retains its archived redo logs until they are applied on all cross-Region replicas. The archived redo logs might result in increased storage consumption.
- A logon trigger on a primary instance must permit access to the `RDS_DATAGUARD` user and to any user whose `AUTHENTICATED_IDENTITY` value is `RDS_DATAGUARD` or `rdsdb`. Also, the trigger must not set the current schema for the `RDS_DATAGUARD` user.
- To avoid disrupting RDS automation, system triggers must permit specific users to log on to the primary and replica database. [System triggers](#) include DDL, logon, and database role triggers. We recommend that you add code to your triggers to exclude the users listed in the following sample code:

```
-- Determine who the user is
SELECT SYS_CONTEXT('USERENV','AUTHENTICATED_IDENTITY') INTO CURRENT_USER FROM DUAL;
-- The following users should always be able to login to either the Primary or Replica
IF CURRENT_USER IN ('master_user', 'SYS', 'SYSTEM', 'RDS_DATAGUARD', 'rdsdb') THEN
RETURN;
END IF;
```

- To avoid blocking connections from the Data Guard broker process, don't enable restricted sessions. For more information about restricted sessions, see [Enabling and disabling restricted sessions \(p. 1038\)](#).
- Block change tracking is supported for read-only replicas, but not for mounted replicas. You can change a mounted replica to a read-only replica, and then enable block change tracking. For more information, see [Enabling and disabling block change tracking \(p. 1075\)](#).

Preparing to create an Oracle replica

Before you can begin using your replica, perform the following tasks.

Topics

- [Enabling automatic backups \(p. 1121\)](#)
- [Enabling force logging mode \(p. 1121\)](#)
- [Changing your logging configuration \(p. 1122\)](#)
- [Setting the MAX_STRING_SIZE parameter \(p. 1122\)](#)
- [Planning compute and storage resources \(p. 1122\)](#)

Enabling automatic backups

Before a DB instance can serve as a source DB instance, make sure to enable automatic backups on the source DB instance. To learn how to perform this procedure, see [Enabling automated backups \(p. 330\)](#).

Enabling force logging mode

We recommend that you enable force logging mode. In force logging mode, the Oracle database writes redo records even when `NOLOGGING` is used with data definition language (DDL) statements.

To enable force logging mode

1. Log in to your Oracle database using a client tool such as SQL Developer.

2. Enable force logging mode by running the following procedure.

```
exec rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

For more information about this procedure, see [Setting force logging \(p. 1063\)](#).

Changing your logging configuration

If you want to change your logging configuration, we recommend that you complete the changes before making a DB instance the source for replicas. Also, we recommend that you not modify the logging configuration after you create the replicas. Modifications can cause the online redo logging configuration to get out of sync with the standby logging configuration.

Modify the logging configuration for a DB instance by using the Amazon RDS procedures `rdsadmin.rdsadmin_util.add_logfile` and `rdsadmin.rdsadmin_util.drop_logfile`. For more information, see [Adding online redo logs \(p. 1064\)](#) and [Dropping online redo logs \(p. 1065\)](#).

Setting the MAX_STRING_SIZE parameter

Before you create an Oracle replica, ensure that the setting of the `MAX_STRING_SIZE` parameter is the same on the source DB instance and the replica. You can do this by associating them with the same parameter group. If you have different parameter groups for the source and the replica, you can set `MAX_STRING_SIZE` to the same value. For more information about setting this parameter, see [Enabling extended data types for a new DB instance \(p. 1104\)](#).

Planning compute and storage resources

Ensure that the source DB instance and its replicas are sized properly, in terms of compute and storage, to suit their operational load. If a replica reaches compute, network, or storage resource capacity, the replica stops receiving or applying changes from its source. Amazon RDS for Oracle doesn't intervene to mitigate high replica lag between a source DB instance and its replicas. You can modify the storage and CPU resources of a replica independently from its source and other replicas.

Creating an Oracle replica in mounted mode

By default, Oracle replicas are read-only. To create a replica in mounted mode, use the console, the AWS CLI, or the RDS API.

Console

To create a mounted replica from a source Oracle DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the Oracle DB instance that you want to use as the source for a mounted replica.
4. For **Actions**, choose **Create replica**.
5. For **Replica mode**, choose **Mounted**.
6. Choose the settings that you want to use. For **DB instance identifier**, enter a name for the read replica. Adjust other settings as needed.
7. For **Regions**, choose the Region where the mounted replica will be launched.
8. Choose your instance size and storage type. We recommend that you use the same DB instance class and storage type as the source DB instance for the read replica.

9. For **Multi-AZ deployment**, choose **Create a standby instance** to create a standby of your replica in another Availability Zone for failover support for the mounted replica. Creating your mounted replica as a Multi-AZ DB instance is independent of whether the source database is a Multi-AZ DB instance.
10. Choose the other settings that you want to use.
11. Choose **Create replica**.

In the **Databases** page, the mounted replica has the role **Replica**.

AWS CLI

To create an Oracle replica in mounted mode, set `--replica-mode` to `mounted` in the AWS CLI command [create-db-instance-read-replica](#).

Example

For Linux, macOS, or Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --replica-mode mounted
```

For Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --source-db-instance-identifier mydbinstance ^
    --replica-mode mounted
```

To change a read-only replica to a mounted state, set `--replica-mode` to `mounted` in the AWS CLI command [modify-db-instance](#). To place a mounted replica in read-only mode, set `--replica-mode` to `open-read-only`.

RDS API

To create an Oracle replica in mounted mode, specify `ReplicaMode=mounted` in the RDS API operation [CreateDBInstanceReadReplica](#).

Modifying the Oracle replica mode

To change the replica mode of an existing replica, use the console, AWS CLI, or RDS API. When you change to mounted mode, the replica disconnects all active connections. When you change to read-only mode, Amazon RDS initializes Active Data Guard.

The change operation can take a few minutes. During the operation, the DB instance status changes to **modifying**. For more information about status changes, see [DB instance status \(p. 404\)](#).

Console

To change the replica mode of an Oracle replica from mounted to read-only

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.

3. Choose the mounted replica database.
4. Choose **Modify**.
5. For **Replica mode**, choose **Read-only**.
6. Choose the other settings that you want to change.
7. Choose **Continue**.
8. For **Scheduling of modifications**, choose **Apply immediately**.
9. Choose **Modify DB instance**.

AWS CLI

To change a read replica to mounted mode, set `--replica-mode` to `mounted` in the AWS CLI command [modify-db-instance](#). To change a mounted replica to read-only mode, set `--replica-mode` to `open-read-only`.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
  --db-instance-identifier myreadreplica \
  --replica-mode mode
```

For Windows:

```
aws rds modify-db-instance ^
  --db-instance-identifier myreadreplica ^
  --replica-mode mode
```

RDS API

To change a read-only replica to mounted mode, set `ReplicaMode=mounted` in [ModifyDBInstance](#). To change a mounted replica to read-only mode, set `ReplicaMode=read-only`.

Troubleshooting Oracle replicas

This section describes possible replication problems and solutions.

Replication lag

To monitor replication lag in Amazon CloudWatch, view the Amazon RDS `ReplicaLag` metric. For information about replication lag time, see [Monitoring read replication \(p. 288\)](#).

If replication lag is too long, query the following views:

- `V$ARCHIVED_LOG` – Shows which commits have been applied to the read replica.
- `V$DATAGUARD_STATS` – Shows a detailed breakdown of the components that make up the `replicaLag` metric.
- `V$DATAGUARD_STATUS` – Shows the log output from Oracle's internal replication processes.

Replication failure after adding or modifying triggers

If you add or modify any triggers, and if replication fails afterward, the problem may be the triggers. Ensure that the trigger excludes the following user accounts, which are required by RDS for replication:

- User accounts with administrator privileges
- **SYS**
- **SYSTEM**
- **RDS_DATAGUARD**
- **rdsdb**

For more information, see[Miscellaneous requirements for Oracle replicas \(p. 1121\)](#).

Adding options to Oracle DB instances

In Amazon RDS, an *option* is an additional feature. Following, you can find a description of options that you can add to Amazon RDS instances running the Oracle DB engine. To enable these options, you add them to an option group, and then associate the option group with your DB instance. For more information, see [Working with option groups \(p. 212\)](#).

Some options require additional memory to run on your DB instance. For example, Oracle Enterprise Manager Database Control uses about 300 MB of RAM. If you enable this option for a small DB instance, you might encounter performance problems due to memory constraints. You can adjust the Oracle parameters so that the database requires less RAM. Alternatively, you can scale up to a larger DB instance.

You can add the following options for Oracle DB instances.

Option	Option ID
Amazon S3 integration (p. 1127)	S3_INTEGRATION
Oracle Application Express (APEX) (p. 1140)	APEX
	APEX-DEV
Oracle Enterprise Manager (p. 1149)	OEM
	OEM_AGENT
Oracle Java virtual machine (p. 1164)	JVM
Oracle Label Security (p. 1167)	OLS
Oracle Locator (p. 1170)	LOCATOR
Oracle Multimedia (p. 1173)	MULTIMEDIA
Oracle native network encryption (p. 1176)	NATIVE_NETWORK_ENCRYPTION
Oracle OLAP (p. 1180)	OLAP
Oracle Secure Sockets Layer (p. 1182)	SSL
Oracle Spatial (p. 1190)	SPATIAL
Oracle SQLT (p. 1193)	SQLT
Oracle Statspack (p. 1198)	STATSPACK
Oracle time zone (p. 1201)	Timezone
Oracle Transparent Data Encryption (p. 1204)	TDE
Oracle UTL_MAIL (p. 1206)	UTL_MAIL
Oracle XML DB (p. 1208)	XMLDB

Amazon S3 integration

You can transfer files between an Amazon RDS for Oracle DB instance and an Amazon S3 bucket. You can use Amazon S3 integration with Oracle features such as Data Pump. For example, you can download Data Pump files from Amazon S3 to the DB instance host.

Note

The DB instance and the Amazon S3 bucket must be in the same AWS Region.

Topics

- [Prerequisites for Amazon RDS for Oracle integration with Amazon S3 \(p. 1127\)](#)
- [Adding the Amazon S3 integration option \(p. 1133\)](#)
- [Transferring files between Amazon RDS for Oracle and an Amazon S3 bucket \(p. 1134\)](#)
- [Removing the Amazon S3 integration option \(p. 1138\)](#)

Prerequisites for Amazon RDS for Oracle integration with Amazon S3

For Amazon RDS for Oracle to integrate with Amazon S3, the Amazon RDS DB instance must have access to an Amazon S3 bucket. Prepare for the integration as follows:

1. Create an AWS Identity and Access Management (IAM) policy with the permissions required to transfer files from your bucket to RDS.

To create the policy, you need the Amazon Resource Name (ARN) value for your bucket. Also, RDS for Oracle supports SSE-KMS and SSE-S3 encryption. If your bucket is encrypted, you need the ARN for your AWS KMS key. For more information, see [Protecting data using server-side encryption](#) in the *Amazon Simple Storage Service Console User Guide*.

Note

An Oracle DB instance can't access Amazon S3 buckets encrypted with SSE-C.

2. Create an IAM role, attach your new policy to it, and then attach the role to your Oracle DB instance. The status of the DB instance must be available.

The Amazon VPC used by your DB instance doesn't need to provide access to the Amazon S3 endpoints.

Console

To create an IAM policy to allow Amazon RDS access to an Amazon S3 bucket

1. Open the [IAM Management Console](#).
2. Under **Access management**, choose **Policies**.
3. Choose **Create policy**.
4. On the **Visual editor** tab, choose **Choose a service**, and then choose **S3**.
5. For **Actions**, choose **Expand all**, and then choose the bucket permissions and object permissions required to transfer files from an Amazon S3 bucket to Amazon RDS. For example, do the following:
 - Expand **List**, and then select **ListBucket**.
 - Expand **Read**, and then select **GetObject**.
 - Expand **Write**, and then select **PutObject**.

Object permissions are permissions for object operations in Amazon S3, and must be granted for objects in a bucket, not the bucket itself. For more information about permissions for object operations in Amazon S3, see [Permissions for object operations](#).

6. Choose **Resources**, and choose **Add ARN for bucket**.
7. In the **Add ARN(s)** dialog box, provide the details about your resource, and choose **Add**.

Specify the Amazon S3 bucket to allow access to. For instance, to allow Amazon RDS to access the Amazon S3 bucket named example-bucket, set the ARN value to arn:aws:s3:::example-bucket.

8. If the **object** resource is listed, choose **Add ARN for object**.
9. In the **Add ARN(s)** dialog box, provide the details about your resource.

For the Amazon S3 bucket, specify the Amazon S3 bucket to allow access to. For the object, you can choose **Any** to grant permissions to any object in the bucket.

Note

You can set **Amazon Resource Name (ARN)** to a more specific ARN value to allow Amazon RDS to access only specific files or folders in an Amazon S3 bucket. For more information about how to define an access policy for Amazon S3, see [Managing access permissions to your Amazon S3 resources](#).

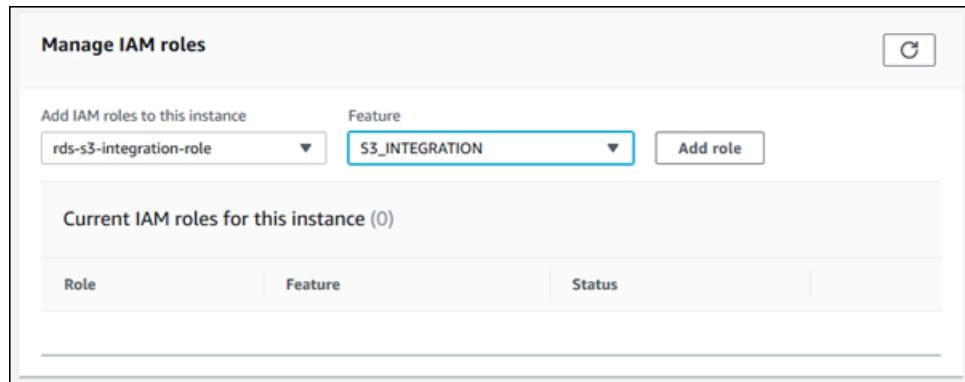
10. (Optional) Choose **Add additional permissions** to add resources to the policy. For example, do the following:
 - If your bucket is encrypted with a custom KMS key, select **KMS** for the service. Select **Encrypt**, **ReEncrypt**, **Decrypt**, **DescribeKey**, and **GenerateDataKey** for actions. Enter the ARN of your custom key as the resource. For more information, see [Protecting Data Using Server-Side Encryption with CMKs Stored in AWS Key Management Service \(SSE-KMS\)](#) in the *Amazon Simple Storage Service Console User Guide*.
 - If you want Amazon RDS to access to access other buckets, add the ARNs for these buckets. Optionally, you can also grant access to all buckets and objects in Amazon S3.
11. Choose **Next: Tags** and then **Next: Review**.
12. For **Name**, enter a name for your IAM policy, for example rds-s3-integration-policy. You use this name when you create an IAM role to associate with your DB instance. You can also add an optional **Description** value.
13. Choose **Create policy**.

To create an IAM role to allow Amazon RDS access to an Amazon S3 bucket

1. In the navigation pane, choose **Roles**.
2. Choose **Create role**.
3. For **AWS service**, choose **RDS**.
4. For **Select your use case**, choose **RDS – Add Role to Database**.
5. Choose **Next: Permissions**.
6. For **Search under Attach permissions policies**, enter the name of the IAM policy you created, and choose the policy when it appears in the list.
7. Choose **Next: Tags** and then **Next: Review**.
8. Set **Role name** to a name for your IAM role, for example rds-s3-integration-role. You can also add an optional **Role description** value.
9. Choose **Create role**.

To associate your IAM role with your DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose **Databases** from the navigation pane.
3. If your database instance is unavailable, choose **Actions** and then **Start**. When the instance status shows **Started**, go to the next step.
4. Choose the Oracle DB instance name to display its details.
5. On the **Connectivity & security** tab, in the **Manage IAM roles** section, choose the role to add under **Add IAM roles to this instance**.
6. For **Feature**, choose **S3_INTEGRATION**.



7. Choose **Add role**.

AWS CLI

To grant Amazon RDS access to an Amazon S3 bucket

1. Create an AWS Identity and Access Management (IAM) policy that grants Amazon RDS access to an Amazon S3 bucket.

Include the appropriate actions in the policy based on the type of access required:

- **GetObject** – Required to transfer files from an Amazon S3 bucket to Amazon RDS.
- **ListBucket** – Required to transfer files from an Amazon S3 bucket to Amazon RDS.
- **PutObject** – Required to transfer files from Amazon RDS to an Amazon S3 bucket.

The following AWS CLI command creates an IAM policy named *rds-s3-integration-policy* with these options. It grants access to a bucket named *your-s3-bucket-arn*.

Example

For Linux, macOS, or Unix:

```
aws iam create-policy \
--policy-name rds-s3-integration-policy \
--policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "s3integration",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::your-s3-bucket-arn/*"
        }
    ]
}'
```

```
        "s3>ListBucket",
        "s3PutObject"
    ],
    "Effect": "Allow",
    "Resource": [
        "arn:aws:s3:::your-s3-bucket-arn",
        "arn:aws:s3:::your-s3-bucket-arn/*"
    ]
}
]'
```

The following example includes permissions for custom KMS keys.

```
aws iam create-policy \
--policy-name rds-s3-integration-policy \
--policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "s3integration",
            "Action": [
                "s3:GetObject",
                "s3>ListBucket",
                "s3PutObject",
                "kmsDecrypt",
                "kmsEncrypt",
                "kmsReEncrypt",
                "kmsGenerateDataKey",
                "kmsDescribeKey",
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::your-s3-bucket-arn",
                "arn:aws:s3:::your-s3-bucket-arn/*",
                "arn:aws:kms:::your-kms-arn"
            ]
        }
    ]
}'
```

For Windows:

```
aws iam create-policy ^
--policy-name rds-s3-integration-policy ^
--policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "s3integration",
            "Action": [
                "s3:GetObject",
                "s3>ListBucket",
                "s3PutObject"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::your-s3-bucket-arn",
                "arn:aws:s3:::your-s3-bucket-arn/*"
            ]
        }
    ]
}'
```

```
}'
```

The following example includes permissions for custom KMS keys.

```
aws iam create-policy ^
--policy-name rds-s3-integration-policy ^
--policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "s3integration",
            "Action": [
                "s3:GetObject",
                "s3>ListBucket",
                "s3:PutObject",
                "kms:Decrypt",
                "kms:Encrypt",
                "kms:ReEncrypt",
                "kms:GenerateDataKey",
                "kms:DescribeKey",
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::your-s3-bucket-arn",
                "arn:aws:s3:::your-s3-bucket-arn/*",
                "arn:aws:kms:::your-kms-arn"
            ]
        }
    ]
}'
```

2. After the policy is created, note the Amazon Resource Name (ARN) of the policy. You need the ARN for a subsequent step.
3. Create an IAM role that Amazon RDS can assume on your behalf to access your Amazon S3 buckets.

The following AWS CLI command creates the **rds-s3-integration-role** for this purpose.

Example

For Linux, macOS, or Unix:

```
aws iam create-role \
--role-name rds-s3-integration-role \
--assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}'
```

For Windows:

```
aws iam create-role ^
--role-name rds-s3-integration-role ^
--assume-role-policy-document '{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "rds.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
]'
```

For more information, see [Creating a role to delegate permissions to an IAM user](#) in the *IAM User Guide*.

4. After the role is created, note the ARN of the role. You need the ARN for a subsequent step.
5. Attach the policy you created to the role you created.

The following AWS CLI command attaches the policy to the role named *rds-s3-integration-role*.

Example

For Linux, macOS, or Unix:

```
aws iam attach-role-policy \
--policy-arn your-policy-arn \
--role-name rds-s3-integration-role
```

For Windows:

```
aws iam attach-role-policy ^
--policy-arn your-policy-arn ^
--role-name rds-s3-integration-role
```

Replace *your-policy-arn* with the policy ARN that you noted in a previous step.

6. Add the role to the Oracle DB instance.

The following AWS CLI command adds the role to an Oracle DB instance named *mydbinstance*.

Example

For Linux, macOS, or Unix:

```
aws rds add-role-to-db-instance \
--db-instance-identifier mydbinstance \
--feature-name S3_INTEGRATION \
--role-arn your-role-arn
```

For Windows:

```
aws rds add-role-to-db-instance ^
--db-instance-identifier mydbinstance ^
--feature-name S3_INTEGRATION ^
--role-arn your-role-arn
```

Replace *your-role-arn* with the role ARN that you noted in a previous step. `S3_INTEGRATION` must be specified for the `--feature-name` option.

Adding the Amazon S3 integration option

To use Amazon RDS for Oracle Integration with Amazon S3, your Amazon RDS for Oracle DB instance must be associated with an option group that includes the `S3_INTEGRATION` option.

Console

To configure an option group for Amazon S3 integration

1. Create a new option group or identify an existing option group to which you can add the `S3_INTEGRATION` option.

For information about creating an option group, see [Creating an option group \(p. 214\)](#).

2. Add the `S3_INTEGRATION` option to the option group.

For information about adding an option to an option group, see [Adding an option to an option group \(p. 216\)](#).

3. Create a new Oracle DB instance and associate the option group with it, or modify an Oracle DB instance to associate the option group with it.

For information about creating a DB instance, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

For information about modifying an Oracle DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

AWS CLI

To configure an option group for Amazon S3 integration

1. Create a new option group or identify an existing option group to which you can add the `S3_INTEGRATION` option.

For information about creating an option group, see [Creating an option group \(p. 214\)](#).

2. Add the `S3_INTEGRATION` option to the option group.

For example, the following AWS CLI command adds the `S3_INTEGRATION` option to an option group named `myoptiongroup`.

Example

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \
--option-group-name myoptiongroup \
--options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

For Windows:

```
aws rds add-option-to-option-group ^
--option-group-name myoptiongroup ^
```

```
--options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

3. Create a new Oracle DB instance and associate the option group with it, or modify an Oracle DB instance to associate the option group with it.

For information about creating a DB instance, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

For information about modifying an Oracle DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Transferring files between Amazon RDS for Oracle and an Amazon S3 bucket

You can use Amazon RDS procedures to transfer files between an Oracle DB instance and an Amazon S3 bucket.

Note

These procedures upload or download the files in a single directory. You can't include subdirectories in these operations.

Topics

- [Uploading files from an Oracle DB instance to an Amazon S3 bucket \(p. 1134\)](#)
- [Downloading files from an Amazon S3 bucket to an Oracle DB instance \(p. 1136\)](#)
- [Monitoring the status of a file transfer \(p. 1138\)](#)

Uploading files from an Oracle DB instance to an Amazon S3 bucket

You can upload files from an Oracle DB instance to an Amazon S3 bucket. For example, you can upload Oracle Recovery Manager (RMAN) backup files. The maximum object size in an Amazon S3 bucket is 5 TB. For more information about working with objects, see [Amazon Simple Storage Service Developer Guide](#). For more information about performing RMAN backups, see [Performing common RMAN tasks for Oracle DB instances \(p. 1069\)](#).

You upload files using the `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` procedure. This procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>p_bucket_name</code>	VARCHAR2	–	required	The name of the Amazon S3 bucket to upload files to.
<code>p_directory_name</code>	VARCHAR2	–	required	The name of the Oracle directory object to upload files from. The directory can be any user-created directory object or the Data Pump directory, such as <code>DATA_PUMP_DIR</code> .

Note

You can only upload files from the specified directory. You can't upload files

Parameter name	Data type	Default	Required	Description
				in subdirectories in the specified directory.
p_s3_prefix	VARCHAR2	–	required	An Amazon S3 file name prefix that files are uploaded to. An empty prefix uploads all files to the top level in the specified Amazon S3 bucket and doesn't add a prefix to the file names. For example, if the prefix is <code>folder_1/oradb</code> , files are uploaded to <code>folder_1</code> . In this case, the <code>oradb</code> prefix is added to each file.
p_prefix	VARCHAR2	–	required	A file name prefix that file names must match to be uploaded. An empty prefix uploads all files in the specified directory.

The return value for the `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` procedure is a task ID.

The following example uploads all of the files in the `DATA_PUMP_DIR` directory to the Amazon S3 bucket named `mys3bucket`.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
    p_bucket_name    => 'mys3bucket',
    p_prefix         => '',
    p_s3_prefix      => '',
    p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

The following example uploads all of the files with the prefix `db` in the `DATA_PUMP_DIR` directory to the Amazon S3 bucket named `mys3bucket`.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
    p_bucket_name    => 'mys3bucket',
    p_prefix         => 'db',
    p_s3_prefix      => '',
    p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

The following example uploads all of the files in the `DATA_PUMP_DIR` directory to the Amazon S3 bucket named `mys3bucket`. The files are uploaded to a `dbfiles` folder.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
    p_bucket_name    => 'mys3bucket',
    p_prefix         => '',
    p_s3_prefix      => 'dbfiles/',
    p_directory_name => 'DATA_PUMP_DIR')
```

```
AS TASK_ID FROM DUAL;
```

The following example uploads all of the files in the `DATA_PUMP_DIR` directory to the Amazon S3 bucket named `mys3bucket`. The files are uploaded to a `dbfiles` folder and `ora` is added to the beginning of each file name.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
    p_bucket_name      => 'mys3bucket',
    p_prefix           => '',
    p_s3_prefix        => 'dbfiles/ora',
    p_directory_name   => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

In each example, the `SELECT` statement returns the ID of the task in a `VARCHAR2` data type.

You can view the result by displaying the task's output file.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP', 'dbtask-task-id.log'));
```

Replace `task-id` with the task ID returned by the procedure.

Note

Tasks are executed asynchronously.

Downloading files from an Amazon S3 bucket to an Oracle DB instance

To download files from an Amazon S3 bucket to an Oracle DB instance, use the Amazon RDS procedure `rdsadmin.rdsadmin_s3_tasks.download_from_s3`. The `rdsadmin.rdsadmin_s3_tasks.download_from_s3` procedure has the following parameters.

Parameter name	Data type	Default	Required	Description
<code>p_bucket_name</code>	<code>VARCHAR2</code>	–	required	The name of the Amazon S3 bucket to download files from.
<code>p_directory_name</code>	<code>VARCHAR2</code>	–	required	The name of the Oracle directory object to download files to. The directory can be any user-created directory object or the Data Pump directory, such as <code>DATA_PUMP_DIR</code> .
<code>p_s3_prefix</code>	<code>VARCHAR2</code>	"	optional	A file name prefix that file names must match to be downloaded. An empty prefix downloads all of the top level files in the specified Amazon S3 bucket, but not the files in folders in the bucket.

Parameter name	Data type	Default	Required	Description
				<p>The procedure downloads Amazon S3 objects only from the first level folder that matches the prefix. Nested directory structures matching the specified prefix are not downloaded.</p> <p>For example, suppose that an Amazon S3 bucket has the folder structure <code>folder_1/folder_2/folder_3</code>. Suppose also that you specify the <code>'folder_1/folder_2/'</code> prefix. In this case, only the files in <code>folder_2</code> are downloaded, not the files in <code>folder_1</code> or <code>folder_3</code>.</p> <p>If, instead, you specify the <code>'folder_1/folder_2'</code> prefix, all files in <code>folder_1</code> that match the <code>'folder_2'</code> prefix are downloaded, and no files in <code>folder_2</code> are downloaded.</p>

The return value for the `rdsadmin.rdsadmin_s3_tasks.download_from_s3` procedure is a task ID.

The following example downloads all of the files in the Amazon S3 bucket named `mys3bucket` to the `DATA_PUMP_DIR` directory.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
    p_bucket_name => 'mys3bucket',
    p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

The following example downloads all of the files with the prefix `db` in the Amazon S3 bucket named `mys3bucket` to the `DATA_PUMP_DIR` directory.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
    p_bucket_name => 'mys3bucket',
    p_s3_prefix => 'db',
    p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

The following example downloads all of the files in the folder `myfolder/` in the Amazon S3 bucket named `mys3bucket` to the `DATA_PUMP_DIR` directory. Use the `prefix` parameter setting to specify the Amazon S3 folder.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
    p_bucket_name => 'mys3bucket',
    p_s3_prefix     => 'myfolder/',
    p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

In each example, the `SELECT` statement returns the ID of the task in a `VARCHAR2` data type.

You can view the result by displaying the task's output file.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP', 'dbtask-task-id.log'));
```

Replace `task-id` with the task ID returned by the procedure.

Note

Tasks are executed asynchronously.

You can use the `UTL_FILE.FREMOVE` Oracle procedure to remove files from a directory. For more information, see [FREMOVE procedure](#) in the Oracle documentation.

Monitoring the status of a file transfer

File transfer tasks publish Amazon RDS events when they start and when they complete. For information about viewing events, see [Viewing Amazon RDS events \(p. 503\)](#).

You can view the status of an ongoing task in a bdump file. The bdump files are located in the `/rdsdbdata/log/trace` directory. Each bdump file name is in the following format.

```
dbtask-task-id.log
```

Replace `task-id` with the ID of the task that you want to monitor.

Note

Tasks are executed asynchronously.

You can use the `rdsadmin.rds_file_util.read_text_file` stored procedure to view the contents of bdump files. For example, the following query returns the contents of the `dbtask-1546988886389-2444.log` bdump file.

```
SELECT text FROM
table(rdsadmin.rds_file_util.read_text_file('BDUMP', 'dbtask-1546988886389-2444.log'));
```

Removing the Amazon S3 integration option

You can remove Amazon S3 integration option from a DB instance.

To remove the Amazon S3 integration option from a DB instance, do one of the following:

- To remove the Amazon S3 integration option from multiple DB instances, remove the `S3_INTEGRATION` option from the option group to which the DB instances belong. This change affects

all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).

- To remove the Amazon S3 integration option from a single DB instance, modify the DB instance and specify a different option group that doesn't include the S3_INTEGRATION option. You can specify the default (empty) option group or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Oracle Application Express (APEX)

Amazon RDS supports Oracle Application Express (APEX) through the use of the `APEX` and `APEX-DEV` options. Oracle APEX can be deployed as a run-time environment or as a full development environment for web-based applications. Using Oracle APEX, developers can build applications entirely within the web browser. For more information, see [Oracle application Express](#) in the Oracle documentation.

Oracle APEX consists of the following main components:

- A *repository* that stores the metadata for APEX applications and components. The repository consists of tables, indexes, and other objects that are installed in your Amazon RDS DB instance.
- A *listener* that manages HTTP communications with Oracle APEX clients. The listener accepts incoming connections from web browsers, forwards them to the Amazon RDS DB instance for processing, and then sends results from the repository back to the browsers. Amazon RDS for Oracle supports the following types of listeners:
 - For APEX version 5.0 and later, use Oracle Rest Data Services (ORDS) version 19.1 and higher. We recommend that you use the latest supported version of Oracle APEX and ORDS. The documentation describes older versions for backwards compatibility only.
 - For APEX version 4.1.1, you can use Oracle APEX Listener version 1.1.4.
 - Oracle HTTP Server and `mod_plsql`.

Note

Amazon RDS doesn't support the Oracle XML DB HTTP server with the embedded PL/SQL gateway; you can't use this as a listener for APEX. In general, Oracle recommends against using the embedded PL/SQL gateway for applications that run on the internet.

For more information about these listener types, see [About choosing a web listener](#) in the Oracle documentation.

When you add the Amazon RDS APEX options to your DB instance, Amazon RDS installs the Oracle APEX repository only. Install your listener on a separate host, such as an Amazon EC2 instance, an on-premises server at your company, or your desktop computer.

The APEX option uses storage on the DB instance class for your DB instance. Following are the supported versions and approximate storage requirements for Oracle APEX.

APEX version	Storage requirements	Supported Oracle database versions	Notes
Oracle APEX version 20.2.v1	148 MiB	All	This version includes patch p32006852_2020_Generic. You can see the patch number and date by running the following query: <pre>SELECT PATCH_VERSION, PATCH_NUMBER FROM APEX_PATCHES;</pre>
Oracle APEX version 20.1.v1	173 MiB	All	This version includes patch 30990551.
Oracle APEX version 19.2.v1	149 MiB	All	
Oracle APEX version 19.1.v1	148 MiB	All	

APEX version	Storage requirements	Supported Oracle database versions	Notes
Oracle APEX version 18.2.v1	146 MiB	All except 19c	
Oracle APEX version 18.1.v1	145 MiB	All except 19c	
Oracle APEX version 5.1.4.v1	220 MiB	All except 19c	
Oracle APEX version 5.1.2.v1	150 MiB	12.1 only	
Oracle APEX version 5.0.4.v1	140 MiB	12.1 only	
Oracle APEX version 4.2.6.v1	160 MiB	12.1 only	

Prerequisites for Oracle APEX and ORDS

To use Oracle APEX and ORDS, make sure you have the following:

- The Java Runtime Environment (JRE)
- An Oracle client installation that includes the following:
 - SQL*Plus or SQL Developer for administration tasks
 - Oracle Net Services for configuring connections to your Oracle instance

Adding the Amazon RDS APEX options

The general process for adding the Amazon RDS APEX options to a DB instance is the following:

1. Create a new option group, or copy or modify an existing option group.
2. Add the options to the option group.
3. Associate the option group with the DB instance.

When you add the Amazon RDS APEX options, a brief outage occurs while your DB instance is automatically restarted.

To add the APEX options to a DB instance

1. Determine the option group you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group with the following settings:
 - a. For **Engine**, choose the Oracle edition that you want to use. The APEX options are supported on all editions.
 - b. For **Major engine version**, choose the version of your DB instance.

For more information, see [Creating an option group \(p. 214\)](#).

2. Add the options to the option group. If you want to deploy only the Oracle APEX run-time environment, add only the **APEX** option. If you want to deploy the full development environment, add both the **APEX** and **APEX-DEV** options. For Oracle Database 12c, add the **APEX** and **APEX-DEV** options.

For **Version**, choose the version of APEX that you want to use. If you don't choose a version, version 4.2.6.v1 is the default for Oracle Database 12c.

Important

If you add the APEX options to an existing option group that is already attached to one or more DB instances, a brief outage occurs. During this outage, all the DB instances are automatically restarted.

For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).

3. Apply the option group to a new or existing DB instance:

- For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- For an existing DB instance, you apply the option group by modifying the instance and attaching the new option group. When you add the APEX options to an existing DB instance, a brief outage occurs while your DB instance is automatically restarted. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Unlocking the public user account

After the Amazon RDS APEX options are installed, you must change the password for the APEX public user account, and then unlock the account. You can do this by using the Oracle SQL*Plus command line utility. Connect to your DB instance as the master user, and issue the following commands. Replace `new_password` with a password of your choice.

```
alter user APEX_PUBLIC_USER identified by new_password;
alter user APEX_PUBLIC_USER account unlock;
```

Configuring RESTful services for Oracle APEX

To configure RESTful services in APEX (not needed for APEX 4.1.1.V1), use SQL*Plus to connect to your DB instance as the master user. After you do this, run the `rdsadmin.rdsadmin_run_apex_rest_config` stored procedure. When you run the stored procedure, you provide passwords for the following users:

- `APEX_LISTENER`
- `APEX_REST_PUBLIC_USER`

The stored procedure runs the `apex_rest_config.sql` script, which creates new database accounts for these users.

Note

Configuration isn't required for Oracle APEX version 4.1.1.v1. For this Oracle APEX version only, you don't need to run the stored procedure.

The following command runs the stored procedure.

```
exec rdsadmin.rdsadmin_run_apex_rest_config('apex_listener_password',
'apex_rest_public_user_password');
```

Setting up ORDS for Oracle APEX

You are now ready to install and configure Oracle Rest Data Services (ORDS) for use with Oracle APEX. For APEX version 5.0 and later, use Oracle Rest Data Services (ORDS) version 19.1 and higher.

Install the listener on a separate host such as an Amazon EC2 instance, an on-premises server at your company, or your desktop computer. For the examples in this section, we assume that the name of your host is `myapexhost.example.com`, and that your host is running Linux.

Preparing to install ORDS

Before you can install ORDS, you need to create a nonprivileged OS user, and then download and unzip the APEX installation file.

To prepare for ORDS installation

1. Log in to `myapexhost.example.com` as `root`.
2. Create a nonprivileged OS user to own the listener installation. The following command creates a new user named `apexuser`.

```
useradd -d /home/apexuser apexuser
```

The following command assigns a password to the new user.

```
passwd apexuser;
```

3. Log in to `myapexhost.example.com` as `apexuser`, and download the APEX installation file from Oracle to your `/home/apexuser` directory:
 - <http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html>
 - [Oracle application Express prior release archives](#)
4. Unzip the file in the `/home/apexuser` directory.

```
unzip apex_<version>.zip
```

After you unzip the file, there is an `apex` directory in the `/home/apexuser` directory.

5. While you are still logged into `myapexhost.example.com` as `apexuser`, download the Oracle REST Data Services file from Oracle to your `/home/apexuser` directory: <http://www.oracle.com/technetwork/developer-tools/apex-listener/downloads/index.html>.

Installing and configuring ORDS

Before you can use APEX, you need to download the `ords.war` file, use Java to install ORDS, and then start the listener.

To install and configure ORDS for use with Oracle APEX

1. Create a new directory based on ORDS, and then unzip the listener file.

```
mkdir /home/apexuser/ORDS
cd /home/apexuser/ORDS
```

2. Download the file `ords.version.number.zip` from [Oracle REST data services](#).
3. Unzip the file into the `/home/apexuser/ORDS` directory.
4. Grant the master user the required privileges to install ORDS.

After the Amazon RDS APEX option is installed, give the master user the required privileges to install the ORDS schema. You can do this by connecting to the database and running the following commands. Replace `master_user` with the name of your master user.

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_OBJECTS', 'master_user', 'SELECT',
true);
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_ROLE_PRIVS', 'master_user', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_COLUMNS', 'master_user', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONS_COLUMNS', 'master_user', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONSTRAINTS', 'master_user', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_OBJECTS', 'master_user', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_PROCEDURES', 'master_user', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TAB_COLUMNS', 'master_user', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TABLES', 'master_user', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_VIEWS', 'master_user', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPIUTL', 'master_user', 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SESSION', 'master_user', 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_UTILITY', 'master_user', 'EXECUTE', true);
```

Note

These commands apply to ORDS version 19.1 and later.

5. Install the ORDS schema using the downloaded ords.war file.

```
java -jar ords.war install advanced
```

The program prompts you for the following information. The default values are in brackets. For more information, see [Introduction to Oracle REST data services](#) in the Oracle documentation.

- Enter the location to store configuration data:

Enter /home/apexuser/ORDS. This is the location of the ORDS configuration files.

- Specify the database connection type to use. Enter number for [1] Basic [2] TNS [3] Custom URL [1]:

Choose the desired connection type.

- Enter the name of the database server [localhost]: *DB_instance_endpoint*

Choose the default or enter the correct value.

- Enter the database listener port [1521]: *DB_instance_port*

Choose the default or enter the correct value.

- Enter 1 to specify the database service name, or 2 to specify the database SID [1]:

Choose 2 to specify the database SID.

- Database SID [xe]

Choose the default or enter the correct value.

- Enter 1 if you want to verify/install Oracle REST Data Services schema or 2 to skip this step [1]:

Choose 1. This step creates the Oracle REST Data Services proxy user named ORDS_PUBLIC_USER.

- Enter the database password for ORDS_PUBLIC_USER:

Enter the password, and then confirm it.

- Requires to login with administrator privileges to verify Oracle REST Data Services schema.

Enter the administrator user name: *master_user*

Enter the database password for *master_user*: *master_user_password*

Confirm the password: *master_user_password*

- Enter the default tablespace for ORDS_METADATA [SYSAUX].

Enter the temporary tablespace for ORDS_METADATA [TEMP].

Enter the default tablespace for ORDS_PUBLIC_USER [USERS].

Enter the temporary tablespace for ORDS_PUBLIC_USER [TEMP].

- Enter 1 if you want to use PL/SQL Gateway or 2 to skip this step. If you're using Oracle Application Express or migrating from mod_plsql, you must enter 1 [1].

Choose the default.

- Enter the PL/SQL Gateway database user name [APEX_PUBLIC_USER]

Choose the default.

- Enter the database password for APEX_PUBLIC_USER:

Enter the password, and then confirm it.

- Enter 1 to specify passwords for Application Express RESTful Services database users (APEX_LISTENER, APEX_REST_PUBLIC_USER) or 2 to skip this step [1]:

Choose 2 for APEX 4.1.1.V1; choose 1 for all other APEX versions.

- [Not needed for APEX 4.1.1.v1] Database password for APEX_LISTENER

Enter the password (if required), and then confirm it.

- [Not needed for APEX 4.1.1.v1] Database password for APEX_REST_PUBLIC_USER

Enter the password (if required), and then confirm it.

- Enter a number to select a feature to enable:

Enter 1 to enable all features: SQL Developer Web, REST Enabled SQL, and Database API.

- Enter 1 if you wish to start in standalone mode or 2 to exit [1]:

Enter 1.

- Enter the APEX static resources location:

If you unzipped APEX installation files into /home/apexuser, enter /home/apexuser/apex/images. Otherwise, enter *unzip_path*/apex/images, where *unzip_path* is the directory where you unzipped the file.

- Enter 1 if using HTTP or 2 if using HTTPS [1]:

If you enter 1, specify the HTTP port. If you enter 2, specify the HTTPS port and the SSL host name. The HTTPS option prompts you to specify how you will provide the certificate:

- Enter 1 to use the self-signed certificate.
- Enter 2 to provide your own certificate. If you enter 2, specify the path for the SSL certificate and the path for the SSL certificate private key.

6. Set a password for the APEX admin user. To do this, use SQL*Plus to connect to your DB instance as the master user, and then run the following commands.

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
```

```
grant APEX_ADMINISTRATOR_ROLE to master;
@/home/apexuser/apex/apxchpwd.sql
```

Replace *master* with your master user name. When the `apxchpwd.sql` script prompts you, enter a new admin password.

7. Start the ORDS listener. Run the following code.

```
java -jar ords.war
```

The first time you start ORDS, you are prompted to provide the location of the APEX Static resources. This images folder is located in the `/apex/images` directory in the installation directory for APEX.

8. Return to the APEX administration window in your browser and choose **Administration**. Next, choose **Application Express Internal Administration**. When you are prompted for credentials, enter the following information:
 - **User name** – `admin`
 - **Password** – the password you set using the `apxchpwd.sql` script

Choose **Login**, and then set a new password for the `admin` user.

Your listener is now ready for use.

Setting up Oracle APEX listener

Note

Oracle APEX Listener is deprecated.

Amazon RDS for Oracle continues to support APEX version 4.1.1 and Oracle APEX Listener version 1.1.4. We recommend that you use the latest supported versions of Oracle APEX and ORDS.

Install Oracle APEX Listener on a separate host such as an Amazon EC2 instance, an on-premises server at your company, or your desktop computer. We assume that the name of your host is `myapexhost.example.com`, and that your host is running Linux.

Preparing to install Oracle APEX listener

Before you can install Oracle APEX Listener, you need to create a nonprivileged OS user, and then download and unzip the APEX installation file.

To prepare for Oracle APEX listener installation

1. Log in to `myapexhost.example.com` as `root`.
2. Create a nonprivileged OS user to own the listener installation. The following command creates a new user named `apexuser`.

```
useradd -d /home/apexuser apexuser
```

The following command assigns a password to the new user.

```
passwd apexuser;
```

3. Log in to `myapexhost.example.com` as `apexuser`, and download the APEX installation file from Oracle to your `/home/apexuser` directory:

- <http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html>
 - [Oracle Application Express prior release archives](#)
4. Unzip the file in the /home/apexuser directory.

```
unzip apex_<version>.zip
```

After you unzip the file, there is an apex directory in the /home/apexuser directory.

5. While you are still logged into myapexhost.example.com as apexuser, download the Oracle APEX Listener file from Oracle to your /home/apexuser directory.

Installing and configuring Oracle APEX listener

Before you can use APEX, you need to download the apex.war file, use Java to install Oracle APEX Listener, and then start the listener.

To install and configure Oracle APEX listener

1. Create a new directory based on Oracle APEX Listener and open the listener file.

Run the following code:

```
mkdir /home/apexuser/apexlistener
cd /home/apexuser/apexlistener
unzip ..//apex_listener.<version>.zip
```

2. Run the following code.

```
java -Dapex.home=../apex -Dapex.images=/home/apexuser/apex/images -Dapex.erase -jar ../apex.war
```

3. Enter information for the program prompts following:

- The APEX Listener Administrator user name. The default is *adminlistener*.
- A password for the APEX Listener Administrator.
- The APEX Listener Manager user name. The default is *managerlistener*.
- A password for the APEX Listener Administrator.

The program prints a URL that you need to complete the configuration, as follows.

```
INFO: Please complete configuration at: http://localhost:8080/apex/listenerConfigure
Database is not yet configured
```

4. Leave Oracle APEX Listener running so that you can use Oracle Application Express. When you have finished this configuration procedure, you can run the listener in the background.
5. From your web browser, go to the URL provided by the APEX Listener program. The Oracle Application Express Listener administration window appears. Enter the following information:
- **Username** – APEX_PUBLIC_USER
 - **Password** – the password for APEX_PUBLIC_USER. This password is the one that you specified earlier when you configured the APEX repository. For more information, see [Unlocking the public user account \(p. 1142\)](#).
 - **Connection type** – Basic

- **Hostname** – the endpoint of your Amazon RDS DB instance, such as `mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com`.
 - **Port** – 1521
 - **SID** – the name of the database on your Amazon RDS DB instance, such as `mydb`.
6. Choose **Apply**. The APEX administration window appears.
 7. Set a password for the APEX admin user. To do this, use SQL*Plus to connect to your DB instance as the master user, and then run the following commands.

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
grant APEX_ADMINISTRATOR_ROLE to master;
@/home/apexuser/apex/apxchpwd.sql
```

Replace `master` with your master user name. When the `apxchpwd.sql` script prompts you, enter a new admin password.

8. Return to the APEX administration window in your browser and choose **Administration**. Next, choose **Application Express Internal Administration**. When you are prompted for credentials, enter the following information:

- **User name** – `admin`
- **Password** – the password you set using the `apxchpwd.sql` script

Choose **Login**, and then set a new password for the `admin` user.

Your listener is now ready for use.

Upgrading the APEX version

Important

Back up your DB instance before you upgrade APEX. For more information, see [Creating a DB snapshot \(p. 346\)](#) and [Testing an Oracle DB upgrade \(p. 1215\)](#).

To upgrade APEX with your DB instance, do the following:

- Create a new option group for the upgraded version of your DB instance.
- Add the upgraded versions of APEX and APEX-DEV to the new option group. Be sure to include any other options that your DB instance uses. For more information, see [Option group considerations \(p. 1213\)](#).
- When you upgrade your DB instance, specify the new option group for your upgraded DB instance.

After you upgrade your version of APEX, the APEX schema for the previous version might still exist in your database. If you don't need it anymore, you can drop the old APEX schema from your database after you upgrade.

If you upgrade the APEX version and RESTful services were not configured in the previous APEX version, we recommend that you configure RESTful services. For more information, see [Configuring RESTful services for Oracle APEX \(p. 1142\)](#).

In some cases when you plan to do a major version upgrade of your DB instance, you might find that you're using an APEX version that isn't compatible with your target database version. In these cases, you can upgrade your version of APEX before you upgrade your DB instance. Upgrading APEX first can reduce the amount of time that it takes to upgrade your DB instance.

Note

After upgrading APEX, install and configure a listener for use with the upgraded version. For instructions, see [Setting up Oracle APEX listener \(p. 1146\)](#).

Removing the APEX option

You can remove the Amazon RDS APEX options from a DB instance. To remove the APEX options from a DB instance, do one of the following:

- To remove the APEX options from multiple DB instances, remove the APEX options from the option group they belong to. This change affects all DB instances that use the option group. When you remove the APEX options from an option group that is attached to multiple DB instances, a brief outage occurs while all the DB instances are restarted.

For more information, see [Removing an option from an option group \(p. 224\)](#).

- To remove the APEX options from a single DB instance, modify the DB instance and specify a different option group that doesn't include the APEX options. You can specify the default (empty) option group, or a different custom option group. When you remove the APEX options, a brief outage occurs while your DB instance is automatically restarted.

For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

When you remove the APEX options from a DB instance, the APEX schema is removed from your database.

Oracle Enterprise Manager

Amazon RDS supports Oracle Enterprise Manager (OEM). OEM is the Oracle product line for integrated management of enterprise information technology.

Amazon RDS supports OEM through the following options.

Option	Option ID	Support for
OEM Database Express (p. 1150)	OEM	OEM Database Express 12c
OEM Management Agent (p. 1154)	OEM_AGENT	OEM Cloud Control for 13c OEM Cloud Control for 12c

Note

You can use OEM Database or OEM Management Agent, but not both.

Oracle Enterprise Manager Database Express

Amazon RDS supports Oracle Enterprise Manager (OEM) Database Express through the use of the OEM option. Amazon RDS supports Oracle Enterprise Manager Database Express for Oracle Database 19c, Oracle Database 18c, and Oracle Database 12c.

OEM Database Express and Database Control are similar tools that have a web-based interface for Oracle database administration. For more information about these tools, see [Accessing Enterprise Manager database Express 18c](#) and [Accessing Enterprise Manager database Express 12c](#) in the Oracle documentation.

The following is a limitation for OEM Database Express:

- OEM Database Express isn't supported on the db.t3.micro or db.t3.small DB instance classes.

For more information about DB instance classes, see [RDS for Oracle instance classes \(p. 992\)](#).

OEM Database option settings

Amazon RDS supports the following settings for the OEM option.

Option setting	Valid values	Description
Port	An integer value	The port on the DB instance that listens for OEM Database. The default for OEM Database Express is 5500.
Security Groups	—	A security group that has access to Port .

Adding the OEM Database option

The general process for adding the OEM option to a DB instance is the following:

1. Create a new option group, or copy or modify an existing option group.
2. Add the option to the option group.
3. Associate the option group with the DB instance.

When you add the OEM option for an Oracle Database 12c or later DB instance, a brief outage occurs while your DB instance is automatically restarted.

To add the OEM option to a DB instance

1. Determine the option group you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group with the following settings:
 - a. For **Engine** choose the oracle edition for your DB instance.
 - b. For **Major engine version** choose the version of your DB instance.

For more information, see [Creating an option group \(p. 214\)](#).

2. Add the OEM option to the option group, and configure the option settings. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#). For more information about each setting, see [OEM Database option settings \(p. 1150\)](#).

Note

If you add the OEM option to an existing option group that is already attached to one or more Oracle Database 19c, Oracle Database 18c, or Oracle Database 12c DB instances, a brief outage occurs while all the DB instances are automatically restarted.

3. Apply the option group to a new or existing DB instance:

- For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- For an existing DB instance, you apply the option group by modifying the instance and attaching the new option group. When you add the OEM option for an Oracle Database 19c, Oracle Database 18c, or Oracle Database 12c DB instance, a brief outage occurs while your DB instance is automatically restarted. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Note

You can also use the AWS CLI to add the OEM option. For examples, see [Adding an option to an option group \(p. 216\)](#).

Using OEM Database

After you enable the OEM option, you can begin using the OEM Database tool from your web browser.

You can access either OEM Database Control or OEM Database Express from your web browser. For example, if the endpoint for your Amazon RDS DB instance is `mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com`, and your OEM port is 1158, then the URL to access the OEM Database Control the following.

```
https://mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com:1158/em
```

When you access either tool from your web browser, a login window appears that prompts you for a user name and password. Type the master user name and master password for your DB instance. You are now ready to manage your Oracle databases.

Modifying OEM Database settings

After you enable OEM Database, you can modify the Security Groups setting for the option.

You can't modify the OEM port number after you have associated the option group with a DB instance. To change the OEM port number for a DB instance, do the following:

1. Create a new option group.
2. Add the OEM option with the new port number to the new option group.
3. Remove the existing option group from the DB instance.
4. Add the new option group to the DB instance.

For more information about how to modify option settings, see [Modifying an option setting \(p. 221\)](#). For more information about each setting, see [OEM Database option settings \(p. 1150\)](#).

Using OEM Database

You can use Amazon RDS procedures to run certain OEM Database Express tasks. By running these procedures, you can do the tasks listed following.

Note

OEM Database Express tasks run asynchronously.

Tasks

- [Switching the website front end for OEM Database Express to Adobe Flash \(p. 1152\)](#)
- [Switching the website front end for OEM Database Express to Oracle JET \(p. 1152\)](#)

[Switching the website front end for OEM Database Express to Adobe Flash](#)

Note

This task is only available on instances running Oracle Database 19c or later.

Starting with Oracle Database 19c, Oracle has deprecated the former OEM Database Express user interface, which was based on Adobe Flash. Instead, OEM Database Express now uses an interface built with Oracle JET. If you experience difficulties with the new interface, you can switch back to the deprecated Flash-based interface. Difficulties you might experience with the new interface include being stuck on a Loading screen after logging in to OEM Database Express. You might also miss certain features that were present in the Flash-based version of OEM Database Express.

To switch the OEM Database Express website front end to Adobe Flash, run the Amazon RDS procedure `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash`. This procedure is equivalent to the `execemx emx` SQL command.

Security best practices discourage the use of Adobe Flash. Although you can revert to the Flash-based OEM Database Express, we recommend the use of the JET-based OEM Database Express websites if possible. If you revert to using Adobe Flash and want to switch back to using Oracle JET, use the `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet` procedure. After an Oracle database upgrade, a newer version of Oracle JET might resolve JET-related issues in OEM Database Express. For more information about switching to Oracle JET, see [Switching the website front end for OEM Database Express to Oracle JET \(p. 1152\)](#).

Note

Running this task from the source DB instance for a read replica also causes the read replica to switch its OEM Database Express website front ends to Adobe Flash.

The following procedure invocation creates a task to switch the OEM Database Express website to Adobe Flash and returns the ID of the task.

```
SELECT rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash() as TASK_ID from DUAL;
```

You can view the result by displaying the task's output file.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP', 'dbtask-task-id.log'));
```

Replace `task-id` with the task ID returned by the procedure. For more information about the Amazon RDS procedure `rdsadmin.rds_file_util.read_text_file`, see [Reading files in a DB instance directory \(p. 1096\)](#).

You can also view the contents of the task's output file in the AWS Management Console by searching the log entries in the **Logs & events** section for the task-id.

[Switching the website front end for OEM Database Express to Oracle JET](#)

Note

This task is only available on Oracle DB instances running version 19c or later.

To switch the OEM Database Express website front end to Oracle JET, run the Amazon RDS procedure `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet`. This procedure is equivalent to the `execemx omx` SQL command.

By default, the OEM Database Express websites for Oracle DB instances running 19c or later use Oracle JET. If you used the `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash` procedure to switch the OEM Database Express website front end to Adobe Flash, you can switch back to Oracle JET. To do this, use the `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet` procedure. For more information about switching to Adobe Flash, see [Switching the website front end for OEM Database Express to Adobe Flash \(p. 1152\)](#).

Note

Running this task from the source DB instance for a read replica also causes the read replica to switch its OEM Database Express website front ends to Oracle JET.

The following procedure invocation creates a task to switch the OEM Database Express website to Oracle JET and returns the ID of the task.

```
SELECT rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet() as TASK_ID from DUAL;
```

You can view the result by displaying the task's output file.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Replace `task-id` with the task ID returned by the procedure. For more information about the Amazon RDS procedure `rdsadmin.rds_file_util.read_text_file`, see [Reading files in a DB instance directory \(p. 1096\)](#)

You can also view the contents of the task's output file in the AWS Management Console by searching the log entries in the **Logs & events** section for the `task-id`.

Removing the OEM Database option

You can remove the OEM option from a DB instance. When you remove the OEM option for an Oracle Database 12c or later DB instance, a brief outage occurs while your instance is automatically restarted. Therefore, after you remove the OEM option, you don't need to restart your DB instance.

To remove the OEM option from a DB instance, do one of the following:

- Remove the OEM option from the option group it belongs to. This change affects all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).
- Modify the DB instance and specify a different option group that doesn't include the OEM option. This change affects a single DB instance. You can specify the default (empty) option group, or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Oracle Management Agent for Enterprise Manager Cloud Control

Oracle Enterprise Manager (OEM) Management Agent is a software component that monitors targets running on hosts and communicates that information to the middle-tier Oracle Management Service (OMS). For more information, see [Overview of Oracle Enterprise Manager cloud control 12c](#) and [Overview of Oracle Enterprise Manager cloud control 13c](#) in the Oracle documentation.

Amazon RDS supports Management Agent through the use of the `OEM_AGENT` option. Management Agent requires an Amazon RDS DB instance running Oracle Database 19c (19.0.0.0), 18.0.0.0, 12.2.0.1, or 12.1.0.2.

Amazon RDS supports Management Agent for the following versions of OEM:

- Oracle Enterprise Manager Cloud Control for 13c
- Oracle Enterprise Manager Cloud Control for 12c

Topics

- [Prerequisites for Management Agent \(p. 1154\)](#)
- [Limitations for Management Agent \(p. 1156\)](#)
- [Option settings for Management Agent \(p. 1156\)](#)
- [Adding the Management Agent option \(p. 1158\)](#)
- [Using the Management Agent \(p. 1159\)](#)
- [Modifying Management Agent settings \(p. 1160\)](#)
- [Performing database tasks with the Management Agent \(p. 1160\)](#)
- [Removing the Management Agent option \(p. 1162\)](#)

Prerequisites for Management Agent

To use Management Agent, ensure that you meet the following prerequisites.

General prerequisites

Following are general prerequisites for using Management Agent:

- You need an Oracle Management Service (OMS) that is configured to connect to your Amazon RDS DB instance.
- In most cases, you must configure your VPC to allow connections from OMS to your DB instance. If you aren't familiar with Amazon Virtual Private Cloud (Amazon VPC), we recommend that you complete the steps in [Tutorial: Create an Amazon VPC for use with a DB instance \(p. 1737\)](#) before continuing.
- Management Agent version 13.4.0.9.v1 requires OMS version 13.4.0.9 or later and patch 32198287.
- Ensure that you have sufficient storage space for your OEM release:
 - At least 8.5 GiB for OEM 13c Release 4
 - At least 8.5 GiB for OEM 13c Release 3
 - At least 5.5 GiB for OEM 13c Release 2
 - At least 4.5 GiB OEM 13c Release 1
 - At least 2.5 GiB for OEM 12c
- If you are using Management Agent versions `OEM_AGENT 13.2.0.0.v3` and `13.3.0.0.v2`, and if you want to use TCPS connectivity, follow the instructions in [Configuring third party CA certificates for communication with target databases](#) in the Oracle documentation. Also, update the JDK on your OMS by following the instructions in the Oracle document with the Oracle Doc ID 2241358.1. This step ensures that OMS supports all the cipher suites that the database supports.

Note

TCPS connectivity between the Management Agent and the DB instance is only supported for Management Agent versions OEM_AGENT_13.2.0.0.v3 and 13.3.0.0.v2.

Oracle Database release prerequisites

Following are the supported Oracle Database versions for each Management Agent version.

Management Agent version	Oracle Database 19c	Oracle Database 18c	Oracle Database 12c Release 2 (12.2)	Oracle Database 12c Release 1 (12.1)
13.4.0.9.v1	Supported	Supported	Supported	Supported
13.3.0.0.v2	Supported	Supported	Supported	Supported
13.3.0.0.v1	Supported	Supported	Supported	Supported
13.2.0.0.v3	Supported	Supported	Supported	Supported
13.2.0.0.v2	Supported	Supported	Supported	Supported
13.2.0.0.v1	Supported	Supported	Supported	Supported
13.1.0.0.v1	Supported	Supported	Supported	Supported
12.1.0.5.v1	Not supported	Supported	Supported	Supported
12.1.0.4.v1	Not supported	Supported	Supported	Supported

Following are prerequisites for different database versions:

- For an Amazon RDS DB instance running Oracle Database 19c (19.0.0.0), the minimum AGENT_VERSION is 13.1.0.0.v1.
 - For an Amazon RDS DB instance running Oracle Database 18c (18.0.0.0) or higher, meet the following requirements:
 - For OMS 13c2, apply the Enterprise Manager 13.2 Master Bundle Patch List, which includes plugins 13.2.1, 13.2.2, 13.2.3, 13.2.4 (Oracle Doc ID 2219797.1).
 - For OMS 13c2, apply the OMS PSU System Patch 28970534.
 - For OMS 13c2, apply the OMS-Side Plugin System 13.2.2.0.190131 Patch 29201709.
 - For an Amazon RDS DB instance running Oracle Database Release 2 (12.2.0.1) or lower, meet the following requirements:
 - For OMS 13c Release 2 with Oracle patch 25163555 applied, use OEM Agent 13.2.0.0.v2 or later.
Use OMSPatcher to apply the patch.
 - For unpatched OMS 13c Release 2, use OEM Agent 13.2.0.0.v1.
- Use OMSPatcher to apply patches.

OMS host communication prerequisites

Make sure that your OMS host and your Amazon RDS DB instance can communicate. Do the following:

- To connect from the Management Agent to your OMS, if your OMS is behind a firewall, add the IP addresses of your DB instances to your OMS.

Make sure the firewall for the OMS allows traffic from both the DB listener port (default 1521) and the OEM Agent port (default 3872), originating from the IP address of the DB instance.

- To connect from your OMS to the Management Agent, if your OMS has a publicly resolvable host name, add the OMS address to a security group. Your security group must have inbound rules that allow access to the DB listener port and the Management Agent port. For an example of creating a security and adding inbound rules, see [Tutorial: Create an Amazon VPC for use with a DB instance \(p. 1737\)](#).
- To connect from your OMS to the Management Agent, if your OMS doesn't have a publicly resolvable host name, use one of the following:
 - If your OMS is hosted on an Amazon Elastic Compute Cloud (Amazon EC2) instance in a private VPC, you can set up VPC peering to connect from OMS to Management Agent. For more information, see [A DB instance in a VPC accessed by an EC2 instance in a different VPC \(p. 1722\)](#).
 - If your OMS is hosted on-premises, you can set up a VPN connection to allow access from OMS to Management Agent. For more information, see [A DB instance in a VPC accessed by a client application through the internet \(p. 1723\)](#) or [VPN connections](#).

Limitations for Management Agent

Following are some limitations to using Management Agent:

- Administrative tasks such as job execution and database patching, that require host credentials, aren't supported.
- Host metrics and the process list aren't guaranteed to reflect the actual system state. Thus, you shouldn't use OEM to monitor the root file system or mount point file system. For more information about monitoring the operating system, see [Using Enhanced Monitoring \(p. 471\)](#).
- Autodiscovery isn't supported. You must manually add database targets.
- OMS module availability depends on your database edition. For example, the database performance diagnosis and tuning module is only available for Oracle Database Enterprise Edition.
- Management Agent consumes additional memory and computing resources. If you experience performance problems after enabling the `OEM_AGENT` option, we recommend that you scale up to a larger DB instance class. For more information, see [DB instance classes \(p. 7\)](#) and [Modifying an Amazon RDS DB instance \(p. 250\)](#).
- The user running the `OEM_AGENT` on the Amazon RDS host doesn't have operating system access to the alert log. Thus, you can't collect metrics for `DB Alert Log` and `DB Alert Log Error Status` in OEM.

Option settings for Management Agent

Amazon RDS supports the following settings for the Management Agent option.

Option setting	Required	Valid values	Description
Version (AGENT_VERSION)	Yes	13.4.0.9.v1 13.3.0.0.v2 13.3.0.0.v1 13.2.0.0.v3 13.2.0.0.v2 13.2.0.0.v1	The version of the Management Agent software. The AWS CLI option name is OptionVersion. Note In the AWS GovCloud (US) Regions, 12.1 and 13.1 versions aren't available.

Option setting	Required	Valid values	Description
		13.1.0.0.v1 12.1.0.5.v1 12.1.0.4.v1	
Port (AGENT_PORT)	Yes	An integer value	The port on the DB instance that listens for the OMS host. The default is 3872. Your OMS host must belong to a security group that has access to this port. The AWS CLI option name is <code>Port</code> .
Security Groups	Yes	Existing security groups	A security group that has access to Port . Your OMS host must belong to this security group. The AWS CLI option name is <code>VpcSecurityGroupMemberships</code> or <code>DBSecurityGroupMemberships</code> .
OMS_HOST	Yes	A string value, for example <i>my.example.oms</i>	The publicly accessible host name or IP address of the OMS. The AWS CLI option name is <code>OMS_HOST</code> .
OMS_PORT	Yes	An integer value	The HTTPS upload port on the OMS Host that listens for the Management Agent. To determine the HTTPS upload port, connect to the OMS host, and run the following command (which requires the <code>SYSMAN</code> password): <code>emctl status oms -details</code> The AWS CLI option name is <code>OMS_PORT</code> .
AGENT_REGISTRATION_PASSWORD	string value		The password that the Management Agent uses to authenticate itself with the OMS. We recommend that you create a persistent password in your OMS before enabling the <code>OEM_AGENT</code> option. With a persistent password you can share a single Management Agent option group among multiple Amazon RDS databases. The AWS CLI option name is <code>AGENT_REGISTRATION_PASSWORD</code> .
ALLOW_TLS_ONLY	No	true, false (default)	A value that configures the OEM Agent to support only the <code>TLSv1</code> protocol while the agent listens as a server. This setting is only supported for 12.1 agent versions. Later agent versions only support Transport Layer Security (TLS) by default.
MINIMUM_TLS_VERSION		<code>TLSv1</code> (default), <code>TLSv1.2</code>	A value that specifies the minimum TLS version supported by the OEM Agent while the agent listens as a server. This setting is only supported for agent versions 13.1.0.0.v1 and higher. Earlier agent versions only support the <code>TLSv1</code> setting.

Option setting	Required	Valid values	Description
TLS_CIPHER_SUITE	No	TLS_RSA_WITH_AES_128_CBC_SHA256 (Default supported by all agent versions) TLS_RSA_WITH_AES_128_CBC_SHA (Requires version 13.1.0.0.v1 or above) TLS_RSA_WITH_AES_256_CBC_SHA (Requires version 13.2.0.0.v3 or above) TLS_RSA_WITH_AES_256_CBC_SHA256 (Requires version 13.2.0.0.v3 or above)	A value that specifies the TLS cipher suite used by the OEM Agent while the agent listens as a server.

Adding the Management Agent option

The general process for adding the Management Agent option to a DB instance is the following:

1. Create a new option group, or copy or modify an existing option group.
2. Add the option to the option group.
3. Associate the option group with the DB instance.

If you encounter errors, check [My Oracle Support](#) documents for information about resolving specific problems.

After you add the Management Agent option, you don't need to restart your DB instance. As soon as the option group is active, the OEM Agent is active.

If your OMS host is using an untrusted third-party certificate, Amazon RDS returns the following error.

You successfully installed the OEM_AGENT option. Your OMS host is using an untrusted third party certificate.
Configure your OMS host with the trusted certificates from your third party.

If this error is returned, the Management Agent option isn't enabled until the problem is corrected. For information about correcting the problem, see the My Oracle Support document [2202569.1](#).

Console

To add the Management Agent option to a DB instance

1. Determine the option group you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group with the following settings:

- a. For **Engine** choose the oracle edition for your DB instance.
- b. For **Major engine version** choose the version of your DB instance.

For more information, see [Creating an option group \(p. 214\)](#).

2. Add the **OEM_AGENT** option to the option group, and configure the option settings. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#). For more information about each setting, see [Option settings for Management Agent \(p. 1156\)](#).
3. Apply the option group to a new or existing DB instance:
 - For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
 - For an existing DB instance, you apply the option group by modifying the instance and attaching the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

AWS CLI

The following example uses the AWS CLI `add-option-to-option-group` command to add the **OEM_AGENT** option to an option group called `myoptiongroup`.

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \
  --option-group-name "myoptiongroup" \
  --options
  OptionName=OEM_AGENT,OptionVersion=13.1.0.0.v1,Port=3872,VpcSecurityGroupMemberships=sg-1234567890,Opt
  {Name=OMS_PORT,Value=4903},{Name=AGENT_REGISTRATION_PASSWORD,Value=password}] \
  --apply-immediately
```

For Windows:

```
aws rds add-option-to-option-group ^
  --option-group-name "myoptiongroup" ^
  --options
  OptionName=OEM_AGENT,OptionVersion=13.1.0.0.v1,Port=3872,VpcSecurityGroupMemberships=sg-1234567890,Opt
  {Name=OMS_PORT,Value=4903},{Name=AGENT_REGISTRATION_PASSWORD,Value=password}] ^
  --apply-immediately
```

Using the Management Agent

After you enable the Management Agent option, take the following steps to begin using it.

To use the Management Agent

1. Unlock and reset the DBSNMP account credential. Do this by running the following code on your target database on your DB instance and using your master user account.

```
ALTER USER dbsnmp IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

2. Add your targets to the OMS console manually:

- a. In your OMS console, choose **Setup**, **Add Target**, **Add Targets Manually**.
- b. Choose **Add Targets Declaratively by Specifying Target Monitoring Properties**.
- c. For **Target Type**, choose **Database Instance**.
- d. For **Monitoring Agent**, choose the agent with the identifier that is the same as your RDS DB instance identifier.

- e. Choose **Add Manually**.
- f. Enter the endpoint for the Amazon RDS DB instance, or choose it from the host name list. Make sure that the specified host name matches the endpoint of the Amazon RDS DB instance.

For information about finding the endpoint for your Amazon RDS DB instance, see [Finding the endpoint of your Oracle DB instance \(p. 1001\)](#).
- g. Specify the following database properties:
 - For **Target name**, enter a name.
 - For **Database system name**, enter a name.
 - For **Monitor username**, enter `dbsnmp`.
 - For **Monitor password**, enter the password from step 1.
 - For **Role**, enter `normal`.
 - For **Oracle home path**, enter `/oracle`.
 - For **Listener Machine name**, the agent identifier already appears.
 - For **Port**, enter the database port. The RDS default port is 1521.
 - For **Database name**, enter the name of your database.
- h. Choose **Test Connection**.
- i. Choose **Next**. The target database appears in your list of monitored resources.

Modifying Management Agent settings

After you enable the Management Agent, you can modify settings for the option. For more information about how to modify option settings, see [Modifying an option setting \(p. 221\)](#). For more information about each setting, see [Option settings for Management Agent \(p. 1156\)](#).

Performing database tasks with the Management Agent

You can use Amazon RDS procedures to run certain EMCTL commands on the Management Agent. By running these procedures, you can do the tasks listed following.

Note

Tasks are executed asynchronously.

Tasks

- [Getting the status of the Management Agent \(p. 1160\)](#)
- [Restarting the Management Agent \(p. 1161\)](#)
- [Listing the targets monitored by the Management Agent \(p. 1161\)](#)
- [Listing the collection threads monitored by the Management Agent \(p. 1161\)](#)
- [Clearing the Management Agent state \(p. 1161\)](#)
- [Making the Management Agent upload its OMS \(p. 1162\)](#)
- [Pinging the OMS \(p. 1162\)](#)
- [Viewing the status of an ongoing task \(p. 1162\)](#)

Getting the status of the Management Agent

To get the status of the Management Agent, run the Amazon RDS procedure `rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent`. This procedure is equivalent to the `emctl status agent` command.

The following procedure creates a task to get the Management Agent's status and returns the ID of the task.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent() as TASK_ID from DUAL;
```

To view the result by displaying the task's output file, see [Viewing the status of an ongoing task \(p. 1162\)](#).

Restarting the Management Agent

To restart the Management Agent, run the Amazon RDS procedure

`rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent`. This procedure is equivalent to running the `emctl stop agent` and `emctl start agent` commands.

The following procedure creates a task to restart the Management Agent and returns the ID of the task.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.restart_oem_agent() as TASK_ID from DUAL;
```

To view the result by displaying the task's output file, see [Viewing the status of an ongoing task \(p. 1162\)](#).

Listing the targets monitored by the Management Agent

To list the targets monitored by the Management Agent, run the Amazon RDS procedure

`rdsadmin.rdsadmin_oem_agent_tasks.list_targets_oem_agent`. This procedure is equivalent to running the `emctl config agent listtargets` command.

The following procedure creates a task to list the targets monitored by the Management Agent and returns the ID of the task.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.list_targets_oem_agent() as TASK_ID from DUAL;
```

To view the result by displaying the task's output file, see [Viewing the status of an ongoing task \(p. 1162\)](#).

Listing the collection threads monitored by the Management Agent

To list of all the running, ready, and scheduled collection threads

monitored by the Management Agent, run the Amazon RDS procedure

`rdsadmin.rdsadmin_oem_agent_tasks.list_clxn_threads_oem_agent`. This procedure is equivalent to the `emctl status agent scheduler` command.

The following procedure creates a task to list the collection threads and returns the ID of the task.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.list_clxn_threads_oem_agent() as TASK_ID from DUAL;
```

To view the result by displaying the task's output file, see [Viewing the status of an ongoing task \(p. 1162\)](#).

Clearing the Management Agent state

To clear the Management Agent's state, run the Amazon RDS procedure

`rdsadmin.rdsadmin_oem_agent_tasks.clearstate_oem_agent`. This procedure is equivalent to running the `emctl clearstate agent` command.

The following procedure creates a task that clears the Management Agent's state and returns the ID of the task.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.clearstate_oem_agent() as TASK_ID from DUAL;
```

To view the result by displaying the task's output file, see [Viewing the status of an ongoing task \(p. 1162\)](#).

Making the Management Agent upload its OMS

To make the Management Agent upload the Oracle Management Server (OMS) associated with it, run the Amazon RDS procedure `rdsadmin.rdsadmin_oem_agent_tasks.upload_oem_agent`. This procedure is equivalent to running the `emctl upload agent` command.

The following procedure creates a task that makes the Management Agent upload its associated OMS and return the ID of the task.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.upload_oem_agent() as TASK_ID from DUAL;
```

To view the result by displaying the task's output file, see [Viewing the status of an ongoing task \(p. 1162\)](#).

Pinging the OMS

To ping the Management Agent's OMS, run the Amazon RDS procedure `rdsadmin.rdsadmin_oem_agent_tasks.ping_oms_oem_agent`. This procedure is equivalent to running the `emctl pingOMS` command.

The following procedure creates a task that pings the Management Agent's OMS and returns the ID of the task.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.ping_oms_oem_agent() as TASK_ID from DUAL;
```

To view the result by displaying the task's output file, see [Viewing the status of an ongoing task \(p. 1162\)](#).

Viewing the status of an ongoing task

You can view the status of an ongoing task in a bdump file. The bdump files are located in the `/rdsdbdata/log/trace` directory. Each bdump file name is in the following format.

```
dbtask-task-id.log
```

When you want to monitor a task, replace `task-id` with the ID of the task that you want to monitor.

To view the contents of bdump files, run the Amazon RDS procedure `rdsadmin.rds_file_util.read_text_file`. The following query returns the contents of the `dbtask-1546988886389-2444.log` bdump file.

```
SELECT text FROM
  table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1546988886389-2444.log'));
```

For more information about the Amazon RDS procedure `rdsadmin.rds_file_util.read_text_file`, see [Reading files in a DB instance directory \(p. 1096\)](#).

Removing the Management Agent option

You can remove the OEM Agent from a DB instance. After you remove the OEM Agent, you don't need to restart your DB instance.

To remove the OEM Agent from a DB instance, do one of the following:

- Remove the OEM Agent option from the option group it belongs to. This change affects all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).
- Modify the DB instance and specify a different option group that doesn't include the OEM Agent option. This change affects a single DB instance. You can specify the default (empty) option group, or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Oracle Java virtual machine

Amazon RDS supports Oracle Java Virtual Machine (JVM) through the use of the `JVM` option. Oracle Java provides a SQL schema and functions that facilitate Oracle Java features in an Oracle database. For more information, see [Introduction to Java in Oracle database](#) in the Oracle documentation.

You can use Oracle JVM with the following Oracle Database versions:

- Oracle Database 19c (19.0.0.0), all versions
- Oracle Database 18c (18.0.0.0), all versions
- Oracle Database 12c Release 2 (12.2), all versions
- Oracle Database 12c Release 1 (12.1), version 12.1.0.2.v13 or later

Java implementation in Amazon RDS has a limited set of permissions. The master user is granted the `RDS_JAVA_ADMIN` role, which grants a subset of the privileges granted by the `JAVA_ADMIN` role. To list the privileges granted to the `RDS_JAVA_ADMIN` role, run the following query on your DB instance:

```
SELECT * FROM dba_java_policy
  WHERE grantee IN ('RDS_JAVA_ADMIN', 'PUBLIC')
    AND enabled = 'ENABLED'
  ORDER BY type_name, name, grantee;
```

Prerequisites for Oracle JVM

The following are prerequisites for using Oracle Java:

- Your DB instance must be inside a virtual private cloud (VPC). For more information, see [Determining whether you are using the EC2-VPC or EC2-Classic platform \(p. 1718\)](#).
- Your DB instance must be of a large enough class. Oracle Java isn't supported for the `db.t3.micro` or `db.t3.small` DB instance classes. For more information, see [DB instance classes \(p. 7\)](#).
- Your DB instance must have **Auto Minor Version Upgrade** enabled. This option enables your DB instance to receive minor DB engine version upgrades automatically when they become available. Amazon RDS uses this option to update your DB instance to the latest Oracle Patch Set Update (PSU) or Release Update (RU). For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Best practices for Oracle JVM

The following are best practices for using Oracle Java:

- For maximum security, use the `JVM` option with Secure Sockets Layer (SSL). For more information, see [Oracle Secure Sockets Layer \(p. 1182\)](#).
- Configure your DB instance to restrict network access. For more information, see [Scenarios for accessing a DB instance in a VPC \(p. 1720\)](#) and [Working with a DB instance in a VPC \(p. 1727\)](#).

Adding the Oracle JVM option

The following is the general process for adding the `JVM` option to a DB instance:

1. Create a new option group, or copy or modify an existing option group.
2. Add the option to the option group.

3. Associate the option group with the DB instance.

There is a brief outage while the **JVM** option is added. After you add the option, you don't need to restart your DB instance. As soon as the option group is active, Oracle Java is available.

Note

During this outage, password verification functions are disabled briefly. You can also expect to see events related to password verification functions during the outage. Password verification functions are enabled again before the Oracle DB instance is available.

To add the JVM option to a DB instance

1. Determine the option group that you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group with the following settings:
 - For **Engine**, choose the DB engine used by the DB instance (**oracle-ee**, **oracle-se**, **oracle-se1**, or **oracle-se2**).
 - For **Major engine version**, choose the version of your DB instance.

For more information, see [Creating an option group \(p. 214\)](#).

2. Add the **JVM** option to the option group. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).
3. Apply the option group to a new or existing DB instance:
 - For a new DB instance, apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
 - For an existing DB instance, apply the option group by modifying the instance and attaching the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
4. Grant the required permissions to users.

The Amazon RDS master user has the permissions to use the **JVM** option by default. If other users require these permissions, connect to the DB instance as the master user in a SQL client and grant the permissions to the users.

The following example grants the permissions to use the **JVM** option to the `test_proc` user.

```
create user test_proc identified by password;
CALL dbms_java.grant_permission('TEST_PROC',
  'oracle.aurora.security.JServerPermission', 'LoadClassInPackage.*', '');
```

After the user is granted the permissions, the following query should return output.

```
select * from dba_java_policy where grantee='TEST_PROC';
```

Note

The Oracle user name is case-sensitive, and it usually has all uppercase characters.

Removing the Oracle JVM option

You can remove the **JVM** option from a DB instance. There is a brief outage while the option is removed. After you remove the **JVM** option, you don't need to restart your DB instance.

Warning

Removing the `JVM` option can result in data loss if the DB instance is using data types that were enabled as part of the option. Back up your data before proceeding. For more information, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#).

To remove the `JVM` option from a DB instance, do one of the following:

- Remove the `JVM` option from the option group it belongs to. This change affects all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).
- Modify the DB instance and specify a different option group that doesn't include the `JVM` option. This change affects a single DB instance. You can specify the default (empty) option group, or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Oracle Label Security

Amazon RDS supports Oracle Label Security for the Enterprise Edition of Oracle Database 12c through the use of the OLS option.

Most database security controls access at the object level. Oracle Label Security provides fine-grained control of access to individual table rows. For example, you can use Label Security to enforce regulatory compliance with a policy-based administration model. You can use Label Security policies to control access to sensitive data, and restrict access to only users with the appropriate clearance level. For more information, see [Introduction to Oracle Label Security](#) in the Oracle documentation.

Important

For Oracle Database 19c, Oracle Database 18c, and Oracle Database 12c Release 2 (12.2) on Amazon RDS, Oracle Label Security is a permanent and persistent option. You can't remove Oracle Label Security from an Oracle Database 19c, Oracle Database 18c, or Oracle Database 12c Release 2 (12.2) DB instance.

Prerequisites for Oracle Label Security

The following are prerequisites for using Oracle Label Security:

- Your DB instance must use the Bring Your Own License model. For more information, see [Oracle licensing options \(p. 990\)](#).
- You must have a valid license for Oracle Enterprise Edition with Software Update License and Support.
- Your Oracle license must include the Label Security option.

Adding the Oracle Label Security option

The general process for adding the Oracle Label Security option to a DB instance is the following:

1. Create a new option group, or copy or modify an existing option group.
2. Add the option to the option group.
3. Associate the option group with the DB instance.

After you add the Label Security option, as soon as the option group is active, Label Security is active.

To add the label security option to a DB instance

1. Determine the option group you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group with the following settings:
 - a. For **Engine**, choose **oracle-ee**.
 - b. For **Major engine version**, choose the version of your DB instance.

For more information, see [Creating an option group \(p. 214\)](#).

2. Add the **OLS** option to the option group. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).

Important

If you add Label Security to an existing option group that is already attached to one or more DB instances, all the DB instances are restarted.

3. Apply the option group to a new or existing DB instance:

- For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- For an existing DB instance, you apply the option group by modifying the instance and attaching the new option group. When you add the Label Security option to an existing DB instance, a brief outage occurs while your DB instance is automatically restarted. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Using Oracle Label Security

To use Oracle Label Security, you create policies that control access to specific rows in your tables. For more information, see [Creating an Oracle Label Security policy](#) in the Oracle documentation.

When you work with Label Security, you perform all actions as the LBAC_DBA role. The master user for your DB instance is granted the LBAC_DBA role. You can grant the LBAC_DBA role to other users so that they can administer Label Security policies.

For Amazon RDS for Oracle Database 19c, Oracle Database 18c, and Oracle Database 12c Release 2 (12.2) DB instances, you must grant access to the OLS_ENFORCEMENT package to any new users who require access to Oracle Label Security. To grant access to the OLS_ENFORCEMENT package, connect to the DB instance as the master user and run the following SQL statement:

```
GRANT ALL ON LBACSYS.OLS_ENFORCEMENT TO username;
```

You can configure Label Security through the Oracle Enterprise Manager (OEM) Cloud Control. Amazon RDS supports the OEM Cloud Control through the Management Agent option. For more information, see [Oracle Management Agent for Enterprise Manager Cloud Control \(p. 1154\)](#).

Removing the Oracle Label Security option

You can remove Oracle Label Security from a DB instance.

To remove Label Security from a DB instance, do one of the following:

- To remove Label Security from multiple DB instances, remove the Label Security option from the option group they belong to. This change affects all DB instances that use the option group. When you remove Label Security from an option group that is attached to multiple DB instances, all the DB instances are restarted. For more information, see [Removing an option from an option group \(p. 224\)](#).
- To remove Label Security from a single DB instance, modify the DB instance and specify a different option group that doesn't include the Label Security option. You can specify the default (empty) option group, or a different custom option group. When you remove the Label Security option, a brief outage occurs while your DB instance is automatically restarted. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Troubleshooting

The following are issues you might encounter when you use Oracle Label Security.

Issue	Troubleshooting suggestions
When you try to create a policy, you see an error message similar to the following: insufficient authorization for the SYSDBA package.	A known issue with Oracle's Label Security feature prevents users with usernames of 16

Issue	Troubleshooting suggestions
	or 24 characters from running Label Security commands. You can create a new user with a different number of characters, grant LBAC_DBAs to the new user, log in as the new user, and run the OLS commands as the new user. For additional information, please contact Oracle support.

Oracle Locator

Amazon RDS supports Oracle Locator through the use of the `LOCATOR` option. Oracle Locator provides capabilities that are typically required to support internet and wireless service-based applications and partner-based GIS solutions. Oracle Locator is a limited subset of Oracle Spatial. For more information, see [Oracle Locator](#) in the Oracle documentation.

Important

If you use Oracle Locator, Amazon RDS automatically updates your DB instance to the latest Oracle PSU if there are security vulnerabilities with a Common Vulnerability Scoring System (CVSS) score of 9+ or other announced security vulnerabilities.

Amazon RDS supports Oracle Locator for the following versions of Oracle Database:

- Oracle Database 19c (19.0.0.0)
- Oracle Database 18c (18.0.0.0)
- Oracle Database 12c Release 2 (12.2.0.1)
- Oracle Database 12c Release 1 (12.1), version 12.1.0.2.v13 or later

Prerequisites for Oracle Locator

The following are prerequisites for using Oracle Locator:

- Your DB instance must be inside a virtual private cloud (VPC). For more information, see [Determining whether you are using the EC2-VPC or EC2-Classic platform \(p. 1718\)](#).
- Your DB instance must be of sufficient class. Oracle Locator is not supported for the db.t3.micro or db.t3.small DB instance classes. For more information, see [RDS for Oracle instance classes \(p. 992\)](#).
- Your DB instance must have **Auto Minor Version Upgrade** enabled. This option enables your DB instance to receive minor DB engine version upgrades automatically when they become available and is required for any options that install the Oracle Java Virtual Machine (JVM). Amazon RDS uses this option to update your DB instance to the latest Oracle Patch Set Update (PSU) or Release Update (RU). For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Best practices for Oracle Locator

The following are best practices for using Oracle Locator:

- For maximum security, use the `LOCATOR` option with Secure Sockets Layer (SSL). For more information, see [Oracle Secure Sockets Layer \(p. 1182\)](#).
- Configure your DB instance to restrict access to your DB instance. For more information, see [Scenarios for accessing a DB instance in a VPC \(p. 1720\)](#) and [Working with a DB instance in a VPC \(p. 1727\)](#).

Adding the Oracle Locator option

The following is the general process for adding the `LOCATOR` option to a DB instance:

1. Create a new option group, or copy or modify an existing option group.
2. Add the option to the option group.
3. Associate the option group with the DB instance.

If Oracle Java Virtual Machine (JVM) is *not* installed on the DB instance, there is a brief outage while the `LOCATOR` option is added. There is no outage if Oracle Java Virtual Machine (JVM) is already installed

on the DB instance. After you add the option, you don't need to restart your DB instance. As soon as the option group is active, Oracle Locator is available.

Note

During this outage, password verification functions are disabled briefly. You can also expect to see events related to password verification functions during the outage. Password verification functions are enabled again before the Oracle DB instance is available.

To add the LOCATOR option to a DB instance

1. Determine the option group that you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group with the following settings:
 - a. For **Engine**, choose the oracle edition for your DB instance.
 - b. For **Major engine version**, choose the version of your DB instance.

For more information, see [Creating an option group \(p. 214\)](#).
2. Add the **LOCATOR** option to the option group. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).
3. Apply the option group to a new or existing DB instance:
 - For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
 - For an existing DB instance, you apply the option group by modifying the instance and attaching the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Using Oracle Locator

After you enable the Oracle Locator option, you can begin using it. You should only use Oracle Locator features. Don't use any Oracle Spatial features unless you have a license for Oracle Spatial.

For a list of features that are supported for Oracle Locator, see [Features Included with Locator](#) in the Oracle documentation.

For a list of features that are not supported for Oracle Locator, see [Features Not Included with Locator](#) in the Oracle documentation.

Removing the Oracle Locator option

After you drop all objects that use data types provided by the **LOCATOR** option, you can remove the option from a DB instance. If Oracle Java Virtual Machine (JVM) is *not* installed on the DB instance, there is a brief outage while the **LOCATOR** option is removed. There is no outage if Oracle Java Virtual Machine (JVM) is already installed on the DB instance. After you remove the **LOCATOR** option, you don't need to restart your DB instance.

To drop the LOCATOR option

1. Back up your data.

Warning

If the instance uses data types that were enabled as part of the option, and if you remove the **LOCATOR** option, you can lose data. For more information, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#).

2. Check whether any existing objects reference data types or features of the **LOCATOR** option.

If LOCATOR options exist, the instance can get stuck when applying the new option group that doesn't have the LOCATOR option. You can identify the objects by using the following queries:

```
SELECT OWNER, SEGMENT_NAME, TABLESPACE_NAME, BYTES/1024/1024 mbytes
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE '%TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
       (SELECT DISTINCT OWNER, TABLE_NAME
        FROM   DBA_TAB_COLUMNS
        WHERE  DATA_TYPE='SDO_GEOGRAPHY'
        AND    OWNER <> 'MDSYS')
ORDER BY 1,2,3,4;

SELECT OWNER, TABLE_NAME, COLUMN_NAME
FROM   DBA_TAB_COLUMNS
WHERE  DATA_TYPE = 'SDO_GEOGRAPHY'
AND    OWNER <> 'MDSYS'
ORDER BY 1,2,3;
```

3. Drop any objects that reference data types or features of the LOCATOR option.
4. Do one of the following:
 - Remove the LOCATOR option from the option group it belongs to. This change affects all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).
 - Modify the DB instance and specify a different option group that doesn't include the LOCATOR option. This change affects a single DB instance. You can specify the default (empty) option group, or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Oracle Multimedia

Amazon RDS supports Oracle Multimedia through the use of the `MULTIMEDIA` option. You can use Oracle Multimedia to store, manage, and retrieve images, audio, video, and other heterogeneous media data. For more information, see [Oracle Multimedia](#) in the Oracle documentation.

Important

If you use Oracle Multimedia, Amazon RDS automatically updates your DB instance to the latest Oracle PSU if there are security vulnerabilities with a Common Vulnerability Scoring System (CVSS) score of 9+ or other announced security vulnerabilities.

Amazon RDS supports Oracle Multimedia for the following editions and versions of Oracle:

- Standard Edition (SE2) or Enterprise Edition of Oracle Database 18c (18.0.0.0), all versions
- Standard Edition (SE2) or Enterprise Edition of Oracle Database 12c Release 2 (12.2), all versions
- Standard Edition (SE2) or Enterprise Edition of Oracle Database 12c Release 1 (12.1), version 12.1.0.2.v13 or later

Note

Oracle desupported Oracle Multimedia in Oracle Database 19c. So, Oracle Multimedia isn't supported for Oracle Database 19c DB instances. For more information, see [Desupport of Oracle Multimedia](#) in the Oracle documentation.

Prerequisites for Oracle Multimedia

The following are prerequisites for using Oracle Multimedia:

- Your DB instance must be inside a virtual private cloud (VPC). For more information, see [Determining whether you are using the EC2-VPC or EC2-Classic platform \(p. 1718\)](#).
- Your DB instance must be of sufficient class. Oracle Multimedia is not supported for the db.t3.micro or db.t3.small DB instance classes. For more information, see [RDS for Oracle instance classes \(p. 992\)](#).
- Your DB instance must have **Auto Minor Version Upgrade** enabled. This option enables your DB instance to receive minor DB engine version upgrades automatically when they become available and is required for any options that install the Oracle Java Virtual Machine (JVM). Amazon RDS uses this option to update your DB instance to the latest Oracle Patch Set Update (PSU) or Release Update (RU). For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Best practices for Oracle Multimedia

The following are best practices for using Oracle Multimedia:

- For maximum security, use the `MULTIMEDIA` option with Secure Sockets Layer (SSL). For more information, see [Oracle Secure Sockets Layer \(p. 1182\)](#).
- Configure your DB instance to restrict access to your DB instance. For more information, see [Scenarios for accessing a DB instance in a VPC \(p. 1720\)](#) and [Working with a DB instance in a VPC \(p. 1727\)](#).

Adding the Oracle Multimedia option

The following is the general process for adding the `MULTIMEDIA` option to a DB instance:

1. Create a new option group, or copy or modify an existing option group.
2. Add the option to the option group.

3. Associate the option group with the DB instance.

If Oracle Java Virtual Machine (JVM) is *not* installed on the DB instance, there is a brief outage while the **MULTIMEDIA** option is added. There is no outage if Oracle Java Virtual Machine (JVM) is already installed on the DB instance. After you add the option, you don't need to restart your DB instance. As soon as the option group is active, Oracle Multimedia is available.

Note

During this outage, password verification functions are disabled briefly. You can also expect to see events related to password verification functions during the outage. Password verification functions are enabled again before the Oracle DB instance is available.

To add the **MULTIMEDIA** option to a DB instance

1. Determine the option group that you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group with the following settings:

- a. For **Engine**, choose **oracle-ee**.
- b. For **Major engine version**, choose the version of your DB instance.

For more information, see [Creating an option group \(p. 214\)](#).

2. Add the **MULTIMEDIA** option to the option group. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).
3. Apply the option group to a new or existing DB instance:
 - For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
 - For an existing DB instance, you apply the option group by modifying the instance and attaching the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Removing the Oracle Multimedia option

After you drop all objects that use data types provided by the **MULTIMEDIA** option, you can remove the option from a DB instance. If Oracle Java Virtual Machine (JVM) is *not* installed on the DB instance, there is a brief outage while the **MULTIMEDIA** option is removed. There is no outage if Oracle Java Virtual Machine (JVM) is already installed on the DB instance. After you remove the **MULTIMEDIA** option, you don't need to restart your DB instance.

To drop the **MULTIMEDIA** option

1. Back up your data.

Warning

If the instance uses data types that were enabled as part of the option, and if you remove the **MULTIMEDIA** option, you can lose data. For more information, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#).

2. Check whether any existing objects reference data types or features of the **MULTIMEDIA** option.
3. Drop any objects that reference data types or features of the **MULTIMEDIA** option.
4. Do one of the following:
 - Remove the **MULTIMEDIA** option from the option group it belongs to. This change affects all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).

- Modify the DB instance and specify a different option group that doesn't include the **MULTIMEDIA** option. This change affects a single DB instance. You can specify the default (empty) option group, or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Oracle native network encryption

Amazon RDS supports Oracle native network encryption (NNE). With native network encryption, you can encrypt data as it moves to and from a DB instance. Amazon RDS supports NNE for all editions of Oracle.

A detailed discussion of Oracle native network encryption is beyond the scope of this guide, but you should understand the strengths and weaknesses of each algorithm and key before you decide on a solution for your deployment. For information about the algorithms and keys that are available through Oracle native network encryption, see [Configuring network data encryption](#) in the Oracle documentation. For more information about AWS security, see the [AWS security center](#).

Note

You can use Native Network Encryption or Secure Sockets Layer, but not both. For more information, see [Oracle Secure Sockets Layer \(p. 1182\)](#).

NNE option settings

Amazon RDS supports the following settings for the NNE option.

Note

When you use commas to separate values for an option setting, don't put a space after the comma.

Option setting	Valid values	Default value	Description
SQLNET.ENCRYPTION_SERVER	Rejected, Requested, Required	Requested	The encryption behavior when a client, or a server acting as a client, connects to the DB instance. Requested indicates that the DB instance does not require traffic from the client to be encrypted.
SQLNET.CRYPTO_CHECKSUM_SERVER	Rejected, Requested, Required	Requested	The data integrity behavior when a client, or a server acting as a client, connects to the DB instance. Requested indicates that the DB instance does not require the client to perform a checksum.
SQLNET.ENCRYPTION_TYPES_SERVER	AES256, AES192, AES128, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	AES256, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	A list of encryption algorithms used by the DB instance. The DB instance uses each algorithm, in order, to attempt to decrypt the client input until an algorithm succeeds or until the end of the list is reached. Amazon RDS uses the following default list from Oracle. You can change the order or limit the algorithms that the DB instance will accept. 1. RC4_256: RSA RC4 (256-bit key size)

Option setting	Valid values	Default value	Description
			<p>2. AES256: AES (256-bit key size) 3. AES192: AES (192-bit key size) 4. 3DES168: 3-key Triple-DES (112-bit effective key size) 5. RC4_128: RSA RC4 (128-bit key size) 6. AES128: AES (128-bit key size) 7. 3DES112: 2-key Triple-DES (80-bit effective key size) 8. RC4_56: RSA RC4 (56-bit key size) 9. DES: Standard DES (56-bit key size) 10RC4_40: RSA RC4 (40-bit key size) 11DES40: DES40 (40-bit key size)</p> <p>You can specify either one value or a comma-separated list of values. If you use a comma, don't insert a space after the comma; otherwise, you receive an <code>InvalidParameterValue</code> error.</p>
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	SHA512, SHA1, SHA384, SHA512, SHA1, MD5	MD5	<p>A list of checksum algorithms.</p> <p>You can specify either one value or a comma-separated list of values. If you use a comma, don't insert a space after the comma; otherwise, you receive an <code>InvalidParameterValue</code> error.</p>

Adding the NNE option

The general process for adding the NNE option to a DB instance is the following:

1. Create a new option group, or copy or modify an existing option group.
2. Add the option to the option group.
3. Associate the option group with the DB instance.

After you add the NNE option, as soon as the option group is active, NNE is active.

To add the NNE option to a DB instance

1. For **Engine**, choose the Oracle edition that you want to use. NNE is supported on all editions.
 2. For **Major engine version**, choose the version of your DB instance.
- For more information, see [Creating an option group \(p. 214\)](#).
3. Add the **NNE** option to the option group. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).

Note

After you add the NNE option, you don't need to restart your DB instances. As soon as the option group is active, NNE is active.

4. Apply the option group to a new or existing DB instance:

- For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- For an existing DB instance, you apply the option group by modifying the instance and attaching the new option group. After you add the NNE option, you don't need to restart your DB instance. As soon as the option group is active, NNE is active. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Using NNE

With Oracle native network encryption, you can also specify network encryption on the client side. On the client (the computer used to connect to the DB instance), you can use the sqlnet.ora file to specify the following client settings: SQLNET.CRYPTO_CHECKSUM_CLIENT, SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT, SQLNET.ENCRYPTION_CLIENT, and SQLNET.ENCRYPTION_TYPES_CLIENT. For information, see [Configuring network data encryption and integrity for Oracle servers and clients](#) in the Oracle documentation.

Sometimes, the DB instance will reject a connection request from an application, for example, if there is a mismatch between the encryption algorithms on the client and on the server.

To test Oracle native network encryption , add the following lines to the sqlnet.ora file on the client:

```
DIAG_ADR_ENABLED=off
TRACE_DIRECTORY_CLIENT=/tmp
TRACE_FILE_CLIENT=nettrace
TRACE_LEVEL_CLIENT=16
```

These lines generate a trace file on the client called /tmp/nettrace* when the connection is attempted. The trace file contains information on the connection. For more information about connection-related issues when you are using Oracle Native Network Encryption, see [About negotiating encryption and integrity](#) in the Oracle documentation.

Modifying NNE settings

After you enable NNE, you can modify settings for the option. For more information about how to modify option settings, see [Modifying an option setting \(p. 221\)](#). For more information about each setting, see [NNE option settings \(p. 1176\)](#).

Removing the NNE option

You can remove NNE from a DB instance.

To remove NNE from a DB instance, do one of the following:

- To remove NNE from multiple DB instances, remove the NNE option from the option group they belong to. This change affects all DB instances that use the option group. After you remove the NNE option, you don't need to restart your DB instances. For more information, see [Removing an option from an option group \(p. 224\)](#).
- To remove NNE from a single DB instance, modify the DB instance and specify a different option group that doesn't include the NNE option. You can specify the default (empty) option group, or a different

custom option group. After you remove the NNE option, you don't need to restart your DB instance. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Oracle OLAP

Amazon RDS supports Oracle OLAP through the use of the `OLAP` option. This option provides On-line Analytical Processing (OLAP) for Oracle DB instances. You can use Oracle OLAP to analyze large amounts of data by creating dimensional objects and cubes in accordance with the OLAP standard. For more information, see [the Oracle documentation](#).

Important

If you use Oracle OLAP, Amazon RDS automatically updates your DB instance to the latest Oracle PSU if there are security vulnerabilities with a Common Vulnerability Scoring System (CVSS) score of 9+ or other announced security vulnerabilities.

Amazon RDS supports Oracle OLAP for the following editions and versions of Oracle:

- Oracle Enterprise Edition, version 19.0.0.0, all versions
- Oracle Enterprise Edition, version 18.0.0.0, all versions
- Oracle Enterprise Edition, version 12.2.0.1, all versions
- Oracle Enterprise Edition, version 12.1.0.2.v13 or later

Prerequisites for Oracle OLAP

The following are prerequisites for using Oracle OLAP:

- You must have an Oracle OLAP license from Oracle. For more information, see [Licensing Information](#) in the Oracle documentation.
- Your DB instance must be inside a virtual private cloud (VPC). For more information, see [Determining whether you are using the EC2-VPC or EC2-Classic platform \(p. 1718\)](#).
- Your DB instance must be of a sufficient instance class. Oracle OLAP isn't supported for the db.t3.micro or db.t3.small DB instance classes. For more information, see [RDS for Oracle instance classes \(p. 992\)](#).
- Your DB instance must have **Auto Minor Version Upgrade** enabled. This option enables your DB instance to receive minor DB engine version upgrades automatically when they become available and is required for any options that install the Oracle Java Virtual Machine (JVM). Amazon RDS uses this option to update your DB instance to the latest Oracle Patch Set Update (PSU) or Release Update (RU). For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
- Your DB instance must not have a user named `OLAPSYS`. If it does, the OLAP option installation fails.

Best practices for Oracle OLAP

The following are best practices for using Oracle OLAP:

- For maximum security, use the `OLAP` option with Secure Sockets Layer (SSL). For more information, see [Oracle Secure Sockets Layer \(p. 1182\)](#).
- Configure your DB instance to restrict access to your DB instance. For more information, see [Scenarios for accessing a DB instance in a VPC \(p. 1720\)](#) and [Working with a DB instance in a VPC \(p. 1727\)](#).

Adding the Oracle OLAP option

The following is the general process for adding the `OLAP` option to a DB instance:

1. Create a new option group, or copy or modify an existing option group.
2. Add the option to the option group.

3. Associate the option group with the DB instance.

If Oracle Java Virtual Machine (JVM) is *not* installed on the DB instance, there is a brief outage while the OLAP option is added. There is no outage if Oracle Java Virtual Machine (JVM) is already installed on the DB instance. After you add the option, you don't need to restart your DB instance. As soon as the option group is active, Oracle OLAP is available.

To add the OLAP option to a DB instance

1. Determine the option group that you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group with the following settings:
 - For **Engine**, choose the Oracle edition for your DB instance.
 - For **Major engine version**, choose the version of your DB instance.For more information, see [Creating an option group \(p. 214\)](#).
2. Add the **OLAP** option to the option group. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).
3. Apply the option group to a new or existing DB instance:
 - For a new DB instance, apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
 - For an existing DB instance, apply the option group by modifying the instance and attaching the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Using Oracle OLAP

After you enable the Oracle OLAP option, you can begin using it. For a list of features that are supported for Oracle OLAP, see [the Oracle documentation](#).

Removing the Oracle OLAP option

After you drop all objects that use data types provided by the OLAP option, you can remove the option from a DB instance. If Oracle Java Virtual Machine (JVM) is *not* installed on the DB instance, there is a brief outage while the OLAP option is removed. There is no outage if Oracle Java Virtual Machine (JVM) is already installed on the DB instance. After you remove the OLAP option, you don't need to restart your DB instance.

To drop the OLAP option

1. Back up your data.

Warning

If the instance uses data types that were enabled as part of the option, and if you remove the OLAP option, you can lose data. For more information, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#).

2. Check whether any existing objects reference data types or features of the OLAP option.
3. Drop any objects that reference data types or features of the OLAP option.
4. Do one of the following:
 - Remove the OLAP option from the option group it belongs to. This change affects all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).

- Modify the DB instance and specify a different option group that doesn't include the OLAP option. This change affects a single DB instance. You can specify the default (empty) option group, or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Oracle Secure Sockets Layer

You enable Secure Sockets Layer (SSL) encryption for an Oracle DB instance by adding the Oracle SSL option to the option group associated with an Oracle DB instance. You specify the port you want to communicate over using SSL. You must configure SQL*Plus as shown in this following section.

You enable SSL encryption for an Oracle DB instance by adding the Oracle SSL option to the option group associated with the DB instance. Amazon RDS uses a second port, as required by Oracle, for SSL connections. This approach allows both clear text and SSL-encrypted communication to occur at the same time between a DB instance and SQL*Plus. For example, you can use the port with clear text communication to communicate with other resources inside a VPC while using the port with SSL-encrypted communication to communicate with resources outside the VPC.

Note

You can use Secure Sockets Layer or Native Network Encryption, but not both. For more information, see [Oracle native network encryption \(p. 1176\)](#).

You can use SSL encryption with the following Oracle database versions and editions:

- 19.0.0.0: All versions, all editions including Standard Edition Two
- 18.0.0.0: All versions, all editions including Standard Edition Two
- 12.2.0.1: All versions, all editions including Standard Edition Two
- 12.1.0.2: All versions, all editions including Standard Edition Two

Note

You cannot use both SSL and Oracle native network encryption (NNE) on the same instance. If you use SSL encryption, you must disable any other connection encryption.

TLS versions for the Oracle SSL option

Amazon RDS for Oracle supports Transport Layer Security (TLS) versions 1.0 and 1.2. To use the Oracle SSL option, use the `SQLNET.SSL_VERSION` option setting. The following values are allowed for this option setting:

- "1.0" – Clients can connect to the DB instance using TLS 1.0 only.
- "1.2" – Clients can connect to the DB instance using TLS 1.2 only.
- "1.2 or 1.0" – Clients can connect to the DB instance using either TLS 1.2 or 1.0.

To use the Oracle SSL option, the `SQLNET.SSL_VERSION` option setting is also required:

- For existing Oracle SSL options, `SQLNET.SSL_VERSION` is set to "1.0" automatically. You can change the setting if necessary.
- When you add a new Oracle SSL option, you must set `SQLNET.SSL_VERSION` explicitly to a valid value.

The following table shows the TLS option settings that are supported for different Oracle engine versions and editions.

Oracle engine version	<code>SQLNET.SSL_VERSION = "1.0"</code>	<code>SQLNET.SSL_VERSION = "1.2"</code>	<code>SQLNET.SSL_VERSION = "1.2 or 1.0"</code>
19.0.0.0 (All editions)	Supported	Supported	Supported
18.0.0.0 (All editions)	Supported	Supported	Supported
12.2.0.1 (All editions)	Supported	Supported	Supported
12.1.0.2 (All editions)	Supported	Supported	Supported

Cipher suites for the Oracle SSL option

Amazon RDS for Oracle supports multiple SSL cipher suites. By default, the Oracle SSL option is configured to use the `SSL_RSA_WITH_AES_256_CBC_SHA` cipher suite. To specify a different cipher suite to use over SSL connections, use the `SQLNET.CIPHER_SUITE` option setting. Following are the allowed values for this option setting:

- "`SSL_RSA_WITH_AES_256_CBC_SHA`" – The default setting, which is compatible with TLS 1.0 and TLS 1.2
- "`SSL_RSA_WITH_AES_256_CBC_SHA256`" – Only compatible with TLS 1.2
- "`SSL_RSA_WITH_AES_256_GCM_SHA384`" – Only compatible with TLS 1.2

For existing Oracle SSL options, `SQLNET.CIPHER_SUITE` is set to "`SSL_RSA_WITH_AES_256_CBC_SHA`" automatically. You can change the setting if necessary.

The following table shows the cipher suite option settings that are supported for different Oracle engine versions and editions.

Oracle engine version	<code>SQLNET.CIPHER_SUITE = "SSL_RSA_WITH_AES_256_GCM_SHA256"</code>	<code>SQLNET.CIPHER_SUITE = "SSL_RSA_WITH_AES_256_CBC_SHA256"</code>	<code>SQLNET.CIPHER_SUITE = "SSL_RSA_WITH_AES_256_CBC_SHA"</code>
19.0.0.0 (All editions)	Supported	Supported	Supported
18.0.0.0 (All editions)	Supported	Supported	Supported
12.2.0.1 (All editions)	Supported	Supported	Supported
12.1.0.2 (All editions)	Supported	Supported	Supported

FIPS support

Amazon RDS for Oracle enables you to use the Federal Information Processing Standard (FIPS) standard for 140-2. FIPS 140-2 is a United States government standard that defines cryptographic module security requirements. You enable the FIPS standard by setting the setting `FIPS.SSLFIPS_140` to `TRUE` for the Oracle SSL option. When FIPS 140-2 is configured for SSL, the cryptographic libraries are designed to encrypt data between the client and the Oracle DB instance.

You can enable the FIPS setting with the following Oracle database versions and editions:

- 19.0.0.0: All versions, all editions including Standard Edition Two
- 18.0.0.0: All versions, all editions including Standard Edition Two
- 12.2.0.1: All versions, all editions including Standard Edition Two

- 12.1.0.2: Version 2 and later, all editions including Standard Edition Two

Clients must use the cipher suite that is FIPS-compliant. When establishing a connection, the client and Oracle DB instance negotiate which cipher suite to use when transmitting messages back and forth. The following table shows the FIPS-compliant SSL cipher suites for each TLS version.

SQLNET.SSL_VERSION	Supported cipher suites
1.0	SSL_RSA_WITH_AES_256_CBC_SHA
1.2	SSL_RSA_WITH_AES_256_CBC_SHA SSL_RSA_WITH_AES_256_GCM_SHA384

For more information, see [Oracle database FIPS 140-2 settings](#) in the Oracle documentation.

Adding the SSL option

To use SSL, your Amazon RDS for Oracle DB instance must be associated with an option group that includes the `SSL` option.

Console

To add the SSL option to an option group

1. Create a new option group or identify an existing option group to which you can add the `SSL` option.

For information about creating an option group, see [Creating an option group \(p. 214\)](#).

2. Add the `SSL` option to the option group.

If you want to use only FIPS-verified cipher suites for SSL connections, set the option `FIPS.SSLFIPS_140` to `TRUE`. For information about the FIPS standard, see [FIPS support \(p. 1183\)](#).

For information about adding an option to an option group, see [Adding an option to an option group \(p. 216\)](#).

3. Create a new Oracle DB instance and associate the option group with it, or modify an Oracle DB instance to associate the option group with it.

For information about creating a DB instance, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

For information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

AWS CLI

To add the SSL option to an option group

1. Create a new option group or identify an existing option group to which you can add the `SSL` option.

For information about creating an option group, see [Creating an option group \(p. 214\)](#).

2. Add the `SSL` option to the option group.

Specify the following option settings:

- `Port` – The SSL port number
- `VpcSecurityGroupMemberships` – The VPC security group for which the option is enabled
- `SQLNET.SSL_VERSION` – The TLS version that client can use to connect to the DB instance

For example, the following AWS CLI command adds the SSL option to an option group named ora-option-group.

Example

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group --option-group-name ora-option-group \
    --options
    'OptionName=SSL,Port=2484,VpcSecurityGroupMemberships="sg-68184619",OptionSettings=[{Name=SQLNET.SSL,Value=1}]]'
```

For Windows:

```
aws rds add-option-to-option-group --option-group-name ora-option-group ^
    --options
    'OptionName=SSL,Port=2484,VpcSecurityGroupMemberships="sg-68184619",OptionSettings=[{Name=SQLNET.SSL,Value=1}]]'
```

3. Create a new Oracle DB instance and associate the option group with it, or modify an Oracle DB instance to associate the option group with it.

For information about creating a DB instance, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

For information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Configuring SQL*Plus to use SSL with an Oracle DB instance

You must configure SQL*Plus before connecting to an Oracle DB instance that uses the Oracle SSL option.

Note

To allow access to the DB instance from the appropriate clients, ensure that your security groups are configured correctly. For more information, see [Controlling access with security groups \(p. 1699\)](#). Also, these instructions are for SQL*Plus and other clients that directly use an Oracle home. For JDBC connections, see [Setting up an SSL connection over JDBC \(p. 1187\)](#).

To configure SQL*Plus to use SSL to connect to an Oracle DB instance

1. Set the ORACLE_HOME environment variable to the location of your Oracle home directory.

The path to your Oracle home directory depends on your installation. The following example sets the ORACLE_HOME environment variable.

```
prompt>export ORACLE_HOME=/home/user/app/user/product/12.1.0/dbhome_1
```

For information about setting Oracle environment variables, see [SQL*Plus environment variables](#) in the Oracle documentation, and also see the Oracle installation guide for your operating system.

2. Append \$ORACLE_HOME/lib to the LD_LIBRARY_PATH environment variable.

The following is an example that sets the LD_LIBRARY_PATH environment variable.

```
prompt>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

3. Create a directory for the Oracle wallet at \$ORACLE_HOME/ssl_wallet.

The following is an example that creates the Oracle wallet directory.

```
prompt>mkdir $ORACLE_HOME/ssl_wallet
```

4. Download the root certificate that works for all AWS Regions and put the file in the ssl_wallet directory.

For information about downloading the root certificate, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

5. In the \$ORACLE_HOME/network/admin directory, modify or create the tnsnames.ora file and include the following entry.

```
<net_service_name>= (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCPS)
(HOST = <endpoint>) (PORT = <ssl port number>)))(CONNECT_DATA = (SID = <database
name>))
(SECURITY = (SSL_SERVER_CERT_DN =
"C=US,ST=Washington,L=Seattle,O=Amazon.com,OU=RDS,CN=<endpoint>")))
```

6. In the same directory, modify or create the sqlnet.ora file and include the following parameters.

Note

To communicate with entities over a TLS secured connection, Oracle requires a wallet with the necessary certificates for authentication. You can use Oracle's ORAPKI utility to create and maintain Oracle wallets, as shown in step 7. For more information, see [Setting up Oracle wallet using ORAPKI](#) in the Oracle documentation.

```
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY = $ORACLE_HOME/
ssl_wallet)))
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_VERSION = 1.0
SSL_CIPHER_SUITES = (SSL_RSA_WITH_AES_256_CBC_SHA)
SSL_SERVER_DN_MATCH = ON
```

Note

You can set SSL_VERSION to a higher value if your DB instance supports it.

7. Run the following commands to create the Oracle wallet.

```
prompt>orapki wallet create -wallet $ORACLE_HOME/ssl_wallet -auto_login_only
prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
$ORACLE_HOME/ssl_wallet/rds-ca-2019-root.pem -auto_login_only
```

Replace the file name with the one you downloaded.

Connecting to an Oracle DB instance using SSL

After you configure SQL*Plus to use SSL as described previously, you can connect to the Oracle DB instance with the SSL option. Optionally, you can first export the TNS_ADMIN value that points to the directory that contains the tnsnames.ora and sqlnet.ora files. Doing so ensures that SQL*Plus can find these files consistently. The following example exports the TNS_ADMIN value.

```
export TNS_ADMIN = ${ORACLE_HOME}/network/admin
```

Connect to the DB instance. For example, you can connect using SQL*Plus and a *<net_service_name>* in a tnsnames.ora file.

```
sqlplus <mydbuser>@<net_service_name>
```

You can also connect to the DB instance using SQL*Plus without using a tnsnames.ora file by using the following command.

```
sqlplus '<mydbuser>@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCPS)(HOST = <endpoint>) (PORT = <ssl port number>))(CONNECT_DATA = (SID = <database name>)))'
```

You can also connect to the Oracle DB instance without using SSL. For example, the following command connects to the DB instance through the clear text port without SSL encryption.

```
sqlplus '<mydbuser>@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = <endpoint>) (PORT = <port number>))(CONNECT_DATA = (SID = <database name>)))'
```

If you want to close Transmission Control Protocol (TCP) port access, create a security group with no IP address ingresses and add it to the instance. This addition closes connections over the TCP port, while still allowing connections over the SSL port that are specified from IP addresses within the range permitted by the SSL option security group.

Setting up an SSL connection over JDBC

To use an SSL connection over JDBC, you must create a keystore, trust the Amazon RDS root CA certificate, and use the code snippet specified following.

To create the keystore in JKS format, use the following command. For more information about creating the keystore, see the [Oracle documentation](#).

```
keytool -keystore clientkeystore -genkey -alias client
```

Next, take the following steps to trust the Amazon RDS root CA certificate.

To trust the Amazon RDS root CA certificate

1. Download the root certificate that works for all AWS Regions and put the file in the ssl_wallet directory.

For information about downloading the root certificate, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

2. Convert the certificate to .der format using the following command.

```
openssl x509 -outform der -in rds-ca-2019-root.pem -out rds-ca-2019-root.der
```

Replace the file name with the one you downloaded.

3. Import the certificate into the keystore using the following command.

```
keytool -import -alias rds-root -keystore clientkeystore.jks -file rds-ca-2019-root.der
```

4. Confirm that the key store was created successfully.

```
keytool -list -v -keystore clientkeystore.jks
```

Enter the keystore password when you are prompted for it.

The following code example shows how to set up the SSL connection using JDBC.

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class OracleSslConnectionTest {
    private static final String DB_SERVER_NAME = "<dns-name-provided-by-amazon-rds>";
    private static final Integer SSL_PORT = "<ssl-option-port-configured-in-option-group>";
    private static final String DB_SID = "<oracle-sid>";
    private static final String DB_USER = "<user name>";
    private static final String DB_PASSWORD = "<password>";
    // This key store has only the prod root ca.
    private static final String KEY_STORE_FILE_PATH = "<file-path-to-keystore>";
    private static final String KEY_STORE_PASS = "<keystore-password>";

    public static void main(String[] args) throws SQLException {
        final Properties properties = new Properties();
        final String connectionString = String.format(
            "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=%d))(CONNECT_DATA=(SID=%s)))",
            DB_SERVER_NAME, SSL_PORT, DB_SID);
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);
        properties.put("oracle.jdbc.J2EE13Compliant", "true");
        properties.put("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        properties.put("javax.net.ssl.trustStoreType", "JKS");
        properties.put("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);
        final Connection connection = DriverManager.getConnection(connectionString,
        properties);
        // If no exception, that means handshake has passed, and an SSL connection can be
        opened
    }
}

```

Enforcing a DN match with an SSL connection

You can use the Oracle parameter `SSL_SERVER_DN_MATCH` to enforce that the distinguished name (DN) for the database server matches its service name. If you enforce the match verifications, then SSL ensures that the certificate is from the server. If you don't enforce the match verification, then SSL performs the check but allows the connection, regardless if there is a match. If you do not enforce the match, you allow the server to potentially fake its identify.

To enforce DN matching, add the DN match property and use the connection string specified below.

Add the property to the client connection to enforce DN matching.

```
properties.put("oracle.net.ssl_server_dn_match", "TRUE");
```

Use the following connection string to enforce DN matching when using SSL.

```

final String connectionString = String.format(
    "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=%d))" +
    "(CONNECT_DATA=(SID=%s))" +
    "(SECURITY = (SSL_SERVER_CERT_DN =
    \"C=US,ST=Washington,L=Seattle,O=Amazon.com,OU=RDS,CN=%s\")))",

```

```
DB_SERVER_NAME, SSL_PORT, DB_SID, DB_SERVER_NAME);
```

Oracle Spatial

Amazon RDS supports Oracle Spatial through the use of the `SPATIAL` option. Oracle Spatial provides a SQL schema and functions that facilitate the storage, retrieval, update, and query of collections of spatial data in an Oracle database. For more information, see [Spatial Concepts](#) in the Oracle documentation.

Important

If you use Oracle Spatial, Amazon RDS automatically updates your DB instance to the latest Oracle PSU when any of the following exist:

- Security vulnerabilities with a Common Vulnerability Scoring System (CVSS) score of 9+
- Other announced security vulnerabilities

Amazon RDS supports Oracle Spatial only in Oracle Enterprise Edition (EE) and Oracle Standard Edition 2 (SE2). The following table shows the versions of the DB engine that support EE and SE2.

Oracle DB Version	EE	SE2
19.0.0.0, all versions	Yes	Yes
18.0.0.0, all versions	Yes	Yes
12.2.0.1, all versions	Yes	Yes
12.1.0.2.v13 or later	Yes	No

Prerequisites for Oracle Spatial

The following are prerequisites for using Oracle Spatial:

- Make sure that your DB instance is inside a virtual private cloud (VPC). For more information, see [Determining whether you are using the EC2-VPC or EC2-Classic platform \(p. 1718\)](#).
- Make sure that your DB instance is of a sufficient instance class. Oracle Spatial isn't supported for the db.t3.micro or db.t3.small DB instance classes. For more information, see [RDS for Oracle instance classes \(p. 992\)](#).
- Make sure that your DB instance has **Auto Minor Version Upgrade** enabled. This option enables your DB instance to receive minor DB engine version upgrades automatically when they become available and is required for any options that install the Oracle Java Virtual Machine (JVM). Amazon RDS uses this option to update your DB instance to the latest Oracle Patch Set Update (PSU) or Release Update (RU). For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Best practices for Oracle Spatial

The following are best practices for using Oracle Spatial:

- For maximum security, use the `SPATIAL` option with Secure Sockets Layer (SSL). For more information, see [Oracle Secure Sockets Layer \(p. 1182\)](#).
- Configure your DB instance to restrict access to your DB instance. For more information, see [Scenarios for accessing a DB instance in a VPC \(p. 1720\)](#) and [Working with a DB instance in a VPC \(p. 1727\)](#).

Adding the Oracle Spatial option

The following is the general process for adding the **SPATIAL** option to a DB instance:

1. Create a new option group, or copy or modify an existing option group.
2. Add the option to the option group.
3. Associate the option group with the DB instance.

If Oracle Java Virtual Machine (JVM) is *not* installed on the DB instance, there is a brief outage while the **SPATIAL** option is added. There is no outage if Oracle Java Virtual Machine (JVM) is already installed on the DB instance. After you add the option, you don't need to restart your DB instance. As soon as the option group is active, Oracle Spatial is available.

Note

During this outage, password verification functions are disabled briefly. You can also expect to see events related to password verification functions during the outage. Password verification functions are enabled again before the Oracle DB instance is available.

To add the **SPATIAL** option to a DB instance

1. Determine the option group that you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group with the following settings:
 - a. For **Engine**, choose the Oracle edition for your DB instance.
 - b. For **Major engine version**, choose the version of your DB instance.

For more information, see [Creating an option group \(p. 214\)](#).
2. Add the **SPATIAL** option to the option group. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).
3. Apply the option group to a new or existing DB instance:
 - For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
 - For an existing DB instance, you apply the option group by modifying the instance and attaching the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Removing the Oracle Spatial option

After you drop all objects that use data types provided by the **SPATIAL** option, you can drop the option from a DB instance. If Oracle Java Virtual Machine (JVM) is *not* installed on the DB instance, there is a brief outage while the **SPATIAL** option is removed. There is no outage if Oracle Java Virtual Machine (JVM) is already installed on the DB instance. After you remove the **SPATIAL** option, you don't need to restart your DB instance.

To drop the **SPATIAL** option

1. Back up your data.

Warning

If the instance uses data types that were enabled as part of the option, and if you remove the **SPATIAL** option, you can lose data. For more information, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#).

2. Check whether any existing objects reference data types or features of the **SPATIAL** option.

If SPATIAL options exist, the instance can get stuck when applying the new option group that doesn't have the SPATIAL option. You can identify the objects by using the following queries:

```
SELECT OWNER, SEGMENT_NAME, TABLESPACE_NAME, BYTES/1024/1024 mbytes
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE '%TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
       (SELECT DISTINCT OWNER, TABLE_NAME
        FROM   DBA_TAB_COLUMNS
        WHERE  DATA_TYPE='SDO_Geometry'
        AND    OWNER <> 'MDSYS')
ORDER BY 1,2,3,4;

SELECT OWNER, TABLE_NAME, COLUMN_NAME
FROM   DBA_TAB_COLUMNS
WHERE  DATA_TYPE = 'SDO_Geometry'
AND    OWNER <> 'MDSYS'
ORDER BY 1,2,3;
```

3. Drop any objects that reference data types or features of the SPATIAL option.
4. Do one of the following:
 - Remove the SPATIAL option from the option group it belongs to. This change affects all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).
 - Modify the DB instance and specify a different option group that doesn't include the SPATIAL option. This change affects a single DB instance. You can specify the default (empty) option group, or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Oracle SQLT

Amazon RDS supports Oracle SQLTXPLAIN (SQLT) through the use of the SQLT option.

The Oracle EXPLAIN PLAN statement can determine the execution plan of a SQL statement. It can verify whether the Oracle optimizer chooses a certain execution plan, such as a nested loops join. It also helps you understand the optimizer's decisions, such as why it chose a nested loops join over a hash join. So EXPLAIN PLAN helps you understand the statement's performance.

SQLT is an Oracle utility that produces a report. The report includes object statistics, object metadata, optimizer-related initialization parameters, and other information that a database administrator can use to tune a SQL statement for optimal performance. SQLT produces an HTML report with hyperlinks to all of the sections in the report.

Unlike Automatic Workload Repository or Statspack reports, SQLT works on individual SQL statements. SQLT is a collection of SQL, PL/SQL, and SQL*Plus files that collect, store, and display performance data.

Following are the supported Oracle versions for each SQLT version.

SQLT version	Oracle Database 19c	Oracle Database 18c	Oracle Database 12c Release 2 (12.2)	Oracle Database 12c Release 1 (12.1)
12.2.180725	Supported	Supported	Supported	Supported
12.2.180331	Not supported	Supported	Supported	Supported
12.1.160429	Not supported	Not supported	Supported	Supported

To download SQLT and access instructions for using it:

- Log in to your My Oracle Support account, and open the following documents:
 - To download SQLT: [Document 215187.1](#)
 - For SQLT usage instructions: [Document 1614107.1](#)
 - For frequently asked questions about SQLT: [Document 1454160.1](#)
 - For information about reading SQLT output: [Document 1456176.1](#)
 - For interpreting the Main report: [Document 1922234.1](#)

You can use SQLT with any edition of the following Oracle Database versions:

- Oracle Database 19c (19.0.0.0)
- Oracle Database 18c (18.0.0.0)
- Oracle Database 12c Release 2 (12.2.0.1)
- Oracle Database 12c Release 1 (12.1.0.2)

Amazon RDS does not support the following SQLT methods:

- XPORE
- XHUME

Prerequisites for SQLT

The following are prerequisites for using SQLT:

- You must remove users and roles that are required by SQLT, if they exist.

The SQLT option creates the following users and roles on a DB instance:

- `SQLTXPLAIN` user
- `SQLTXADMIN` user
- `SQLT_USER_ROLE` role

If your DB instance has any of these users or roles, log in to the DB instance using a SQL client, and drop them using the following statements:

```
DROP USER SQLTXPLAIN CASCADE;
DROP USER SQLTXADMIN CASCADE;
DROP ROLE SQLT_USER_ROLE CASCADE;
```

- You must remove tablespaces that are required by SQLT, if they exist.

The SQLT option creates the following tablespaces on a DB instance:

- `RDS_SQLT_TS`
- `RDS_TEMP_SQLT_TS`

If your DB instance has these tablespaces, log in to the DB instance using a SQL client, and drop them.

SQLT option settings

SQLT can work with licensed features that are provided by the Oracle Tuning Pack and the Oracle Diagnostics Pack. The Oracle Tuning Pack includes the SQL Tuning Advisor, and the Oracle Diagnostics Pack includes the Automatic Workload Repository. The SQLT settings enable or disable access to these features from SQLT.

Amazon RDS supports the following settings for the SQLT option.

Option setting	Valid values	Default value	Description
LICENSE_PACK	T, D, N	N	<p>The Oracle Management Packs that you want to access with SQLT. Enter one of the following values:</p> <ul style="list-style-type: none"> • T indicates that you have a license for the Oracle Tuning Pack and the Oracle Diagnostics Pack, and you want to access the SQL Tuning Advisor and Automatic Workload Repository from SQLT. • D indicates that you have a license for the Oracle Diagnostics Pack, and you want to access the Automatic Workload Repository from SQLT. • N indicates that you don't have a license for the Oracle Tuning Pack and the Oracle Diagnostics Pack, or that you have a license for one or both of them, but you don't want SQLT to access them. <p>Note Amazon RDS does not provide licenses for these Oracle Management Packs. If you indicate that</p>

Option setting	Valid values	Default value	Description
			you want to use a pack that is not included in your DB instance, you can use SQLT with the DB instance. However, SQLT can't access the pack, and the SQLT report doesn't include the data for the pack. For example, if you specify <code>T</code> , but the DB instance doesn't include the Oracle Tuning Pack, SQLT works on the DB instance, but the report it generates doesn't contain data related to the Oracle Tuning Pack.
VERSION	2016-04-29, 2018-03-31.v1, 2018-07-25.v1	2016-04-29.v1	The version of SQLT that you want to install. Note For Oracle Database 19c (19.0.0.0), the only supported version is 2018-07-25.v1. This version is also the default for Oracle Database 19c.

Adding the SQLT option

The following is the general process for adding the SQLT option to a DB instance:

1. Create a new option group, or copy or modify an existing option group.
2. Add the SQLT option to the option group.
3. Associate the option group with the DB instance.

After you add the SQLT option, as soon as the option group is active, SQLT is active.

To add the SQLT option to a DB instance

1. Determine the option group that you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group with the following settings:
 - a. For **Engine**, choose the Oracle edition that you want to use. The SQLT option is supported on all editions.
 - b. For **Major engine version**, choose the version of your DB instance.

For more information, see [Creating an option group \(p. 214\)](#).

2. Add the **SQLT** option to the option group. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).
3. Apply the option group to a new or existing DB instance:
 - For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
 - For an existing DB instance, you apply the option group by modifying the instance and attaching the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
4. (Optional) Verify the SQLT installation on each DB instance with the SQLT option.
 - a. Use a SQL client to connect to the DB instance as the master user.

For information about connecting to an Oracle DB instance using a SQL client, see [Connecting to your Oracle DB instance \(p. 1001\)](#).

- b. Run the following query:

```
SELECT sqlxplain.sqlt$a.get_param('tool_version') sqlt_version FROM DUAL;
```

The query returns the current version of the SQLT option on Amazon RDS. 12.1.160429 is an example of a version of SQLT that is available on Amazon RDS.

5. Change the passwords of the users that are created by the SQLT option.

- a. Use a SQL client to connect to the DB instance as the master user.
- b. Run the following SQL statement to change the password for the SQLTXADMIN user:

```
ALTER USER SQLTXADMIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

- c. Run the following SQL statement to change the password for the SQLTXPLAIN user:

```
ALTER USER SQLTXPLAIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

Note

Upgrading SQLT requires uninstalling an older version of SQLT and then installing the new version. So, all SQLT metadata can be lost when you upgrade SQLT. A major version upgrade of a database also uninstalls and re-installs SQLT. An example of a major version upgrade is an upgrade from Oracle Database 18c to Oracle Database 19c.

Using SQLT

SQLT works with the Oracle SQL*Plus utility.

To use SQLT

1. Download the SQLT .zip file from [Document 215187.1](#) on the My Oracle Support site.

Note

You can't download SQLT 12.1.160429 from the My Oracle Support site. Oracle has deprecated this older version.

2. Unzip the SQLT .zip file.
3. From a command prompt, change to the `sqlt/run` directory on your file system.
4. From the command prompt, open SQL*Plus, and connect to the DB instance as the master user.

For information about connecting to a DB instance using SQL*Plus, see [Connecting to your Oracle DB instance \(p. 1001\)](#).
5. Get the SQL ID of a SQL statement:

```
SELECT SQL_ID FROM V$SQL WHERE SQL_TEXT='sql_statement';
```

Your output is similar to the following:

```
SQL_ID
-----
chvsmttqjzjkn
```

6. Analyze a SQL statement with SQLT:

```
START sqltxtract.sql sql_id sqltxplain_user_password
```

For example, for the SQL ID *chvsmttqjzjkn*, enter the following:

```
START sqltxtract.sql chvsmttqjzjkn sqltxplain_user_password
```

SQLT generates the HTML report and related resources as a .zip file in the directory from which the SQLT command was run.

7. (Optional) To enable application users to diagnose SQL statements with SQLT, grant *SQLT_USER_ROLE* to each application user with the following statement:

```
GRANT SQLT_USER_ROLE TO application_user_name;
```

Note

Oracle does not recommend running SQLT with the *SYS* user or with users that have the *DBA* role. It is a best practice to run SQLT diagnostics using the application user's account, by granting *SQLT_USER_ROLE* to the application user.

Upgrading the SQLT option

With Amazon RDS for Oracle, you can upgrade the SQLT option from your existing version to a higher version. To upgrade the SQLT option, complete steps 1–3 in [Using SQLT \(p. 1196\)](#) for the new version of SQLT. Also, if you granted privileges for the previous version of SQLT in step 7 of that section, grant the privileges again for the new SQLT version.

Upgrading the SQLT option results in the loss of the older SQLT version's metadata. The older SQLT version's schema and related objects are dropped, and the newer version of SQLT is installed. For more information about the changes in the latest SQLT version, see [Document 1614201.1](#) on the My Oracle Support site.

Note

Version downgrades are not supported.

Modifying SQLT settings

After you enable SQLT, you can modify the *LICENSE_PACK* and *VERSION* settings for the option.

For more information about how to modify option settings, see [Modifying an option setting \(p. 221\)](#). For more information about each setting, see [SQL option settings \(p. 1194\)](#).

Removing the SQL option

You can remove SQL from a DB instance.

To remove SQL from a DB instance, do one of the following:

- To remove SQL from multiple DB instances, remove the SQL option from the option group to which the DB instances belong. This change affects all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).
- To remove SQL from a single DB instance, modify the DB instance and specify a different option group that doesn't include the SQL option. You can specify the default (empty) option group or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Oracle Statspack

The Oracle Statspack option installs and enables the Oracle Statspack performance statistics feature. Oracle Statspack is a collection of SQL, PL/SQL, and SQL*Plus scripts that collect, store, and display performance data. For information about using Oracle Statspack, see [Oracle Statspack](#) in the Oracle documentation.

Note

Oracle Statspack is no longer supported by Oracle and has been replaced by the more advanced Automatic Workload Repository (AWR). AWR is available only for Oracle Enterprise Edition customers who have purchased the Diagnostics Pack. You can use Oracle Statspack with any Oracle DB engine on Amazon RDS. You can't run Oracle Statspack on Amazon RDS read replicas.

Setting up Oracle Statspack

To run Statspack scripts, you must add the Statspack option.

To set up Oracle Statspack

1. In a SQL client, log in to the Oracle DB with an administrative account.
2. Do either of the following actions, depending on whether Statspack is installed:
 - If Statspack is installed, and the `PERFSTAT` account is associated with Statspack, skip to Step 4.
 - If Statspack is not installed, and the `PERFSTAT` account exists, drop the account as follows:

```
DROP USER PERFSTAT CASCADE;
```

Otherwise, attempting to add the Statspack option generates an error and `RDS-Event-0058`.

3. Add the Statspack option to an option group. See [Adding an option to an option group \(p. 216\)](#).
4. Amazon RDS automatically installs the Statspack scripts on the DB instance and then sets up the `PERFSTAT` account.

4. Reset the password using the following SQL statement, replacing `pwd` with your new password:

```
ALTER USER PERFSTAT IDENTIFIED BY pwd ACCOUNT UNLOCK;
```

You can log in using the `PERFSTAT` user account and run the Statspack scripts.

5. Do either of the following actions, depending on your DB engine version:
 - If you are using Oracle Database 18c or lower, skip this step.
 - If you are using Oracle Database 19c or higher, grant the `CREATE JOB` privilege to the `PERFSTAT` account using the following statement:

```
GRANT CREATE JOB TO PERFSTAT;
```

6. Ensure that idle wait events in the `PERFSTAT.STATS$IDLE_EVENT` table are populated.

Because of Oracle Bug 28523746, the idle wait events in `PERFSTAT.STATS$IDLE_EVENT` may not be populated. To ensure all idle events are available, run the following statement:

```
INSERT INTO PERFSTAT.STATS$IDLE_EVENT (EVENT)
SELECT NAME FROM V$EVENT_NAME WHERE WAIT_CLASS='Idle'
MINUS
SELECT EVENT FROM PERFSTAT.STATS$IDLE_EVENT;
COMMIT;
```

Generating Statspack reports

A Statspack report compares two snapshots.

To generate Statspack reports

1. In a SQL client, log in to the Oracle DB with the `PERFSTAT` account.
2. Create a snapshot using either of the following techniques:
 - Create a Statspack snapshot manually.
 - Create a job that takes a Statspack snapshot after a given time interval. For example, the following job creates a Statspack snapshot every hour:

```
VARIABLE jn NUMBER;
exec dbms_job.submit(:jn, 'statspack.snap;',SYSDATE,'TRUNC(SYSDATE+1/24,''HH24'')");
COMMIT;
```

3. View the snapshots using the following query:

```
SELECT SNAP_ID, SNAP_TIME FROM STATS$SNAPSHOT ORDER BY 1;
```

4. Run the Amazon RDS procedure `rdsadmin.rds_run_spreport`, replacing `begin_snap` and `end_snap` with the snapshot IDs:

```
exec rdsadmin.rds_run_spreport(begin_snap,end_snap);
```

For example, the following command creates a report based on the interval between Statspack snapshots 1 and 2:

```
exec rdsadmin.rds_run_spreport(1,2);
```

The file name of the Statspack report includes the number of the two snapshots. For example, a report file created using Statspack snapshots 1 and 2 would be named `ORCL_spreport_1_2.lst`.

5. Monitor the output for errors.

Oracle Statspack performs checks before running the report. Therefore, you could also see error messages in the command output. For example, you might try to generate a report based on an invalid range, where the beginning Statspack snapshot value is larger than the ending value. In this case, the output shows the error message, but the DB engine does not generate an error file.

```
exec rdsadmin.rds_run_spreport(2,1);
*
ERROR at line 1:
ORA-20000: Invalid snapshot IDs. Find valid ones in perfstat.stats$snapshot.
```

If you use an invalid number a Statspack snapshot, the output shows an error. For example, if you try to generate a report for snapshots 1 and 50, but snapshot 50 doesn't exist, the output shows an error.

```
exec rdsadmin.rds_run_spreport(1,50);
*
ERROR at line 1:
ORA-20000: Could not find both snapshot IDs
```

6. (Optional)

To retrieve the report, call the trace file procedures, as explained in [Working with Oracle trace files \(p. 527\)](#).

Alternatively, download the Statspack report from the RDS console. Go to the **Log** section of the DB instance details and choose **Download**:

Name	Last written	Size
trace/ORCL_mmon_11800.trc	Thu Jan 18 09:39:14 GMT-800 2018	68.2 kB
trace/ORCL_mmon_11800.trm	Thu Jan 18 09:39:14 GMT-800 2018	6.7 kB
trace/ORCL_spreport_1_2.trc	Thu Jan 18 09:38:03 GMT-800 2018	107.5 kB
trace/alert_ORCL.log	Thu Jan 18 09:37:39 GMT-800 2018	60.5 kB
audit/ORCL_ora_26710_2018011817315766624143795.aud	Thu Jan 18 09:31:57 GMT-800 2018	5.5 kB

If an error occurs while generating a report, the DB engine uses the same naming conventions as for a report but with an extension of .err. For example, if an error occurred while creating a report using Statspack snapshots 1 and 7, the report file would be named ORCL_spreport_1_7.err. You can download the error report using the same techniques as for a standard Snapshot report.

Removing Statspack files

To remove Oracle Statspack files, use the following command:

```
exec statspack.purge(begin snap, end snap);
```

Oracle time zone

You can use the time zone option to change the system time zone used by your Oracle DB instance. For example, you might change the time zone of a DB instance to be compatible with an on-premises environment, or a legacy application. The time zone option changes the time zone at the host level. Changing the time zone impacts all date columns and values, including SYSDATE and SYSTIMESTAMP.

The time zone option differs from the `rdsadmin_util.alter_db_time_zone` command. The `alter_db_time_zone` command changes the time zone only for certain data types. The time zone option changes the time zone for all date columns and values. For more information about `alter_db_time_zone`, see [Setting the database time zone \(p. 1052\)](#). For more information about upgrade considerations, see [Time zone considerations \(p. 1213\)](#).

Considerations for setting the time zone

The time zone option is a permanent and persistent option. Therefore, you can't do the following:

- Remove the option from an option group after you add the option.
- Remove the option group from a DB instance after you add the group.
- Modify the time zone setting of the option to a different time zone.

If you accidentally set the time zone incorrectly, you need to recover the DB instance to its previous time zone setting. We strongly urge you to use one of the following strategies, depending on your situation:

- Your DB instance currently uses the default option group.

Take a snapshot of your DB instance, and then add the time zone option to your DB instance. For more information, see [Creating a DB snapshot \(p. 346\)](#).

- Your DB instance currently uses a nondefault option group.

Take a snapshot of your DB instance, create a new option group with the time zone option, and then add the option group to your instance.

We strongly urge you to test the time zone option on a test DB instance before you add it to a production DB instance. Adding the time zone option can cause problems with tables that use system date to add dates or times. We recommend that you analyze your data and applications to determine the impact of changing the time zone.

Time zone option settings

Amazon RDS supports the following settings for the time zone option.

Option setting	Valid values	Description
TIME_ZONE	One of the available time zones. For the full list, see Available time zones (p. 1203) .	The new time zone for your DB instance.

Adding the time zone option

The general process for adding the time zone option to a DB instance is the following:

1. Create a new option group, or copy or modify an existing option group.

2. Add the option to the option group.
3. Associate the option group with the DB instance.

When you add the time zone option, a brief outage occurs while your DB instance is automatically restarted.

Console

To add the time zone option to a DB instance

1. Determine the option group you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group with the following settings:
 - a. For **Engine** choose the oracle edition for your DB instance.
 - b. For **Major engine version** choose the version of your DB instance.

For more information, see [Creating an option group \(p. 214\)](#).

2. Add the **Timezone** option to the option group, and configure the option settings.

Important

If you add the time zone option to an existing option group that is already attached to one or more DB instances, a brief outage occurs while all the DB instances are automatically restarted.

For more information about adding options, see [Adding an option to an option group \(p. 216\)](#). For more information about each setting, see [Time zone option settings \(p. 1201\)](#).

3. Apply the option group to a new or existing DB instance:

- For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- For an existing DB instance, you apply the option group by modifying the instance and attaching the new option group. When you add the time zone option to an existing DB instance, a brief outage occurs while your DB instance is automatically restarted. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

AWS CLI

The following example uses the AWS CLI [add-option-to-option-group](#) command to add the **Timezone** option and the **TIME_ZONE** option setting to an option group called **myoptiongroup**. The time zone is set to **Africa/Cairo**.

For Linux, macOS, or Unix:

```
aws rds add-option-to-option-group \
--option-group-name "myoptiongroup" \
--options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/Cairo}]" \
--apply-immediately
```

For Windows:

```
aws rds add-option-to-option-group ^
--option-group-name "myoptiongroup" ^
--options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/Cairo}]" ^
--apply-immediately
```

Modifying time zone settings

The time zone option is a permanent and persistent option. You can't remove the option from an option group after you add it. You can't remove the option group from a DB instance after you add it. You can't modify the time zone setting of the option to a different time zone. If you set the time zone incorrectly, restore a snapshot of your DB instance from before you added the time zone option.

Removing the time zone option

The time zone option is a permanent and persistent option. You can't remove the option from an option group after you add it. You can't remove the option group from a DB instance after you add it. To remove the time zone option, restore a snapshot of your DB instance from before you added the time zone option.

Available time zones

The following values can be used for the time zone option.

Zone	Time zone
Africa	Africa/Cairo, Africa/Casablanca, Africa/Harare, Africa/Lagos, Africa/Luanda, Africa/Monrovia, Africa/Nairobi, Africa/Tripoli, Africa/Windhoek
America	America/Araguaina, America/Argentina/Buenos_Aires, America/Asuncion, America/Bogota, America/Caracas, America/Chicago, America/Chihuahua, America/Cuiaba, America/Denver, America/Detroit, America/Fortaleza, America/Godthab, America/Guatemala, America/Halifax, America/Lima, America/Los_Angeles, America/Manaus, America/Matamoros, America/Mexico_City, America/Monterrey, America/Montevideo, America/New_York, America/Phoenix, America/Santiago, America/Sao_Paulo, America/Tijuana, America/Toronto
Asia	Asia/Amman, Asia/Ashgabat, Asia/Baghdad, Asia/Baku, Asia/Bangkok, Asia/Beirut, Asia/Calcutta, Asia/Damascus, Asia/Dhaka, Asia/Hong_Kong, Asia/Irkutsk, Asia/Jakarta, Asia/Jerusalem, Asia/Kabul, Asia/Karachi, Asia/Kathmandu, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Magadan, Asia/Manila, Asia/Muscat, Asia/Novosibirsk, Asia/Rangoon, Asia/Riyadh, Asia/Seoul, Asia/Shanghai, Asia/Singapore, Asia/Taipei, Asia/Tehran, Asia/Tokyo, Asia/Ulaanbaatar, Asia/Vladivostok, Asia/Yakutsk, Asia/Yerevan
Atlantic	Atlantic/Azores, Atlantic/Cape_Verde
Australia	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Eucla, Australia/Hobart, Australia/Lord_Howe, Australia/Perth, Australia/Sydney
Brazil	Brazil/DeNoronha, Brazil/East
Canada	Canada/Newfoundland, Canada/Saskatchewan
Etc	Etc/GMT-3
Europe	Europe/Amsterdam, Europe/Athens, Europe/Berlin, Europe/Dublin, Europe/Helsinki, Europe/Kaliningrad, Europe/London, Europe/Madrid, Europe/Moscow, Europe/Paris, Europe/Prague, Europe/Rome, Europe/Sarajevo
Pacific	Pacific/Apia, Pacific/Auckland, Pacific/Chatham, Pacific/Fiji, Pacific/Guam, Pacific/Honolulu, Pacific/Kiritimati, Pacific/Marquesas, Pacific/Samoa, Pacific/Tongatapu, Pacific/Wake

Zone	Time zone
US	US/Alaska, US/Central, US/East-Indiana, US/Eastern, US/Pacific
UTC	UTC

Related topics

- [Working with option groups \(p. 212\)](#)
- [Adding options to Oracle DB instances \(p. 1126\)](#)
- [Setting the time zone for Oracle Scheduler jobs \(p. 1086\)](#)

Oracle Transparent Data Encryption

Amazon RDS supports Oracle Transparent Data Encryption (TDE), a feature of the Oracle Advanced Security option available in Oracle Enterprise Edition. This feature automatically encrypts data before it is written to storage and automatically decrypts data when the data is read from storage.

Oracle Transparent Data Encryption is used in scenarios where you need to encrypt sensitive data in case data files and backups are obtained by a third party or when you need to address security-related regulatory compliance issues.

The TDE option is a permanent option that can't be removed from an option group. You can't disable TDE from a DB instance once that instance is associated with an option group with the Oracle TDE option. You can change the option group of a DB instance that is using the TDE option, but the option group associated with the DB instance must include the TDE option. You can also modify an option group that includes the TDE option by adding or removing other options.

A detailed explanation about Oracle Transparent Data Encryption is beyond the scope of this guide. For information about using Oracle Transparent Data Encryption, see [Securing stored data using Transparent Data Encryption](#). For more information about Oracle Advanced Security, see [Oracle advanced security](#) in the Oracle documentation. For more information on AWS security, see the [AWS security center](#).

Note

You can't share a DB snapshot that uses this option. For more information about sharing DB snapshots, see [Sharing a DB snapshot \(p. 365\)](#).

TDE encryption modes

Oracle Transparent Data Encryption supports two encryption modes: TDE tablespace encryption and TDE column encryption. TDE tablespace encryption is used to encrypt entire application tables. TDE column encryption is used to encrypt individual data elements that contain sensitive data. You can also apply a hybrid encryption solution that uses both TDE tablespace and column encryption.

Note

Amazon RDS manages the Oracle Wallet and TDE master key for the DB instance. You do not need to set the encryption key using the command `ALTER SYSTEM set encryption key`.

For information about TDE best practices, see [Oracle advanced security Transparent Data Encryption best practices](#).

Once the option is enabled, you can check the status of the Oracle Wallet by using the following command:

```
SELECT * FROM v$encryption_wallet;
```

To create an encrypted tablespace, use the following command:

```
CREATE TABLESPACE encrypt_ts ENCRYPTION DEFAULT STORAGE (ENCRYPT);
```

To specify the encryption algorithm, use the following command:

```
CREATE TABLESPACE encrypt_ts ENCRYPTION USING 'AES256' DEFAULT STORAGE (ENCRYPT);
```

Note that the previous commands for encrypting a tablespace are the same as the commands you would use with an Oracle installation not on Amazon RDS, and the ALTER TABLE syntax to encrypt a column is also the same as the commands you would use for an Oracle installation not on Amazon RDS.

You should determine if your DB instance is associated with an option group that has the **TDE** option. To view the option group that a DB instance is associated with, you can use the RDS console, the [describe-db-instance](#) AWS CLI command, or the API operation [DescribeDBInstances](#).

To comply with several security standards, Amazon RDS is working to implement automatic periodic master key rotation.

Adding the TDE option

The process for using Oracle Transparent Data Encryption (TDE) with Amazon RDS is as follows:

1. If the DB instance is not associated with an option group that has the **TDE** option enabled, you must either create an option group and add the **TDE** option or modify the associated option group to add the **TDE** option. For information about creating or modifying an option group, see [Working with option groups \(p. 212\)](#). For information about adding an option to an option group, see [Adding an option to an option group \(p. 216\)](#).
2. Associate the DB instance with the option group with the **TDE** option. For information about associating a DB instance with an option group, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Removing the TDE option

If you no longer want to use the TDE option with a DB instance, you must decrypt all your data on the DB instance, copy the data to a new DB instance that is not associated with an option group with TDE enabled, and then delete the original instance. You can rename the new instance to be the same name as the previous DB instance if you prefer.

Using TDE with Oracle Data Pump

You can use Oracle Data Pump to import or export encrypted dump files. Amazon RDS supports the password encryption mode (ENCRYPTION_MODE=PASSWORD) for Oracle Data Pump. Amazon RDS does not support transparent encryption mode (ENCRYPTION_MODE=TRANSPARENT) for Oracle Data Pump. For more information about using Oracle Data Pump with Amazon RDS, see [Importing using Oracle Data Pump \(p. 1106\)](#).

Oracle UTL_MAIL

Amazon RDS supports Oracle UTL_MAIL through the use of the UTL_MAIL option and SMTP servers. You can send email directly from your database by using the UTL_MAIL package. Amazon RDS supports UTL_MAIL for the following versions of Oracle:

- Oracle Database 19c (19.0.0.0), all versions
- Oracle Database 18c (18.0.0.0), all versions
- Oracle Database 12c Release 2 (12.2), all versions
- Oracle Database 12c Release 1 (12.1), version 12.1.0.2.v5 and later

The following are some limitations to using UTL_MAIL:

- UTL_MAIL does not support Transport Layer Security (TLS) and therefore emails are not encrypted.

To connect securely to remote SSL/TLS resources by creating and uploading custom Oracle wallets, follow the instructions in [Configuring outbound network access on your Oracle DB instance \(p. 1025\)](#).

The specific certificates that are required for your wallet vary by service. For AWS services, these can typically be found in the [Amazon trust services repository](#).

- UTL_MAIL does not support authentication with SMTP servers.
- You can only send a single attachment in an email.
- You can't send attachments larger than 32 K.
- You can only use ASCII and Extended Binary Coded Decimal Interchange Code (EBCDIC) character encodings.
- SMTP port (25) is throttled based on the elastic network interface owner's policies.

When you enable UTL_MAIL, only the master user for your DB instance is granted the execute privilege. If necessary, the master user can grant the execute privilege to other users so that they can use UTL_MAIL.

Important

We recommend that you enable Oracle's built-in auditing feature to track the use of UTL_MAIL procedures.

Prerequisites for Oracle UTL_MAIL

The following are prerequisites for using Oracle UTL_MAIL:

- One or more SMTP servers, and the corresponding IP addresses or public or private Domain Name Server (DNS) names. For more information about private DNS names resolved through a custom DNS server, see [Setting up a custom DNS server \(p. 1045\)](#).
- For Oracle versions prior to 12c, your DB instance must also use the XML DB option. For more information, see [Oracle XML DB \(p. 1208\)](#).

Adding the Oracle UTL_MAIL option

The general process for adding the Oracle UTL_MAIL option to a DB instance is the following:

1. Create a new option group, or copy or modify an existing option group.
2. Add the option to the option group.
3. Associate the option group with the DB instance.

After you add the UTL_MAIL option, as soon as the option group is active, UTL_MAIL is active.

To add the UTL_MAIL option to a DB instance

1. Determine the option group you want to use. You can create a new option group or use an existing option group. If you want to use an existing option group, skip to the next step. Otherwise, create a custom DB option group with the following settings:
 - a. For **Engine**, choose the edition of Oracle you want to use.
 - b. For **Major engine version**, choose the version of your DB instance.

For more information, see [Creating an option group \(p. 214\)](#).
2. Add the **UTL_MAIL** option to the option group. For more information about adding options, see [Adding an option to an option group \(p. 216\)](#).
3. Apply the option group to a new or existing DB instance:
 - For a new DB instance, you apply the option group when you launch the instance. For more information, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
 - For an existing DB instance, you apply the option group by modifying the instance and attaching the new option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Using Oracle UTL_MAIL

After you enable the UTL_MAIL option, you must configure the SMTP server before you can begin using it.

You configure the SMTP server by setting the `SMTP_OUT_SERVER` parameter to a valid IP address or public DNS name. For the `SMTP_OUT_SERVER` parameter, you can specify a comma-separated list of the addresses of multiple servers. If the first server is unavailable, UTL_MAIL tries the next server, and so on.

You can set the default `SMTP_OUT_SERVER` for a DB instance by using a [DB parameter group](#). You can set the `SMTP_OUT_SERVER` parameter for a session by running the following code on your database on your DB instance.

```
ALTER SESSION SET smtp_out_server = mailserver.domain.com:25;
```

After the UTL_MAIL option is enabled, and your `SMTP_OUT_SERVER` is configured, you can send mail by using the `SEND` procedure. For more information, see [UTL_MAIL](#) in the Oracle documentation.

Removing the Oracle UTL_MAIL option

You can remove Oracle UTL_MAIL from a DB instance.

To remove UTL_MAIL from a DB instance, do one of the following:

- To remove UTL_MAIL from multiple DB instances, remove the UTL_MAIL option from the option group they belong to. This change affects all DB instances that use the option group. For more information, see [Removing an option from an option group \(p. 224\)](#).
- To remove UTL_MAIL from a single DB instance, modify the DB instance and specify a different option group that doesn't include the UTL_MAIL option. You can specify the default (empty) option group, or a different custom option group. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Troubleshooting

The following are issues you might encounter when you use UTL_MAIL with Amazon RDS.

- Throttling. SMTP port (25) is throttled based on the elastic network interface owner's policies. If you can successfully send email by using UTL_MAIL, and you see the error ORA-29278: `SMTP transient error: 421 Service not available`, you are possibly being throttled. If you experience throttling with email delivery, we recommend that you implement a backoff algorithm. For more information about backoff algorithms, see [Error retries and exponential backoff in AWS](#) and [How to handle a "throttling – Maximum sending rate exceeded" error](#).

You can request that this throttle be removed. For more information, see [How do I remove the throttle on port 25 from my EC2 instance?](#).

Oracle XML DB

Oracle XML DB adds native XML support to your DB instance. With XML DB, you can store and retrieve structured or unstructured XML, in addition to relational data. XML DB is preinstalled on Oracle version 12c and later.

Upgrading the Oracle DB engine

When Amazon RDS supports a new version of Oracle, you can upgrade your DB instances to the new version. For information about which Oracle versions are available on Amazon RDS, see [Oracle database engine release notes \(p. 1245\)](#).

Important

RDS for Oracle Database 11g is deprecated. If you maintain Oracle Database 11g snapshots, you can upgrade them to a later release. For more information, see [Upgrading an Oracle DB snapshot \(p. 1217\)](#).

Topics

- [Overview of Oracle DB engine upgrades \(p. 1209\)](#)
- [Major version upgrades \(p. 1211\)](#)
- [Oracle minor version upgrades \(p. 1212\)](#)
- [Oracle SE2 upgrade paths \(p. 1212\)](#)
- [Considerations for Oracle DB upgrades \(p. 1213\)](#)
- [Preparing for the automatic upgrade of Oracle Database 18c \(p. 1214\)](#)
- [Testing an Oracle DB upgrade \(p. 1215\)](#)
- [Upgrading an Oracle DB instance \(p. 1216\)](#)

Overview of Oracle DB engine upgrades

Before upgrading your Oracle DB instance, familiarize yourself with the following concepts.

Topics

- [Major and minor version upgrades \(p. 1209\)](#)
- [Oracle engine version management \(p. 1210\)](#)
- [Automatic snapshots during engine upgrades \(p. 1210\)](#)
- [Oracle upgrades in a Multi-AZ deployment \(p. 1210\)](#)
- [Oracle upgrades of read replicas \(p. 1210\)](#)
- [Oracle upgrades of micro DB instances \(p. 1211\)](#)

Major and minor version upgrades

Amazon RDS supports the following upgrades to an Oracle DB instance:

- Major version upgrades

In general, a *major version upgrade* for a database engine can introduce changes that aren't compatible with existing applications. To upgrade your DB instance to a major version, you must perform the action manually.

- Minor version upgrades

A *minor version upgrade* includes only changes that are backward-compatible with existing applications. If you enable auto minor version upgrades on your DB instance, minor version upgrades occur automatically. In all other cases, you upgrade the DB instance manually.

When you upgrade the DB engine, an outage occurs. The duration of the outage depends on your engine version and instance size.

Oracle engine version management

With DB engine version management, you control when and how the database engine is patched and upgraded. You get the flexibility to maintain compatibility with database engine patch versions. You can also test new patch versions to ensure they work with your application before deploying them in production. In addition, you upgrade the versions on your own terms and timelines.

Note

Amazon RDS periodically aggregates official Oracle database patches using an Amazon RDS-specific DB engine version. To see a list of which Oracle patches are contained in an Amazon RDS Oracle-specific engine version, go to [Oracle database engine release notes \(p. 1245\)](#).

Automatic snapshots during engine upgrades

During upgrades of an Oracle DB instance, snapshots offer protection against upgrade issues. If the backup retention period for your DB instance is greater than 0, Amazon RDS takes the following DB snapshots during the upgrade:

1. A snapshot of the DB instance before any upgrade changes have been made. If the upgrade fails, you can restore this snapshot to create a DB instance running the old version.
2. A snapshot of the DB instance after the upgrade completes.

Note

To change your backup retention period, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

After an upgrade, you can't revert to the previous engine version. However, you can create a new Oracle DB instance by restoring the pre-upgrade snapshot.

Oracle upgrades in a Multi-AZ deployment

If your DB instance is in a Multi-AZ deployment, Amazon RDS upgrades both the primary and standby replicas. If no operating system updates are required, the primary and standby upgrades occur simultaneously. The instances are not available until the upgrade completes.

If operating system updates are required in a Multi-AZ deployment, Amazon RDS applies the updates when you request the DB upgrade. Amazon RDS performs the following steps:

1. Updates the operating system on the standby DB instance.
2. Upgrades the standby DB instance.
3. Fails over the primary instance to the standby DB instance.
4. Upgrades the operating system on the new standby DB instance, which was formerly the primary instance.
5. Upgrades the new standby DB instance.

Oracle upgrades of read replicas

The Oracle DB engine version of the source DB instance and all of its read replicas must be the same. Amazon RDS performs the upgrade in the following stages:

1. Upgrades the source DB instance. The read replicas are available during this stage.
2. Upgrades the read replicas in parallel, regardless of the replica maintenance windows. The source DB is available during this stage.

For major version upgrades of cross-Region read replicas, Amazon RDS performs additional actions:

- Generates an option group for the target version automatically
- Copies all options and option settings from the original option group to the new option group
- Associates the upgraded cross-Region read replica with the new option group

Oracle upgrades of micro DB instances

We don't recommend upgrading databases running on micro DB instances. Because these instances have limited CPU, the upgrade can take hours to complete.

You can upgrade micro DB instances with small amounts of storage (10–20 GiB) by copying your data using Data Pump. Before you migrate your production DB instances, we recommend that you test by copying data using Data Pump.

Major version upgrades

Amazon RDS supports the following major version upgrades.

To perform a major version upgrade, modify the DB instance manually. Major version upgrades don't occur automatically.

Supported versions for major upgrades

Amazon RDS supports the following major version upgrades.

Current version	Upgrade supported
18.0.0.0	19.0.0.0
12.2.0.1	19.0.0.0
	18.0.0.0
12.1.0.2	19.0.0.0
	18.0.0.0
	12.2.0.1

A major version upgrade of Oracle Database must upgrade to a Release Update (RU) that was released in the same month or later. Major version downgrades aren't supported for any Oracle versions.

Supported instance classes for major upgrades

Your current Oracle DB instance might run on a DB instance class that isn't supported for the version to which you are upgrading. In this case, before you upgrade, migrate the DB instance to a supported DB instance class. For more information about the supported DB instance classes for each version and edition of Amazon RDS for Oracle, see [DB instance classes \(p. 7\)](#).

Gathering statistics before major upgrades

Before you perform a major version upgrade, Oracle recommends that you gather optimizer statistics on the DB instance that you are upgrading. This action can reduce DB instance downtime during the upgrade.

To gather optimizer statistics, connect to the DB instance as the master user, and run the `DBMS_STATS.GATHER_DICTIONARY_STATS` procedure, as in the following example.

```
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS;
```

For more information, see [Gathering optimizer statistics to decrease Oracle database downtime](#) in the Oracle documentation.

Allowing major upgrades

A major engine version upgrade might be incompatible with your application. The upgrade is irreversible. If you specify a major version for the `EngineVersion` parameter that is different from the current major version, you must allow major version upgrades.

If you upgrade a major version using the CLI command [modify-db-instance](#), specify `--allow-major-version-upgrade`. This setting isn't persistent, so you must specify `--allow-major-version-upgrade` whenever you perform a major upgrade. This parameter has no impact on upgrades of minor engine versions. For more information, see [Upgrading a DB instance engine version \(p. 271\)](#).

If you upgrade a major version using the console, you don't need to choose an option to allow the upgrade. Instead, the console displays a warning that major upgrades are irreversible.

Oracle minor version upgrades

A minor version upgrade applies an Oracle Database Patch Set Update (PSU) or Release Update (RU) in a major version.

An Amazon RDS for Oracle DB instance is scheduled to be upgraded automatically during its next maintenance window when it meets the following conditions:

- The DB instance has the **Auto minor version upgrade** option enabled.
- The DB instance is not running the latest minor DB engine version.

The DB instance is upgraded to the latest quarterly PSU or RU four to six weeks after it is made available by Amazon RDS for Oracle. For more information about PSUs and RUs, see [Oracle database engine release notes \(p. 1245\)](#).

The following minor version upgrades aren't supported.

Current version	Upgrade not supported
12.1.0.2.v6	12.1.0.2.v7
12.1.0.2.v5	12.1.0.2.v7
12.1.0.2.v5	12.1.0.2.v6

Note

Minor version downgrades aren't supported.

Oracle SE2 upgrade paths

The following table shows supported upgrade paths to Standard Edition Two (SE2). For more information about the License Included and Bring Your Own License (BYOL) models, see [Oracle licensing options \(p. 990\)](#).

Your existing configuration	Supported SE2 configuration
12.2.0.1 SE2, BYOL	12.2.0.1 SE2, BYOL or License Included
12.1.0.2 SE2, BYOL	12.2.0.1 SE2, BYOL or License Included 12.1.0.2 SE2, BYOL or License Included

To upgrade from your existing configuration to a supported SE2 configuration, use a supported upgrade path. For more information, see [Major version upgrades \(p. 1211\)](#).

Considerations for Oracle DB upgrades

Before upgrading, review the implications for option groups, parameter groups, and time zones.

Option group considerations

If your DB instance uses a custom option group, sometimes Amazon RDS can't automatically assign a new option group. For example, this situation occurs when you upgrade to a new major version. In such cases, specify a new option group when you upgrade. We recommend that you create a new option group, and add the same options to it as in your existing custom option group.

For more information, see [Creating an option group \(p. 214\)](#) or [Copying an option group \(p. 215\)](#).

If your DB instance uses a custom option group that contains the APEX option, you can sometimes reduce the upgrade time. To do this, upgrade your version of APEX at the same time as your DB instance. For more information, see [Upgrading the APEX version \(p. 1148\)](#).

Parameter group considerations

If your DB instance uses a custom parameter group, sometimes Amazon RDS can't automatically assign your DB instance a new parameter group. For example, this situation occurs when you upgrade to a new major version. In such cases, make sure to specify a new parameter group when you upgrade. We recommend that you create a new parameter group, and configure the parameters as in your existing custom parameter group.

For more information, see [Creating a DB parameter group \(p. 229\)](#) or [Copying a DB parameter group \(p. 236\)](#).

Time zone considerations

You can use the time zone option to change the *system time zone* used by your Oracle DB instance. For example, you might change the time zone of a DB instance to be compatible with an on-premises environment, or a legacy application. The time zone option changes the time zone at the host level. Amazon RDS for Oracle updates the system time zone automatically throughout the year. For more information about the system time zone, see [Oracle time zone \(p. 1201\)](#).

When you create an Oracle DB instance, the database automatically sets the *database time zone*. The database time zone is also known as the Daylight Saving Time (DST) time zone. The database time zone is distinct from the system time zone.

Between Oracle Database releases, patch sets or individual patches may include new DST versions. These patches reflect the changes in transition rules for various time zone regions. For example, a government

might change when DST takes effect. Changes to DST rules may affect existing data of the `TIMESTAMP WITH TIME ZONE` data type.

If you upgrade an RDS for Oracle instance, Amazon RDS does not upgrade the database time zone automatically. To upgrade the database time zone manually, create a new Oracle DB instance that has the desired DST patch. Then migrate the data from your current instance to the new instance. You can migrate data using several techniques, including the following:

- Oracle GoldenGate
- AWS Database Migration Service
- Oracle Data Pump
- Original Export/Import (desupported for general use)

Note

When you migrate data using Oracle Data Pump, the utility raises the error ORA-39405 when the target time zone version is lower than the source time zone version.

For more information, see [TIMESTAMP WITH TIMEZONE restrictions](#) in the Oracle documentation.

Preparing for the automatic upgrade of Oracle Database 18c

On July 1, 2021, Amazon RDS plans to begin automatically upgrading Oracle Database 18c instances to Oracle Database 19c. The automatic upgrades are not guaranteed to occur in your maintenance window. All Oracle Database 18c instances, including reserved instances, will move to the latest available Release Update (RU).

Before the automatic upgrades begin, we highly recommend that you upgrade your existing Oracle Database 18c DB instances to Oracle Database 19c manually. When you upgrade manually, you can validate that your applications work correctly. To avoid the automatic upgrade, use one of the following strategies before July 1, 2021.

Topics

- [Upgrade your Oracle Database 18c DB instance \(p. 1214\)](#)
- [Upgrade your Oracle Database 18c DB snapshots \(p. 1214\)](#)
- [Downgrade your Oracle Database 18c DB instance \(p. 1215\)](#)

Upgrade your Oracle Database 18c DB instance

You can upgrade your Oracle Database 18c instance to Oracle Database 19c. Before upgrading, consider the following:

- Your SQL statements might perform differently after the upgrade. If so, you can use the `OPTIMIZER_FEATURES_ENABLE` parameter to retain the behavior of the Oracle Database 18c optimizer. For more information, see [Influencing the Optimizer](#) in the Oracle documentation.
- If you have Extended Support for Oracle Database 18c on the BYOL model, consider the implications. In this case, you must have Extended Support agreements from Oracle Support for Oracle Database 19c. For details on licensing and support requirements for BYOL, see [Amazon RDS for Oracle FAQs](#).

Upgrade your Oracle Database 18c DB snapshots

You can upgrade your existing snapshots to Oracle Database 19c, and then restore them. For more information, see [Upgrading an Oracle DB snapshot \(p. 1217\)](#).

If you plan to upgrade using snapshots, the planned deadline to avoid the automatic upgrade is June 30, 2021.

Downgrade your Oracle Database 18c DB instance

You may decide not to upgrade your DB instances to Oracle Database 19c. In this case, you can downgrade your instance to Oracle Database Release 1 (12.1.0.2) or Release 2 (12.2.0.1). Use any of the following techniques:

- Oracle Data Pump
- AWS Database Migration Service (DMS)
- Any supported logical replication tool

For more information about import options, see [Importing data into Oracle on Amazon RDS \(p. 1106\)](#).

Testing an Oracle DB upgrade

Before you upgrade your DB instance to a major version, thoroughly test your database and all applications that access the database for compatibility with the new version. We recommend that you use the following procedure.

To test a major version upgrade

1. Review the Oracle upgrade documentation for the new version of the database engine to see if there are compatibility issues that might affect your database or applications. For more information, see [Database Upgrade Guide](#) in the Oracle documentation.
2. If your DB instance uses a custom option group, create a new option group compatible with the new version you are upgrading to. For more information, see [Option group considerations \(p. 1213\)](#).
3. If your DB instance uses a custom parameter group, create a new parameter group compatible with the new version you are upgrading to. For more information, see [Parameter group considerations \(p. 1213\)](#).
4. Create a DB snapshot of the DB instance to be upgraded. For more information, see [Creating a DB snapshot \(p. 346\)](#).
5. Restore the DB snapshot to create a new test DB instance. For more information, see [Restoring from a DB snapshot \(p. 349\)](#).
6. Modify this new test DB instance to upgrade it to the new version, by using one of the following methods:
 - [Console \(p. 271\)](#)
 - [AWS CLI \(p. 272\)](#)
 - [RDS API \(p. 272\)](#)
7. Perform testing:
 - Run as many of your quality assurance tests against the upgraded DB instance as needed to ensure that your database and application work correctly with the new version.
 - Implement any new tests needed to evaluate the impact of any compatibility issues that you identified in step 1.
 - Test all stored procedures, functions, and triggers.
 - Direct test versions of your applications to the upgraded DB instance. Verify that the applications work correctly with the new version.
 - Evaluate the storage used by the upgraded instance to determine if the upgrade requires additional storage. You might need to choose a larger instance class to support the new version in production. For more information, see [DB instance classes \(p. 7\)](#).

8. If all tests pass, upgrade your production DB instance. We recommend that you confirm that the DB instance working correctly before allowing write operations to the DB instance.

Upgrading an Oracle DB instance

For information about manually or automatically upgrading an Oracle DB instance, see [Upgrading a DB instance engine version \(p. 271\)](#).

Upgrading an Oracle DB snapshot

If you have existing manual DB snapshots, you can upgrade them to a later version of the Oracle database engine.

When Oracle stops providing patches for a version, and Amazon RDS deprecates the version, you can upgrade your snapshots that correspond to the deprecated version. For more information, see [Oracle engine version management \(p. 1210\)](#).

The following snapshot upgrades are currently supported.

Current snapshot version	Supported snapshot upgrade
12.1.0.1	12.1.0.2.v8
11.2.0.4	12.1.0.2, 12.2.0.1, 18c, and 19c when the following conditions are met: <ul style="list-style-type: none">• The minor version is not a downgrade.• You upgrade the snapshot to the RU, RUR, or PSU for July 2020 or later. For specific version numbers, see Oracle database engine release notes (p. 1245).
11.2.0.3	11.2.0.4.v11
11.2.0.2	11.2.0.4.v12

Amazon RDS supports upgrading snapshots in all AWS Regions.

Console

To upgrade an Oracle DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**, and then select the DB snapshot that you want to upgrade.
3. For **Actions**, choose **Upgrade snapshot**. The **Upgrade snapshot** page appears.
4. Choose the **New engine version** to upgrade the snapshot to.
5. (Optional) For **Option group**, choose the option group for the upgraded DB snapshot. The same option group considerations apply when upgrading a DB snapshot as when upgrading a DB instance. For more information, see [Option group considerations \(p. 1213\)](#).
6. Choose **Save changes** to save your changes.

During the upgrade process, all snapshot actions are disabled for this DB snapshot. Also, the DB snapshot status changes from **available** to **upgrading**, and then changes to **active** upon completion. If the DB snapshot can't be upgraded because of snapshot corruption issues, the status changes to **unavailable**. You can't recover the snapshot from this state.

Note

If the DB snapshot upgrade fails, the snapshot is rolled back to the original state with the original version.

AWS CLI

To upgrade an Oracle DB snapshot by using the AWS CLI, call the [modify-db-snapshot](#) command with the following parameters:

- `--db-snapshot-identifier` – The name of the DB snapshot.
- `--engine-version` – The version to upgrade the snapshot to.

You might also need to include the following parameter. The same option group considerations apply when upgrading a DB snapshot as when upgrading a DB instance. For more information, see [Option group considerations \(p. 1213\)](#).

- `--option-group-name` – The option group for the upgraded DB snapshot.

Example

The following example upgrades a DB snapshot.

For Linux, macOS, or Unix:

```
aws rds modify-db-snapshot \
  --db-snapshot-identifier mydbsnapshot \
  --engine-version 11.2.0.4.v12 \
  --option-group-name default:oracle-se1-11-2
```

For Windows:

```
aws rds modify-db-snapshot ^
  --db-snapshot-identifier mydbsnapshot ^
  --engine-version 11.2.0.4.v12 ^
  --option-group-name default:oracle-se1-11-2
```

RDS API

To upgrade an Oracle DB snapshot by using the Amazon RDS API, call the [ModifyDBSnapshot](#) operation with the following parameters:

- `DBSnapshotIdentifier` – The name of the DB snapshot.
- `EngineVersion` – The version to upgrade the snapshot to.

You might also need to include the `OptionGroupName` parameter. The same option group considerations apply when upgrading a DB snapshot as when upgrading a DB instance. For more information, see [Option group considerations \(p. 1213\)](#).

Using your Oracle DB instance with third-party software

This section provides information about tools and third-party software for Oracle DB instances on Amazon RDS.

Topics

- [Setting up Amazon RDS to host tools and third-party software for Oracle \(p. 1219\)](#)
- [Using Oracle GoldenGate with Amazon RDS \(p. 1225\)](#)
- [Using the Oracle Repository Creation Utility on Amazon RDS for Oracle \(p. 1237\)](#)
- [Installing a Siebel database on Oracle on Amazon RDS \(p. 1242\)](#)

Setting up Amazon RDS to host tools and third-party software for Oracle

You can use Amazon RDS to host an Oracle DB instance that supports software and components such as the following:

- Siebel Customer Relationship Management (CRM)
- Oracle Fusion Middleware Metadata — installed by the Repository Creation Utility (RCU)

The following procedures help you create an Oracle DB instance on Amazon RDS that you can use to host additional software and components for Oracle.

Creating a VPC for use with an Oracle database

In the following procedure, you create a virtual private cloud (VPC) based on the Amazon VPC service, a private subnet, and a security group. Your Amazon RDS DB instance needs to be available only to your middle-tier components, and not to the public internet. Thus, your Amazon RDS DB instance is hosted in a private subnet, providing greater security.

To create a VPC based on Amazon VPC

1. Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the upper-right corner of the AWS Management Console, choose the AWS Region for your VPC. This example uses the US West (Oregon) region.
3. In the upper-left corner, choose **VPC Dashboard**, and then choose **Start VPC Wizard**.
4. On the page **Step 1: Select a VPC Configuration**, choose **VPC with Public and Private Subnets**, and then choose **Select**.
5. On the page **Step 2: VPC with Public and Private Subnets**, shown following, set the following values.

Option	Value
IPv4 CIDR block	10.0.0.0/16 For more information about selecting CIDR blocks for your VPC, see VPC sizing .

Option	Value
IPv6 CIDR block	No IPv6 CIDR Block
VPC name	The name for your VPC, for example vpc-1 .
Public subnet's IPv4 CIDR	10.0.0.0/24 For more information about subnet sizing, see Subnet sizing .
Availability Zone	An Availability Zone for your AWS Region.
Public subnet name	The name for your public subnet, for example subnet-public-1 .
Private subnet's IPv4 CIDR	10.0.1.0/24 For more information about subnet sizing, see Subnet sizing .
Availability Zone	An Availability Zone for your AWS Region.
Private subnet name	The name for your private subnet, for example subnet-private-1 .
Instance type	An instance type for your NAT instance, for example t2.small . Note If you don't see Instance type in the console, choose Use a NAT instance instead .
Key pair name	No key pair
Service endpoints	None
Enable DNS hostnames	Yes
Hardware tenancy	Default

Step 2: VPC with Public and Private Subnets

IPv4 CIDR block: <input type="text" value="10.0.0.0/16"/>	(65531 IP addresses available)
IPv6 CIDR block:	<input checked="" type="radio"/> No IPv6 CIDR Block <input type="radio"/> Amazon provided IPv6 CIDR block
VPC name:	<input type="text" value="vpc-1"/>
Public subnet's IPv4 CIDR: <input type="text" value="10.0.0.0/24"/>	(251 IP addresses available)
Availability Zone: <input type="text" value="us-west-2a"/>	
Public subnet name:	<input type="text" value="subnet-public-1"/>
Private subnet's IPv4 CIDR: <input type="text" value="10.0.1.0/24"/>	(251 IP addresses available)
Availability Zone: <input type="text" value="us-west-2a"/>	
Private subnet name:	<input type="text" value="subnet-private-1"/>
You can add more subnets after AWS creates the VPC.	
Specify the details of your NAT instance (Instance rates apply). Use a NAT gateway instead	
Instance type: <input type="text" value="t2.small"/>	
Key pair name: <input type="text" value="No key pair"/>	
Service endpoints	
<input type="button" value="Add Endpoint"/>	
Enable DNS hostnames: <input checked="" type="radio"/> Yes <input type="radio"/> No	
Hardware tenancy: <input type="text" value="Default"/>	
<input type="button" value="Cancel and Exit"/> <input type="button" value="Back"/> <input type="button" value="Create VPC"/>	

- Choose **Create VPC**.

An Amazon RDS DB instance in a VPC requires at least two private subnets or at least two public subnets, to support Multi-AZ deployment. For more information about working with multiple Availability Zones, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#). Because your database is private, add a second private subnet to your VPC.

To create an additional subnet

- Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
- In the upper-right corner of the AWS Management Console, confirm that you are in the correct AWS Region for your VPC.
- In the upper-left corner, choose **VPC Dashboard**, choose **Subnets**, and then choose **Create Subnet**.

4. On the **Create Subnet** page, set the following values.

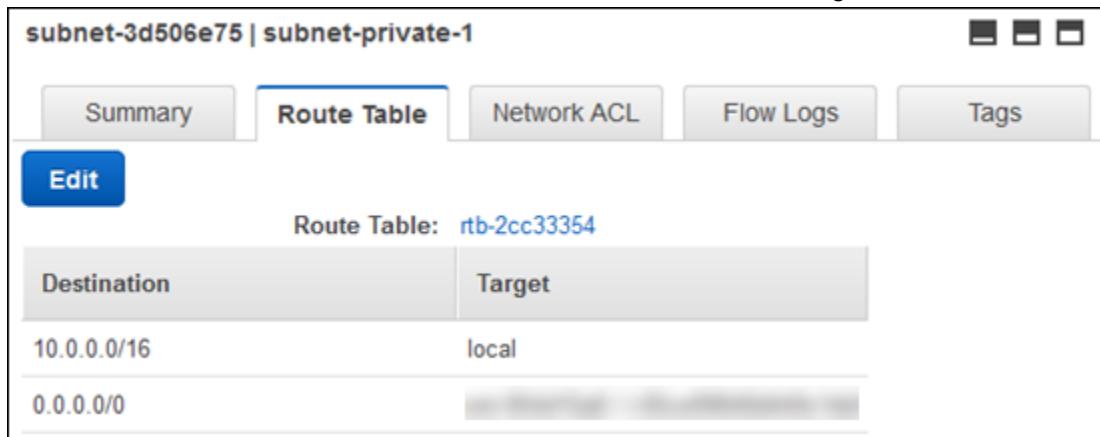
Option	Value
Name tag	The name for your second private subnet, for example subnet-private-2 .
VPC	Your VPC, for example vpc-1 .
Availability Zone	An Availability Zone for your AWS Region. Note Choose an Availability Zone different from the one that you chose for the first private subnet.
CIDR block	10.0.2.0/24

5. Choose **Yes, Create**.

Both private subnets must use the same route table. In the following procedure, you check to make sure the route tables match, and if not you edit one of them.

To ensure the subnets use the same route table.

1. Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the upper-right corner of the AWS Management Console, confirm that you are in the correct AWS Region for your VPC.
3. In the upper-left corner, choose **VPC Dashboard**, choose **Subnets**, and then choose your first private subnet, for example **subnet-private-1**.
4. At the bottom of the console, choose the **Route Table** tab, shown following.



5. Make a note of the route table, for example **rtb-0d9fc668**.
6. In the list of subnets, choose the second private subnet, for example **subnet-private-2**.
7. At the bottom of the console, choose the **Route Table** tab.
8. If the route table for the second subnet is not the same as the route table for the first subnet, edit it to match:
 - a. Choose **Edit**.
 - b. For **Change to**, choose the route table that matches your first subnet.
 - c. Choose **Save**.

A security group acts as a virtual firewall for your DB instance to control inbound and outbound traffic. In the following procedure, you create a security group for your DB instance. For more information about security groups, see [Security groups for your VPC](#).

To create a VPC security group for a private Amazon RDS DB instance

1. Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the upper-right corner of the AWS Management Console, confirm that you are in the correct AWS Region for your VPC.
3. In the upper-left corner, choose **VPC Dashboard**, choose **Security Groups**, and then choose **Create Security Group**.
4. On the page **Create Security Group**, set the following values.

Option	Value
Name tag	The name for your security group, for example sgdb-1 .
Group name	The name for your security group, for example sgdb-1 .
Description	A description for your security group.
VPC	Your VPC, for example vpc-1 .

5. Choose **Yes, Create**.

In the following procedure, you add rules to your security group to control inbound traffic to your DB instance. For more information about inbound rules, see [Security group rules](#).

To add inbound rules to the security group

1. Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the upper-right corner of the AWS Management Console, confirm that you are in the correct AWS Region for your VPC.
3. In the upper-left corner, choose **VPC Dashboard**, choose **Security Groups**, and then choose your security group, for example **sgdb-1**.
4. At the bottom of the console, choose the **Inbound Rules** tab, and then choose **Edit**.
5. Set these values, as shown following.

Option	Value
Type	Oracle (1521)
Protocol	TCP (6)
Port Range	1521
Source	The identifier of your security group. When you choose the box, you see the name of your security group, for example sgdb-1 .



6. Choose **Save**.

Creating an Oracle DB instance

You can use Amazon RDS to host an Oracle DB instance. When you create the new DB instance, specify the VPC and security group you created previously using the instructions in [Creating a VPC for use with an Oracle database \(p. 1219\)](#). Also, choose **No** for **Publicly accessible**.

For information about creating a DB instance, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

Additional Amazon RDS interfaces

In the preceding procedures, we use the AWS Management Console to perform tasks. Amazon Web Services also provides the AWS Command Line Interface (AWS CLI), and an application programming interface (API). You can use the AWS CLI or the API to automate many of the tasks for managing Amazon RDS, including tasks to manage an Oracle DB instance with Amazon RDS.

For more information, see [AWS Command Line Interface reference for Amazon RDS](#) and [Amazon RDS API Reference](#).

Using Oracle GoldenGate with Amazon RDS

Oracle GoldenGate (GoldenGate) collects, replicates, and manages transactional data between databases. It is a log-based change data capture (CDC) and replication software package used with Oracle databases for online transaction processing (OLTP) systems. GoldenGate creates trail files that contain the most recent changed data from the source database and then pushes these files to the target database. Amazon RDS supports GoldenGate for Oracle Database Standard Edition Two (SE2) and Enterprise Edition (EE).

You can use GoldenGate with Amazon RDS to do the following:

- Active-Active database replication
- Zero-downtime migration and upgrades
- Disaster recovery
- Data protection
- In-region and cross-region replication

When working with GoldenGate on Amazon RDS, consider the following:

- You are responsible for setting up and managing GoldenGate for use with Amazon RDS.
- You are responsible for managing GoldenGate licensing (BYOL) for use with Amazon RDS in all AWS regions. For more information, see [Oracle licensing options \(p. 990\)](#).
- Amazon RDS supports GoldenGate version 11.2.1 and later.
- Amazon RDS supports migration and replication across Oracle databases using GoldenGate. We do not support nor prevent customers from migrating or replicating across heterogeneous databases.
- You can use GoldenGate on Amazon RDS for Oracle DB instances that use Oracle Transparent Data Encryption (TDE). To maintain the integrity of replicated data, you should configure encryption on the GoldenGate hub using EBS encrypted volumes or trail file encryption. You should also configure encryption for data sent between the GoldenGate hub and the source and target database instances. Amazon RDS for Oracle DB instances support encryption with [Oracle Secure Sockets Layer \(p. 1182\)](#) or [Oracle native network encryption \(p. 1176\)](#).
- GoldenGate DDL is supported with GoldenGate version 12.1 and later when using Integrated capture mode.

Overview

The GoldenGate architecture for use with Amazon RDS consists of three decoupled modules. The source database can be either an on-premises Oracle database, an Oracle database on an Amazon EC2 instance, or an Oracle database on an Amazon RDS DB instance. You also work with a GoldenGate hub, which moves transaction information from the source database to the target database. Your hub can be either of these:

- An Amazon EC2 instance with Oracle Database and GoldenGate installed
- An on-premises Oracle installation.

You can have more than one Amazon EC2 hub, and we recommend that you use two hubs if you are using GoldenGate for cross-region replication.

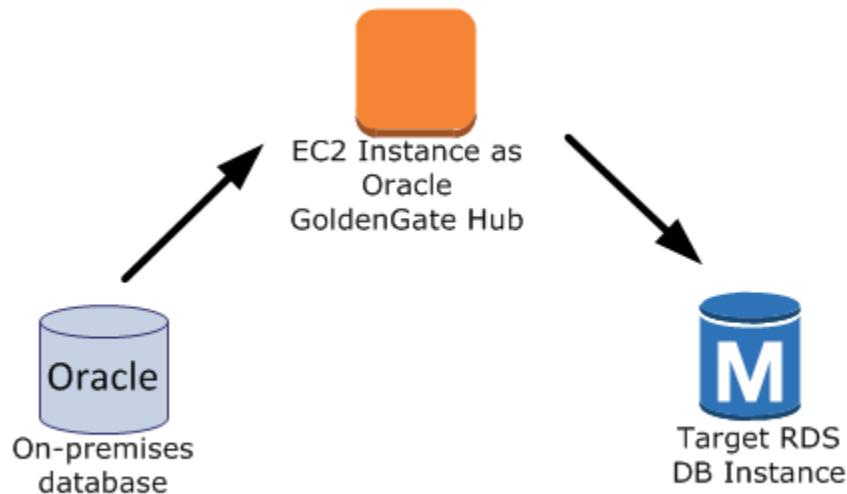
Your target database can be on either an Amazon RDS DB instance, an Amazon EC2 instance, or an on-premises location.

GoldenGate on Amazon RDS supports the following common scenarios:

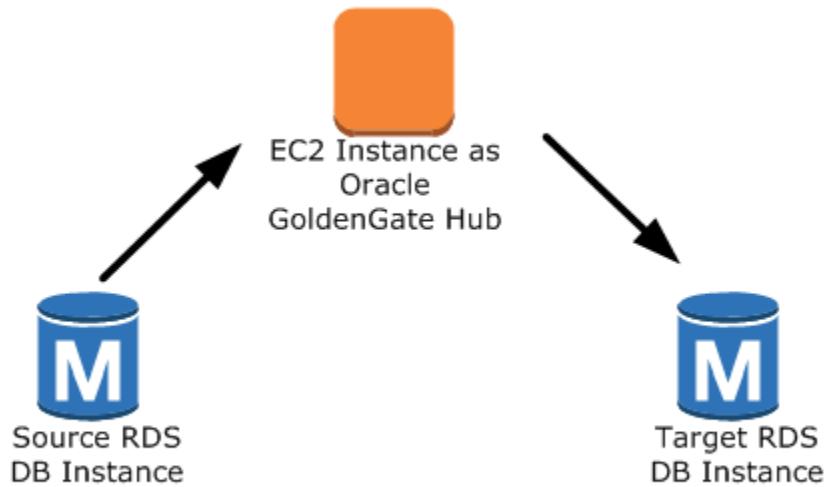
Scenario 1: An on-premises Oracle source database and on-premises GoldenGate hub, that provides data to a target Amazon RDS DB instance.



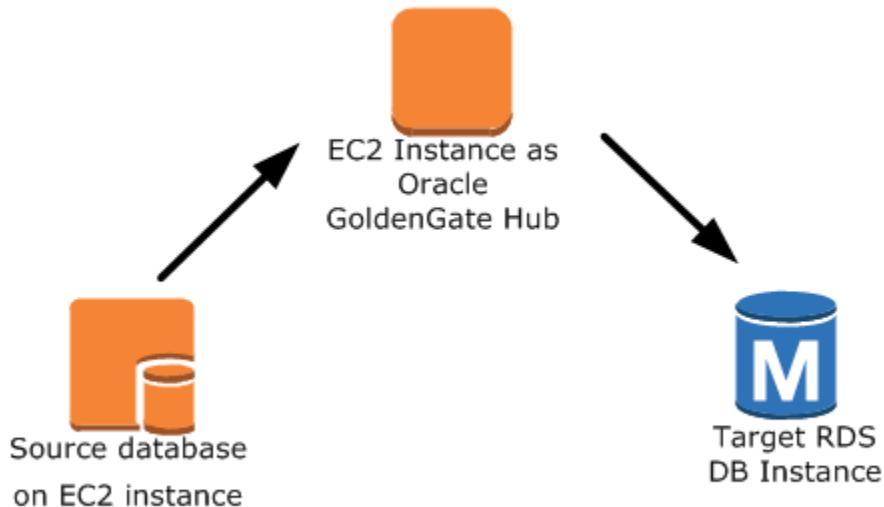
Scenario 2: An on-premises Oracle database that acts as the source database, connected to an Amazon EC2 instance hub that provides data to a target Amazon RDS DB instance.



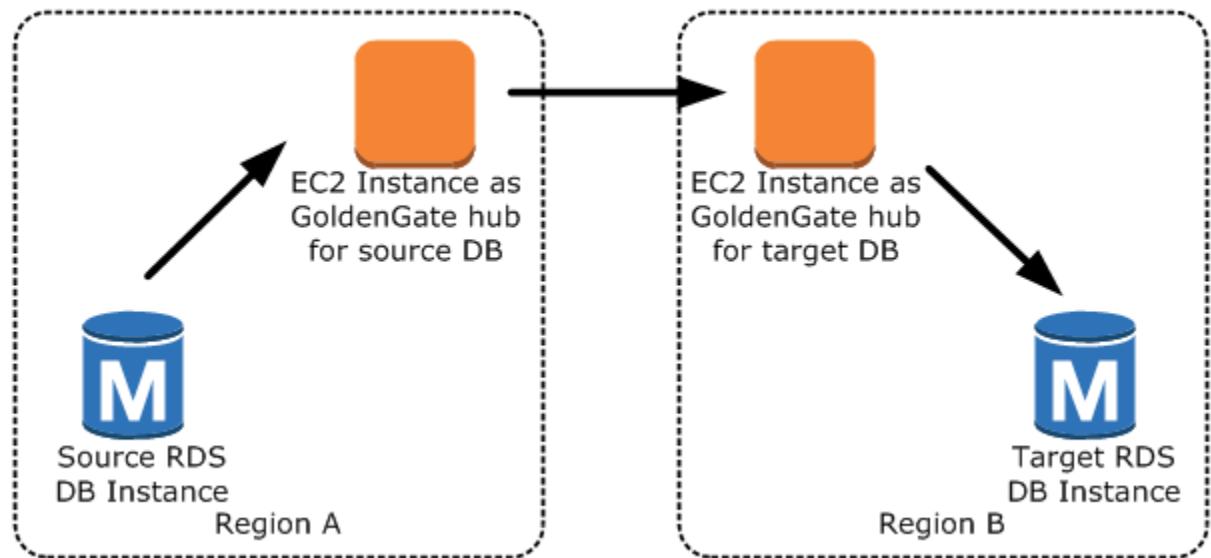
Scenario 3: An Oracle database on an Amazon RDS DB instance that acts as the source database, connected to an Amazon EC2 instance hub that provides data to a target Amazon RDS DB instance.



Scenario 4: An Oracle database on an Amazon EC2 instance that acts as the source database, connected to an Amazon EC2 instance hub that provides data to a target Amazon RDS DB instance.



Scenario 5: An Oracle database on an Amazon RDS DB instance connected to an Amazon EC2 instance hub in the same AWS Region. In this scenario, the hub is connected to an Amazon EC2 instance hub in a different AWS Region. This second hub provides data to the target Amazon RDS DB instance in the same AWS Region as the second Amazon EC2 instance hub.



Note

Any issues that affect running GoldenGate on an on-premises environment also affect running GoldenGate on AWS. We strongly recommend that you monitor the GoldenGate hub to ensure that EXTRACT and REPLICAT are resumed if a failover occurs. Because the GoldenGate hub is run on an Amazon EC2 instance, Amazon RDS does not manage the GoldenGate hub and cannot ensure that it is running.

You can use GoldenGate using Amazon RDS to upgrade to major versions of Oracle. For example, you can use GoldenGate with Amazon RDS to upgrade from an Oracle version 8 on-premises database to Oracle Database 19c on an Amazon RDS DB instance.

To set up GoldenGate using Amazon RDS, you configure the hub on the Amazon EC2 instance, and then configure the source and target databases. The following steps show how to set up GoldenGate for use with Amazon RDS. Each step is explained in detail in the following sections:

- Setting up a GoldenGate hub on Amazon EC2 (p. 1228)
- Setting up a source database for use with GoldenGate on Amazon RDS (p. 1229)
- Setting up a target database for use with GoldenGate on Amazon RDS (p. 1231)
- Working with the EXTRACT and REPLICAT utilities of GoldenGate (p. 1233)

Setting up a GoldenGate hub on Amazon EC2

To create a GoldenGate hub on an Amazon EC2 instance, you complete several steps. First, you create an Amazon EC2 instance with a full client installation of Oracle RDBMS. The Amazon EC2 instance must also have Oracle GoldenGate software installed. The exact software versions depend on the source and target database versions. For more information about installing GoldenGate, see the [Oracle documentation](#).

The Amazon EC2 instance that serves as the GoldenGate hub stores and processes the transaction information from the source database into trail files. To support this process, make sure that you meet the following conditions:

- You have allocated enough storage for the trail files
- The Amazon EC2 instance has enough processing power to manage the amount of data.

- The EC2 instance has enough memory to store the transaction information before it's written to the trail file.

The following tasks set up a GoldenGate hub on an Amazon EC2 instance; each task is explained in detail in this section:

1. Create the GoldenGate subdirectories.
2. Update the GLOBALS parameter file.
3. Configure the *mgr.prm* file and start the manager.

Create subdirectories in the GoldenGate directory using the Amazon EC2 command line shell and *ggsci*, the GoldenGate command interpreter. The subdirectories are created under the *gg* directory and include directories for parameter, report, and checkpoint files.

```
prompt$ cd /gg
prompt$ ./ggsci
GGSCI> CREATE SUBDIRS
```

Create a GLOBALS parameter file using the Amazon EC2 command line shell. Parameters that affect all GoldenGate processes are defined in the GLOBALS parameter file. The following example creates the necessary file:

```
$ cd $GGHOME
$ vi GLOBALS
CheckpointTable oggadm1.oggchkpt
```

The last step in setting up and configuring the GoldenGate hub is to configure the *manager*. Add the following lines to the *mgr.prm* file, then start the *manager* using *ggsci*:

```
PORt 8199
PurgeOldExtracts ./dirdat/*, UseCheckpoints, MINKEEPDAYS 5
```

```
GGSCI> start mgr
```

Once you have completed these steps, the GoldenGate hub is ready for use. Next, you set up the source and target databases.

Setting up a source database for use with GoldenGate on Amazon RDS

When your source database is running Oracle Database 12c or later, complete the following tasks to set up a source database for use with GoldenGate:

1. Set the `ENABLE_GOLDENGATE_REPLICATION` parameter to *True*. This parameter turns on supplemental logging for the source database. If your source database is on an Amazon RDS DB instance, make sure that you have a parameter group assigned to the DB instance with the `ENABLE_GOLDENGATE_REPLICATION` parameter set to *true*. For more information about the `ENABLE_GOLDENGATE_REPLICATION` parameter, see the [Oracle documentation](#).
2. Set the retention period for archived redo logs for the GoldenGate source database.
3. Create a GoldenGate user account on the source database.
4. Grant the necessary privileges to the GoldenGate user.
5. Add a TNS alias for the source database to the `tnsnames.ora` file on the GoldenGate hub.

Enable supplemental logging on the source DB

The `ENABLE_GOLDENGATE_REPLICATION` parameter, when set to *True*, turns on supplemental logging for the source database and configures the required GoldenGate permissions. If your source database is on an Amazon RDS DB instance, make sure that you have a parameter group assigned to the DB instance with `ENABLE_GOLDENGATE_REPLICATION` set to *true*. For more information about `ENABLE_GOLDENGATE_REPLICATION`, see the [Oracle documentation](#).

Set the log retention period on the source DB

The source database must also retain archived redo logs. For example, set the retention period for archived redo logs to 24 hours.

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours', 24);
```

Specify the duration for log retention in hours. The duration should exceed any potential downtime of the source instance, any potential period of communication, and any potential period of networking issues for the source instance. Such a duration lets Oracle GoldenGate recover logs from the source instance as needed.

The absolute minimum value required is one hour of logs retained. If you don't have log retention enabled, or if the retention value is too small, you receive the following message.

```
2014-03-06 06:17:27  ERROR    OGG-00446  error 2 (No such file or directory)
opening redo log /rdsdbdata/db/GGTEST3_A/onlinelog/o1_mf_2_9k4bp1n6_.log
for sequence 1306Not able to establish initial position for begin time 2014-03-06
06:16:55.
```

The logs are retained on your DB instance. Ensure that you have sufficient storage on your instance for the files. To see how much space you have used in the last X hours, use the following query, replacing X with the number of hours.

```
SELECT SUM(BLOCKS * BLOCK_SIZE) BYTES FROM V$ARCHIVED_LOG
WHERE NEXT_TIME>=SYSDATE-X/24 AND DEST_ID=1;
```

Create a user account on the source

GoldenGate runs as a database user and requires the appropriate database privileges to access the redo and archive logs for the source database. To provide these, create a GoldenGate user account on the source database. For more information about the permissions for a GoldenGate user account, see the sections 4, section 4.4, and table 4.1 in the [Oracle documentation](#).

The following statements create a user account named `oggadm1`.

```
CREATE TABLESPACE administrator;
CREATE USER oggadm1 IDENTIFIED BY "password"
DEFAULT TABLESPACE ADMINISTRATOR TEMPORARY TABLESPACE TEMP;
```

Grant account privileges on the source DB

Grant the necessary privileges to the GoldenGate user account using the SQL command `grant` and the `rdsadmin.rdsadmin_util` procedure `grant_sys_object`. The following statements grant privileges to a user named `oggadm1`.

```
GRANT CREATE SESSION, ALTER SESSION TO oggadm1;
GRANT RESOURCE TO oggadm1;
```

```
GRANT SELECT ANY DICTIONARY TO oggadm1;
GRANT FLASHBACK ANY TABLE TO oggadm1;
GRANT SELECT ANY TABLE TO oggadm1;
GRANT SELECT_CATALOG_ROLE TO rds_master_user_name WITH ADMIN OPTION;
exec rdsadmin.rdsadmin_util.grant_sys_object ('DBA_CLUSTERS', 'OGGADM1');
GRANT EXECUTE ON DBMS_FLASHBACK TO oggadm1;
GRANT SELECT ON SYS.V_$DATABASE TO oggadm1;
GRANT ALTER ANY TABLE TO oggadm1;
```

Finally, grant the privileges needed by a user account to be a GoldenGate administrator. The package that you use to perform the grant, dbms_goldengate_auth or rdsadmin_dbms_goldengate_auth, depends on the Oracle DB engine version.

- For Oracle DB versions that are *earlier than* Oracle Database 12c Release 2 (12.2), run the following PL/SQL program.

```
exec dbms_goldengate_auth.grant_admin_privilege (grantee=>'OGGADM1',
    privilege_type=>'capture',
    grant_select_privileges=>true,
    do_grants=>TRUE);
```

- For Oracle DB versions that are *later than or equal to* Oracle Database 12c Release 2 (12.2), which requires patch level 12.2.0.1.ru-2019-04.rur-2019-04.r1 or later, run the following PL/SQL program.

```
exec rdsadmin.rdsadmin_dbms_goldengate_auth.grant_admin_privilege (grantee=>'OGGADM1',
    privilege_type=>'capture',
    grant_select_privileges=>true,
    do_grants=>TRUE);
```

To revoke privileges, use the procedure `revoke_admin_privilege` in the same package.

Add a TNS alias for the source DB

Add the following entry to `$ORACLE_HOME/network/admin/tnsnames.ora` in the Oracle Home to be used by the EXTRACT process. For more information on the `tnsnames.ora` file, see the [Oracle documentation](#).

```
OGGSOURCE=
  (DESCRIPTION=
    (ENABLE=BROKEN)
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-source.abcdef12345.us-
west-2.rds.amazonaws.com)(PORT=8200)))
      (CONNECT_DATA=(SID=ORCL))
    )
```

Setting up a target database for use with GoldenGate on Amazon RDS

The following tasks set up a target DB instance for use with GoldenGate:

1. Set the `ENABLE_GOLDENGATE_REPLICATION` parameter to `TRUE`. If your target database is on an Amazon RDS DB instance, make sure that you have a parameter group assigned to the DB instance with the `ENABLE_GOLDENGATE_REPLICATION` parameter set to `TRUE`. For more information about the `ENABLE_GOLDENGATE_REPLICATION` parameter, see the [Oracle documentation](#).
2. Create and manage a GoldenGate user account on the target database

3. Grant the necessary privileges to the GoldenGate user
4. Add a TNS alias for the target database to tnsnames.ora on the GoldenGate hub.

Create a user account on the target DB

GoldenGate runs as a database user and requires the appropriate database privileges. To make sure it has these, create a GoldenGate user account on the target database.

The following statements create a user named *oggadm1*:

```
CREATE TABLESPACE administrator;
CREATE TABLESPACE administrator_idx;
CREATE USER oggadm1 IDENTIFIED BY "password"
    DEFAULT TABLESPACE administrator
    TEMPORARY TABLESPACE temp;
ALTER USER oggadm1 QUOTA UNLIMITED ON administrator;
ALTER USER oggadm1 QUOTA UNLIMITED ON administrator_idx;
```

Grant account privileges on the target DB

Grant necessary privileges to the GoldenGate user account on the target DB. In the following example, you grant privileges to *oggadm1*.

```
GRANT CREATE SESSION      TO oggadm1;
GRANT ALTER SESSION       TO oggadm1;
GRANT CREATE CLUSTER     TO oggadm1;
GRANT CREATE INDEXTYPE   TO oggadm1;
GRANT CREATE OPERATOR    TO oggadm1;
GRANT CREATE PROCEDURE   TO oggadm1;
GRANT CREATE SEQUENCE    TO oggadm1;
GRANT CREATE TABLE        TO oggadm1;
GRANT CREATE TRIGGER     TO oggadm1;
GRANT CREATE TYPE         TO oggadm1;
GRANT SELECT ANY DICTIONARY TO oggadm1;
GRANT CREATE ANY TABLE   TO oggadm1;
GRANT ALTER ANY TABLE    TO oggadm1;
GRANT LOCK ANY TABLE     TO oggadm1;
GRANT SELECT ANY TABLE   TO oggadm1;
GRANT INSERT ANY TABLE   TO oggadm1;
GRANT UPDATE ANY TABLE   TO oggadm1;
GRANT DELETE ANY TABLE   TO oggadm1;
```

Finally, grant the privileges needed by a user account to be a GoldenGate administrator. The package that you use to perform the grant, dbms_goldengate_auth or rdsadmin_dbms_goldengate_auth, depends on the Oracle DB engine version.

- For Oracle DB versions that are *earlier than* Oracle Database 12c Release 2 (12.2), run the following PL/SQL program.

```
exec dbms_goldengate_auth.grant_admin_privilege (grantee=>'OGGADM1',
    privilege_type=>'capture',
    grant_select_privileges=>true,
    do_grants=>TRUE);
```

- For Oracle DB versions that are *later than or equal to* Oracle Database 12c Release 2 (12.2), which requires patch level 12.2.0.1.ru-2019-04.rur-2019-04.r1 or later, run the following PL/SQL program.

```
exec rdsadmin.rdsadmin_dbms_goldengate_auth.grant_admin_privilege (grantee=>'OGGADM1',
```

```
privilege_type=>'capture',
grant_select_privileges=>true,
do_grants=>TRUE);
```

To revoke privileges, use the procedure `revoke_admin_privilege` in the same package.

Add a TNS alias for the target DB

Add the following entry to `$ORACLE_HOME/network/admin/tnsnames.ora` in the Oracle Home to be used by the `REPLICAT` process. For more information on the `tnsname.ora` file, see the [Oracle documentation](#).

```
OGGTARGET=
  (DESCRIPTION=
    (ENABLE=BROKEN)
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-target.abcdef12345.us-west-2.rds.amazonaws.com)(PORT=8200)))
    (CONNECT_DATA=(SID=ORCL))
  )
```

Working with the EXTRACT and REPLICAT utilities of GoldenGate

The GoldenGate utilities `EXTRACT` and `REPLICAT` work together to keep the source and target databases in sync via incremental transaction replication using trail files. All changes that occur on the source database are automatically detected by `EXTRACT`, then formatted and transferred to trail files on the GoldenGate on-premises or EC2-instance hub. After initial load is completed, the data is read from these files and replicated to the target database by the `REPLICAT` utility.

Running the GoldenGate EXTRACT utility

The `EXTRACT` utility retrieves, converts, and outputs data from the source database to trail files. `EXTRACT` queues transaction details to memory or to temporary disk storage. When the transaction is committed to the source database, `EXTRACT` flushes all of the transaction details to a trail file. The trail file routes these details to the GoldenGate on-premises or the EC2 instance hub and then to the target database.

The following tasks enable and start the `EXTRACT` utility:

1. Configure the `EXTRACT` parameter file on the GoldenGate hub (on-premises or EC2 instance). The following listing shows an example `EXTRACT` parameter file.

```
EXTRACT EABC
SETENV (ORACLE_SID=ORCL)
SETENV (NLSLANG=AL32UTF8)

USERID oggadm1@OGGSOURCE, PASSWORD XXXXXX
EXTTRAIL /path/to/goldengate/dirdat/ab

IGNOREREPLICATES
GETAPLOPS
TRANLOGOPTIONS EXCLUDEUSER OGGADM1

TABLE EXAMPLE.TABLE;
```

2. On the GoldenGate hub, launch the GoldenGate command line interface (*ggsci*). Log into the source database. The following example shows the format for logging in:

```
dblogin userid <user>@<db tnsname>
```

3. Add a checkpoint table for the database:

```
add checkpointtable
```

4. Add transdata to turn on supplemental logging for the database table:

```
add transdata <user>.<table>
```

Alternatively, you can add transdata to turn on supplemental logging for all tables in the database:

```
add transdata <user>.*
```

5. Using the *ggsci* command line, enable the **EXTRACT** utility using the following commands:

```
add extract <extract name> tranlog, INTEGRATED tranlog, begin now
add exttrail <path-to-trail-from-the param-file>
    extract <extractname-from-paramfile>,
    MEGABYTES Xm
```

6. Register the **EXTRACT** utility with the database so that the archive logs are not deleted. This allows you to recover old, uncommitted transactions if necessary. To register the **EXTRACT** utility with the database, use the following command:

```
register EXTRACT <extract process name>, DATABASE
```

7. To start the **EXTRACT** utility, use the following command:

```
start <extract process name>
```

Running the GoldenGate REPLICAT utility

The **REPLICAT** utility is used to "push" transaction information in the trail files to the target database.

The following tasks enable and start the **REPLICAT** utility:

1. Configure the **REPLICAT** parameter file on the GoldenGate hub (on-premises or EC2 instance). The following listing shows an example **REPLICAT** parameter file.

```
REPLICAT RABC
SETENV (ORACLE_SID=ORCL)
SETENV (NLSLANG=AL32UTF8)

USERID oggadm1@OGGTARGET, password XXXXXX

ASSUMETARGETDEFS
MAP EXAMPLE.TABLE, TARGET EXAMPLE.TABLE;
```

2. Launch the GoldenGate command line interface (*ggsci*). Log into the target database. The following example shows the format for logging in.

```
dblogin userid <user>@<db tnsname>
```

3. Using the *ggsci* command line, add a checkpoint table. The user indicated should be the GoldenGate user account, not the target table schema owner. The following example creates a checkpoint table named *gg_checkpoint*.

```
add checkpointtable <user>.gg_checkpoint
```

4. To enable the REPLICAT utility, use the following command.

```
add replicat <replicat name> EXTTRAIL <extract trail file> CHECKPOINTTABLE  
<user>.gg_checkpoint
```

5. To start the REPLICAT utility, use the following command.

```
start <replicat name>
```

Troubleshooting issues when using GoldenGate with Amazon RDS

This section explains the most common issues when using Oracle GoldenGate with Amazon RDS.

Topics

- [Log retention \(p. 1235\)](#)
- [GoldenGate appears to be properly configured but replication is not working \(p. 1235\)](#)
- [Integrated REPLICAT slow due to query on sys."_DBA_APPLY_CDR_INFO" \(p. 1236\)](#)

Log retention

To work with Oracle GoldenGate with Amazon RDS, make sure that you have log retention enabled.

Specify the duration for log retention in hours. The duration should exceed any potential downtime of the source instance, any potential period of communication, and any potential period of networking issues for the source instance. Such a duration lets Oracle GoldenGate recover logs from the source instance as needed.

The absolute minimum value required is one hour of logs retained. If you don't have log retention enabled, or if the retention value is too small, you receive the following message.

```
2014-03-06 06:17:27  ERROR    OGG-00446  error 2 (No such file or directory)  
opening redo log /rdsdbdata/db/GGTEST3_A/onlinelog/o1_mf_2_9k4bp1n6_.log  
for sequence 1306Not able to establish initial position for begin time 2014-03-06  
06:16:55.
```

GoldenGate appears to be properly configured but replication is not working

For pre-existing tables, GoldenGate must be told which SCN it should work from. Take the following steps to fix this issue:

1. Launch the GoldenGate command line interface (*ggsci*). Log into the source database. The following example shows the format for logging.

```
dblogin userid <user>@<db tnsname>
```

2. Using the *ggsci* command line, set up the start SCN for the EXTRACT process. The following example sets the SCN to 223274 for the extract.

```
ALTER EXTRACT <extract process name> SCN 223274
start <extract process name>
```

3. Log in to the target database. The following example shows the format for logging in.

```
dblogin userid <user>@<db tnsname>
```

4. Using the ggsci command line, set up the start SCN for the REPLICAT process. The following example sets the SCN to 223274 for the REPLICAT.

```
start <replicat process name> atcsn 223274
```

Integrated REPLICAT slow due to query on sys."_DBA_APPLY_CDR_INFO"

Oracle GoldenGate Conflict Detection and Resolution (CDR) provides basic conflict resolution routines. For example, CDR can resolve a unique conflict for an `INSERT` statement.

When CDR resolves a collision, it can insert records into the exception table `_DBA_APPLY_CDR_INFO` temporarily. Integrated REPLICAT deletes these records later. In a rare scenario, the integrated REPLICAT can process a large number of collisions, but a new integrated REPLICAT does not replace it. Instead of being removed, the existing rows in `_DBA_APPLY_CDR_INFO` are orphaned. Any new integrated REPLICAT processes slow down because they are querying orphaned rows in `_DBA_APPLY_CDR_INFO`.

To remove all rows from `_DBA_APPLY_CDR_INFO`, use the Amazon RDS procedure `rdsadmin_util.truncate_apply$_cdr_info`. This procedure is released as part of the October 2020 release and patch update. The procedure is available in the following database versions:

- [Version 19.0.0.0.ru-2020-10.rur-2020-10.r1 \(p. 1266\)](#)
- [Version 18.0.0.0.ru-2020-10.rur-2020-10.r1 \(p. 1298\)](#)
- [Version 12.2.0.1.ru-2020-10.rur-2020-10.r1 \(p. 1326\)](#)
- [Version 12.1.0.2.v22 \(p. 1364\)](#)

The following example truncates the table `_DBA_APPLY_CDR_INFO`.

```
SET SERVEROUTPUT ON SIZE 2000
EXEC rdsadmin.rdsadmin_util.truncate_apply$_cdr_info;
```

Using the Oracle Repository Creation Utility on Amazon RDS for Oracle

You can use Amazon RDS to host an Oracle DB instance that holds the schemas to support your Fusion Middleware components. Before you can use Fusion Middleware components, you must create and populate schemas for them in your database. You create and populate the schemas by using the Oracle Repository Creation Utility (RCU).

You can store the schemas for any Fusion Middleware components in your Amazon RDS DB instance. The following is a list of schemas that have been verified to install correctly:

- Analytics (ACTIVITIES)
- Audit Services (IAU)
- Audit Services Append (IAU_APPEND)
- Audit Services Viewer (IAU_VIEWER)
- Discussions (DISCUSSIONS)
- Metadata Services (MDS)
- Oracle Business Intelligence (BIPLATFORM)
- Oracle Platform Security Services (OPSS)
- Portal and Services (WEBCENTER)
- Portlet Producers (PORTLET)
- Service Table (STB)
- SOA Infrastructure (SOAINFRA)
- User Messaging Service (UCSUMS)
- WebLogic Services (WLS)

Licensing and versions

Amazon RDS supports Oracle Repository Creation Utility (RCU) version 12c only. You can use the RCU in the following configurations:

- RCU 12c with Oracle database 12.2.0.1
- RCU 12c with Oracle database 12.1.0.2.v4 or later

Before you can use RCU, you need a license for Oracle Fusion Middleware. You also need to follow the Oracle licensing guidelines for the Oracle database that hosts the repository. For more information, see [Oracle fusion middleware licensing information user manual](#) in the Oracle documentation.

Fusion MiddleWare supports repositories on Oracle Database Enterprise Edition and Standard Editions (SE, SE One, or SE Two). Oracle recommends Enterprise Edition for production installations that require partitioning and installations that require online index rebuild.

Before you create your Oracle DB instance, confirm the Oracle database version that you need to support the components that you want to deploy. You can use the Certification Matrix to find the requirements for the Fusion Middleware components and versions you want to deploy. For more information, see [Oracle fusion middleware supported system configurations](#) in the Oracle documentation.

Amazon RDS supports Oracle database version upgrades as needed. For more information, see [Upgrading a DB instance engine version \(p. 271\)](#).

Before you begin

Before you begin, you need an Amazon VPC. Because your Amazon RDS DB instance needs to be available only to your Fusion Middleware components, and not to the public Internet, your Amazon RDS DB instance is hosted in a private subnet, providing greater security. For information about how to create an Amazon VPC for use with an Oracle DB instance, see [Creating a VPC for use with an Oracle database \(p. 1219\)](#).

Before you begin, you also need an Oracle DB instance. For information about how to create an Oracle DB instance for use with Fusion Middleware metadata, see [Creating an Oracle DB instance \(p. 1224\)](#).

Recommendations

The following are some recommendations for working with your DB instance in this scenario:

- We recommend that you use Multi-AZ for production workloads. For more information about working with multiple Availability Zones, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).
- For additional security, Oracle recommends that you use Transparent Data Encryption (TDE) to encrypt your data at rest. If you have an Enterprise Edition license that includes the Advanced Security Option, you can enable encryption at rest by using the TDE option. For more information, see [Oracle Transparent Data Encryption \(p. 1204\)](#).

Amazon RDS also provides an encryption at rest option for all database editions. For more information, see [Encrypting Amazon RDS resources \(p. 1630\)](#).

- Configure your VPC Security Groups to allow communication between your application servers and your Amazon RDS DB instance. The application servers that host the Fusion Middleware components can be on Amazon EC2 or on-premises.

Using the Oracle Repository Creation Utility

You use the Oracle Repository Creation Utility (RCU) to create and populate the schemas to support your Fusion Middleware components.

Running RCU using the command line in one step

If you don't need to edit any of your schemas before populating them, you can run RCU in a single step. Otherwise, see the following section for running RCU in multiple steps.

You can run the RCU in silent mode by using the command-line parameter `-silent`. When you run RCU in silent mode, you can avoid typing passwords on the command line by creating a text file containing the passwords. Create a text file with the password for `dbUser` on the first line, and the password for each component on subsequent lines. You specify the name of the password file as the last parameter to the RCU command.

Example

The following example creates and populates schemas for the SOA Infrastructure component (and its dependencies) in a single step.

For Linux, macOS, or Unix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-createRepository \
-connectString ${dbhost}:${dbport}:${dbname} \
```

```
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-f < /tmp/passwordfile.txt
```

For more information, see [Running Repository Creation Utility from the command line](#) in the Oracle documentation.

Running RCU using the command line in multiple steps

If you need to manually edit your schema scripts, you can run the RCU in multiple steps:

1. Run RCU in **Prepare Scripts for System Load** mode by using the `-generateScript` command-line parameter to create the scripts for your schemas.
2. Manually edit and run the generated script `script_systemLoad.sql`.
3. Run RCU again in **Perform Product Load** mode by using the `-dataLoad` command-line parameter to populate the schemas.
4. Run the generated clean-up script `script_postDataLoad.sql`.

You can run the RCU in silent mode by using the command-line parameter `-silent`. When you run RCU in silent mode, you can avoid typing passwords on the command line by creating a text file containing the passwords. Create a text file with the password for `dbUser` on the first line, and the password for each component on subsequent lines. You specify the name of the password file as the last parameter to the RCU command.

Example

The following example creates schema scripts for the SOA Infrastructure component (and its dependencies).

For Linux, macOS, or Unix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-generateScript \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
[-encryptTablespace true] \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
```

```
-component WLS \
-component SOAINFRA \
-scriptLocation /tmp/rcuscripts \
-f < /tmp/passwordfile.txt
```

Now you can edit the generated script, connect to your Oracle DB instance, and run the script. The generated script is named `script_systemLoad.sql`. For information about connecting to your Oracle DB instance, see [Connecting to your sample Oracle DB instance \(p. 97\)](#).

The following example populates the schemas for the SOA Infrastructure component (and its dependencies).

For Linux, macOS, or Unix:

```
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-dataLoad \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-f < /tmp/passwordfile.txt
```

To finish, you connect to your Oracle DB instance, and run the clean-up script. The script is named `script_postDataLoad.sql`.

For more information, see [Running Repository Creation Utility from the command line](#) in the Oracle documentation.

Running RCU in interactive mode

To use the RCU graphical user interface, you can run RCU in interactive mode. To run RCU in interactive mode, include the `-interactive` parameter and omit the `-silent` parameter. For more information, see [Understanding Repository Creation Utility screens](#) in the Oracle documentation.

Example

The following example starts RCU in interactive mode and pre-populates the connection information.

For Linux, macOS, or Unix:

```
export ORACLE_HOME=u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-interactive \
-createRepository \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal
```

Known issues

The following are some known issues for working with RCU, with some troubleshooting suggestions:

- Oracle Managed Files (OMF) — Amazon RDS uses OMF data files to simplify storage management. You can customize tablespace attributes, such as size and extent management. However, specifying a data file name when you run RCU causes tablespace code to fail with ORA-20900. The RCU can be used with OMF in the following ways:
 - In RCU 12.2.1.0 and later, use the `-honorOMF` command-line parameter.
 - In RCU 12.1.0.3 and later, use multiple steps and edit the generated script. For more information, see [Running RCU using the command line in multiple steps \(p. 1239\)](#).
- SYSDBA — Because Amazon RDS is a managed service, you don't have full SYSDBA access to your Oracle DB instance. However, RCU 12c supports users with lower privileges. In most cases, the master user privilege is sufficient to create repositories. In some cases, the RCU might fail with ORA-01031 when attempting to grant SYS object privileges. You can retry and run the `RDSADMIN_UTIL.GRANT_SYS_OBJECT()` stored procedure, or contact AWS Support.
- Dropping Enterprise Scheduler Service — When you use the RCU to drop an Enterprise Scheduler Service repository, the RCU might fail with `Error: Component drop check failed`.

Related topics

- [Oracle licensing options \(p. 990\)](#)

Installing a Siebel database on Oracle on Amazon RDS

You can use Amazon RDS to host a Siebel Database on an Oracle DB instance. The Siebel Database is part of the Siebel Customer Relationship Management (CRM) application architecture. For an illustration, see [Generic architecture of Siebel business application](#).

Use the following topic to help set up a Siebel Database on an Oracle DB instance on Amazon RDS. You can also find out how to use Amazon Web Services to support the other components required by the Siebel CRM application architecture.

Note

To install a Siebel Database on Oracle on Amazon RDS, you need to use the master user account. You don't need SYSDBA privilege; master user privilege is sufficient. For more information, see [Master user account privileges \(p. 1712\)](#).

Licensing and versions

To install a Siebel Database on Amazon RDS, you must use your own Oracle Database license, and your own Siebel license. You must have the appropriate Oracle Database license (with Software Update License and Support) for the DB instance class and Oracle Database edition. For more information, see [Oracle licensing options \(p. 990\)](#).

Oracle Database Enterprise Edition is the only edition certified by Siebel for this scenario. Amazon RDS supports Siebel CRM version 15.0 or 16.0. Use Oracle Database 12c Release 1 (12.1.0.2.0). For the procedures following, we use Siebel CRM version 15.0 and Oracle Database Release 1 (12.1.0.2) or Oracle Database Release 2 (12.2.0.1). For more information, see [Oracle Database 12c with Amazon RDS \(p. 981\)](#).

Amazon RDS supports database version upgrades. For more information, see [Upgrading a DB instance engine version \(p. 271\)](#).

Before you begin

Before you begin, you need an Amazon VPC. Because your Amazon RDS DB instance needs to be available only to your Siebel Enterprise Server, and not to the public Internet, your Amazon RDS DB instance is hosted in a private subnet, providing greater security. For information about how to create an Amazon VPC for use with Siebel CRM, see [Creating a VPC for use with an Oracle database \(p. 1219\)](#).

Before you begin, you also need an Oracle DB instance. For information about how to create an Oracle DB instance for use with Siebel CRM, see [Creating an Oracle DB instance \(p. 1224\)](#).

Installing and configuring a Siebel database

After you create your Oracle DB instance, you can install your Siebel Database. You install the database by creating table owner and administrator accounts, installing stored procedures and functions, and then running the Siebel Database Configuration Wizard. For more information, see [Installing the Siebel database on the RDBMS](#).

To run the Siebel Database Configuration Wizard, you need to use the master user account. You don't need SYSDBA privilege; master user privilege is sufficient. For more information, see [Master user account privileges \(p. 1712\)](#).

Using other Amazon RDS features with a Siebel database

After you create your Oracle DB instance, you can use additional Amazon RDS features to help you customize your Siebel Database.

Collecting statistics with the Oracle Statspack option

You can add features to your DB instance through the use of options in DB option groups. When you created your Oracle DB instance, you used the default DB option group. If you want to add features to your database, you can create a new option group for your DB instance.

If you want to collect performance statistics on your Siebel Database, you can add the Oracle Statspack feature. For more information, see [Oracle Statspack \(p. 1198\)](#).

Some option changes are applied immediately, and some option changes are applied during the next maintenance window for the DB instance. For more information, see [Working with option groups \(p. 212\)](#). After you create a customized option group, modify your DB instance to attach it. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Performance tuning with parameters

You manage your DB engine configuration through the use of parameters in a DB parameter group. When you created your Oracle DB instance, you used the default DB parameter group. If you want to customize your database configuration, you can create a new parameter group for your DB instance.

When you change a parameter, depending on the type of the parameter, the changes are applied either immediately or after you manually reboot the DB instance. For more information, see [Working with DB parameter groups \(p. 228\)](#). After you create a customized parameter group, modify your DB instance to attach it. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

To optimize your Oracle DB instance for Siebel CRM, you can customize certain parameters. The following table shows some recommended parameter settings. For more information about performance tuning Siebel CRM, see [Siebel CRM Performance Tuning Guide](#).

Parameter name	Default value	Guidance for optimal Siebel CRM performance
<code>_always_semi_join</code>	<code>CHOOSE</code>	OFF
<code>_b_tree_bitmap_p</code>	<code>TRUE</code>	FALSE
<code>_like_with_bind_</code>	<code>EQUALITY</code>	TRUE
<code>_no_or_expansion</code>	<code>FALSE</code>	FALSE
<code>_optimizer_join_</code>	<code>TRUE</code>	<code>sanity_check</code> TRUE
<code>_optimizer_max_parallelizations</code>	<code>2000</code>	100
<code>_optimizer_sortmerge_join_enabled</code>	<code>TRUE</code>	FALSE
<code>_partition_view_enabled</code>	<code>TRUE</code>	FALSE
<code>open_cursors</code>	300	At least 2000 .

Creating snapshots

After you create your Siebel Database, you can copy the database by using the snapshot features of Amazon RDS. For more information, see [Creating a DB snapshot \(p. 346\)](#) and [Restoring from a DB snapshot \(p. 349\)](#).

Support for other Siebel CRM components

In addition to your Siebel Database, you can also use Amazon Web Services to support the other components of your Siebel CRM application architecture. You can find more information about the support provided by Amazon AWS for additional Siebel CRM components in the following table.

Siebel CRM component	Amazon AWS Support
Siebel Enterprise (with one or more Siebel Servers)	You can host your Siebel Servers on Amazon Elastic Compute Cloud (Amazon EC2) instances. You can use Amazon EC2 to launch as many or as few virtual servers as you need. Using Amazon EC2, you can scale up or down easily to handle changes in requirements. For more information, see What is Amazon EC2? You can put your servers in the same VPC with your DB instance and use the VPC security group to access the database. For more information, see Working with a DB instance in a VPC (p. 1727) .
Web Servers (with Siebel Web Server Extensions)	You can install multiple Web Servers on multiple EC2 instances. You can then use Elastic Load Balancing to distribute incoming traffic among the instances. For more information, see What is Elastic Load Balancing?
Siebel Gateway Name Server	You can host your Siebel Gateway Name Server on an EC2 instance. You can then put your server in the same VPC with the DB instance and use the VPC security group to access the database. For more information, see Working with a DB instance in a VPC (p. 1727) .

Oracle database engine release notes

Updates to your Amazon RDS for Oracle DB instances keep them current. If you apply updates, you can be confident that your DB instance is running a version of the database software that has been tested by both Oracle and Amazon. We don't support applying one-off patches to individual DB instances.

You can specify any currently supported Oracle version when creating a new DB instance. You can specify the major version, such as Oracle Database 12c Release 1 (12.1), and any supported minor version for the specified major version. If no version is specified, Amazon RDS defaults to a supported version, typically the most recent version. If a major version is specified but a minor version is not, Amazon RDS defaults to a recent release of the major version that you have specified. To see a list of supported versions and defaults for newly created DB instances, use the [describe-db-engine-versions](#) AWS CLI command.

Oracle Database 19c (19.0.0), Oracle Database 18c (18.0.0), and Oracle Database 12c Release 2 (12.2.0.1)

For Amazon RDS for Oracle Database 19c (19.0.0.0), Oracle Database 18c (18.0.0.0), and Oracle Database 12c Release 2 (12.2.0.1), Amazon RDS incorporates bug fixes from Oracle by using Release Updates (RUs) and Release Updates Revisions (RURs). We don't support applying one-off patches to individual DB instances.

To find what RUs and RURs are applied to Amazon RDS for Oracle Database 19c (19.0.0), Oracle Database 18c (18.0.0.0), and Oracle Database 12c Release 2 (12.2.0.1), see the following table.

RU and RUR	Oracle Database 19c (19.0.0.0)	Oracle Database 18c (18.0.0.0)	Oracle Database 12c Release 2 (12.2.0.1)
2021 January	Version 19.0.0.0.ru-2021-01.rur-202108.010:0.(p-2021)01.rur-202101.11u(p2021)01.rur-2021-01 and Version 19.0.0.0.ru-2021-01.rur-2021-01.r1 (p. 1256)	Version 19.0.0.0.ru-2021-01.rur-202108.010:0.(p-2021)01.rur-202101.11u(p2021)01.rur-2021-01	Version 19.0.0.0.ru-2021-01.rur-202108.010:0.(p-2021)01.rur-202101.11u(p2021)01.rur-2021-01
2020 October	19.0.0.0.ru-2020-10.rur-202008.000:0.(p-2020)10.rur-202001.11u(p2020)0.rur-2020-10	19.0.0.0.ru-2020-10.rur-202008.000:0.(p-2020)10.rur-202001.11u(p2020)0.rur-2020-10	19.0.0.0.ru-2020-10.rur-202008.000:0.(p-2020)10.rur-202001.11u(p2020)0.rur-2020-10
2020 July	19.0.0.0.ru-2020-07.rur-202008.070:0.(p-2020)07.rur-202001.11u(p2020)07.rur-2020-07	19.0.0.0.ru-2020-07.rur-202008.070:0.(p-2020)07.rur-202001.11u(p2020)07.rur-2020-07	19.0.0.0.ru-2020-07.rur-202008.070:0.(p-2020)07.rur-202001.11u(p2020)07.rur-2020-07
2020 April	19.0.0.0.ru-2020-04.rur-202008.040:0.(p-2020)04.rur-202001.11u(p2020)04.rur-2020-04	19.0.0.0.ru-2020-04.rur-202008.040:0.(p-2020)04.rur-202001.11u(p2020)04.rur-2020-04	19.0.0.0.ru-2020-04.rur-202008.040:0.(p-2020)04.rur-202001.11u(p2020)04.rur-2020-04
2020 January	19.0.0.0.ru-2020-01.rur-202008.010:0.(p-2020)01.rur-202001.11u(p2020)01.rur-2020-01	19.0.0.0.ru-2020-01.rur-202008.010:0.(p-2020)01.rur-202001.11u(p2020)01.rur-2020-01	19.0.0.0.ru-2020-01.rur-202008.010:0.(p-2020)01.rur-202001.11u(p2020)01.rur-2020-01
2019 October	19.0.0.0.ru-2019-10.rur-201908.000:0.(p-2019)10.rur-201901.11u(p2019)0.rur-2019-10	19.0.0.0.ru-2019-10.rur-201908.000:0.(p-2019)10.rur-201901.11u(p2019)0.rur-2019-10	19.0.0.0.ru-2019-10.rur-201908.000:0.(p-2019)10.rur-201901.11u(p2019)0.rur-2019-10
2019 July	19.0.0.0.ru-2019-07.rur-201908.070:0.(p-2019)07.rur-201901.11u(p2019)07.rur-2019-07	19.0.0.0.ru-2019-07.rur-201908.070:0.(p-2019)07.rur-201901.11u(p2019)07.rur-2019-07	19.0.0.0.ru-2019-07.rur-201908.070:0.(p-2019)07.rur-201901.11u(p2019)07.rur-2019-07
2019 April	—	—	12.2.0.1.ru-2019-04.rur-2019-04
2019 January	—	—	12.2.0.1.ru-2019-01.rur-2019-01
2018 October	—	—	12.2.0.1.ru-2018-10.rur-2018-10

Oracle versions 12.1.0.2 and 11.2.0.4

For Amazon RDS for Oracle versions 12.1.0.2 and 11.2.0.4, Amazon RDS incorporates bug fixes from Oracle by using their quarterly Database Patch Set Updates (PSUs). If you apply updates, you can be confident that your DB instance is running a version of the database software that has been tested by both Oracle and Amazon. We don't support applying one-off patches to individual DB instances.

Note

RDS for Oracle Database 11g is deprecated. The 11.2.0.4 information in this section is only relevant when you want to upgrade an 11g snapshot.

To find what Oracle Patch Set Updates (PSUs) are applied to Amazon RDS for Oracle versions 12.1.0.2 and 11.2.0.4, see the following table.

PSU	Oracle Database 12c Release 1 (12.1.0.2)	Oracle Database 11g (11.2.0.4)
2021 January	12.1.0.2.v23 (p. 1360)	N/A
2020 October	12.1.0.2.v22 (p. 1364)	11.2.0.4.v26 (p. 1411)
2020 July	12.1.0.2.v21 (p. 1368)	11.2.0.4.v25 (p. 1414)
2020 April	12.1.0.2.v20 (p. 1372)	11.2.0.4.v24 (p. 1417)
2020 January	12.1.0.2.v19 (p. 1375)	11.2.0.4.v23 (p. 1420)
2019 October	12.1.0.2.v18 (p. 1379)	11.2.0.4.v22 (p. 1422)
2019 July	12.1.0.2.v17 (p. 1382)	11.2.0.4.v21 (p. 1424)
2019 April	12.1.0.2.v16 (p. 1384)	11.2.0.4.v20 (p. 1426)
2019 January	12.1.0.2.v15 (p. 1387)	11.2.0.4.v19 (p. 1428)
2018 October	12.1.0.2.v14 (p. 1389)	11.2.0.4.v18 (p. 1430)
2018 July	12.1.0.2.v13 (p. 1392)	11.2.0.4.v17 (p. 1432)
2018 April	12.1.0.2.v12 (p. 1394)	11.2.0.4.v16 (p. 1433)
2018 January	12.1.0.2.v11 (p. 1396)	11.2.0.4.v15 (p. 1435)
2017 October	12.1.0.2.v10 (p. 1398)	11.2.0.4.v14 (p. 1437)
2017 July	12.1.0.2.v9 (p. 1400)	11.2.0.4.v13 (p. 1438)
2017 April	12.1.0.2.v8 (p. 1401)	11.2.0.4.v12 (p. 1440)
2017 January	12.1.0.2.v7 (p. 1403)	11.2.0.4.v11 (p. 1441)
2016 October	12.1.0.2.v6 (p. 1405)	11.2.0.4.v10 (p. 1443)
2016 July	12.1.0.2.v5 (p. 1406)	11.2.0.4.v9 (p. 1444)
2016 April	12.1.0.2.v4 (p. 1407)	11.2.0.4.v8 (p. 1445)
2016 January	12.1.0.2.v3 (p. 1408)	11.2.0.4.v7 (p. 1447)
2015 October	12.1.0.2.v2 (p. 1409)	11.2.0.4.v6 (p. 1448) 11.2.0.4.v5 (p. 1448)

PSU	Oracle Database 12c Release 1 (12.1.0.2)	Oracle Database 11g (11.2.0.4)
2015 April	12.1.0.2.v1 (p. 1410)	11.2.0.4.v4 (p. 1449)
2014 October	—	11.2.0.4.v3 (p. 1451)
2014 July	—	11.2.0.4.v2 (p. 1451) (Deprecated)
2014 January	—	11.2.0.4.v1 (p. 1452)

Database engine: 19.0.0.0

The following versions are available for Oracle database engine 19.0.0.0:

- [Version 19.0.0.0.ru-2021-01.rur-2021-01.r2 \(p. 1247\)](#)
- [Version 19.0.0.0.ru-2021-01.rur-2021-01.r1 \(p. 1256\)](#)
- [Version 19.0.0.0.ru-2020-10.rur-2020-10.r1 \(p. 1266\)](#)
- [Version 19.0.0.0.ru-2020-07.rur-2020-07.r1 \(p. 1273\)](#)
- [Version 19.0.0.0.ru-2020-04.rur-2020-04.r1 \(p. 1279\)](#)
- [Version 19.0.0.0.ru-2020-01.rur-2020-01.r1 \(p. 1284\)](#)
- [Version 19.0.0.0.ru-2019-10.rur-2019-10.r1 \(p. 1287\)](#)
- [Version 19.0.0.0.ru-2019-07.rur-2019-07.r1 \(p. 1291\)](#)

Version 19.0.0.0.ru-2021-01.rur-2021-01.r2

Important

We recommend that you upgrade your DB instance to this version rather than to 19.0.0.0.ru-2021-01.rur-2021-01.r1. The application of the patch for minor engine upgrades to 19.0.0.0.ru-2021-01.rur-2021-01.r1 encountered an issue. Release update 19.10.0.0.210119 (32218454) didn't register correctly in the DBA_REGISTRY_SQLPATCH table.

Version 19.0.0.0.ru-2021-01.rur-2021-01.r2 includes the following:

- Patch 32218454: DATABASE RELEASE UPDATE 19.10.0.0.210119
- Patch 32067171: OJVM RELEASE UPDATE 19.10.0.0.210119
- Patch 31335037: RDBMS - DSTV35 UPDATE - TZDATA2020A
- Patch 31335142: DSTV35 UPDATE - TZDATA2020A - NEED OJVM FIX
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER TABLE
- Patch 29782284: ORA-06508:"MDSYS.MDPRVT_IDX" WHILE UPGRADING DATABASE TO 18.3
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 29374604: IE not starting against 18c Oracle RDBMS Standard Edition
- PreUpgrade Jar: preupgrade_19_cbuild_9_lf.zip
- Support for [Managing advisor tasks \(p. 1099\)](#) using procedures in the rdsadmin.rdsadmin_util package

Combined patches for version 19.0.0.0.ru-2021-01.rur-2021-01.r2, Released January 2021

Bugs fixed:

```
7391838, 8460502, 8476681, 14735102, 16664572, 17428816, 17468475
19080742, 19138896, 19697993, 20313356, 20867658, 21232786, 21374587
21528318, 21629064, 21639146, 21888352, 21965541, 22580355, 22729345
22748979, 23294761, 23296836, 23606241, 23645975, 23734075, 23763462
24596874, 24669730, 24687075, 24833686, 24957575, 24971456, 25030027
25092651, 25093917, 25404117, 25416731, 25560538, 25607406, 25756945
25792962, 25804387, 25806201, 25809128, 25883179, 25905368, 25986062
25997810, 26001677, 26127355, 26173091, 26284288, 26352615, 26440142
26476244, 26499997, 26611353, 26668264, 26716835, 26739322, 26777814
26819036, 26872233, 27004828, 27036163, 27044169, 27101798, 27126122
27126938, 27130348, 27166935, 27195575, 27195935, 27221350, 27222128
27244999, 27254335, 27260704, 27261477, 27359766, 27362994, 27369515
27378053, 27392968, 27406105, 27411022, 27423500, 27439716, 27453490
27458357, 27489107, 27572040, 27582210, 27589260, 27604329, 27622946
27629928, 27661222, 27666312, 27684864, 27692173, 27700413, 27710072
27729678, 27742354, 27745728, 27760043, 27801144, 27828892, 27829722
27846298, 27873364, 27877830, 27880025, 27929509, 27934711, 27935464
27941110, 27957203, 27967484, 27998559, 28007516, 28064977, 28072567
28078186, 28092783, 28104176, 28109326, 28125947, 28127569, 28129791
28133903, 28138847, 28144569, 28145995, 28181021, 28187837, 28189466
28204262, 28205555, 28209985, 28210681, 28263142, 28271258, 28271693
28276054, 28279456, 28294563, 28302580, 28313275, 28319114, 28322973
28326928, 28338211, 28350595, 28370061, 28371123, 28373960, 28375383
28379065, 28381939, 28386259, 28390273, 28395302, 28397317, 28402823
28403019, 28406374, 28410431, 28431445, 28435333, 28436414, 28442896
28454215, 28463226, 28470673, 28475242, 28482048, 28484299, 28489419
28492006, 28498976, 28502773, 28504631, 28513333, 28521330, 28530171
28534475, 28535127, 28537481, 28538439, 28541606, 28542455, 28546290
28547068, 28547926, 28558645, 28561704, 28564479, 28565296, 28567417
28567819, 28569897, 28572407, 28572533, 28572544, 28572667, 28572834
28578945, 28587723, 28589509, 28593682, 28594086, 28597221, 28601957
28602253, 28605066, 28606598, 28608211, 28612239, 28618343, 28620697
28621543, 28622202, 28625580, 28625862, 28627033, 28628592, 28630381
28632796, 28636532, 28639299, 28640772, 28642469, 28642899, 28643583
28643654, 28643718, 28644549, 28645570, 28646200, 28646939, 28649388
28655209, 28661333, 28663289, 28663782, 28672457, 28673945, 28681153
28689483, 28690694, 28692103, 28692275, 28694639, 28694872, 28696373
28697526, 28698087, 28699321, 28700945, 28703812, 28705231, 28707931
28708400, 28709063, 28710385, 28710469, 28710663, 28710734, 28714461
28715655, 28715727, 28718469, 28719348, 28720204, 28720418, 28721497
28722229, 28730079, 28730253, 28734355, 28740708, 28740799, 28742555
28745367, 28749853, 28751498, 28752923, 28755011, 28755084, 28755846
28758722, 28760206, 28763291, 28765983, 28767240, 28769456, 28771947
28772390, 28772816, 28774416, 28776431, 28776811, 28777214, 28778754
28781599, 28781754, 28785273, 28785321, 28785531, 28789531, 28791852
28793062, 28794230, 28795551, 28795734, 28800508, 28802734, 28804517
28805242, 28806517, 28808314, 28808652, 28808656, 28810381, 28811560
28813931, 28815123, 28815355, 28815557, 28816871, 28817449, 28818063
28819640, 28820669, 28821847, 28824482, 28827682, 28831971, 28833912
28835937, 28836716, 28837979, 28838385, 28844738, 28845346, 28846759
28847541, 28847572, 28849776, 28850084, 28852325, 28854004, 28854733
28855520, 28855922, 28857552, 28861861, 28862532, 28863263, 28863432
28863487, 28865569, 28867698, 28867992, 28870496, 28871040, 28872645
28872829, 28873575, 28874416, 28875089, 28876253, 28876639, 28876926
28877252, 28878865, 28881191, 28881848, 28882784, 28884931, 28887305
28888083, 28888327, 28889389, 28889730, 28892794, 28897123, 28897512
28899663, 28900506, 28901126, 28905390, 28905457, 28905615, 28907196
28910498, 28910586, 28911140, 28912691, 28915561, 28917080, 28918429
```

28919145, 28921844, 28922227, 28922532, 28922608, 28925250, 28925460
28925634, 28925880, 28927452, 28928462, 28932914, 28933158, 28935293
28935956, 28936114, 28937717, 28938422, 28938698, 28940179, 28940281
28940472, 28941901, 28942455, 28942694, 28945421, 28945994, 28946233
28948554, 28949888, 28950868, 28951332, 28951533, 28952168, 28954762
28955606, 28955883, 28956908, 28957260, 28957292, 28957723, 28958088
28959493, 28960863, 28962775, 28963036, 28965084, 28965095, 28965231
28965376, 28966444, 28968779, 28974083, 28974999, 28977322, 28980448
28981871, 28983095, 28983486, 28984313, 28985114, 28985272, 28985362
28985478, 28986207, 28986231, 28986257, 28986326, 28986481, 28986696
28988482, 28988864, 28989306, 28993295, 28993353, 28994307, 28994542
28995287, 28996376, 28999046, 29000000, 29001305, 29001888, 29002488
29002784, 29002927, 29003207, 29003407, 29003617, 29003738, 29006318
29006621, 29007321, 29007353, 29007775, 29008035, 29008669, 29009513
29010126, 29010517, 29011936, 29012609, 29013475, 29013832, 29014076
29015118, 29016294, 29017265, 29018655, 29018680, 29019121, 29020423
29021063, 29021352, 29022986, 29024028, 29024054, 29024448, 29024552
29024732, 29024876, 29026154, 29026582, 29026606, 29027456, 29027694
29027933, 29027940, 29030184, 29030927, 29031575, 29031600, 29032234
29032276, 29032457, 29032607, 29033052, 29033145, 29033200, 29033280
29034587, 29036278, 29037290, 29038528, 29038728, 29039089, 29039510
29040739, 29041739, 29041775, 29043554, 29043651, 29043725, 29044086
29044763, 29044954, 29046482, 29047127, 29047850, 29048178, 29048289
29048498, 29048605, 29048728, 29049673, 29050357, 29050560, 29050765
29050886, 29051263, 29051702, 29051953, 29052726, 29052850, 29053783
29053902, 29055644, 29056024, 29056270, 29056560, 29056767, 29056894
29058476, 29059011, 29060216, 29061016, 29061959, 29062692, 29062848
29062860, 29062868, 29110526, 29110783, 29110790, 29110797, 29110802
29110805, 29111598, 29111631, 29112455, 29113282, 29113305, 29115857
29117337, 29117526, 29117642, 29118543, 29119077, 29120223, 29122224
29122254, 29122367, 29123297, 29123432, 29123444, 29123482, 29124368
29125036, 29125374, 29125380, 29125708, 29125786, 29126345, 29127957
29128693, 29128935, 29129450, 29129476, 29129497, 29129691, 29129712
29130219, 29131539, 29131772, 29132456, 29132869, 29132938, 29133470
29134447, 29135383, 29135649, 29136111, 29138641, 29139070, 29139727
29139761, 29139956, 29141316, 29141341, 29141685, 29141886, 29142609
29142667, 29143516, 29144995, 29145214, 29145730, 29146077, 29146157
29146810, 29147849, 29148799, 29149170, 29149829, 29150338, 29151520
29152357, 29152603, 29152752, 29154631, 29154636, 29154725, 29154829
29155099, 29157051, 29157389, 29158680, 29158899, 29159216, 29159661
29159909, 29159936, 29160174, 29160462, 29161597, 29161923, 29162095
29163073, 29163156, 29163415, 29163437, 29163524, 29163567, 29164376
29165682, 29167111, 29167342, 29167374, 29167940, 29168137, 29168219
29168433, 29169073, 29169215, 29169540, 29169739, 29170232, 29170717
29171683, 29171942, 29172618, 29172826, 29173140, 29173373, 29173618
29173817, 29174004, 29174753, 29175638, 29176318, 29177466, 29177543
29177886, 29178385, 29179097, 29180313, 29180455, 29180559, 29180721
29180893, 29181078, 29181153, 29181231, 29181568, 29181620, 29181743
29181923, 29182019, 29182517, 29182901, 29182920, 29183298, 29183912
29184297, 29184666, 29185193, 29186091, 29186456, 29186605, 29188255
29189302, 29189307, 29189889, 29190235, 29190474, 29190663, 29190740
29191541, 29191827, 29192419, 29192468, 29192685, 29193207, 29194205
29194367, 29194493, 29194827, 29194981, 29195279, 29195337, 29195758
29196725, 29198092, 29198913, 29199163, 29199635, 29199733, 29200316
29200700, 29201143, 29201494, 29201539, 29201695, 29201787, 29202104
29202461, 29202850, 29203041, 29203122, 29203166, 29203227, 29203425
29203443, 29203604, 29205281, 29205323, 29205419, 29205463, 29205767
29205918, 29206109, 29206605, 29207073, 29208260, 29208732, 29209545
29210577, 29210610, 29210624, 29210683, 29211457, 29211724, 29212012
29212433, 29212611, 29213320, 29213351, 29213613, 29213641, 29213775
29213850, 29213879, 29213893, 29214561, 29214960, 29216312, 29216723
29216746, 29216984, 29217294, 29217472, 29217828, 29217848, 29217856
29218570, 29219205, 29219273, 29219627, 29220079, 29221248, 29221891
29221942, 29222031, 29222784, 29223833, 29223859, 29223967, 29224065
29224294, 29224605, 29224710, 29225076, 29225168, 29225758, 29225861
29227602, 29228869, 29229164, 29229754, 29229839, 29229844, 29229955

29230252, 29230565, 29231133, 29232117, 29232154, 29232449, 29232653
29233415, 29233810, 29233953, 29234123, 29235934, 29236573, 29237538
29237575, 29237744, 29240307, 29240668, 29240759, 29241345, 29241651
29242017, 29242884, 29242906, 29243749, 29243958, 29244495, 29244766
29244968, 29245063, 29245137, 29245160, 29246163, 29247183, 29247415
29247712, 29247906, 29248495, 29248552, 29248723, 29248835, 29248858
29249289, 29249412, 29249583, 29249991, 29250059, 29250317, 29251259
29251564, 29253184, 29253871, 29254031, 29254623, 29254930, 29255178
29255273, 29255431, 29255435, 29255616, 29255705, 29255718, 29255973
29256426, 29259119, 29259320, 29260224, 29260452, 29260956, 29261547
29261548, 29261695, 29261906, 29262512, 29262887, 29265448, 29266248
29266899, 29267292, 29268412, 29269171, 29269228, 29269825, 29270585
29271019, 29273168, 29273360, 29273539, 29273570, 29273735, 29273812
29273847, 29274428, 29274564, 29274627, 29275461, 29276272, 29277317
29278218, 29278684, 29279658, 29279751, 29279854, 29281112, 29281527
29281691, 29281796, 29282090, 29282233, 29282666, 29282898, 29285197
29285453, 29285503, 29285621, 29285788, 29285956, 29286037, 29286220
29286229, 29287130, 29287705, 29290110, 29290235, 29292232, 29292837
29293072, 29293574, 29293806, 29294753, 29296257, 29297863, 29297915
29298220, 29299049, 29299082, 29299830, 29299844, 29301463, 29301566
29302565, 29302614, 29302963, 29303918, 29304314, 29304692, 29304781
29304853, 29305093, 29306226, 29306713, 29307090, 29307109, 29307638
29309698, 29311336, 29311528, 29311588, 29311927, 29312310, 29312672
29312734, 29312753, 29312889, 29313347, 29313417, 29313525, 29314539
29314636, 29317756, 29318410, 29319441, 29320900, 29321489, 29321689
29323946, 29324568, 29324735, 29325087, 29325105, 29325257, 29325765
29325993, 29326233, 29327044, 29327892, 29329042, 29329087, 29329675
29329807, 29329848, 29330361, 29330791, 29331066, 29331209, 29331380
29331493, 29332292, 29332395, 29332763, 29332771, 29333500, 29336843
29336899, 29337294, 29337310, 29337742, 29338315, 29338348, 29338453
29338780, 29338913, 29339101, 29339155, 29339299, 29340333, 29341209
29342099, 29343086, 29343156, 29343861, 29344541, 29345937, 29346057
29346211, 29346943, 29347620, 29348176, 29348358, 29350052, 29350712
29350762, 29350868, 29351044, 29351386, 29351662, 29351716, 29351735
29351749, 29351771, 29352298, 29352724, 29352867, 29352947, 29353271
29353432, 29353718, 29353821, 29353960, 29355654, 29356547, 29356704
29356711, 29356752, 29356782, 29357821, 29358509, 29358828, 29360252
29360285, 29360467, 29360672, 29360775, 29360911, 29360950, 29361319
29361472, 29361801, 29362596, 29363151, 29364171, 29364177, 29366406
29366940, 29367019, 29367561, 29367971, 29368253, 29368310, 29368725
29372069, 29372541, 29372562, 29373418, 29373588, 29374179, 29374604
29375355, 29375941, 29375984, 29376346, 29377804, 29377986, 29378029
29378287, 29378834, 29378913, 29379299, 29379381, 29379750, 29379978
29380527, 29381000, 29382296, 29382641, 29382784, 29382815, 29383695
29384781, 29384854, 29384864, 29385339, 29385429, 29385652, 29386502
29386557, 29386635, 29386660, 29386835, 29387073, 29387274, 29387310
29387337, 29388020, 29388072, 29388094, 29388524, 29388830, 29389408
29389889, 29390011, 29390435, 29390785, 29391030, 29391237, 29391301
29391438, 29391849, 29391925, 29392554, 29392966, 29393291, 29393649
29394014, 29394140, 29394749, 29395657, 29396481, 29397841, 29397954
29397996, 29398488, 29398863, 29399046, 29399100, 29399121, 29399336
29399938, 29402110, 29402131, 29404483, 29405012, 29405462, 29405651
29405996, 29407488, 29407804, 29408853, 29409149, 29409455, 29410311
29410834, 29411037, 29411469, 29411931, 29412066, 29412269, 29413360
29413382, 29413517, 29413544, 29413634, 29413956, 29415493, 29416688
29416700, 29417084, 29417173, 29417719, 29417884, 29418165, 29418341
29420254, 29420834, 29421059, 29423003, 29423016, 29423156, 29423491
29423826, 29424999, 29426241, 29426320, 29428230, 29429017, 29429087
29429264, 29429466, 29429566, 29429895, 29430524, 29430866, 29431192
29431402, 29431485, 29432176, 29434301, 29434869, 29435474, 29435652
29436454, 29436514, 29436522, 29436727, 29437029, 29437379, 29437594
29437712, 29438150, 29438277, 29438736, 29439522, 29440651, 29441196
29442400, 29442936, 29443187, 29443250, 29443559, 29444072, 29444282
29444602, 29445548, 29446319, 29446669, 29448498, 29449477, 29449845
29449852, 29450162, 29450193, 29450273, 29450421, 29450812, 29450936
29451085, 29451386, 29452251, 29452576, 29452936, 29452953, 29454450

29454978, 29455424, 29455773, 29456538, 29456714, 29457312, 29457319
29457370, 29457502, 29457807, 29457978, 29458132, 29460252, 29461420
29461791, 29461971, 29462594, 29462767, 29462957, 29463047, 29463528
29463798, 29464616, 29464779, 29465047, 29465177, 29466674, 29467622
29469563, 29469565, 29470059, 29470291, 29471633, 29471832, 29471857
29471860, 29472618, 29473708, 29476473, 29477015, 29481584, 29482021
29483452, 29483532, 29483626, 29483672, 29483685, 29483712, 29483723
29483771, 29485099, 29485877, 29486181, 29486848, 29487150, 29487189
29487407, 29488894, 29489436, 29489546, 29490256, 29491784, 29492127
29492939, 29493122, 29494245, 29495057, 29495171, 29495684, 29497311
29497588, 29497696, 29498198, 29500257, 29500826, 29500963, 29501218
29502561, 29503543, 29503631, 29503827, 29504103, 29504492, 29504682
29505225, 29505589, 29505668, 29506942, 29507270, 29507616, 29508681
29509777, 29510278, 29511064, 29511611, 29511980, 29512125, 29512890
29514479, 29515134, 29515240, 29515476, 29515766, 29515834, 29516300
29516727, 29516766, 29517168, 29517883, 29518604, 29518767, 29519131
29521187, 29521688, 29521748, 29521862, 29522358, 29522561, 29522662
29523055, 29523216, 29523511, 29524599, 29524985, 29525366, 29525467
29525886, 29526966, 29527595, 29527610, 29528368, 29529147, 29530440
29530515, 29530812, 29530909, 29531654, 29531836, 29532112, 29532532
29536342, 29536445, 29536794, 29537829, 29538631, 29539413, 29540327
29541742, 29541769, 29541973, 29542084, 29542449, 29542580, 29542643
29543034, 29543956, 29544552, 29546817, 29547010, 29547867, 29548413
29548427, 29548592, 29548687, 29548722, 29549040, 29549071, 29549104
29549154, 29549730, 29550530, 29552402, 29552773, 29553141, 29554092
29555105, 29557144, 29557261, 29557336, 29557556, 29558238, 29558452
29558926, 29558975, 29559187, 29559395, 29559446, 29559908, 29559981
29564592, 29564593, 29565611, 29579919, 29580394, 29580983, 29581771
29584261, 29584693, 29586143, 29587299, 29587720, 29587765, 29588732
29589544, 29591343, 29591641, 29592011, 29592215, 29597536, 29597716
29597754, 29598039, 29598046, 29598226, 29598233, 29599008, 29599243
29599300, 29599552, 29601461, 29602831, 29603460, 29603884, 29604002
29604257, 29606261, 29607136, 29607797, 29608000, 29608023, 29610506
29611020, 29611991, 29614206, 29614931, 29614987, 29615824, 29616244
29616414, 29618074, 29618190, 29620042, 29622936, 29623323, 29623592
29624124, 29625065, 29625804, 29625876, 29626154, 29626732, 29628200
29628647, 29629430, 29629650, 29629681, 29629745, 29631749, 29632095
29632265, 29632611, 29633697, 29633753, 29633936, 29634643, 29635427
29635717, 29635990, 29637362, 29637526, 29637560, 29638285, 29641736
29642451, 29643721, 29644426, 29644464, 29645167, 29645349, 29647176
29647770, 29648928, 29649694, 29651183, 29651520, 29652809, 29653132
29653246, 29655164, 29655668, 29656400, 29656819, 29656843, 29657399
29657422, 29657744, 29657960, 29658056, 29661028, 29661065, 29661722
29663191, 29663368, 29663494, 29663601, 29664087, 29664161, 29665168
29665940, 29666451, 29667527, 29667994, 29668005, 29669413, 29670713
29670782, 29671363, 29672507, 29675446, 29676089, 29677051, 29677173
29677733, 29677927, 29679856, 29680700, 29681987, 29683039, 29683211
29684518, 29685137, 29685276, 29687214, 29687220, 29687459, 29687718
29687727, 29687763, 29688867, 29689145, 29689255, 29692694, 29694869
29695425, 29695821, 29695841, 29695964, 29695987, 29696310, 29697928
29700125, 29700460, 29700770, 29701720, 29703932, 29705793, 29706160
29707099, 29707493, 29707896, 29708324, 29708353, 29708876, 29708915
29710188, 29710858, 29713810, 29715220, 29715703, 29716194, 29716227
29716491, 29716602, 29716871, 29717659, 29717901, 29718198, 29719146
29720133, 29720373, 29721418, 29721576, 29722167, 29724658, 29725476
29725781, 29726695, 29737941, 29738374, 29738400, 29739576, 29741319
29741976, 29742223, 29744225, 29744400, 29744637, 29745288, 29746962
29747493, 29747648, 29747653, 29748285, 29748325, 29748336, 29748513
29749471, 29750673, 29751094, 29753244, 29754196, 29754951, 29755821
29756274, 29756444, 29757099, 29757264, 29757651, 29757687, 29758203
29758217, 29758661, 29761678, 29761837, 29761911, 29763158, 29764644
29765035, 29765219, 29765347, 29765393, 29765493, 29766207, 29766435
29766503, 29766679, 29768487, 29768899, 29769901, 29770750, 29771032
29771242, 29772514, 29772761, 29773197, 29773205, 29773842, 29774362
29775393, 29779196, 29780140, 29782211, 29782284, 29782823, 29782866
29783142, 29784106, 29785239, 29785311, 29785831, 29787292, 29787766

29789911, 29791152, 29791880, 29792213, 29792433, 29793318, 29794174
29794462, 29795712, 29795957, 29796335, 29796378, 29796916, 29797209
29797726, 29801164, 29802382, 29802695, 29804875, 29805368, 29805772
29806390, 29806964, 29807964, 29809792, 29809837, 29811616, 29812084
29812489, 29813503, 29813650, 29813671, 29814995, 29815341, 29815713
29816887, 29817278, 29817547, 29817784, 29820341, 29821130, 29821582
29822714, 29825525, 29827647, 29827852, 29828644, 29829339, 29831196
29833984, 29834506, 29836096, 29838337, 29838485, 29838740, 29838773
29839715, 29840619, 29841267, 29841687, 29842369, 29843277, 29843692
29843831, 29844131, 29844226, 29844275, 29845530, 29846126, 29846525
29846645, 29846688, 29848084, 29848849, 29849100, 29850930, 29851733
29853485, 29856420, 29856859, 29856890, 29858121, 29858376, 29858420
29859068, 29860022, 29860994, 29861075, 29864203, 29864261, 29865188
29865590, 29865658, 29869052, 29869086, 29869149, 29869404, 29869887
29869906, 29870065, 29870533, 29871098, 29871312, 29871360, 29872401
29872937, 29872983, 29873665, 29874090, 29874761, 29875200, 29875459
29875565, 29876358, 29876989, 29877608, 29878076, 29881050, 29881478
29881575, 29881643, 29881839, 29882427, 29882454, 29882729, 29884958
29885182, 29885890, 29886809, 29887045, 29887111, 29888621, 29889184
29889358, 29890740, 29891075, 29891853, 29891916, 29892604, 29893132
29894021, 29896510, 29897418, 29897863, 29900203, 29900824, 29901419
29901961, 29902299, 29902327, 29902330, 29902659, 29903190, 29903299
29903357, 29903454, 29904002, 29906678, 29907942, 29908389, 29908777
29909658, 29910218, 29910402, 29912135, 29912286, 29913805, 29913966
29914449, 29914544, 29915217, 29915848, 29916975, 29919789, 29920025
29920376, 29920804, 29921318, 29922225, 29922461, 29923452, 29924181
29926466, 29927756, 29928210, 29928340, 29928427, 29928564, 29930457
29931956, 29932202, 29932310, 29932430, 29932780, 29934048, 29934052
29935685, 29937565, 29937655, 29937956, 29938225, 29939400, 29939795
29940373, 29941062, 29942096, 29942275, 29942554, 29943670, 29943879
29944035, 29944159, 29944660, 29944963, 29945645, 29946388, 29947145
29947428, 29948165, 29950220, 29951620, 29951759, 29952700, 29956016
29956222, 29957412, 29957493, 29958925, 29960884, 29961353, 29961360
29961609, 29961847, 29962160, 29962248, 29962834, 29962927, 29962939
29965052, 29965603, 29965888, 29966768, 29967223, 29968085, 29969557
29970081, 29970261, 29970298, 29970587, 29971027, 29971481, 29971888
29971936, 29971951, 29972134, 29972176, 29973012, 29989783, 29989845
29990779, 29991257, 29993717, 29997326, 29997553, 29997937, 30000664
30001105, 30001331, 30003187, 30004660, 30004856, 30006159, 30006985
30007450, 30007536, 30007797, 30008125, 30008198, 30008214, 30009710
30012181, 30014200, 30015070, 30017836, 30018017, 30018903, 30019864
30021830, 30024618, 30025814, 30026016, 30026596, 30027614, 30027649
30028182, 30028599, 30029519, 30029806, 30031027, 30032233, 30032376
30032484, 30033040, 30033547, 30034456, 30035598, 30036258, 30038392
30039800, 30039959, 30040157, 30041501, 30041514, 30042490, 30043398
30043610, 30043930, 30044108, 30044507, 300445273, 300445389, 300445484
30046497, 30047531, 30047702, 30047765, 30047931, 30048688, 30049966
30051176, 30051783, 30051804, 30052928, 30053036, 30053501, 30053748
30054980, 30056058, 30057718, 30057799, 30058149, 30058453, 30059106
30059109, 30060267, 30060330, 30062364, 30062819, 30064268, 30066352
30067565, 30068384, 30068792, 30068871, 30071446, 30072018, 30072905
30073314, 30073422, 30073744, 30074296, 30074349, 30074469, 30074472
30074820, 30075037, 30076058, 30076197, 30076253, 30076604, 30078675
30078934, 30079949, 30080111, 30080266, 30081546, 30081580, 30082145
30083100, 30083216, 30083488, 30083807, 30084971, 30085897, 30086596
30086992, 30087165, 30087509, 30088229, 30090568, 30092280, 30092859
30094929, 30095591, 30095952, 30097092, 30097115, 30098251, 30099302
30099420, 30099454, 30100354, 30101186, 30102774, 30103551, 30103553
30104348, 30104378, 30104555, 30106748, 30106901, 30108012, 30109365
30110224, 30110370, 30110518, 30114477, 30114489, 30114534, 30116085
30116203, 30116854, 30117209, 30117335, 30117469, 30117593, 30118261
30118279, 30120608, 30122523, 30122583, 30123138, 30125765, 30126145
30126470, 30127145, 30127522, 30127805, 30127904, 30128047, 30130240
30131286, 30131645, 30132708, 30133841, 30134746, 30135396, 30135731
30135942, 30136346, 30137792, 30139392, 30142907, 30143470, 30143593
30146593, 30146969, 30147195, 30147307, 30147473, 30147928, 30148929

30148999, 30149035, 30149658, 30150606, 30150710, 30153552, 30153885
30154633, 30155241, 30155489, 30155814, 30155837, 30155999, 30156569
30157526, 30158313, 30159329, 30159511, 30159536, 30159752, 30159760
30160625, 30161094, 30163243, 30164714, 30165493, 30165503, 30165897
30167787, 30169254, 30170104, 30172925, 30173113, 30173370, 30173556
30174401, 30175291, 30175587, 30177597, 30178250, 30178839, 30178990
30179038, 30179644, 30180208, 30180643, 30181756, 30182498, 30183367
30183696, 30183715, 30183920, 30184102, 30185852, 30186319, 30186476
30186706, 30187627, 30187866, 30189516, 30189535, 30190090, 30191274
30192691, 30192729, 30192853, 30193165, 30193262, 30193505, 30193506
30193584, 30193736, 30194612, 30194710, 30194972, 30195667, 30195668
30195684, 30196195, 30196358, 30196629, 30198239, 30198861, 30198905
30199890, 30200034, 30200132, 30200237, 30200680, 30200758, 30202349
30202388, 30203929, 30204042, 30204542, 30206493, 30206675, 30207473
30208327, 30208690, 30208723, 30209222, 30209736, 30210429, 30210753
30210884, 30213031, 30213540, 30214769, 30214826, 30215130, 30215302
30215351, 30217206, 30217562, 30217982, 30218044, 30218317, 30219222
30221237, 30221298, 30222512, 30222669, 30222975, 30223712, 30223847
30224544, 30224650, 30224725, 30224868, 30224950, 30225265, 30225439
30225443, 30225718, 30225844, 30226244, 30228567, 30229683, 30232638
30233934, 30234132, 30234227, 30235919, 30235979, 30236554, 30236964
30237477, 30238211, 30238715, 30239480, 30240010, 30240547, 30240858
30240930, 30240972, 30241567, 30241807, 30241920, 30242120, 30242724
30243216, 30244340, 30246053, 30246179, 30247305, 30248531, 30249432
30251003, 30252005, 30252098, 30252156, 30252458, 30252977, 30253035
30253090, 30253608, 30253705, 30253835, 30254206, 30254525, 30254576
30254726, 30255143, 30255528, 30256542, 30257412, 30257908, 30259120
30259469, 30260595, 30264405, 30265523, 30265608, 30265615, 30265703
30266791, 30267155, 30269428, 30269748, 30270647, 30270744, 30271114
30272329, 30274090, 30274188, 30274324, 30274662, 30275548, 30275569
30275578, 30276144, 30276243, 30277120, 30277451, 30277589, 30277733
30277887, 30278402, 30281428, 30282501, 30282591, 30282918, 30283296
30283577, 30283579, 30283581, 30283932, 30284219, 30284369, 30285026
30285166, 30285251, 30285457, 30285540, 30285843, 30288343, 30288491
30289074, 30289458, 30292305, 30293345, 30294267, 30294465, 30294671
30295110, 30295137, 30295549, 30295790, 30295808, 30297905, 30299367
30299817, 30299934, 30300030, 30300342, 30300363, 30300523, 30300538
30305264, 30305395, 30305568, 30305880, 30307814, 30307883, 30308368
30308624, 30308772, 30308947, 30309098, 30309798, 30311826, 30312094
30312546, 30313749, 30313848, 30313989, 30314079, 30314198, 30314837
30316667, 30316897, 30317209, 30317397, 30318638, 30318943, 30319080
30319099, 30320029, 30321076, 30321398, 30322980, 30323658, 30323849
30324180, 30324466, 30325407, 30326882, 30327149, 30327810, 30328168
30328690, 30329209, 30329751, 30330123, 30331356, 30331759, 30332505
30334484, 30334563, 30335127, 30335832, 30335987, 30336032, 30336383
30336530, 30336742, 30336996, 30337245, 30338591, 30339103, 30341713
30342371, 30342878, 30344614, 30345201, 30345432, 30345809, 30345926
30346330, 30346867, 30347410, 30349714, 30350177, 30350543, 30352532
30352581, 30352623, 30352715, 30355490, 30357463, 30357698, 30357897
30358416, 30359614, 30360383, 30361635, 30362003, 30362850, 30363088
30363311, 30363716, 30364329, 30364481, 30364613, 30365745, 30367193
30368048, 30368482, 30368534, 30368668, 30368917, 30371264, 30371623
30371909, 30372081, 30373419, 30373550, 30374345, 30374570, 30374739
30375109, 30376986, 30377347, 30377692, 30380907, 30381207, 30381525
30382646, 30382982, 30383286, 30384121, 30384152, 30387628, 30387666
30388853, 30389229, 30389414, 30389507, 30389821, 30390635, 30391272
30392011, 30392870, 30392987, 30393110, 30393653, 30394738, 30394974
30396120, 30396946, 30397100, 30398257, 30398422, 30399906, 30402386
30403412, 30403763, 30403881, 30403902, 30403989, 30404117, 30404153
30404639, 30406709, 30408515, 30408808, 30409207, 30409339, 30409472
30409590, 30412188, 30412772, 30412863, 30412885, 30412921, 30413137
30413294, 30414491, 30414679, 30414714, 30416034, 30416603, 30417648
30417732, 30419024, 30421026, 30421204, 30421439, 30421476, 30421706
30422487, 30423135, 30423218, 30424347, 30430921, 30431274, 30431504
30431698, 30431703, 30431717, 30431867, 30433177, 30436399, 30437003
30437149, 30439985, 30440651, 30441277, 30441687, 30441959, 30442266

30442749, 30442884, 30443393, 30446583, 30446820, 30447060, 30447589
30447994, 30448182, 30448917, 30449194, 30449837, 30450787, 30453442
30454090, 30457633, 30458568, 30458593, 30460095, 30460922, 30461123
30461458, 30463938, 30464250, 30464655, 30466081, 30469777, 30472891
30473634, 30474167, 30474774, 30475115, 30476768, 30477588, 30477685
30477691, 30477767, 30479252, 30479715, 30480872, 30483065, 30483140
30483521, 30484042, 30484801, 30485255, 30486436, 30486896, 30487387
30489582, 30490014, 30490578, 30492380, 30493518, 30494259, 30494900
30495035, 30495078, 30495133, 30495483, 30496957, 30497057, 30497765
30498824, 30500224, 30500297, 30500344, 30500582, 30501574, 30502415
30503943, 30505029, 30505497, 30506794, 30506991, 30507032, 30508100
30509277, 30510347, 30510527, 30512690, 30513285, 30513480, 30513848
30515886, 30516868, 30517214, 30517516, 30517635, 30518349, 30519188
30522285, 30522998, 30523137, 30523538, 30523601, 30523750, 30524736
30528547, 30528687, 30528704, 30528935, 30529790, 30529940, 30530585
30532811, 30533132, 30534351, 30534549, 30534662, 30534827, 30536237
30537405, 30537533, 30537584, 30539519, 30540109, 30540407, 30544247
30544595, 30544629, 30545281, 30545556, 30549255, 30549368, 30549637
30549789, 30549881, 30551000, 30551123, 30554178, 30556326, 30556581
30556807, 30557386, 30558561, 30559252, 30560365, 30560513, 30561404
30561590, 30561737, 30564139, 30564343, 30564898, 30565004, 30565595
30565805, 30566054, 30567372, 30571306, 30573236, 30573703, 30576112
30576393, 30576853, 30577071, 30577591, 30578221, 30579051, 30580813
30581448, 30582221, 30582500, 30588738, 30591028, 30591475, 30592859
30593046, 30593104, 30593863, 30594167, 30595114, 30595408, 30595860
30596488, 30596694, 30598682, 30598746, 30598919, 30599405, 30599407
30600173, 30600184, 30602230, 30602828, 30605215, 30605676, 30606345
30606451, 30608583, 30609799, 30610406, 30610667, 30611603, 30612199
30613937, 30613971, 30614411, 30616406, 30616738, 30619138, 30619525
30619728, 30619787, 30620805, 30621255, 30622528, 30622755, 30623138
30623142, 30624243, 30624792, 30624864, 30624874, 30625121, 30628834
30628899, 30629139, 30629643, 30629799, 30631393, 30631523, 30633259
30633938, 30634548, 30635183, 30635302, 30635326, 30637270, 30637319
30641541, 30641755, 30641900, 30644530, 30644766, 30644889, 30645896
30647133, 30650404, 30651231, 30651570, 30651621, 30651674, 30652515
30652853, 30654558, 30655906, 30657196, 30657365, 30657566, 30657624
30657706, 30657875, 30657906, 30657940, 30658533, 30658555, 30658702
30659940, 30660412, 30661000, 30661939, 30662651, 30662736, 30663646
30665399, 30668407, 30670328, 30670584, 30671720, 30671813, 30671958
30674373, 30674959, 30676209, 30677633, 30679595, 30679771, 30681462
30681516, 30681521, 30684902, 30686017, 30686131, 30687047, 30689557
30690686, 30691604, 30691731, 30691857, 30692462, 30692473, 30693791
30694947, 30696566, 30698289, 30703610, 30704826, 30705448, 30708735
30710807, 30711370, 30712670, 30713133, 30714151, 30714715, 30716863
30718841, 30718862, 30719327, 30719419, 30720736, 30720844, 30722705
30723671, 30724679, 30724881, 30727701, 30727759, 30729278, 30729604
30730026, 30732711, 30734707, 30735153, 30735736, 30739876, 30740669
30740997, 30741263, 30748707, 30749644, 30749722, 30750219, 30750991
30751521, 30751527, 30751639, 30751968, 30753432, 30755348, 30758836
30758854, 30758943, 30761871, 30761878, 30763272, 30763305, 30763639
30763754, 30764405, 30764663, 30765486, 30765995, 30767277, 30768636
30769312, 30770717, 30773164, 30773797, 30776416, 30776929, 30777759
30778855, 30779240, 30781032, 30781041, 30782266, 30782300, 30782414
30783395, 30783551, 30785101, 30786237, 30786641, 30786655, 30788973
30789904, 30790441, 30801296, 30801510, 30803210, 30804646, 30806757
30806984, 30807723, 30807888, 30808869, 30809087, 30810765, 30812574
30814266, 30814285, 30815495, 30815852, 30816760, 30816938, 30819340
30819629, 30821297, 30823744, 30825391, 30825419, 30825656, 30826474
30828350, 30829779, 30830555, 30832775, 30833454, 30834068, 30834110
30835184, 30835853, 30836129, 30838605, 30839451, 30839836, 30841241
30842277, 30843271, 30844839, 30846063, 30846782, 30847442, 30847871
30848028, 30848097, 30848773, 30851448, 30851951, 30852954, 30855101
30856358, 30857501, 30857721, 30858877, 30858919, 30860803, 30861988
30863115, 30864607, 30865805, 30866141, 30866988, 30869131, 30870439
30871716, 30871792, 30873527, 30874270, 30874337, 30874660, 30879169
30879708, 30880774, 30880913, 30881407, 30883715, 30883785, 30883877

30886188, 30887501, 30887777, 30889443, 30889607, 30889723, 30890720
30890971, 30891760, 30891792, 30895426, 30895577, 30896620, 30896685
30898381, 30898748, 30898939, 30902655, 30904672, 30906274, 30906407
30909596, 30909918, 30910264, 30913399, 30914272, 30914674, 30915781
30919587, 30919691, 30919804, 30921136, 30922936, 30922996, 30923514
30923517, 30923597, 30923940, 30925316, 30927821, 30930149, 30930339
30931311, 30931981, 30932674, 30936251, 30936831, 30936942, 30937340
30937391, 30937410, 30938413, 30939307, 30939317, 30939934, 30940259
30940868, 30940869, 30941056, 30944643, 30945005, 30946072, 30946768
30946876, 30952104, 30952191, 30953266, 30953836, 30956571, 30956647
30957739, 30960356, 30960736, 30964194, 30965554, 30965649, 30968737
30968781, 30970518, 30972817, 30972841, 30972887, 30972947, 30972951
30972959, 30972966, 30973085, 30973113, 30973127, 30973137, 30973143
30973197, 30973698, 30973877, 30974813, 30977411, 30978554, 30980115
30980317, 30980733, 30981240, 30985027, 30985906, 30987088, 30988444
30990034, 30992330, 30992597, 30993198, 30993518, 30994996, 30996991
30997375, 30998035, 30998662, 30998759, 30998847, 31001017, 31001455
31001490, 31001859, 31002223, 31002923, 31003137, 31003659, 31004077
31004719, 31004844, 31006792, 31008240, 31008907, 31009545, 31009590
31009680, 31010218, 31010976, 31011361, 31013127, 31014323, 31015330
31016413, 31019249, 31021068, 31021157, 31021324, 31021542, 31022858
31025520, 31025531, 31025859, 31026220, 31026591, 31026860, 31027747
31028986, 31029936, 31030898, 31031955, 31032904, 31034794, 31035287
31035916, 31037421, 31038220, 31038447, 31039627, 31039928, 31042208
31043483, 31043630, 31044145, 31044951, 31045929, 31046188, 31046619
31047022, 31047169, 31048025, 31048741, 31049215, 31051056, 31051075
31052735, 31052809, 31055142, 31056909, 31058548, 31061482, 31061504
31062010, 31063380, 31063769, 31064025, 31065838, 31066082, 31066250
31066265, 31066554, 31067892, 31069059, 31071080, 31073586, 31074032
31075323, 31075960, 31077117, 31077365, 31078391, 31078757, 31079204
31080474, 31081558, 31084921, 31086869, 31087361, 31087679, 31088115
31088341, 31089270, 31090262, 31091868, 31092129, 31092233, 31092581
31092921, 31094183, 31094228, 31094688, 31096846, 31097760, 31097961
31100172, 31101386, 31103065, 31104809, 31106140, 31106577, 31107577
31109506, 31112530, 31112972, 31113089, 31113249, 31114265, 31114671
31115201, 31115502, 31118809, 31119057, 31119846, 31120361, 31122876
31124363, 31124914, 31125773, 31126053, 31126058, 31127043, 31127457
31127969, 31130156, 31132732, 31134430, 31138106, 31139643, 31141792
31142749, 31145403, 31145804, 31153120, 31153485, 31155634, 31156383
31158341, 31158380, 31159382, 31162711, 31162915, 31163379, 31165038
31165577, 31165722, 31168440, 31171096, 31171631, 31172207, 31172642
31175365, 31177193, 31177204, 31177221, 31178103, 31180519, 31181380
31182159, 31182756, 31182793, 31185224, 31188038, 31188398, 31190624
31192039, 31193292, 31193936, 31194264, 31195090, 31195430, 31195838
31200845, 31201001, 31201366, 31202536, 31204412, 31204878, 31208287
31213034, 31214119, 31215422, 31215438, 31216995, 31217946, 31218837
31219047, 31219975, 31220549, 31220881, 31220912, 31221454, 31222780
31223382, 31226448, 31228670, 31230775, 31233170, 31234765, 31234790
31235797, 31240626, 31244968, 31249008, 31249406, 31249696, 31254297
31254535, 31254929, 31255369, 31255869, 31257740, 31258101, 31258995
31260692, 31261641, 31265651, 31265773, 31268557, 31270711, 31271032
31287871, 31289115, 31290300, 31292298, 31293484, 31298871, 31301460
31303032, 31304573, 31305114, 31305624, 31306248, 31306261, 31306867
31306927, 31309379, 31309867, 31310564, 31310624, 31311830, 31312450
31312976, 31313117, 31313444, 31314885, 31315495, 31315876, 31316250
31321092, 31322720, 31325584, 31326608, 31326977, 31326998, 31327259
31327278, 31327349, 31327391, 31327896, 31331354, 31331372, 31333156
31334606, 31334961, 31335037, 31335142, 31336298, 31338249, 31338673
31338769, 31339457, 31339643, 31339744, 31343110, 31344046, 31348711
31350348, 31353610, 31356601, 31357581, 31357737, 31358308, 31359215
31359366, 31360146, 31360323, 31360469, 31360529, 31366716, 31367188
31367364, 31369444, 31372498, 31373825, 31373837, 31373843, 31376708
31377129, 31377487, 31377808, 31380443, 31381701, 31383396, 31383464
31383814, 31386394, 31387426, 31387443, 31388288, 31390936, 31391991
31393600, 31394341, 31394347, 31394365, 31395247, 31396027, 31396695
31398663, 31399131, 31401831, 31402078, 31403177, 31403565, 31404014

31404130, 31404263, 31408636, 31409483, 31411163, 31414023, 31414524
31417192, 31421316, 31422620, 31424838, 31425167, 31425761, 31429501
31429590, 31429770, 31430722, 31431005, 31433092, 31433579, 31434805
31434870, 31437030, 31440426, 31440813, 31442332, 31442714, 31444353
31444516, 31446431, 31447733, 31448680, 31449354, 31450392, 31450653
31454972, 31455597, 31458049, 31466433, 31468060, 31475635, 31476093
31476736, 31477424, 31477695, 31479272, 31479772, 31483949, 31484385
31484603, 31485386, 31485507, 31486557, 31487441, 31487491, 31489137
31490604, 31491634, 31493840, 31494264, 31496174, 31498559, 31499370
31499700, 31500971, 31501139, 31503349, 31507107, 31508712, 31509279
31510891, 31512044, 31513011, 31523548, 31525783, 31526903, 31527103
31527199, 31528962, 31533274, 31533817, 31533833, 31535955, 31536401
31536731, 31537521, 31539566, 31541864, 31544097, 31545477, 31546864
31547220, 31548675, 31549221, 31553674, 31553813, 31555539, 31557663
31559085, 31560592, 31561886, 31567124, 31567441, 31570054, 31570161
31572006, 31572267, 31574244, 31574267, 31576738, 31577569, 31578994
31581627, 31585351, 31585789, 31586381, 31591384, 31591400, 31591409
31591421, 31595632, 31597727, 31600023, 31600894, 31601385, 31603199
31605119, 31607937, 31609974, 31616104, 31620748, 31625579, 31625618
31626572, 31627587, 31628311, 31628753, 31630551, 31633224, 31637607
31637680, 31640240, 31644775, 31648120, 31649819, 31650202, 31652641
31653080, 31655807, 31658464, 31658943, 31661865, 31663189, 31663788
31668061, 31668694, 31668872, 31670014, 31670353, 31672605, 31674380
31675568, 31676941, 31677460, 31682766, 31683044, 31684494, 31686979
31688978, 31691030, 31695062, 31696577, 31697741, 31700234, 31701910
31706595, 31707190, 31708133, 31709647, 31709739, 31709777, 31711889
31711997, 31715935, 31718134, 31718346, 31721863, 31721880, 31722646
31723651, 31727560, 31728160, 31734583, 31735662, 31743771, 31747935
31747989, 31748000, 31748944, 31752502, 31753202, 31753425, 31753692
31754887, 31755245, 31756415, 31757357, 31757775, 31757824, 31758083
31758846, 31760592, 31763707, 31764866, 31765257, 31765296, 31766696
31767237, 31769373, 31770289, 31771370, 31771410, 31771468, 31775101
31776994, 31781897, 31783451, 31783782, 31785445, 31786838, 31787655
31788704, 31788761, 31790500, 31792465, 31792615, 31793713, 31796208
31796277, 31796882, 31798742, 31800757, 31807516, 31815099, 31816158
31816631, 31820632, 31820859, 31823051, 31827605, 31827912, 31829617
31829639, 31833172, 31833948, 31835854, 31836454, 31839779, 31842545
31843462, 31847489, 31849859, 31851383, 31852574, 31854692, 31855526
31860193, 31862359, 31863118, 31866141, 31867037, 31869601, 31871692
31872230, 31876368, 31878314, 31880154, 31881527, 31883124, 31886547
31886695, 31887130, 31888148, 31888731, 31889222, 31895670, 31897786
31897854, 31900585, 31903523, 31904933, 31905033, 31907137, 31907565
31909295, 31913650, 31921267, 31927930, 31935717, 31942144, 31943497
31952052, 31953989, 31958958, 31961940, 31965542, 31974597, 31974693
31986836, 31988079, 31991705, 31996264, 32002411, 32003551, 32005048
32007698, 32008586, 32010707, 32017301, 32032733, 32032887, 32048412
32050048, 32057639, 32061648, 32069696, 32069834, 32079739, 32082098
32089820, 32097882, 32101305, 32113113, 32118727, 32121326, 32129659
32130083, 32130504, 32150818, 32165759, 32169151, 32172777, 32174571
32207088, 32212635, 32221141, 32234161, 32290399, 32296941, 32321765

Version 19.0.0.0.ru-2021-01.rur-2021-01.r1

Important

We recommend that you upgrade your DB instance to 19.0.0.0.ru-2021-01.rur-2021-01.r2 rather than to 19.0.0.0.ru-2021-01.rur-2021-01.r1. The application of the patch for minor engine upgrades to 19.0.0.0.ru-2021-01.rur-2021-01.r1 encountered an issue. Release update 19.10.0.0.210119 (32218454) didn't register correctly in the DBA_REGISTRY_SQLPATCH table.

Version 19.0.0.0.ru-2021-01.rur-2021-01.r1 includes the following:

- Patch 32218454: DATABASE RELEASE UPDATE 19.10.0.0.210119
- Patch 32067171: OJVM RELEASE UPDATE 19.10.0.0.210119

- Patch 31335037: RDBMS - DSTV35 UPDATE - TZDATA2020A
- Patch 31335142: DSTV35 UPDATE - TZDATA2020A - NEED OJVM FIX
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER TABLE
- Patch 29782284: ORA-06508:"MDSYS.MDPRVT_IDX" WHILE UPGRADING DATABASE TO 18.3
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 29374604: IE not starting against 18c Oracle RDBMS Standard Edition
- PreUpgrade Jar: preupgrade_19_cbuild_9_lf.zip
- Support for [Managing advisor tasks \(p. 1099\)](#) using procedures in the rdsadmin.rdsadmin_util package

Combined patches for version 19.0.0.0.ru-2021-01.rur-2021-01.r1, released January 2021

Bugs fixed:

```
7391838, 8460502, 8476681, 14735102, 16664572, 17428816, 17468475
19080742, 19138896, 19697993, 20313356, 20867658, 21232786, 21374587
21528318, 21629064, 21639146, 21888352, 21965541, 22580355, 22729345
22748979, 23294761, 23296836, 23606241, 23645975, 23734075, 23763462
24596874, 24669730, 24687075, 24833686, 24957575, 24971456, 25030027
25092651, 25093917, 25404117, 25416731, 25560538, 25607406, 25756945
25792962, 25804387, 25806201, 25809128, 25883179, 25905368, 25986062
25997810, 26001677, 26127355, 26173091, 26284288, 26352615, 26440142
26476244, 26499997, 26611353, 26668264, 26716835, 26739322, 26777814
26819036, 26872233, 27004828, 27036163, 27044169, 27101798, 27126122
27126938, 27130348, 27166935, 27195575, 27195935, 27221350, 27222128
27244999, 27254335, 27260704, 27261477, 27359766, 27362994, 27369515
27378053, 27392968, 27406105, 27411022, 27423500, 27439716, 27453490
27458357, 27489107, 27572040, 27582210, 27589260, 27604329, 27622946
27629928, 27661222, 27666312, 27684864, 27692173, 27700413, 27710072
27729678, 27742354, 27745728, 27760043, 27801144, 27828892, 27829722
27846298, 27873364, 27877830, 27880025, 27929509, 27934711, 27935464
27941110, 27957203, 27967484, 27998559, 28007516, 28064977, 28072567
28078186, 28092783, 28104176, 28109326, 28125947, 28127569, 28129791
28133903, 28138847, 28144569, 28145995, 28181021, 28187837, 28189466
28204262, 28205555, 28209985, 28210681, 28263142, 28271258, 28271693
28276054, 28279456, 28294563, 28302580, 28313275, 28319114, 28322973
28326928, 28338211, 28350595, 28370061, 28371123, 28373960, 28375383
28379065, 28381939, 28386259, 28390273, 28395302, 28397317, 28402823
28403019, 28406374, 28410431, 28431445, 28435333, 28436414, 28442896
28454215, 28463226, 28470673, 28475242, 28482048, 28484299, 28489419
28492006, 28498976, 28502773, 28504631, 28513333, 28521330, 28530171
28534475, 28535127, 28537481, 28538439, 28541606, 28542455, 28546290
28547068, 28547926, 28558645, 28561704, 28564479, 28565296, 28567417
28567819, 28569897, 28572407, 28572533, 28572544, 28572667, 28572834
28578945, 28587723, 28589509, 28593682, 28594086, 28597221, 28601957
28602253, 28605066, 28606598, 28608211, 28612239, 28618343, 28620697
28621543, 28622202, 28625580, 28625862, 28627033, 28628592, 28630381
28632796, 28636532, 28639299, 28640772, 28642469, 28642899, 28643583
28643654, 28643718, 28644549, 28645570, 28646200, 28646939, 28649388
28655209, 28661333, 28663289, 28663782, 28672457, 28673945, 28681153
28689483, 28690694, 28692103, 28692275, 28694639, 28694872, 28696373
28697526, 28698087, 28699321, 28700945, 28703812, 28705231, 28707931
28708400, 28709063, 28710385, 28710469, 28710663, 28710734, 28714461
28715655, 28715727, 28718469, 28719348, 28720204, 28720418, 28721497
28722229, 28730079, 28730253, 28734355, 28740708, 28740799, 28742555
28745367, 28749853, 28751498, 28752923, 28755011, 28755084, 28755846
```

28758722, 28760206, 28763291, 28765983, 28767240, 28769456, 28771947
28772390, 28772816, 28774416, 28776431, 28776811, 28777214, 28778754
28781599, 28781754, 28785273, 28785321, 28785531, 28789531, 28791852
28793062, 28794230, 28795551, 28795734, 28800508, 28802734, 28804517
28805242, 28806517, 28808314, 28808652, 28808656, 28810381, 28811560
28813931, 28815123, 28815355, 28815557, 28816871, 28817449, 28818063
28819640, 28820669, 28821847, 28824482, 28827682, 28831971, 28833912
28835937, 28836716, 28837979, 28838385, 28844738, 28845346, 28846759
28847541, 28847572, 28849776, 28850084, 28852325, 28854004, 28854733
28855520, 28855922, 28857552, 28861861, 28862532, 28863263, 28863432
28863487, 28865569, 28867698, 28867992, 28870496, 28871040, 28872645
28872829, 28873575, 28874416, 28875089, 28876253, 28876639, 28876926
28877252, 28878865, 28881191, 28881848, 28882784, 28884931, 28887305
28888083, 28888327, 28889389, 28889730, 28892794, 28897123, 28897512
28899663, 28900506, 28901126, 28905390, 28905457, 28905615, 28907196
28910498, 28910586, 28911140, 28912691, 28915561, 28917080, 28918429
28919145, 28921844, 28922227, 28922532, 28922608, 28925250, 28925460
28925634, 28925880, 28927452, 28928462, 28932914, 28933158, 28935293
28935956, 28936114, 28937717, 28938422, 28938698, 28940179, 28940281
28940472, 28941901, 28942455, 28942694, 28945421, 28945994, 28946233
28948554, 28949888, 28950868, 28951332, 28951533, 28952168, 28954762
28955606, 28955883, 28956908, 28957260, 28957292, 28957723, 28958088
28959493, 28960863, 28962775, 28963036, 28965084, 28965095, 28965231
28965376, 28966444, 28968779, 28974083, 28974999, 28977322, 28980448
28981871, 28983095, 28983486, 28984313, 28985114, 28985272, 28985362
28985478, 28986207, 28986231, 28986257, 28986326, 28986481, 28986696
28988482, 28988864, 28989306, 28993295, 28993353, 28994307, 28994542
28995287, 28996376, 28999046, 29000000, 29001305, 29001888, 29002488
29002784, 29002927, 29003207, 29003407, 29003617, 29003738, 29006318
29006621, 29007321, 29007353, 29007775, 29008035, 29008669, 29009513
29010126, 29010517, 29011936, 29012609, 29013475, 29013832, 29014076
29015118, 29016294, 29017265, 29018655, 29018680, 29019121, 29020423
29021063, 29021352, 29022986, 29024028, 29024054, 29024448, 29024552
29024732, 29024876, 29026154, 29026582, 29026606, 29027456, 29027694
29027933, 29027940, 29030184, 29030927, 29031575, 29031600, 29032234
29032276, 29032457, 29032607, 29033052, 29033145, 29033200, 29033280
29034587, 29036278, 29037290, 29038528, 29038728, 29039089, 29039510
29040739, 29041739, 29041775, 29043554, 29043651, 29043725, 29044086
29044763, 29044954, 29046482, 29047127, 29047850, 29048178, 29048289
29048498, 29048605, 29048728, 29049673, 29050357, 29050560, 29050765
29050886, 29051263, 29051702, 29051953, 29052726, 29052850, 29053783
29053902, 29055644, 29056024, 29056270, 29056560, 29056767, 29056894
29058476, 29059011, 29060216, 29061016, 29061959, 29062692, 29062848
29062860, 29062868, 29110526, 29110783, 29110790, 29110797, 29110802
29110805, 29111598, 29111631, 29112455, 29113282, 29113305, 29115857
29117337, 29117526, 29117642, 29118543, 29119077, 29120223, 29122224
29122254, 29122367, 29123297, 29123432, 29123444, 29123482, 29124368
29125036, 29125374, 29125380, 29125708, 29125786, 29126345, 29127957
29128693, 29128935, 29129450, 29129476, 29129497, 29129691, 29129712
29130219, 29131539, 29131772, 29132456, 29132869, 29132938, 29133470
29134447, 29135383, 29135649, 29136111, 29138641, 29139070, 29139727
29139761, 29139956, 29141316, 29141341, 29141685, 29141886, 29142609
29142667, 29143516, 29144995, 29145214, 29145730, 29146077, 29146157
29146810, 29147849, 29148799, 29149170, 29149829, 29150338, 29151520
29152357, 29152603, 29152752, 29154631, 29154636, 29154725, 29154829
29155099, 29157051, 29157389, 29158680, 29158899, 29159216, 29159661
29159909, 29159936, 29160174, 29160462, 29161597, 29161923, 29162095
29163073, 29163156, 29163415, 29163437, 29163524, 29163567, 29164376
29165682, 29167111, 29167342, 29167374, 29167940, 29168137, 29168219
29168433, 29169073, 29169215, 29169540, 29169739, 29170232, 29170717
29171683, 29171942, 29172618, 29172826, 29173140, 29173373, 29173618
29173817, 29174004, 29174753, 29175638, 29176318, 29177466, 29177543
29177886, 29178385, 29179097, 29180313, 29180455, 29180559, 29180721
29180893, 29181078, 29181153, 29181231, 29181568, 29181620, 29181743
29181923, 29182019, 29182517, 29182901, 29182920, 29183298, 29183912
29184297, 29184666, 29185193, 29186091, 29186456, 29186605, 29188255

29189302, 29189307, 29189889, 29190235, 29190474, 29190663, 29190740
29191541, 29191827, 29192419, 29192468, 29192685, 29193207, 29194205
29194367, 29194493, 29194827, 29194981, 29195279, 29195337, 29195758
29196725, 29198092, 29198913, 29199163, 29199635, 29199733, 29200316
29200700, 29201143, 29201494, 29201539, 29201695, 29201787, 29202104
29202461, 29202850, 29203041, 29203122, 29203166, 29203227, 29203425
29203443, 29203604, 29205281, 29205323, 29205419, 29205463, 29205767
29205918, 29206109, 29206605, 29207073, 29208260, 29208732, 29209545
29210577, 29210610, 29210624, 29210683, 29211457, 29211724, 29212012
29212433, 29212611, 29213320, 29213351, 29213613, 29213641, 29213775
29213850, 29213879, 29213893, 29214561, 29214960, 29216312, 29216723
29216746, 29216984, 29217294, 29217472, 29217828, 29217848, 29217856
29218570, 29219205, 29219273, 29219627, 29220079, 29221248, 29221891
29221942, 29222031, 29222784, 29223833, 29223859, 29223967, 29224065
29224294, 29224605, 29224710, 29225076, 29225168, 29225758, 29225861
29227602, 29228869, 29229164, 29229754, 29229839, 29229844, 29229955
29230252, 29230565, 29231133, 29232117, 29232154, 29232449, 29232653
29233415, 29233810, 29233953, 29234123, 29235934, 29236573, 29237538
29237575, 29237744, 29240307, 29240668, 29240759, 29241345, 29241651
29242017, 29242884, 29242906, 29243749, 29243958, 29244495, 29244766
29244968, 29245063, 29245137, 29245160, 29246163, 29247183, 29247415
29247712, 29247906, 29248495, 29248552, 29248723, 29248835, 29248858
29249289, 29249412, 29249583, 29249991, 29250059, 29250317, 29251259
29251564, 29253184, 29253871, 29254031, 29254623, 29254930, 29255178
29255273, 29255431, 29255435, 29255616, 29255705, 29255718, 29255973
29256426, 29259119, 29259320, 29260224, 29260452, 29260956, 29261547
29261548, 29261695, 29261906, 29262512, 29262887, 29265448, 29266248
29266899, 29267292, 29268412, 29269171, 29269228, 29269825, 29270585
29271019, 29273168, 29273360, 29273539, 29273570, 29273735, 29273812
29273847, 29274428, 29274564, 29274627, 29275461, 29276272, 29277317
29278218, 29278684, 29279658, 29279751, 29279854, 29281112, 29281527
29281691, 29281796, 29282090, 29282233, 29282666, 29282898, 29285197
29285453, 29285503, 29285621, 29285788, 29285956, 29286037, 29286220
29286229, 29287130, 29287705, 29290110, 29290235, 29292232, 29292837
29293072, 29293574, 29293806, 29294753, 29296257, 29297863, 29297915
29298220, 29299049, 29299082, 29299830, 29299844, 29301463, 29301566
29302565, 29302614, 29302963, 29303918, 29304314, 29304692, 29304781
29304853, 29305093, 29306226, 29306713, 29307090, 29307109, 29307638
29309698, 29311336, 29311528, 29311588, 29311927, 29312310, 29312672
29312734, 29312753, 29312889, 29313347, 29313417, 29313525, 29314539
29314636, 29317756, 29318410, 29319441, 29320900, 29321489, 29321689
29323946, 29324568, 29324735, 29325087, 29325105, 29325257, 29325765
29325993, 29326233, 29327044, 29327892, 29329042, 29329087, 29329675
29329807, 29329848, 29330361, 29330791, 29331066, 29331209, 29331380
29331493, 29332292, 29332395, 29332763, 29332771, 29333500, 29336843
29336899, 29337294, 29337310, 29337742, 29338315, 29338348, 29338453
29338780, 29338913, 29339101, 29339155, 29339299, 29340333, 29341209
29342099, 29343086, 29343156, 29343861, 29344541, 29345937, 29346057
29346211, 29346943, 29347620, 29348176, 29348358, 29350052, 29350712
29350762, 29350868, 29351044, 29351386, 29351662, 29351716, 29351735
29351749, 29351771, 29352298, 29352724, 29352867, 29352947, 29353271
29353432, 29353718, 29353821, 29353960, 29355654, 29356547, 29356704
29356711, 29356752, 29356782, 29357821, 29358509, 29358828, 29360252
29360285, 29360467, 29360672, 29360775, 29360911, 29360950, 29361319
29361472, 29361801, 29362596, 29363151, 29364171, 29364177, 29366406
29366940, 29367019, 29367561, 29367971, 29368253, 29368310, 29368725
29372069, 29372541, 29372562, 29373418, 29373588, 29374179, 29374604
29375355, 29375941, 29375984, 29376346, 29377804, 29377986, 29378029
29378287, 29378834, 29378913, 29379299, 29379381, 29379750, 29379978
29380527, 29381000, 29382296, 29382641, 29382784, 29382815, 29383695
29384781, 29384854, 29384864, 29385339, 29385429, 29385652, 29386502
29386557, 29386635, 29386660, 29386835, 29387073, 29387274, 29387310
29387337, 29388020, 29388072, 29388094, 29388524, 29388830, 29389408
29389889, 29390011, 29390435, 29390785, 29391030, 29391237, 29391301
29391438, 29391849, 29391925, 29392554, 29392966, 29393291, 29393649
29394014, 29394140, 29394749, 29395657, 29396481, 29397841, 29397954

29397996, 29398488, 29398863, 29399046, 29399100, 29399121, 29399336
29399938, 29402110, 29402131, 29404483, 29405012, 29405462, 29405651
29405996, 29407488, 29407804, 29408853, 29409149, 29409455, 29410311
29410834, 29411037, 29411469, 29411931, 29412066, 29412269, 29413360
29413382, 29413517, 29413544, 29413634, 29413956, 29415493, 29416688
29416700, 29417084, 29417173, 29417719, 29417884, 29418165, 29418341
29420254, 29420834, 29421059, 29423003, 29423016, 29423156, 29423491
29423826, 29424999, 29426241, 29426320, 29428230, 29429017, 29429087
29429264, 29429466, 29429566, 29429895, 29430524, 29430866, 29431192
29431402, 29431485, 29432176, 29434301, 29434869, 29435474, 29435652
29436454, 29436514, 29436522, 29436727, 29437029, 29437379, 29437594
29437712, 29438150, 29438277, 29438736, 29439522, 29440651, 29441196
29442400, 29442936, 29443187, 29443250, 29443559, 29444072, 29444282
29444602, 29445548, 29446319, 29446669, 29448498, 29449477, 29449845
29449852, 29450162, 29450193, 29450273, 29450421, 29450812, 29450936
29451085, 29451386, 29452251, 29452576, 29452936, 29452953, 29454450
29454978, 29455424, 29455773, 29456538, 29456714, 29457312, 29457319
29457370, 29457502, 29457807, 29457978, 29458132, 29460252, 29461420
29461791, 29461971, 29462594, 29462767, 29462957, 29463047, 29463528
29463798, 29464616, 29464779, 29465047, 29465177, 29466674, 29467622
29469563, 29469565, 29470059, 29470291, 29471633, 29471832, 29471857
29471860, 29472618, 29473708, 29476473, 29477015, 29481584, 29482021
29483452, 29483532, 29483626, 29483672, 29483685, 29483712, 29483723
29483771, 29485099, 29485877, 29486181, 29486848, 29487150, 29487189
29487407, 29488894, 29489436, 29489546, 29490256, 29491784, 29492127
29492939, 29493122, 29494245, 29495057, 29495171, 29495684, 29497311
29497588, 29497696, 29498198, 29500257, 29500826, 29500963, 29501218
29502561, 29503543, 29503631, 29503827, 29504103, 29504492, 29504682
29505225, 29505589, 29505668, 29506942, 29507270, 29507616, 29508681
29509777, 29510278, 29511064, 29511611, 29511980, 29512125, 29512890
29514479, 29515134, 29515240, 29515476, 29515766, 29515834, 29516300
29516727, 29516766, 29517168, 29517883, 29518604, 29518767, 29519131
29521187, 29521688, 29521748, 29521862, 29522358, 29522561, 29522662
29523055, 29523216, 29523511, 29524599, 29524985, 29525366, 29525467
29525886, 29526966, 29527595, 29527610, 29528368, 29529147, 29530440
29530515, 29530812, 29530909, 29531654, 29531836, 29532112, 29532532
29536342, 29536445, 29536794, 29537829, 29538631, 29539413, 29540327
29541742, 29541769, 29541973, 29542084, 29542449, 29542580, 29542643
29543034, 29543956, 29544552, 29546817, 29547010, 29547867, 29548413
29548427, 29548592, 29548687, 29548722, 29549040, 29549071, 29549104
29549154, 29549730, 29550530, 29552402, 29552773, 29553141, 29554092
29555105, 29557144, 29557261, 29557336, 29557556, 29558238, 29558452
29558926, 29558975, 29559187, 29559395, 29559446, 29559908, 29559981
29564592, 29564593, 29565611, 29579919, 29580394, 29580983, 29581771
29584261, 29584693, 29586143, 29587299, 29587720, 29587765, 29588732
29589544, 29591343, 29591641, 29592011, 29592215, 29597536, 29597716
29597754, 29598039, 29598046, 29598226, 29598233, 29599008, 29599243
29599300, 29599552, 29601461, 29602831, 29603460, 29603884, 29604002
29604257, 29606261, 29607136, 29607797, 29608000, 29608023, 29610506
29611020, 29611991, 29614206, 29614931, 29614987, 29615824, 29616244
29616414, 29618074, 29618190, 29620042, 29622936, 29623323, 29623592
29624124, 29625065, 29625804, 29625876, 29626154, 29626732, 29628200
29628647, 29629430, 29629650, 29629681, 29629745, 29631749, 29632095
29632265, 29632611, 29633697, 29633753, 29633936, 29634643, 29635427
29635717, 29635990, 29637362, 29637526, 29637560, 29638285, 29641736
29642451, 29643721, 29644426, 29644464, 29645167, 29645349, 29647176
29647770, 29648928, 29649694, 29651183, 29651520, 29652809, 29653132
29653246, 29655164, 29655668, 29656400, 29656819, 29656843, 29657399
29657422, 29657744, 29657960, 29658056, 29661028, 29661065, 29661722
29663191, 29663368, 29663494, 29663601, 29664087, 29664161, 29665168
29665940, 29666451, 29667527, 29667994, 29668005, 29669413, 29670713
29670782, 29671363, 29672507, 29675446, 29676089, 29677051, 29677173
29677733, 29677927, 29679856, 29680700, 29681987, 29683039, 29683211
29684518, 29685137, 29685276, 29687214, 29687220, 29687459, 29687718
29687727, 29687763, 29688867, 29689145, 29689255, 29692694, 29694869
29695425, 29695821, 29695841, 29695964, 29695987, 29696310, 29697928

29700125, 29700460, 29700770, 29701720, 29703932, 29705793, 29706160
29707099, 29707493, 29707896, 29708324, 29708353, 29708876, 29708915
29710188, 29710858, 29713810, 29715220, 29715703, 29716194, 29716227
29716491, 29716602, 29716871, 29717659, 29717901, 29718198, 29719146
29720133, 29720373, 29721418, 29721576, 29722167, 29724658, 29725476
29725781, 29726695, 29737941, 29738374, 29738400, 29739576, 29741319
29741976, 29742223, 29744225, 29744400, 29744637, 29745288, 29746962
29747493, 29747648, 29747653, 29748285, 29748325, 29748336, 29748513
29749471, 29750673, 29751094, 29753244, 29754196, 29754951, 29755821
29756274, 29756444, 29757099, 29757264, 29757651, 29757687, 29758203
29758217, 29758661, 29761678, 29761837, 29761911, 29763158, 29764644
29765035, 29765219, 29765347, 29765393, 29765493, 29766207, 29766435
29766503, 29766679, 29768487, 29768899, 29769901, 29770750, 29771032
29771242, 29772514, 29772761, 29773197, 29773205, 29773842, 29774362
29775393, 29779196, 29780140, 29782211, 29782284, 29782823, 29782866
29783142, 29784106, 29785239, 29785311, 29785831, 29787292, 29787766
29789911, 29791152, 29791880, 29792213, 29792433, 29793318, 29794174
29794462, 29795712, 29795957, 29796335, 29796378, 29796916, 29797209
29797726, 29801164, 29802382, 29802695, 29804875, 29805368, 29805772
29806390, 29806964, 29807964, 29809792, 29809837, 29811616, 29812084
29812489, 29813503, 29813650, 29813671, 29814995, 29815341, 29815713
29816887, 29817278, 29817547, 29817784, 29820341, 29821130, 29821582
29822714, 29825525, 29827647, 29827852, 29828644, 29829339, 29831196
29833984, 29834506, 29836096, 29838337, 29838485, 29838740, 29838773
29839715, 29840619, 29841267, 29841687, 29842369, 29843277, 29843692
29843831, 29844131, 29844226, 29844275, 29845530, 29846126, 29846525
29846645, 29846688, 29848084, 29848849, 29849100, 29850930, 29851733
29853485, 29856420, 29856859, 29856890, 29858121, 29858376, 29858420
29859068, 29860022, 29860994, 29861075, 29864203, 29864261, 29865188
29865590, 29865658, 29869052, 29869086, 29869149, 29869404, 29869887
29869906, 29870065, 29870533, 29871098, 29871312, 29871360, 29872401
29872937, 29872983, 29873665, 29874090, 29874761, 29875200, 29875459
29875565, 29876358, 29876989, 29877608, 29878076, 29881050, 29881478
29881575, 29881643, 29881839, 29882427, 29882454, 29882729, 29884958
29885182, 29885890, 29886809, 29887045, 29887111, 29888621, 29889184
29889358, 29890740, 29891075, 29891853, 29891916, 29892604, 29893132
29894021, 29896510, 29897418, 29897863, 29900203, 29900824, 29901419
29901961, 29902299, 29902327, 29902330, 29902659, 29903190, 29903299
29903357, 29903454, 29904002, 29906678, 29907942, 29908389, 29908777
29909658, 29910218, 29910402, 29912135, 29912286, 29913805, 29913966
29914449, 29914544, 29915217, 29915848, 29916975, 29919789, 29920025
29920376, 29920804, 29921318, 29922225, 29922461, 29923452, 29924181
29926466, 29927756, 29928210, 29928340, 29928427, 29928564, 29930457
29931956, 29932202, 29932310, 29932430, 29932780, 29934048, 29934052
29935685, 29937565, 29937655, 29937956, 29938225, 29939400, 29939795
29940373, 29941062, 29942096, 29942275, 29942554, 29943670, 29943879
29944035, 29944159, 29944660, 29944963, 29945645, 29946388, 29947145
29947428, 29948165, 29950220, 29951620, 29951759, 29952700, 29956016
29956222, 29957412, 29957493, 29958925, 29960884, 29961353, 29961360
29961609, 29961847, 29962160, 29962248, 29962834, 29962927, 29962939
29965052, 29965603, 29965888, 29966768, 29967223, 29968085, 29969557
29970081, 29970261, 29970298, 29970587, 29971027, 29971481, 29971888
29971936, 29971951, 29972134, 29972176, 29973012, 29989783, 29989845
29990779, 29991257, 29993717, 29997326, 29997553, 29997937, 30000664
30001105, 30001331, 30003187, 30004660, 30004856, 30006159, 30006985
30007450, 30007536, 30007797, 30008125, 30008198, 30008214, 30009710
30012181, 30014200, 30015070, 30017836, 30018017, 30018903, 30019864
30021830, 30024618, 30025814, 30026016, 30026596, 30027614, 30027649
30028182, 30028599, 30029519, 30029806, 30031027, 30032233, 30032376
30032484, 30033040, 30033547, 30034456, 30035598, 30036258, 30038392
30039800, 30039959, 30040157, 30041501, 30041514, 30042490, 30043398
30043610, 30043930, 30044108, 30044507, 30045273, 30045389, 30045484
30046497, 30047531, 30047702, 30047765, 30047931, 30048688, 30049966
30051176, 30051783, 30051804, 30052928, 30053036, 30053501, 30053748
30054980, 30056058, 30057718, 30057799, 30058149, 30058453, 30059106
30059109, 30060267, 30060330, 30062364, 30062819, 30064268, 30066352

30067565, 30068384, 30068792, 30068871, 30071446, 30072018, 30072905
30073314, 30073422, 30073744, 30074296, 30074349, 30074469, 30074472
30074820, 30075037, 30076058, 30076197, 30076253, 30076604, 30078675
30078934, 30079949, 30080111, 30080266, 30081546, 30081580, 30082145
30083100, 30083216, 30083488, 30083807, 30084971, 30085897, 30086596
30086992, 30087165, 30087509, 30088229, 30090568, 30092280, 30092859
30094929, 30095591, 30095952, 30097092, 30097115, 30098251, 30099302
30099420, 30099454, 30100354, 30101186, 30102774, 30103551, 30103553
30104348, 30104378, 30104555, 30106748, 30106901, 30108012, 30109365
30110224, 30110370, 30110518, 30114477, 30114489, 30114534, 30116085
30116203, 30116854, 30117209, 30117335, 30117469, 30117593, 30118261
30118279, 30120608, 30122523, 30122583, 30123138, 30125765, 30126145
30126470, 30127145, 30127522, 30127805, 30127904, 30128047, 30130240
30131286, 30131645, 30132708, 30133841, 30134746, 30135396, 30135731
30135942, 30136346, 30137792, 30139392, 30142907, 30143470, 30143593
30146593, 30146969, 30147195, 30147307, 30147473, 30147928, 30148929
30148999, 30149035, 30149658, 30150606, 30150710, 30153552, 30153885
30154633, 30155241, 30155489, 30155814, 30155837, 30155999, 30156569
30157526, 30158313, 30159329, 30159511, 30159536, 30159752, 30159760
30160625, 30161094, 30163243, 30164714, 30165493, 30165503, 30165897
30167787, 30169254, 30170104, 30172925, 30173113, 30173370, 30173556
30174401, 30175291, 30175587, 30177597, 30178250, 30178839, 30178990
30179038, 30179644, 30180208, 30180643, 30181756, 30182498, 30183367
30183696, 30183715, 30183920, 30184102, 30185852, 30186319, 30186476
30186706, 30187627, 30187866, 30189516, 30189535, 30190090, 30191274
30192691, 30192729, 30192853, 30193165, 30193262, 30193505, 30193506
30193584, 30193736, 30194612, 30194710, 30194972, 30195667, 30195668
30195684, 30196195, 30196358, 30196629, 30198239, 30198861, 30198905
30199890, 30200034, 30200132, 30200237, 30200680, 30200758, 30202349
30202388, 30203929, 30204042, 30204542, 30206493, 30206675, 30207473
30208327, 30208690, 30208723, 30209222, 30209736, 30210429, 30210753
30210884, 30213031, 30213540, 30214769, 30214826, 30215130, 30215302
30215351, 30217206, 30217562, 30217982, 30218044, 30218317, 30219222
30221237, 30221298, 30222512, 30222669, 30222975, 30223712, 30223847
30224544, 30224650, 30224725, 30224868, 30224950, 30225265, 30225439
30225443, 30225718, 30225844, 30226244, 30228567, 30229683, 30232638
30233934, 30234132, 30234227, 30235919, 30235979, 30236554, 30236964
30237477, 30238211, 30238715, 30239480, 30240010, 30240547, 30240858
30240930, 30240972, 30241567, 30241807, 30241920, 30242120, 30242724
30243216, 30244340, 30246053, 30246179, 30247305, 30248531, 30249432
30251003, 30252005, 30252098, 30252156, 30252458, 30252977, 30253035
30253090, 30253608, 30253705, 30253835, 30254206, 30254525, 30254576
30254726, 30255143, 30255528, 30256542, 30257412, 30257908, 30259120
30259469, 30260595, 30264405, 30265523, 30265608, 30265615, 30265703
30266791, 30267155, 30269428, 30269748, 30270647, 30270744, 30271114
30272329, 30274090, 30274188, 30274324, 30274662, 30275548, 30275569
30275578, 30276144, 30276243, 30277120, 30277451, 30277589, 30277733
30277887, 30278402, 30281428, 30282501, 30282591, 30282918, 30283296
30283577, 30283579, 30283581, 30283932, 30284219, 30284369, 30285026
30285166, 30285251, 30285457, 30285540, 30285843, 30288343, 30288491
30289074, 30289458, 30292305, 30293345, 30294267, 30294465, 30294671
30295110, 30295137, 30295549, 30295790, 30295808, 30297905, 30299367
30299817, 30299934, 30300030, 30300342, 30300363, 30300523, 30300538
30305264, 30305395, 30305568, 30305880, 30307814, 30307883, 30308368
30308624, 30308772, 30308947, 30309098, 30309798, 30311826, 30312094
30312546, 30313749, 30313848, 30313989, 30314079, 30314198, 30314837
30316667, 30316897, 30317209, 30317397, 30318638, 30318943, 30319080
30319099, 30320029, 30321076, 30321398, 30322980, 30323658, 30323849
30324180, 30324466, 30325407, 30326882, 30327149, 30327810, 30328168
30328690, 30329209, 30329751, 30330123, 30331356, 30331759, 30332505
30334484, 30334563, 30335127, 30335832, 30335987, 30336032, 30336383
30336530, 30336742, 30336996, 30337245, 30338591, 30339103, 30341713
30342371, 30342878, 30344614, 30345201, 30345432, 30345809, 30345926
30346330, 30346867, 30347410, 30349714, 30350177, 30350543, 30352532
30352581, 30352623, 30352715, 30355490, 30357463, 30357698, 30357897
30358416, 30359614, 30360383, 30361635, 30362003, 30362850, 30363088

30363311, 30363716, 30364329, 30364481, 30364613, 30365745, 30367193
30368048, 30368482, 30368534, 30368668, 30368917, 30371264, 30371623
30371909, 30372081, 30373419, 30373550, 30374345, 30374570, 30374739
30375109, 30376986, 30377347, 30377692, 30380907, 30381207, 30381525
30382646, 30382982, 30383286, 30384121, 30384152, 30387628, 30387666
30388853, 30389229, 30389414, 30389507, 30389821, 30390635, 30391272
30392011, 30392870, 30392987, 30393110, 30393653, 30394738, 30394974
30396120, 30396946, 30397100, 30398257, 30398422, 30399906, 30402386
30403412, 30403763, 30403881, 30403902, 30403989, 30404117, 30404153
30404639, 30406709, 30408515, 30408808, 30409207, 30409339, 30409472
30409590, 30412188, 30412772, 30412863, 30412885, 30412921, 30413137
30413294, 30414491, 30414679, 30414714, 30416034, 30416603, 30417648
30417732, 30419024, 30421026, 30421204, 30421439, 30421476, 30421706
30422487, 30423135, 30423218, 30424347, 30430921, 30431274, 30431504
30431698, 30431703, 30431717, 30431867, 30433177, 30436399, 30437003
30437149, 30439985, 30440651, 30441277, 30441687, 30441959, 30442266
30442749, 30442884, 30443393, 30446583, 30446820, 30447060, 30447589
30447994, 30448182, 30448917, 30449194, 30449837, 30450787, 30453442
30454090, 30457633, 30458568, 30458593, 30460095, 30460922, 30461123
30461458, 30463938, 30464250, 30464655, 30466081, 30469777, 30472891
30473634, 30474167, 30474774, 30475115, 30476768, 30477588, 30477685
30477691, 30477767, 30479252, 30479715, 30480872, 30483065, 30483140
30483521, 30484042, 30484801, 30485255, 30486436, 30486896, 30487387
30489582, 30490014, 30490578, 30492380, 30493518, 30494259, 30494900
30495035, 30495078, 30495133, 30495483, 30496957, 30497057, 30497765
30498824, 30500224, 30500297, 30500344, 30500582, 30501574, 30502415
30503943, 30505029, 30505497, 30506794, 30506991, 30507032, 30508100
30509277, 30510347, 30510527, 30512690, 30513285, 30513480, 30513848
30515886, 30516868, 30517214, 30517516, 30517635, 30518349, 30519188
30522285, 30522998, 30523137, 30523538, 30523601, 30523750, 30524736
30528547, 30528687, 30528704, 30528935, 30529790, 30529940, 30530585
30532811, 30533132, 30534351, 30534549, 30534662, 30534827, 30536237
30537405, 30537533, 30537584, 30539519, 30540109, 30540407, 30544247
30544595, 30544629, 30545281, 30545556, 30549255, 30549368, 30549637
30549789, 30549881, 30551000, 30551123, 30554178, 30556326, 30556581
30556807, 30557386, 30558561, 30559252, 30560365, 30560513, 30561404
30561590, 30561737, 30564139, 30564343, 30564898, 30565004, 30565595
30565805, 30566054, 30567372, 30571306, 30573236, 30573703, 30576112
30576393, 30576853, 30577071, 30577591, 30578221, 30579051, 30580813
30581448, 30582221, 30582500, 30588738, 30591028, 30591475, 30592859
30593046, 30593104, 30593863, 30594167, 30595114, 30595408, 30595860
30596488, 30596694, 30598682, 30598746, 30598919, 30599405, 30599407
30600173, 30600184, 30602230, 30602828, 30605215, 30605676, 30606345
30606451, 30608583, 30609799, 30610406, 30610667, 30611603, 30612199
30613937, 30613971, 30614411, 30616406, 30616738, 30619138, 30619525
30619728, 30619787, 30620805, 30621255, 30622528, 30622755, 30623138
30623142, 30624243, 30624792, 30624864, 30624874, 30625121, 30628834
30628899, 30629139, 30629643, 30629799, 30631393, 30631523, 30633259
30633938, 30634548, 30635183, 30635302, 30635326, 30637270, 30637319
30641541, 30641755, 30641900, 30644530, 30644766, 30644889, 30645896
30647133, 30650404, 30651231, 30651570, 30651621, 30651674, 30652515
30652853, 30654558, 30655906, 30657196, 30657365, 30657566, 30657624
30657706, 30657875, 30657906, 30657940, 30658533, 30658555, 30658702
30659940, 30660412, 30661000, 30661939, 30662651, 30662736, 30663646
30665399, 30668407, 30670328, 30670584, 30671720, 30671813, 30671958
30674373, 30674959, 30676209, 30677633, 30679595, 30679771, 30681462
30681516, 30681521, 30684902, 30686017, 30686131, 30687047, 30689557
30690686, 30691604, 30691731, 30691857, 30692462, 30692473, 30693791
30694947, 30696566, 30698289, 30703610, 30704826, 30705448, 30708735
30710807, 30711370, 30712670, 30713133, 30714151, 30714715, 30716863
30718841, 30718862, 30719327, 30719419, 30720736, 30720844, 30722705
30723671, 30724679, 30724881, 30727701, 30727759, 30729278, 30729604
30730026, 30732711, 30734707, 30735153, 30735736, 30739876, 30740669
30740997, 30741263, 30748707, 30749644, 30749722, 30750219, 30750991
30751521, 30751527, 30751639, 30751968, 30753432, 30755348, 30758836
30758854, 30758943, 30761871, 30761878, 30763272, 30763305, 30763639

30763754, 30764405, 30764663, 30765486, 30765995, 30767277, 30768636
30769312, 30770717, 30773164, 30773797, 30776416, 30776929, 30777759
30778855, 30779240, 30781032, 30781041, 30782266, 30782300, 30782414
30783395, 30783551, 30785101, 30786237, 30786641, 30786655, 30788973
30789904, 30790441, 30801296, 30801510, 30803210, 30804646, 30806757
30806984, 30807723, 30807888, 30808869, 30809087, 30810765, 30812574
30814266, 30814285, 30815495, 30815852, 30816760, 30816938, 30819340
30819629, 30821297, 30823744, 30825391, 30825419, 30825656, 30826474
30828350, 30829779, 30830555, 30832775, 30833454, 30834068, 30834110
30835184, 30835853, 30836129, 30838605, 30839451, 30839836, 30841241
30842277, 30843271, 30844839, 30846063, 30846782, 30847442, 30847871
30848028, 30848097, 30848773, 30851448, 30851951, 30852954, 30855101
30856358, 30857501, 30857721, 30858877, 30858919, 30860803, 30861988
30863115, 30864607, 30865805, 30866141, 30866988, 30869131, 30870439
30871716, 30871792, 30873527, 30874270, 30874337, 30874660, 30879169
30879708, 30880774, 30880913, 30881407, 30883715, 30883785, 30883877
30886188, 30887501, 30887777, 30889443, 30889607, 30889723, 30890720
30890971, 30891760, 30891792, 30895426, 30895577, 30896620, 30896685
30898381, 30898748, 30898939, 30902655, 30904672, 30906274, 30906407
30909596, 30909918, 30910264, 30913399, 30914272, 30914674, 30915781
30919587, 30919691, 30919804, 30921136, 30922936, 30922996, 30923514
30923517, 30923597, 30923940, 30925316, 30927821, 30930149, 30930339
30931311, 30931981, 30932674, 30936251, 30936831, 30936942, 30937340
30937391, 30937410, 30938413, 30939307, 30939317, 30939934, 30940259
30940868, 30940869, 30941056, 30944643, 30945005, 30946072, 30946768
30946876, 30952104, 30952191, 30953266, 30953836, 30956571, 30956647
30957739, 30960356, 30960736, 30964194, 30965554, 30965649, 30968737
30968781, 30970518, 30972817, 30972841, 30972887, 30972947, 30972951
30972959, 30972966, 30973085, 30973113, 30973127, 30973137, 30973143
30973197, 30973698, 30973877, 30974813, 30977411, 30978554, 30980115
30980317, 30980733, 30981240, 30985027, 30985906, 30987088, 30988444
30990034, 30992330, 30992597, 30993198, 30993518, 30994996, 30996991
30997375, 30998035, 30998662, 30998759, 30998847, 31001017, 31001455
31001490, 31001859, 31002223, 31002923, 31003137, 31003659, 31004077
31004719, 31004844, 31006792, 31008240, 31008907, 31009545, 31009590
31009680, 31010218, 31010976, 31011361, 31013127, 31014323, 31015330
31016413, 31019249, 31021068, 31021157, 31021324, 31021542, 31022858
31025520, 31025531, 31025859, 31026220, 31026591, 31026860, 31027747
31028986, 31029936, 31030898, 31031955, 31032904, 31034794, 31035287
31035916, 31037421, 31038220, 31038447, 31039627, 31039928, 31042208
31043483, 31043630, 31044145, 31044951, 31045929, 31046188, 31046619
31047022, 31047169, 31048025, 31048741, 31049215, 31051056, 31051075
31052735, 31052809, 31055142, 31056909, 31058548, 31061482, 31061504
31062010, 31063380, 31063769, 31064025, 31065838, 31066082, 31066250
31066265, 31066554, 31067892, 31069059, 31071080, 31073586, 31074032
31075323, 31075960, 31077117, 31077365, 31078391, 31078757, 31079204
31080474, 31081558, 31084921, 31086869, 31087361, 31087679, 31088115
31088341, 31089270, 31090262, 31091868, 31092129, 31092233, 31092581
31092921, 31094183, 31094228, 31094688, 31096846, 31097760, 31097961
31100172, 31101386, 31103065, 31104809, 31106140, 31106577, 31107577
31109506, 31112530, 31112972, 31113089, 31113249, 31114265, 31114671
31115201, 31115502, 31118809, 31119057, 31119846, 31120361, 31122876
31124363, 31124914, 31125773, 31126053, 31126058, 31127043, 31127457
31127969, 31130156, 31132732, 31134430, 31138106, 31139643, 31141792
31142749, 31145403, 31145804, 31153120, 31153485, 31155634, 31156383
31158341, 31158380, 31159382, 31162711, 31162915, 31163379, 31165038
31165577, 31165722, 31168440, 31171096, 31171631, 31172207, 31172642
31175365, 31177193, 31177204, 31177221, 31178103, 31180519, 31181380
31182159, 31182756, 31182793, 31185224, 31188038, 31188398, 31190624
31192039, 31193292, 31193936, 31194264, 31195090, 31195430, 31195838
31200845, 31201001, 31201366, 31202536, 31204412, 31204878, 31208287
31213034, 31214119, 31215422, 31215438, 31216995, 31217946, 31218837
31219047, 31219975, 31220549, 31220881, 31220912, 31221454, 31222780
31223382, 31226448, 31228670, 31230775, 31233170, 31234765, 31234790
31235797, 31240626, 31244968, 31249008, 31249406, 31249696, 31254297
31254535, 31254929, 31255369, 31255869, 31257740, 31258101, 31258995

31260692, 31261641, 31265651, 31265773, 31268557, 31270711, 31271032
31287871, 31289115, 31290300, 31292298, 31293484, 31298871, 31301460
31303032, 31304573, 31305114, 31305624, 31306248, 31306261, 31306867
31306927, 31309379, 31309867, 31310564, 31310624, 31311830, 31312450
31312976, 31313117, 31313444, 31314885, 31315495, 31315876, 31316250
31321092, 31322720, 31325584, 31326608, 31326977, 31326998, 31327259
31327278, 31327349, 31327391, 31327896, 31331354, 31331372, 31333156
31334606, 31334961, 31335037, 31335142, 31336298, 31338249, 31338673
31338769, 31339457, 31339643, 31339744, 31343110, 31344046, 31348711
31350348, 31353610, 31356601, 31357581, 31357737, 31358308, 31359215
31359366, 31360146, 31360323, 31360469, 31360529, 31366716, 31367188
31367364, 31369444, 31372498, 31373825, 31373837, 31373843, 31376708
31377129, 31377487, 31377808, 31380443, 31381701, 31383396, 31383464
31383814, 31386394, 31387426, 31387443, 31388288, 31390936, 31391991
31393600, 31394341, 31394347, 31394365, 31395247, 31396027, 31396695
31398663, 31399131, 31401831, 31402078, 31403177, 31403565, 31404014
31404130, 31404263, 31408636, 31409483, 31411163, 31414023, 31414524
31417192, 31421316, 31422620, 31424838, 31425167, 31425761, 31429501
31429590, 31429770, 31430722, 31431005, 31433092, 31433579, 31434805
31434870, 31437030, 31440426, 31440813, 31442332, 31442714, 31444353
31444516, 31446431, 31447733, 31448680, 31449354, 31450392, 31450653
31454972, 31455597, 31458049, 31466433, 31468060, 31475635, 31476093
31476736, 31477424, 31477695, 31479272, 31479772, 31483949, 31484385
31484603, 31485386, 31485507, 31486557, 31487441, 31487491, 31489137
31490604, 31491634, 31493840, 31494264, 31496174, 31498559, 31499370
31499700, 31500971, 31501139, 31503349, 31507107, 31508712, 31509279
31510891, 31512044, 31513011, 31523548, 31525783, 31526903, 31527103
31527199, 31528962, 31533274, 31533817, 31533833, 31535955, 31536401
31536731, 31537521, 31539566, 31541864, 31544097, 31545477, 31546864
31547220, 31548675, 31549221, 31553674, 31553813, 31555539, 31557663
31559085, 31560592, 31561886, 31567124, 31567441, 31570054, 31570161
31572006, 31572267, 31574244, 31574267, 31576738, 31577569, 31578994
31581627, 31585351, 31585789, 31586381, 31591384, 31591400, 31591409
31591421, 31595632, 31597727, 31600023, 31600894, 31601385, 31603199
31605119, 31607937, 31609974, 31616104, 31620748, 31625579, 31625618
31626572, 31627587, 31628311, 31628753, 31630551, 31633224, 31637607
31637680, 31640240, 31644775, 31648120, 31649819, 31650202, 31652641
31653080, 31655807, 31658464, 31658943, 31661865, 31663189, 31663788
31668061, 31668694, 31668872, 31670014, 31670353, 31672605, 31674380
31675568, 31676941, 31677460, 31682766, 31683044, 31684494, 31686979
31688978, 31691030, 31695062, 31696577, 31697741, 31700234, 31701910
31706595, 31707190, 31708133, 31709647, 31709739, 31709777, 31711889
31711997, 31715935, 31718134, 31718346, 31721863, 31721880, 31722646
31723651, 31727560, 31728160, 31734583, 31735662, 31743771, 31747935
31747989, 31748000, 31748944, 31752502, 31753202, 31753425, 31753692
31754887, 31755245, 31756415, 31757357, 31757775, 31757824, 31758083
31758846, 31760592, 31763707, 31764866, 31765257, 31765296, 31766696
31767237, 31769373, 31770289, 31771370, 31771410, 31771468, 31775101
31776994, 31781897, 31783451, 31783782, 31785445, 31786838, 31787655
31788704, 31788761, 31790500, 31792465, 31792615, 31793713, 31796208
31796277, 31796882, 31798742, 31800757, 31807516, 31815099, 31816158
31816631, 31820632, 31820859, 31823051, 31827605, 31827912, 31829617
31829639, 31833172, 31833948, 31835854, 31836454, 31839779, 31842545
31843462, 31847489, 31849859, 31851383, 31852574, 31854692, 31855526
31860193, 31862359, 31863118, 31866141, 31867037, 31869601, 31871692
31872230, 31876368, 31878314, 31880154, 31881527, 31883124, 31886547
31886695, 31887130, 31888148, 31888731, 31889222, 31895670, 31897786
31897854, 31900585, 31903523, 31904933, 31905033, 31907137, 31907565
31909295, 31913650, 31921267, 31927930, 31935717, 31942144, 31943497
31952052, 31953989, 31958958, 31961940, 31965542, 31974597, 31974693
31986836, 31988079, 31991705, 31996264, 32002411, 32003551, 32005048
32007698, 32008586, 32010707, 32017301, 32032733, 32032887, 32048412
32050048, 32057639, 32061648, 32069696, 32069834, 32079739, 32082098
32089820, 32097882, 32101305, 32113113, 32118727, 32121326, 32129659
32130083, 32130504, 32150818, 32165759, 32169151, 32172777, 32174571
32207088, 32212635, 32221141, 32234161, 32290399, 32296941, 32321765

Version 19.0.0.0.ru-2020-10.rur-2020-10.r1

Version 19.0.0.0.ru-2020-10.rur-2020-10.r1 includes the following:

- Patch 31771877: Database Release Update: 19.9.0.0.201020 (31771877)
- Patch 31668882: OJVM RELEASE UPDATE: 19.9.0.0.201020 (31668882)
- Patch 31335037: RDBMS - DSTV35 UPDATE - TZDATA2020A
- Patch 31335142: DSTV35 UPDATE - TZDATA2020A - NEED OJVM FIX
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- Patch 29374604: IE not starting against 18c Oracle RDBMS Standard Edition.
- PreUpgrade Jar: preupgrade_19_cbuild_8_lf.zip
- Support for [Setting and unsetting system diagnostic events \(p. 1046\)](#) using procedures in the rdsadmin.rdsadmin_util package
- Support for the procedure rdsadmin_util.truncate_apply\$_cdr_info described in [Integrated REPLICAT slow due to query on sys."_DBA_APPLY_CDR_INFO" \(p. 1236\)](#)

Combined patches for version 19.0.0.0.ru-2020-10.rur-2020-10.r1, released October 2020

Bugs fixed:

7391838, 8460502, 8476681, 14735102, 17428816, 17468475, 19080742
19697993, 20313356, 21374587, 21639146, 21888352, 21965541, 22580355
22729345, 22748979, 23294761, 23296836, 23606241, 23645975, 23734075
23763462, 24596874, 24669730, 24687075, 24833686, 24971456, 25030027
25092651, 25093917, 25404117, 25416731, 25560538, 25756945, 25804387
25806201, 25809128, 25883179, 25905368, 25986062, 25997810, 26001677
26284288, 26352615, 26440142, 26476244, 26499997, 26611353, 26668264
26716835, 26739322, 26777814, 26819036, 26872233, 27036163, 27044169
27101798, 27126122, 27126938, 2716935, 27195935, 27221350, 27222128
27244999, 27254335, 27260704, 27261477, 27359766, 27369515, 27378053
27392968, 27406105, 27411022, 27423500, 27439716, 27453490, 27458357
27489107, 27572040, 27582210, 27589260, 27604329, 27629928, 27666312
27692173, 27700413, 27710072, 27729678, 27745728, 27760043, 27801144
27828892, 27846298, 27873364, 27877830, 27880025, 27929509, 27934711
27935464, 27941110, 27957203, 27967484, 28064977, 28072567, 28078186
28092783, 28104176, 28109326, 28125947, 28129791, 28138847, 28144569
28181021, 28189466, 28204262, 28205555, 28209985, 28210681, 28263142
28271258, 28271693, 28276054, 28279456, 28294563, 28302580, 28313275
28319114, 28322973, 28326928, 28350595, 28371123, 28373960, 28375383
28379065, 28381939, 28386259, 28390273, 28395302, 28397317, 28402823
28406374, 28410431, 28431445, 28435333, 28436414, 28442896, 28454215
28463226, 28470673, 28475242, 28482048, 28484299, 28489419, 28492006
28498976, 28502773, 28504631, 28513333, 28521330, 28530171, 28534475
28535127, 28537481, 28538439, 28541606, 28542455, 28546290, 28547068
28547926, 28558645, 28561704, 28564479, 28565296, 28567417, 28567819
28569897, 28572407, 28572533, 28572544, 28572667, 28572834, 28578945
28587723, 28589509, 28593682, 28594086, 28597221, 28601957, 28602253
28605066, 28606598, 28608211, 28612239, 28618343, 28620697, 28621543
28622202, 28625862, 28627033, 28628592, 28632796, 28636532, 28639299
28642469, 28642899, 28643583, 28643654, 28643718, 28644549, 28645570
28646200, 28646939, 28649388, 28655209, 28661333, 28663289, 28663782

28672457, 28673945, 28681153, 28689483, 28690694, 28692103, 28692275
28694639, 28694872, 28696373, 28697526, 28699321, 28703812, 28705231
28707931, 28708400, 28709063, 28710385, 28710469, 28710663, 28710734
28714461, 28715655, 28715727, 28718469, 28719348, 28720204, 28720418
28721497, 28722229, 28730079, 28730253, 28734355, 28740708, 28740799
28742555, 28745367, 28749853, 28752923, 28755011, 28755846, 28758722
28760206, 28765983, 28767240, 28769456, 28771947, 28772390, 28772816
28774416, 28776431, 28776811, 28777214, 28778754, 28781599, 28781754
28785321, 28785531, 28789531, 28791852, 28793062, 28794230, 28795551
28795734, 28800508, 28802734, 28804517, 28805242, 28808314, 28808652
28808656, 28810381, 28811560, 28813931, 28815123, 28815355, 28815557
28817449, 28819640, 28820669, 28821847, 28824482, 28827682, 28831971
28833912, 28835937, 28836716, 28838385, 28844738, 28845346, 28846759
28847541, 28847572, 28849776, 28850084, 28852325, 28854004, 28854733
28855520, 28855922, 28857552, 28861861, 28862532, 28863263, 28863432
28863487, 28865569, 28867698, 28867992, 28870496, 28871040, 28872645
28873575, 28874416, 28875089, 28876253, 28876639, 28876926, 28877252
28878865, 28881191, 28881848, 28882784, 28884931, 28887305, 28888083
28888327, 28889389, 28889730, 28892794, 28897512, 28899663, 28900506
28901126, 28905390, 28905457, 28905615, 28907196, 28910498, 28910586
28911140, 28912691, 28915561, 28917080, 28918429, 28919145, 28921844
28922227, 28922532, 28922608, 28925250, 28925460, 28925634, 28925880
28927452, 28928462, 28932914, 28933158, 28935293, 28935956, 28936114
28937717, 28938422, 28938698, 28940179, 28940281, 28940472, 28941901
28942455, 28942694, 28945421, 28945994, 28946233, 28948554, 28949888
28950868, 28951332, 28951533, 28952168, 28954762, 28955606, 28955883
28956908, 28957292, 28957723, 28958088, 28959493, 28960863, 28962775
28963036, 28965084, 28965095, 28965231, 28965376, 28966444, 28968779
28974083, 28974999, 28977322, 28980448, 28981871, 28983095, 28983486
28984313, 28985478, 28986207, 28986231, 28986257, 28986326, 28986481
28986696, 28988482, 28988864, 28989306, 28993295, 28993353, 28994307
28994542, 28995287, 28996376, 29000000, 29001305, 29001888, 29002488
29002784, 29002927, 29003207, 29003407, 29003617, 29003738, 29006318
29006621, 29007321, 29007353, 29007775, 29008035, 29008669, 29009513
29010126, 29010517, 29011936, 29012609, 29013475, 29013832, 29014076
29015118, 29016294, 29017265, 29018655, 29018680, 29019121, 29021063
29021352, 29022986, 29024054, 29024448, 29024552, 29024732, 29024876
29026154, 29026582, 29026606, 29027456, 29027694, 29027933, 29027940
29031575, 29031600, 29032234, 29032276, 29032457, 29032607, 29033052
29033145, 29033200, 29033280, 29034587, 29036278, 29037290, 29038528
29038728, 29039089, 29039510, 29040739, 29041739, 29041775, 29043554
29043651, 29043725, 29044086, 29044763, 29044954, 29046482, 29047127
29047850, 29048178, 29048289, 29048498, 29048605, 29048728, 29049673
29050357, 29050560, 29050765, 29050886, 29051263, 29051702, 29051953
29052726, 29052850, 29053783, 29055644, 29056024, 29056270, 29056560
29056767, 29056894, 29058476, 29059011, 29060216, 29061016, 29061959
29062692, 29062848, 29062860, 29062868, 29110526, 29110783, 29110790
29110797, 29110802, 29110805, 29111598, 29111631, 29112455, 29113282
29113305, 29115857, 29117337, 29117526, 29117642, 29118543, 29119077
29120223, 29122224, 29122254, 29122367, 29123297, 29123432, 29123444
29123482, 29124368, 29125036, 29125374, 29125380, 29125708, 29125786
29126345, 29127957, 29128693, 29128935, 29129450, 29129476, 29129497
29129691, 29129712, 29130219, 29131539, 29131772, 29132456, 29132869
29132938, 29133470, 29134447, 29135383, 29135649, 29136111, 29138641
29139070, 29139727, 29139761, 29139956, 29141316, 29141341, 29141685
29141886, 29142609, 29142667, 29143516, 29144995, 29145214, 29145730
29146157, 29146810, 29147849, 29149170, 29149829, 29150338, 29151520
29152357, 29152603, 29152752, 29154631, 29154636, 29154725, 29154829
29155099, 29157051, 29157389, 29158680, 29158899, 29159216, 29159661
29159909, 29159936, 29160174, 29160462, 29161597, 29161923, 29162095
29163073, 29163156, 29163415, 29163437, 29163524, 29163567, 29164376
29165682, 29167111, 29167342, 29167374, 29167940, 29168137, 29168219
29168433, 29169073, 29169215, 29169540, 29169739, 29170232, 29170717
29171683, 29171942, 29172618, 29172826, 29173140, 29173373, 29173618
29173817, 29174004, 29174753, 29175638, 29176318, 29177466, 29177543
29177886, 29178385, 29179097, 29180313, 29180455, 29180559, 29180721

29180893, 29181078, 29181153, 29181231, 29181568, 29181620, 29181743
29181923, 29182019, 29182517, 29182901, 29182920, 29183298, 29183912
29184297, 29184666, 29185193, 29186091, 29186456, 29186605, 29188255
29189302, 29189307, 29189889, 29190235, 29190474, 29190663, 29190740
29191541, 29191827, 29192419, 29192468, 29192685, 29193207, 29194205
29194367, 29194493, 29194827, 29194981, 29195279, 29195337, 29195758
29196725, 29198092, 29198913, 29199635, 29199733, 29200316, 29200700
29201143, 29201494, 29201539, 29201695, 29201787, 29202104, 29202461
29202850, 29203122, 29203166, 29203227, 29203425, 29203443, 29203604
29205281, 29205323, 29205419, 29205463, 29205767, 29205918, 29206109
29206605, 29207073, 29208260, 29208732, 29209545, 29210577, 29210610
29210624, 29210683, 29211457, 29211724, 29212012, 29212433, 29212611
29213320, 29213351, 29213613, 29213641, 29213775, 29213850, 29213879
29213893, 29214561, 29214960, 29216312, 29216723, 29216746, 29216984
29217294, 29217472, 29217828, 29217848, 29218570, 29219205, 29219273
29219627, 29220079, 29221248, 29221891, 29221942, 29222031, 29222784
29223833, 29223859, 29223967, 29224065, 29224294, 29224605, 29224710
29225076, 29225168, 29225758, 29225861, 29227602, 29228869, 29229164
29229754, 29229839, 29229844, 29229955, 29230252, 29230565, 29231133
29232117, 29232154, 29232449, 29232653, 29233415, 29233810, 29233953
29234123, 29235934, 29236573, 29237538, 29237575, 29237744, 29240307
29240668, 29240759, 29241345, 29241651, 29242017, 29242884, 29242906
29243749, 29243958, 29244495, 29244766, 29244968, 29245063, 29245137
29245160, 29246163, 29247415, 29247712, 29247906, 29248495, 29248552
29248723, 29248835, 29248858, 29249289, 29249412, 29249583, 29249991
29250059, 29250317, 29251259, 29251564, 29253184, 29253871, 29254031
29254623, 29254930, 29255178, 29255273, 29255431, 29255435, 29255616
29255718, 29255973, 29256426, 29259119, 29259320, 29260224, 29260452
29260956, 29261547, 29261548, 29261695, 29261906, 29262512, 29262887
29265448, 29266248, 29266899, 29267292, 29268412, 29269171, 29269228
29269825, 29270585, 29271019, 29273360, 29273539, 29273570, 29273735
29273812, 29273847, 29274428, 29274564, 29274627, 29275461, 29276272
29277317, 29278218, 29278684, 29279658, 29279751, 29279854, 29281527
29281691, 29281796, 29282090, 29282233, 29282666, 29282898, 29285197
29285453, 29285503, 29285621, 29285788, 29285956, 29286037, 29286229
29287130, 29287705, 29290110, 29290235, 29292232, 29292837, 29293072
29293574, 29293806, 29294753, 29296257, 29297863, 29297915, 29298220
29299049, 29299082, 29299830, 29299844, 29301463, 29301566, 29302963
29303918, 29304314, 29304692, 29304781, 29304853, 29306226, 29306713
29307090, 29307109, 29307638, 29309698, 29311336, 29311528, 29311588
29311927, 29312310, 29312672, 29312734, 29312753, 29312889, 29313347
29313417, 29313525, 29314539, 29314636, 29317756, 29318410, 29319441
29320900, 29321489, 29323946, 29324568, 29324735, 29325087, 29325105
29325257, 29325765, 29325993, 29327044, 29327892, 29329042, 29329087
29329675, 29329807, 29330361, 29330791, 29331066, 29331209, 29331380
29331493, 29332292, 29332395, 29332763, 29332771, 29333500, 29336843
29336899, 29337294, 29337310, 29337742, 29338315, 29338348, 29338453
29338780, 29338913, 29339101, 29339155, 29339299, 29341209, 29342099
29343086, 29343156, 29343861, 29344541, 29345937, 29346057, 29346211
29346943, 29347620, 29348176, 29348358, 29350052, 29350712, 29350762
29350868, 29351044, 29351386, 29351662, 29351716, 29351735, 29351749
29351771, 29352298, 29352724, 29352867, 29352947, 29353271, 29353432
29353718, 29353821, 29353960, 29355654, 29356547, 29356704, 29356711
29356752, 29356782, 29357821, 29358509, 29358828, 29360252, 29360285
29360467, 29360672, 29360775, 29360911, 29360950, 29361319, 29361472
29361801, 29362596, 29363151, 29364171, 29364177, 29366406, 29366940
29367019, 29367561, 29367971, 29368253, 29368310, 29368725, 29372069
29372541, 29372562, 29373418, 29373588, 29374179, 29374604, 29375355
29375941, 29375984, 29376346, 29377804, 29377986, 29378029, 29378287
29378834, 29378913, 29379299, 29379381, 29379750, 29379978, 29380527
29381000, 29382296, 29382641, 29382784, 29382815, 29383695, 29384781
29384854, 29384864, 29385339, 29385429, 29385652, 29386502, 29386557
29386635, 29386660, 29386835, 29387073, 29387274, 29387310, 29387337
29388020, 29388072, 29388094, 29388524, 29388830, 29389408, 29389889
29390011, 29390435, 29390785, 29391030, 29391237, 29391301, 29391438
29391849, 29391925, 29392554, 29392966, 29393291, 29393649, 29394014

29394140, 29394749, 29395657, 29397954, 29397996, 29398488, 29398863
29399046, 29399100, 29399121, 29399336, 29399938, 29402110, 29402131
29404483, 29405012, 29405462, 29405651, 29405996, 29407488, 29407804
29408853, 29409149, 29409455, 29410311, 29410834, 29411037, 29411469
29411931, 29412066, 29412269, 29413360, 29413382, 29413517, 29413544
29413634, 29413956, 29416688, 29416700, 29417084, 29417173, 29417719
29417884, 29418165, 29420254, 29420834, 29421059, 29423003, 29423016
29423156, 29423491, 29423826, 29424999, 29426241, 29426320, 29428230
29429017, 29429087, 29429264, 29429466, 29429566, 29429895, 29430524
29430866, 29431192, 29431485, 29432176, 29434301, 29434869, 29435474
29435652, 29436454, 29436514, 29436522, 29436727, 29437379, 29437594
29437712, 29438150, 29438277, 29438736, 29439522, 29440651, 29441196
29442936, 29443187, 29443250, 29443559, 29444072, 29444282, 29444602
29445548, 29446319, 29446669, 29448498, 29449477, 29449845, 29449852
29450162, 29450193, 29450421, 29450812, 29450936, 29451386, 29452251
29452576, 29452936, 29452953, 29454450, 29454978, 29455424, 29455773
29456538, 29456714, 29457312, 29457319, 29457370, 29457502, 29457807
29457978, 29460252, 29461420, 29461791, 29461971, 29462594, 29462767
29462957, 29463047, 29463528, 29463798, 29464616, 29464779, 29465047
29465177, 29466674, 29467622, 29469563, 29469565, 29470059, 29470291
29471633, 29471832, 29471860, 29472618, 29473708, 29476473, 29477015
29481584, 29482021, 29483452, 29483532, 29483626, 29483672, 29483685
29483712, 29483723, 29483771, 29485099, 29485877, 29486181, 29486848
29487150, 29487189, 29488894, 29489436, 29489546, 29490256, 29492127
29492939, 29493122, 29494245, 29495057, 29495684, 29497311, 29497588
29497696, 29498198, 29500257, 29500826, 29500963, 29501218, 29502561
29503543, 29503631, 29503827, 29504103, 29504492, 29504682, 29505668
29506942, 29507270, 29507616, 29508681, 29509777, 29510278, 29511064
29511611, 29511980, 29512125, 29512890, 29514479, 29515134, 29515240
29515476, 29515766, 29515834, 29516300, 29516727, 29516766, 29517168
29517883, 29519131, 29521187, 29521688, 29521748, 29521862, 29522358
29522561, 29522662, 29523055, 29523216, 29523511, 29524599, 29524985
29525366, 29525467, 29525886, 29526966, 29527595, 29527610, 29528368
29529147, 29530440, 29530515, 29530812, 29530909, 29531654, 29531836
29532532, 29536342, 29536445, 29537829, 29538631, 29540327, 29541742
29541769, 29541973, 29542084, 29542449, 29542580, 29542643, 29543034
29543956, 29544552, 29546817, 29547010, 29547867, 29548413, 29548427
29548592, 29548687, 29548722, 29549040, 29549071, 29549104, 29549154
29549730, 29552402, 29552773, 29553141, 29554092, 29555105, 29557144
29557261, 29557336, 29557556, 29558238, 29558452, 29558975, 29559187
29559395, 29559446, 29559908, 29559981, 29564592, 29564593, 29565611
29579919, 29580394, 29580983, 29581771, 29584261, 29584693, 29586143
29587299, 29587765, 29588732, 29589544, 29591343, 29592011, 29592215
29597536, 29597754, 29598039, 29598046, 29598226, 29598233, 29599008
29599300, 29601461, 29602831, 29603460, 29603884, 29604002, 29604257
29606261, 29607136, 29607797, 29608000, 29608023, 29610506, 29611020
29611991, 29614206, 29614987, 29615824, 29616244, 29616414, 29618074
29618190, 29620042, 29622936, 29623323, 29623592, 29624124, 29625065
29625804, 29625876, 29626154, 29626732, 29628200, 29629430, 29629650
29629681, 29629745, 29631749, 29632095, 29632265, 29632611, 29633697
29633753, 29633936, 29634643, 29635427, 29635717, 29635990, 29637362
29637526, 29637560, 29638285, 29641736, 29643721, 29644426, 29644464
29645167, 29645349, 29647176, 29647770, 29648928, 29651183, 29651520
29652809, 29653132, 29653246, 29655164, 29655668, 29656400, 29656819
29656843, 29657399, 29657422, 29657744, 29657960, 29658056, 29661028
29661065, 29661722, 29663191, 29663368, 29663494, 29663601, 29664087
29664161, 29665168, 29665940, 29667527, 29667994, 29668005, 29669413
29670782, 29671363, 29672507, 29675446, 29676089, 29677051, 29677173
29677733, 29677927, 29679856, 29681987, 29683039, 29683211, 29684518
29685137, 29685276, 29687214, 29687220, 29687459, 29687718, 29687727
29687763, 29688867, 29689145, 29689255, 29692694, 29694869, 29695425
29695821, 29695841, 29695964, 29696310, 29700125, 29700460, 29700770
29701720, 29703932, 29705793, 29707099, 29707493, 29707896, 29708353
29708876, 29708915, 29710188, 29710858, 29713810, 29715220, 29715703
29716194, 29716227, 29716491, 29716602, 29716871, 29717659, 29717901
29719146, 29720133, 29721418, 29721576, 29722167, 29724658, 29725476

29725781, 29726695, 29738374, 29738400, 29739576, 29741319, 29741976
29742223, 29744225, 29744400, 29745288, 29746962, 29747493, 29747648
29747653, 29748285, 29748336, 29748513, 29749471, 29750673, 29751094
29753244, 29754196, 29754951, 29755821, 29756274, 29756444, 29757099
29757264, 29757651, 29757687, 29758203, 29758217, 29758661, 29761678
29761837, 29761911, 29763158, 29765035, 29765393, 29766207, 29766435
29766503, 29766679, 29768487, 29768899, 29769901, 29770750, 29771032
29771242, 29773197, 29773205, 29773842, 29774362, 29775393, 29779196
29780140, 29782211, 29782823, 29782866, 29784106, 29785239, 29785311
29787292, 29787766, 29789911, 29791152, 29791880, 29792213, 29792433
29793318, 29794174, 29794462, 29795712, 29795957, 29796335, 29796378
29797209, 29797726, 29802382, 29802695, 29804875, 29805368, 29805772
29806390, 29807964, 29809792, 29809837, 29812084, 29812489, 29813503
29813650, 29813671, 29815341, 29815713, 29817278, 29817547, 29817784
29821130, 29821582, 29822714, 29825525, 29827647, 29827852, 29828644
29831196, 29833984, 29834506, 29836096, 29838337, 29838485, 29838773
29839715, 29840619, 29841267, 29841687, 29843277, 29843692, 29843831
29844131, 29844226, 29844275, 29845530, 29846126, 29846645, 29846688
29848084, 29848849, 29849100, 29850930, 29851733, 29853485, 29856859
29858121, 29858376, 29859068, 29860994, 29861075, 29864203, 29864261
29865188, 29865590, 29865658, 29869086, 29869149, 29869404, 29869887
29869906, 29870065, 29871098, 29871312, 29871360, 29872401, 29872937
29872983, 29873665, 29874090, 29874761, 29875459, 29875565, 29876358
29876989, 29877608, 29878076, 29881050, 29881478, 29881575, 29881643
29881839, 29882427, 29882454, 29882729, 29884958, 29885182, 29885890
29886809, 29887045, 29887111, 29888621, 29889184, 29889358, 29890740
29891075, 29891853, 29891916, 29892604, 29893132, 29896510, 29897418
29897863, 29900203, 29900824, 29901419, 29902299, 29902327, 29902330
29902659, 29903190, 29903299, 29903357, 29903454, 29904002, 29906678
29907942, 29908389, 29908777, 29909658, 29910402, 29912286, 29913805
29913966, 29914449, 29914544, 29915217, 29915848, 29916975, 29919789
29920025, 29920376, 29920804, 29921318, 29922225, 29922461, 29923452
29924181, 29926466, 29927756, 29928210, 29928340, 29928427, 29928564
29930457, 29932202, 29932430, 29932780, 29934052, 29935685, 29937565
29937655, 29937956, 29938225, 29939400, 29939795, 29940373, 29941062
29942096, 29942275, 29942554, 29943670, 29943879, 29944035, 29944159
29944660, 29944963, 29945645, 29946388, 29947145, 29948165, 29950220
29951620, 29951759, 29956016, 29956222, 29957412, 29957493, 29958925
29960884, 29961353, 29961609, 29961847, 29962160, 29962248, 29962834
29962927, 29962939, 29965052, 29965603, 29965888, 29966768, 29967223
29968085, 29969557, 29970081, 29970298, 29971027, 29971481, 29971888
29971936, 29971951, 29972176, 29973012, 29989783, 29989845, 29991257
29993717, 29997326, 29997553, 29997937, 30000664, 30001331, 30003187
30006159, 30006985, 30007450, 30007797, 30008125, 30008198, 30008214
30009710, 30012181, 30015070, 30017836, 30018017, 30018903, 30019864
30024618, 30025814, 30026016, 30027614, 30028599, 30029519, 30029806
30031027, 30032376, 30033040, 30033547, 30034456, 30035598, 30036258
30038392, 30039800, 30039959, 30040157, 30041501, 30041514, 30042490
30043398, 30043610, 30043930, 30044108, 30044507, 30045389, 30045484
30046497, 30047531, 30047702, 30047765, 30047931, 30048688, 30049966
30051176, 30051783, 30051804, 30052928, 30053036, 30053501, 30053748
30054980, 30056058, 30057718, 30057799, 30058149, 30058453, 30059106
30059109, 30060267, 30060330, 30062364, 30062819, 30064268, 30066352
30067565, 30068384, 30068871, 30071446, 30072905, 30073314, 30073744
30074296, 30074349, 30074469, 30074472, 30074820, 30075037, 30076058
30076197, 30076253, 30076604, 30078675, 30078934, 30079949, 30080266
30081546, 30081580, 30082145, 30083100, 30083216, 30083488, 30083807
30084971, 30085897, 30086596, 30086992, 30090568, 30092280, 30092859
30095591, 30095952, 30097092, 30097115, 30098251, 30099302, 30099420
30099454, 30100354, 30101186, 30103551, 30103553, 30104378, 30104555
30106748, 30109365, 30110224, 30110370, 30110518, 30114477, 30114489
30114534, 30116085, 30116203, 30116854, 30117209, 30117335, 30117469
30117593, 30118261, 30118279, 30120608, 30122583, 30125765, 30126145
30127145, 30127522, 30127805, 30127904, 30128047, 30130240, 30131286
30131645, 30132708, 30133841, 30134746, 30135396, 30135731, 30135942
30136346, 30139392, 30142907, 30143470, 30143593, 30146593, 30146969

30147473, 30147928, 30148999, 30149035, 30149658, 30150606, 30150710
30153552, 30153885, 30154633, 30155241, 30155814, 30155837, 30159329
30159511, 30159536, 30159752, 30159760, 30160625, 30161094, 30163243
30164714, 30165493, 30165503, 30165897, 30167787, 30169254, 30170104
30172925, 30173113, 30173556, 30174401, 30175291, 30177597, 30178250
30178839, 30178990, 30179644, 30180208, 30180643, 30181756, 30182498
30183696, 30183715, 30183920, 30184102, 30185852, 30186319, 30186476
30186706, 30187866, 30189516, 30190090, 30191274, 30192691, 30193165
30193505, 30193736, 30194612, 30194710, 30194972, 30195667, 30195668
30195684, 30196195, 30196358, 30196629, 30198861, 30198905, 30199890
30200034, 30200132, 30200237, 30200680, 30200758, 30202349, 30202388
30203929, 30204042, 30204542, 30206493, 30206675, 30207473, 30208327
30208723, 30209736, 30210884, 30213031, 30213540, 30215130, 30215302
30215351, 30217206, 30217562, 30217982, 30218044, 30218317, 30219222
30221237, 30221298, 30222512, 30223712, 30223847, 30224650, 30224868
30224950, 30225265, 30225439, 30225443, 30225718, 30225844, 30226244
30228567, 30229683, 30232638, 30235919, 30235979, 30236554, 30237477
30238211, 30238715, 30239480, 30240010, 30240547, 30240930, 30241567
30241807, 30241920, 30242120, 30242724, 30243216, 30244340, 30246053
30246179, 30247305, 30249432, 30251003, 30252005, 30252098, 30252156
30252458, 30252977, 30253035, 30253090, 30253608, 30253705, 30253835
30254206, 30254525, 30254576, 30254726, 30255143, 30255528, 30256542
30257412, 30257908, 30260595, 30264405, 30265523, 30265608, 30265615
30265703, 30266791, 30267155, 30269428, 30269748, 30270647, 30270744
30271114, 30272329, 30274090, 30274188, 30274324, 30274662, 30275548
30275569, 30275578, 30276243, 30277120, 30277451, 30277589, 30277733
30281428, 30282501, 30282591, 30283296, 30283577, 30283579, 30283581
30283932, 30284219, 30284369, 30285457, 30285540, 30285843, 30288343
30288491, 30289074, 30289458, 30293345, 30294267, 30294671, 30295110
30295137, 30295549, 30295790, 30295808, 30297905, 30299367, 30299817
30299934, 30300030, 30300342, 30300363, 30300538, 30305264, 30305395
30305568, 30305880, 30307814, 30307883, 30308368, 30308624, 30308772
30308947, 30309098, 30309798, 3031826, 30312094, 30312546, 30313848
30313989, 30314079, 30314198, 30314837, 30316667, 30316897, 30317209
30318638, 30318943, 30319080, 30319099, 30320029, 30322980, 30323658
30323849, 30324180, 30325407, 30326882, 30327149, 30328168, 30328690
30329209, 30330123, 30331356, 30331759, 30332505, 30334484, 30334563
30335127, 30335832, 30335987, 30336032, 30336742, 30339103, 30341713
30342371, 30342878, 30344614, 30345201, 30345432, 30345809, 30346330
30346867, 30347410, 30349714, 30350543, 30352581, 30352623, 30352715
30355490, 30357463, 30357897, 30360383, 30362003, 30362850, 30363088
30363716, 30364329, 30364613, 30365745, 30367193, 30368048, 30368482
30368534, 30368668, 30371264, 30371623, 30371909, 30372081, 30373419
30373550, 30374345, 30374570, 30374739, 30375109, 30376986, 30377347
30381207, 30381525, 30382982, 30383286, 30384121, 30384152, 30387666
30389229, 30389414, 30389507, 30391272, 30392011, 30392987, 30393110
30393653, 30394738, 30394974, 30396946, 30397100, 30398257, 30398422
30399906, 30402386, 30403763, 30403881, 30403902, 30403989, 30404117
30404153, 30406709, 30408515, 30408808, 30409207, 30409339, 30409590
30412188, 30412863, 30412885, 30412921, 30413137, 30413294, 30414491
30414679, 30414714, 30416034, 30416603, 30417648, 30417732, 30419024
30421204, 30421439, 30421476, 30421706, 30422487, 30423135, 30423218
30424347, 30430921, 30431274, 30431504, 30431698, 30431703, 30431717
30431867, 30433177, 30437149, 30441687, 30441959, 30442266, 30442749
30442884, 30443393, 30446583, 30447060, 30447589, 30448182, 30448917
30449194, 30449837, 30450787, 30453442, 30454090, 30457633, 30458568
30458593, 30460922, 30461458, 30463938, 30464250, 30464655, 30466081
30469777, 30472891, 30473634, 30474167, 30474774, 30475115, 30476768
30477588, 30477685, 30477691, 30477767, 30479252, 30479715, 30480872
30483065, 30483140, 30483521, 30484042, 30484801, 30485255, 30486436
30487387, 30490014, 30490578, 30493518, 30495035, 30495078, 30495133
30495483, 30496957, 30497057, 30497765, 30498824, 30500224, 30500297
30500344, 30500582, 30501574, 30502415, 30503943, 30505497, 30506794
30506991, 30507032, 30509277, 30510347, 30510527, 30513285, 30513848
30515886, 30516868, 30517214, 30517516, 30519188, 30522285, 30522998
30523137, 30523538, 30523601, 30523750, 30528547, 30528704, 30528935

30529940, 30532811, 30533132, 30534351, 30534549, 30534662, 30534827
30537405, 30537533, 30539519, 30540109, 30540407, 30544247, 30544595
30544629, 30545281, 30545556, 30549255, 30549637, 30549789, 30549881
30551000, 30551123, 30551478, 30556581, 30556807, 30557386, 30559252
30560365, 30560513, 30561590, 30561737, 30564139, 30564343, 30565805
30573236, 30573703, 30576112, 30576393, 30576853, 30577071, 30577591
30578221, 30579051, 30580813, 30581448, 30582221, 30582500, 30588738
30591028, 30592859, 30593046, 30595114, 30595860, 30596488, 30598682
30598746, 30598919, 30599405, 30599407, 30600173, 30600184, 30602230
30602828, 30605215, 30606345, 30606451, 30609799, 30610667, 30611603
30612199, 30613937, 30613971, 30614411, 30619525, 30619787, 30620805
30621255, 30622528, 30623138, 30623142, 30624792, 30624864, 30625121
30628899, 30629643, 30629799, 30631393, 30631523, 30633259, 30633938
30635183, 30635302, 30635326, 30637270, 30637319, 30641755, 30641900
30644889, 30647133, 30650404, 30651231, 30651621, 30651674, 30652515
30652853, 30654558, 30655906, 30657365, 30657624, 30657706, 30657875
30658533, 30658555, 30658702, 30660412, 30661939, 30662651, 30662736
30663646, 30668407, 30670584, 30671720, 30671813, 30674959, 30676209
30679595, 30679771, 30681462, 30681516, 30686131, 30687047, 30690686
30691604, 30691731, 30691857, 30692462, 30692473, 30694947, 30696566
30698289, 30703610, 30704826, 30708735, 30710807, 30711370, 30714151
30714715, 30716863, 30718841, 30718862, 30719419, 30720736, 30720844
30722705, 30723671, 30724679, 30724881, 30727701, 30729278, 30729604
30730026, 30732711, 30734707, 30735153, 30735736, 30740669, 30740997
30741263, 30749644, 30749722, 30750991, 30751521, 30751968, 30755348
30758943, 30761878, 30763272, 30763305, 30763639, 30763754, 30764663
30765486, 30769312, 30770717, 30773164, 30773797, 30776416, 30776929
30777759, 30778855, 30779240, 30781032, 30781041, 30782414, 30783551
30785101, 30786655, 30789904, 30790441, 30801296, 30801510, 30803210
30807723, 30808869, 30812574, 30814266, 30814285, 30815852, 30816760
30816938, 30821297, 30823744, 30825391, 30825419, 30825656, 30826474
30828350, 30829779, 30832775, 30833454, 30834110, 30835853, 30838605
30844839, 30847442, 30848097, 30848773, 30851951, 30855101, 30856358
30857501, 30857721, 30858919, 30861988, 30865805, 30866141, 30866988
30869131, 30870439, 30871716, 30871792, 30873527, 30880774, 30880913
30881407, 30883785, 30883877, 30886188, 30887501, 30887777, 30889723
30890720, 30890971, 30895577, 30896620, 30904672, 30906274, 30906407
30909918, 30910264, 30913399, 30914272, 30914674, 30919691, 30919804
30922936, 30922996, 30923517, 30923597, 30923940, 30927821, 30930339
30936831, 30937340, 30937410, 30939307, 30939934, 30940259, 30940868
30941056, 30944643, 30945005, 30946072, 30946876, 30952104, 30952191
30953266, 30953836, 30957739, 30964194, 30965649, 30968737, 30968781
30970518, 30972841, 30972887, 30972966, 30973113, 30973137, 30973143
30973698, 30978554, 30980317, 30980733, 30981240, 30985027, 30987088
30990034, 30992597, 30993198, 30994996, 30996991, 30997375, 30998759
30998847, 31001017, 31001455, 31001859, 31003659, 31004077, 31004719
31004844, 31008240, 31008907, 31009680, 31010976, 31013127, 31015330
31016413, 31019249, 31021157, 31021324, 31021542, 31022858, 31025859
31026220, 31028986, 31029936, 31031955, 31032904, 31034794, 31035916
31038220, 31039627, 31039928, 31042208, 31043483, 31051075, 31056909
31061482, 31061504, 31062010, 31066250, 31066265, 31066554, 31067892
31071080, 31077117, 31077365, 31079204, 31080474, 31084921, 31086869
31092129, 31094688, 31097760, 31100172, 31103065, 31104809, 31106577
31109506, 31113089, 31113249, 31115502, 31118809, 31119057, 31124914
31134430, 31141792, 31153120, 31153485, 31155634, 31156383, 31163379
31172207, 31177193, 31178103, 31180519, 31182159, 31182793, 31188038
31192039, 31193936, 31194264, 31200845, 31201001, 31202536, 31208287
31214119, 31215438, 31217946, 31219975, 31220912, 31221454, 31222780
31223382, 31228670, 31234765, 31249406, 31254297, 31254535, 31258101
31265773, 31292298, 31301460, 31305624, 31306248, 31306261, 31309867
31312976, 31315876, 31321092, 31325584, 31326608, 31331354, 31334606
31335037, 31335142, 31338249, 31338673, 31359215, 31383396, 31386394
31387443, 31393600, 31394365, 31401831, 31414023, 31414524, 31417192
31429770, 31430722, 31431005, 31433092, 31455597, 31475635, 31477424
31486557, 31500971, 31509279, 31513011, 31523548, 31527103, 31536731
31537521, 31544097, 31570161, 31591384, 31591400, 31591409, 31600023

31609974, 31628753, 31658464, 31668061, 31668872, 31672605, 31683044
31718134, 31718346, 31747989, 31758846, 31781897, 31792615, 31796208
31796277, 31820859, 31833172, 31867037, 31876368, 31886547, 31888148
31897786, 31905033, 31935717

Version 19.0.0.0.ru-2020-07.rur-2020-07.r1

Version 19.0.0.0.ru-2020-07.rur-2020-07.r1 includes the following:

- Patch 31281355: Database Release Update 19.8.0.0.200714
- Patch 31219897: Oracle JVM Release Update 19.8.0.0.200714
- Patch 31335037: DSTV35 for RDBMS (TZDATA2020A)
- Patch 31335142: DSTV35 for OJVM (TZDATA2020A)
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- Patch 29374604: IE not starting against 18c Oracle RDBMS Standard Edition
- PreUpgrade Jar: preupgrade_19_cbuild_7_lf.zip
- Patch 30417732: ORA-00600 [KQRHASHTABLEREMOVE: X LOCK] AND INSTANCE CRASH

Combined patches for version 19.0.0.0.ru-2020-07.rur-2020-07.r1, released July 2020

Bugs fixed:

7391838, 8476681, 14735102, 17428816, 19080742, 19697993, 20313356
21374587, 21965541, 22580355, 22729345, 23296836, 23606241, 23645975
23763462, 24596874, 24669730, 24687075, 24971456, 25030027, 25092651
25416731, 25560538, 25756945, 25804387, 25806201, 25883179, 25986062
25997810, 26001677, 26284288, 26440142, 26476244, 26611353, 26668264
26739322, 26777814, 26872233, 27036163, 27044169, 27101798, 27126122
27126938, 27166935, 27195935, 27221350, 27222128, 27244999, 27254335
27359766, 27369515, 27378053, 27392968, 27406105, 27411022, 27423500
27439716, 27453490, 27458357, 27489107, 27572040, 27582210, 27589260
27604329, 27629928, 27666312, 27692173, 27700413, 27710072, 27729678
27760043, 27801144, 27828892, 27846298, 27877830, 27880025, 27929509
27934711, 27935464, 27941110, 27957203, 27967484, 28064977, 28072567
28078186, 28092783, 28104176, 28109326, 28125947, 28129791, 28138847
28144569, 28181021, 28189466, 28204262, 28205555, 28209985, 28210681
28263142, 28271258, 28271693, 28276054, 28279456, 28294563, 28302580
28313275, 28319114, 28322973, 28326928, 28350595, 28371123, 28373960
28375383, 28379065, 28381939, 28386259, 28390273, 28395302, 28397317
28402823, 28410431, 28431445, 28435333, 28442896, 28454215, 28463226
28470673, 28475242, 28484299, 28489419, 28492006, 28498976, 28502773
28504631, 28513333, 28521330, 28530171, 28534475, 28535127, 28538439
28541606, 28542455, 28546290, 28547068, 28547926, 28558645, 28561704
28564479, 28565296, 28567417, 28567819, 28569897, 28572407, 28572533
28572544, 28572667, 28572834, 28578945, 28587723, 28589509, 28593682
28594086, 28597221, 28601957, 28605066, 28606598, 28608211, 28612239
28618343, 28620697, 28621543, 28622202, 28625862, 28627033, 28628592
28632796, 28636532, 28639299, 28642469, 28642899, 28643654, 28643718
28644549, 28645570, 28646200, 28646939, 28649388, 28655209, 28663289
28663782, 28672457, 28673945, 28681153, 28689483, 28692103, 28692275
28694639, 28694872, 28696373, 28699321, 28703812, 28705231, 28707931
28708400, 28709063, 28710385, 28710469, 28710663, 28710734, 28714461
28715655, 28715727, 28718469, 28719348, 28720204, 28720418, 28721497

28722229, 28730079, 28730253, 28734355, 28740708, 28742555, 28745367
28749853, 28752923, 28755011, 28755846, 28758722, 28760206, 28765983
28767240, 28769456, 28771947, 28772390, 28772816, 28774416, 28776431
28776811, 28777214, 28778754, 28781599, 28781754, 28785321, 28785531
28789531, 28791852, 28793062, 28795551, 28795734, 28800508, 28802734
28804517, 28805242, 28808314, 28808652, 28808656, 28810381, 28811560
28813931, 28815123, 28815355, 28815557, 28817449, 28819640, 28820669
28821847, 28824482, 28827682, 28831971, 28833912, 28835937, 28836716
28838385, 28844738, 28845346, 28846759, 28847541, 28847572, 28849776
28850084, 28852325, 28854004, 28855520, 28855922, 28857552, 28861861
28862532, 28863263, 28863432, 28863487, 28865569, 28867992, 28870496
28871040, 28872645, 28873575, 28874416, 28875089, 28876253, 28876639
28876926, 28877252, 28878865, 28881191, 28881848, 28882784, 28884931
28887305, 28888083, 28888327, 28889730, 28892794, 28897512, 28899663
28900506, 28901126, 28905390, 28905457, 28905615, 28907196, 28910498
28911140, 28912691, 28915561, 28917080, 28918429, 28919145, 28922227
28922532, 28922608, 28925250, 28925460, 28925634, 28925880, 28927452
28932914, 28933158, 28935293, 28935956, 28936114, 28937717, 28938422
28938698, 28940179, 28940281, 28940472, 28941901, 28942455, 28942694
28945421, 28945994, 28946233, 28949888, 28950868, 28951332, 28951533
28952168, 28954762, 28955606, 28955883, 28956908, 28957292, 28957723
28958088, 28959493, 28960863, 28962775, 28963036, 28965084, 28965095
28965231, 28965376, 28966444, 28968779, 28974083, 28974999, 28977322
28980448, 28981871, 28983095, 28983486, 28985478, 28986207, 28986231
28986257, 28986326, 28986481, 28988482, 28988864, 28989306, 28993295
28993353, 28994307, 28994542, 28995287, 28996376, 29000000, 29001305
29001888, 29002488, 29002784, 29002927, 29003207, 29003407, 29003617
29003738, 29006318, 29006621, 29007321, 29007353, 29007775, 29008035
29008669, 29009513, 29010126, 29011936, 29012609, 29013475, 29013832
29014076, 29015118, 29016294, 29017265, 29018655, 29018680, 29019121
29021063, 29021352, 29022986, 29024054, 29024448, 29024552, 29024732
29024876, 29026154, 29026582, 29026606, 29027456, 29027694, 29027933
29027940, 29031575, 29031600, 29032234, 29032276, 29032457, 29032607
29033052, 29033145, 29033200, 29033280, 29034587, 29036278, 29037290
29038528, 29038728, 29039089, 29039510, 29040739, 29043554, 29043651
29043725, 29044763, 29044954, 29046482, 29047127, 29047850, 29048178
29048289, 29048498, 29048605, 29048728, 29050357, 29050560, 29050765
29050886, 29051702, 29051953, 29052726, 29052850, 29053783, 29055644
29056024, 29056270, 29056560, 29056767, 29056894, 29058476, 29059011
29060216, 29061016, 29061959, 29062692, 29062848, 29062860, 29062868
29110526, 29110783, 29110790, 29110797, 29110802, 29110805, 29111598
29111631, 29112455, 29113282, 29113305, 29115857, 29117337, 29117526
29117642, 29118543, 29119077, 29120223, 29122224, 29122254, 29123297
29123432, 29123444, 29123482, 29124368, 29125036, 29125374, 29125380
29125708, 29125786, 29126345, 29127957, 29128693, 29128935, 29129450
29129476, 29129497, 29129712, 29130219, 29131539, 29131772, 29132456
29132869, 29132938, 29133470, 29134447, 29135383, 29135649, 29136111
29138641, 29139070, 29139727, 29139761, 29139956, 29141316, 29141341
29141685, 29142609, 29142667, 29143516, 29144995, 29145214, 29145730
29146157, 29146810, 29147849, 29149170, 29149829, 29150338, 29151520
29152357, 29152603, 29152752, 29154631, 29154636, 29154725, 29154829
29155099, 29157051, 29157389, 29158680, 29158899, 29159216, 29159661
29159909, 29159936, 29160174, 29160462, 29161597, 29161923, 29162095
29163073, 29163156, 29163415, 29163437, 29163524, 29163567, 29164376
29165682, 29167111, 29167342, 29167374, 29167940, 29168137, 29168219
29168433, 29169073, 29169215, 29169540, 29169739, 29170232, 29170717
29171683, 29171942, 29172618, 29172826, 29173140, 29173373, 29173618
29173817, 29174004, 29174753, 29176318, 29177466, 29177543, 29177886
29178385, 29179097, 29180313, 29180455, 29180559, 29180893, 29181078
29181153, 29181231, 29181568, 29181620, 29181743, 29181923, 29182019
29182517, 29182901, 29182920, 29183298, 29183912, 29184297, 29184666
29185193, 29186091, 29186456, 29186605, 29188255, 29189302, 29189307
29189889, 29190235, 29190474, 29190663, 29190740, 29191541, 29191827
29192419, 29192468, 29192685, 29193207, 29194205, 29194367, 29194493
29194827, 29194981, 29195279, 29195337, 29195758, 29196725, 29198092
29198913, 29199635, 29199733, 29200316, 29200700, 29201143, 29201494

29201539, 29201695, 29201787, 29202104, 29202461, 29202850, 29203122
29203166, 29203227, 29203425, 29203443, 29203604, 29205281, 29205323
29205419, 29205463, 29205767, 29205918, 29206109, 29206605, 29207073
29208260, 29208732, 29209545, 29210577, 29210610, 29210624, 29210683
29211457, 29211724, 29212012, 29212433, 29212611, 29213320, 29213351
29213613, 29213641, 29213775, 29213850, 29213879, 29213893, 29214561
29214960, 29216312, 29216723, 29216746, 29216984, 29217294, 29217472
29217828, 29217848, 29218570, 29219205, 29219273, 29219627, 29220079
29221248, 29221891, 29221942, 29222031, 29222784, 29223833, 29223859
29223967, 29224065, 29224294, 29224605, 29225076, 29225168, 29225758
29225861, 29227602, 29228869, 29229164, 29229754, 29229839, 29229844
29229955, 29230252, 29230565, 29231133, 29232117, 29232154, 29232449
29233415, 29233810, 29233953, 29234123, 29235934, 29236573, 29237538
29237575, 29237744, 29240307, 29240759, 29241345, 29241651, 29242017
29242884, 29242906, 29243749, 29243958, 29244495, 29244766, 29244968
29245063, 29245137, 29245160, 29246163, 29247415, 29247712, 29247906
29248495, 29248552, 29248723, 29248835, 29248858, 29249289, 29249412
29249583, 29249991, 29250059, 29250317, 29251259, 29251564, 29253184
29253871, 29254031, 29254623, 29254930, 29255178, 29255273, 29255431
29255435, 29255616, 29255718, 29255973, 29256426, 29259119, 29259320
29260224, 29260452, 29260956, 29261547, 29261548, 29261695, 29261906
29262512, 29262887, 29265448, 29266248, 29266899, 29267292, 29268412
29269171, 29269228, 29269825, 29270585, 29271019, 29273360, 29273539
29273570, 29273735, 29273812, 29273847, 29274428, 29274564, 29274627
29275461, 29276272, 29277317, 29278218, 29278684, 29279658, 29279751
29279854, 29281527, 29281691, 29281796, 29282090, 29282233, 29282666
29282898, 29285197, 29285453, 29285503, 29285621, 29285788, 29285956
29286037, 29286229, 29287130, 29287705, 29290110, 29290235, 29292232
29292837, 29293072, 29293574, 29293806, 29294753, 29296257, 29297863
29297915, 29298220, 29299049, 29299082, 29299830, 29299844, 29301463
29301566, 29302963, 29303918, 29304314, 29304781, 29306226, 29306713
29307090, 29307109, 29307638, 29309698, 29311336, 29311528, 29311588
29311927, 29312310, 29312672, 29312734, 29312753, 29312889, 29313347
29313417, 29313525, 29314539, 29314636, 29317756, 29318410, 29319441
29321489, 29323946, 29324568, 29324735, 29325087, 29325105, 29325257
29325765, 29325993, 29327044, 29327892, 29329042, 29329087, 29329675
29329807, 29330361, 29330791, 29331066, 29331209, 29331380, 29331493
29332292, 29332395, 29332763, 29332771, 29333500, 29336843, 29337294
29337310, 29337742, 29338315, 29338348, 29338453, 29338780, 29338913
29339101, 29339155, 29339299, 29341209, 29343086, 29343156, 29343861
29344541, 29345937, 29346057, 29346211, 29346943, 29347620, 29348176
29350052, 29350762, 29350886, 29351044, 29351386, 29351662, 29351716
29351735, 29351749, 29351771, 29352298, 29352724, 29352867, 29352947
29353271, 29353432, 29353718, 29353821, 29353960, 29355654, 29356547
29356704, 29356711, 29356752, 29356782, 29357821, 29358509, 29358828
29360252, 29360285, 29360467, 29360672, 29360775, 29360911, 29360950
29361319, 29361472, 29361801, 29362596, 29363151, 29364171, 29364177
29366406, 29366940, 29367019, 29367561, 29367971, 29368253, 29368310
29368725, 29372069, 29372541, 29373418, 29373588, 29374179, 29374604
29375355, 29375941, 29375984, 29376346, 29377804, 29377986, 29378029
29378287, 29378834, 29378913, 29379299, 29379381, 29379750, 29379978
29380527, 29381000, 29382296, 29382641, 29382784, 29382815, 29383695
29384781, 29384854, 29384864, 29385339, 29385429, 29385652, 29386502
29386557, 29386635, 29386660, 29387073, 29387274, 29387310, 29387337
29388020, 29388072, 29388094, 29388524, 29388830, 29389408, 29389889
29390011, 29390435, 29390785, 29391030, 29391237, 29391301, 29391438
29391849, 29391925, 29392554, 29392966, 29393291, 29393649, 29394014
29394140, 29394749, 29395657, 29397954, 29397996, 29398488, 29398863
29399046, 29399100, 29399121, 29399336, 29399938, 29402110, 29402131
29404483, 29405012, 29405462, 29405651, 29405996, 29407488, 29407804
29408853, 29409149, 29409455, 29410311, 29410834, 29411037, 29411469
29411931, 29412066, 29412269, 29413360, 29413382, 29413517, 29413544
29413634, 29413956, 29416688, 29416700, 29417084, 29417173, 29417719
29417884, 29418165, 29420254, 29420834, 29421059, 29423003, 29423016
29423156, 29423826, 29424999, 29426241, 29426320, 29428230, 29429017
29429087, 29429264, 29429466, 29429566, 29430524, 29430866, 29431192

29431485, 29432176, 29434301, 29434869, 29435474, 29435652, 29436454
29436514, 29436522, 29436727, 29437594, 29437712, 29438150, 29438277
29438736, 29439522, 29441196, 29442936, 29443187, 29443250, 29444072
29444282, 29444602, 29445548, 29446669, 29448498, 29449477, 29449845
29449852, 29450162, 29450193, 29450421, 29450812, 29450936, 29451386
29452251, 29452576, 29452936, 29452953, 29454978, 29455424, 29455773
29456538, 29456714, 29457312, 29457319, 29457370, 29457502, 29457807
29457978, 29460252, 29461420, 29461791, 29461971, 29462594, 29462767
29462957, 29463047, 29463528, 29463798, 29464616, 29464779, 29465047
29465177, 29466674, 29467622, 29469565, 29470291, 29471832, 29471860
29472618, 29473708, 29476473, 29481584, 29483452, 29483532, 29483626
29483672, 29483685, 29483712, 29483723, 29483771, 29485099, 29486181
29486848, 29487150, 29487189, 29488894, 29489436, 29489546, 29490256
29492127, 29492939, 29493122, 29494245, 29495057, 29495684, 29497311
29497588, 29497696, 29498198, 29500257, 29500826, 29500963, 29501218
29502561, 29503543, 29503631, 29503827, 29504492, 29504682, 29505668
29506942, 29507270, 29507616, 29508681, 29509777, 29510278, 29511064
29511611, 29511980, 29512125, 29512890, 29514479, 29515134, 29515240
29515476, 29515766, 29515834, 29516300, 29516727, 29516766, 29517168
29517883, 29519131, 29521187, 29521688, 29521748, 29521862, 29522358
29522561, 29522662, 29523055, 29523216, 29523511, 29524599, 29524985
29525366, 29525467, 29525886, 29526966, 29527595, 29527610, 29528368
29529147, 29530440, 29530515, 29530812, 29530909, 29531654, 29531836
29532532, 29536342, 29536445, 29537829, 29538631, 29540327, 29541742
29541769, 29542084, 29542449, 29542580, 29542643, 29543034, 29543956
29546817, 29547010, 29547867, 29548413, 29548427, 29548592, 29548687
29548722, 29549040, 29549071, 29549104, 29549154, 29549730, 29552773
29553141, 29554092, 29557144, 29557261, 29557336, 29557556, 29558238
29558452, 29558975, 29559187, 29559395, 29559446, 29559908, 29559981
29564592, 29564593, 29565611, 29579919, 29580394, 29580983, 29581771
29584261, 29584693, 29586143, 29587299, 29587765, 29589544, 29591343
29592011, 29592215, 29597536, 29597754, 29598039, 29598046, 29598226
29598233, 29599008, 29599300, 29601461, 29602831, 29603460, 29603884
29604002, 29604257, 29606261, 29607136, 29607797, 29608000, 29608023
29610506, 29611020, 29611991, 29614206, 29614987, 29615824, 29616244
29616414, 29618074, 29618190, 29620042, 29622936, 29623323, 29625065
29625804, 29625876, 29626154, 29626732, 29628200, 29629430, 29629650
29629681, 29629745, 29631749, 29632095, 29632265, 29632611, 29633697
29633753, 29633936, 29634643, 29635427, 29635717, 29635990, 29637362
29637526, 29637560, 29638285, 29641736, 29643721, 29644426, 29644464
29645167, 29645349, 29647176, 29648928, 29651183, 29651520, 29653132
29653246, 29655164, 29655668, 29656400, 29656819, 29656843, 29657399
29657422, 29657744, 29657960, 29658056, 29661028, 29661065, 29661722
29663191, 29663368, 29663494, 29663601, 29664087, 29664161, 29665940
29667527, 29667994, 29668005, 29669413, 29670782, 29671363, 29672507
29675446, 29676089, 29677051, 29677173, 29677733, 29677927, 29679856
29681987, 29683039, 29683211, 29684518, 29685137, 29685276, 29687214
29687220, 29687459, 29687718, 29687727, 29687763, 29688867, 29689145
29689255, 29692694, 29694869, 29695425, 29695821, 29695841, 29696310
29700125, 29700460, 29700770, 29703932, 29705793, 29707099, 29707493
29707896, 29708876, 29708915, 29710188, 29710858, 29713810, 29715220
29715703, 29716194, 29716227, 29716491, 29716602, 29716871, 29717659
29717901, 29719146, 29720133, 29721418, 29721576, 29722167, 29724658
29725476, 29725781, 29726695, 29738400, 29739576, 29741319, 29741976
29742223, 29744225, 29746962, 29747493, 29747648, 29747653, 29748285
29748336, 29748513, 29749471, 29750673, 29751094, 29753244, 29754196
29754951, 29755821, 29756274, 29756444, 29757099, 29757264, 29757651
29757687, 29758203, 29758217, 29758661, 29761678, 29761837, 29761911
29763158, 29765035, 29765393, 29766207, 29766435, 29766503, 29766679
29768899, 29770750, 29771032, 29771242, 29773197, 29773842, 29774362
29775393, 29779196, 29780140, 29782211, 29782823, 29782866, 29784106
29785239, 29785311, 29787292, 29787766, 29789911, 29791152, 29791880
29792213, 29793318, 29794174, 29794462, 29795712, 29795957, 29796378
29797726, 29802695, 29804875, 29805772, 29806390, 29807964, 29809792
29809837, 29812084, 29812489, 29813503, 29813650, 29813671, 29815341
29815713, 29817278, 29817547, 29817784, 29821582, 29822714, 29825525

29827647, 29827852, 29831196, 29834506, 29836096, 29838485, 29838773
29839715, 29840619, 29841267, 29841687, 29843277, 29843692, 29843831
29844226, 29844275, 29845530, 29846126, 29846645, 29846688, 29848084
29848849, 29849100, 29850930, 29851733, 29853485, 29856859, 29858121
29858376, 29859068, 29860994, 29861075, 29864203, 29864261, 29865188
29865590, 29865658, 29869086, 29869404, 29869887, 29869906, 29870065
29871098, 29871312, 29872401, 29872937, 29872983, 29873665, 29874090
29874761, 29875459, 29875565, 29876358, 29876989, 29877608, 29878076
29881050, 29881478, 29881575, 29881643, 29881839, 29882427, 29882454
29882729, 29884958, 29885890, 29886809, 29887111, 29888621, 29889184
29889358, 29890740, 29891075, 29891853, 29891916, 29892604, 29893132
29896510, 29897418, 29897863, 29900203, 29900824, 29901419, 29902299
29902327, 29902330, 29902659, 29903190, 29903299, 29903357, 29903454
29904002, 29906678, 29907942, 29908389, 29908777, 29909658, 29910402
29912286, 29913805, 29913966, 29914449, 29914544, 29915217, 29915848
29916975, 29919789, 29920025, 29920376, 29920804, 29921318, 29922225
29923452, 29924181, 29926466, 29927756, 29928210, 29928427, 29928564
29930457, 29932202, 29932430, 29932780, 29934052, 29937565, 29937956
29938225, 29939400, 29939795, 29940373, 29942096, 29942275, 29942554
29943670, 29943879, 29944035, 29944660, 29944963, 29945645, 29946388
29947145, 29950220, 29951620, 29951759, 29956016, 29956222, 29957412
29957493, 29958925, 29960884, 29961353, 29961609, 29961847, 29962160
29962248, 29962834, 29962927, 29962939, 29965052, 29965603, 29965888
29966768, 29967223, 29968085, 29969557, 29970081, 29970298, 29971027
29971481, 29971888, 29971936, 29971951, 29972176, 29973012, 29989783
29989845, 29991257, 29993717, 29997326, 29997553, 29997937, 30000664
30001331, 30003187, 30006159, 30006985, 30007450, 30007797, 30008125
30008198, 30008214, 30009710, 30012181, 30015070, 30017836, 30018017
30018903, 30019864, 30024618, 30025814, 30026016, 30027614, 30028599
30029519, 30029806, 30031027, 30032376, 30033040, 30033547, 30034456
30035598, 30038392, 30039800, 30039959, 30040157, 30041501, 30042490
30043398, 30043610, 30043930, 30044507, 30045389, 30045484, 30046497
30047531, 30047702, 30047765, 30047931, 30049966, 30051176, 30051783
30051804, 30052928, 30053036, 30053501, 30053748, 30054980, 30056058
30057718, 30057799, 30058149, 30058453, 30059106, 30059109, 30060267
30060330, 30062364, 30062819, 30064268, 30066352, 30067565, 30068871
30071446, 30072905, 30073314, 30073744, 30074296, 30074349, 30074469
30074472, 30074820, 30075037, 30076058, 30076197, 30076253, 30078675
30078934, 30079949, 30080266, 30081546, 30081580, 30082145, 30083100
30083216, 30083488, 30083807, 30084971, 30085897, 30086596, 30086992
30090568, 30092280, 30092859, 30095591, 30095952, 30097092, 30097115
30098251, 30099302, 30099420, 30099454, 30100354, 30101186, 30103551
30103553, 30104378, 30104555, 30106748, 30109365, 30110224, 30110370
30110518, 30114477, 30114489, 30114534, 30116085, 30116203, 30116854
30117209, 30117335, 30117593, 30118261, 30118279, 30120608, 30122583
30125765, 30126145, 30127522, 30127805, 30127904, 30128047, 30131286
30131645, 30132708, 30133841, 30134746, 30135396, 30135731, 30135942
30136346, 30139392, 30142907, 30143470, 30143593, 30146593, 30146969
30147473, 30147928, 30149035, 30149658, 30150606, 30153552, 30153885
30154633, 30155814, 30155837, 30159329, 30159511, 30159536, 30159752
30159760, 30160625, 30163243, 30164714, 30165493, 30165503, 30165897
30169254, 30170104, 30172925, 30173113, 30173556, 30174401, 30175291
30177597, 30178250, 30178839, 30178990, 30179644, 30180208, 30181756
30183696, 30183715, 30183920, 30184102, 30185852, 30186319, 30186476
30186706, 30187866, 30189516, 30190090, 30191274, 30193165, 30193505
30193736, 30194612, 30194710, 30194972, 30195667, 30195668, 30195684
30196195, 30196358, 30196629, 30198861, 30198905, 30200034, 30200132
30200237, 30200680, 30200758, 30202349, 30202388, 30203929, 30204042
30204542, 30206493, 30206675, 30207473, 30208327, 30208723, 30209736
30210884, 30213031, 30213540, 30215130, 30215302, 30215351, 30217206
30217982, 30218044, 30218317, 30219222, 30221237, 30221298, 30222512
30223712, 30223847, 30224650, 30224868, 30224950, 30225265, 30225443
30225718, 30225844, 30226244, 30228567, 30229683, 30232638, 30235919
30235979, 30236554, 30238211, 30238715, 30239480, 30240010, 30240547
30241567, 30241920, 30242120, 30242724, 30243216, 30244340, 30246053
30246179, 30247305, 30249432, 30251003, 30252005, 30252098, 30252156

30252458, 30252977, 30253035, 30253090, 30253608, 30253705, 30253835
30254206, 30254525, 30254726, 30255143, 30255528, 30256542, 30257412
30264405, 30265523, 30265608, 30265615, 30265703, 30266791, 30267155
30269428, 30269748, 30270647, 30270744, 30271114, 30272329, 30274090
30274188, 30274324, 30275548, 30275569, 30275578, 30276243, 30277120
30277451, 30277733, 30281428, 30282501, 30282591, 30283296, 30283577
30283579, 30283581, 30283932, 30284219, 30284369, 30285457, 30285843
30288343, 30288491, 30289458, 30293345, 30294267, 30294671, 30295110
30295549, 30295808, 30297905, 30299367, 30299817, 30299934, 30300030
30300538, 30305264, 30305395, 30305568, 30305880, 30307814, 30307883
30308368, 30308624, 30308772, 30308947, 30309098, 30309798, 30311826
30312094, 30312546, 30313848, 30314079, 30314198, 30314837, 30316667
30316897, 30317209, 30318638, 30318943, 30319080, 30319099, 30322980
30323658, 30323849, 30324180, 30325407, 30326882, 30327149, 30328168
30328690, 30329209, 30330123, 30331759, 30332505, 30334484, 30334563
30335127, 30335832, 30335987, 30336032, 30336742, 30339103, 30341713
30342371, 30342878, 30344614, 30345201, 30345432, 30345809, 30346330
30346867, 30349714, 30352581, 30352623, 30355490, 30357463, 30357897
30360383, 30362003, 30362850, 30363716, 30364329, 30364613, 30365745
30367193, 30368048, 30368482, 30368534, 30368668, 30371264, 30371623
30371909, 30372081, 30373550, 30374345, 30374570, 30374739, 30375109
30376986, 30377347, 30381207, 30381525, 30382982, 30383286, 30384121
30384152, 30389229, 30389414, 30389507, 30391272, 30392011, 30393110
30393653, 30394738, 30394974, 30396946, 30397100, 30398257, 30398422
30399906, 30402386, 30403763, 30403881, 30403902, 30403989, 30404117
30408515, 30408808, 30409207, 30409339, 30409590, 30412188, 30412863
30412885, 30412921, 30413137, 30414679, 30414714, 30416034, 30417648
30417732, 30419024, 30421439, 30421476, 30422487, 30423135, 30423218
30424347, 30430921, 30431274, 30431504, 30431698, 30431703, 30431717
30433177, 30437149, 30441687, 30441959, 30442749, 30442884, 30443393
30446583, 30447060, 30447589, 30448182, 30448917, 30449194, 30450787
30453442, 30454090, 30458568, 30458593, 30460922, 30461458, 30463938
30464250, 30464655, 30466081, 30469777, 30472891, 30474167, 30474774
30475115, 30476768, 30477588, 30477685, 30477691, 30477767, 30479715
30480872, 30483140, 30483521, 30484042, 30484801, 30485255, 30486436
30487387, 30490014, 30493518, 30495035, 30495133, 30495483, 30496957
30497057, 30500224, 30500297, 30500344, 30500582, 30501574, 30502415
30503943, 30505497, 30506794, 30506991, 30507032, 30509277, 30510347
30510527, 30513285, 30513848, 30515886, 30516868, 30517214, 30519188
30522285, 30522998, 30523137, 30523538, 30523601, 30523750, 30528547
30528704, 30529940, 30533132, 30534351, 30534549, 30534662, 30534827
30537405, 30537533, 30539519, 30540109, 30540407, 30544247, 30544595
30545281, 30545556, 30549637, 30549789, 30549881, 30551000, 30551123
30556581, 30556807, 30557386, 30560513, 30561590, 30564139, 30564343
30573703, 30576393, 30577071, 30579051, 30581448, 30582500, 30588738
30591028, 30592859, 30595114, 30595860, 30596488, 30598682, 30598746
30598919, 30599405, 30599407, 30600173, 30600184, 30602230, 30605215
30609799, 30612199, 30613937, 30613971, 30614411, 30619525, 30619787
30622528, 30623138, 30623142, 30624792, 30624864, 30625121, 30628899
30629643, 30629799, 30631393, 30633259, 30633938, 30635183, 30635302
30635326, 30637270, 30637319, 30641755, 30641900, 30644889, 30651231
30651621, 30651674, 30652515, 30652853, 30654558, 30657365, 30657624
30657706, 30657875, 30658702, 30661939, 30662651, 30663646, 30668407
30671813, 30674959, 30676209, 30681462, 30681516, 30686131, 30690686
30691604, 30691731, 30691857, 30692473, 30698289, 30703610, 30704826
30708735, 30710807, 30714151, 30714715, 30718862, 30719419, 30720736
30720844, 30722705, 30724679, 30724881, 30727701, 30729604, 30730026
30732711, 30734707, 30735736, 30740997, 30741263, 30749722, 30750991
30751968, 30758943, 30761878, 30763272, 30763305, 30763639, 30765486
30769312, 30773164, 30776416, 30776929, 30778855, 30781032, 30781041
30783551, 30785101, 30789904, 30790441, 30801296, 30801510, 30803210
30808869, 30812574, 30814285, 30815852, 30816938, 30823744, 30825391
30825419, 30825656, 30826474, 30828350, 30829779, 30833454, 30834110
30838605, 30847442, 30848773, 30851951, 30855101, 30857721, 30858919
30866141, 30866988, 30871716, 30871792, 30880913, 30881407, 30883877
30886188, 30887501, 30890720, 30890971, 30896620, 30904672, 30906274

30909918, 30913399, 30914272, 30914674, 30919691, 30922936, 30922996
30923517, 30923940, 30927821, 30930339, 30936831, 30937410, 30939934
30940259, 30941056, 30944643, 30945005, 30952104, 30953836, 30957739
30964194, 30968737, 30970518, 30972841, 30972966, 30973113, 30973137
30973143, 30980317, 30980733, 30987088, 30990034, 30992597, 30993198
30996991, 30998759, 30998847, 31001017, 31001455, 31001859, 31004077
31004719, 31004844, 31008240, 31010976, 31013127, 31016413, 31019249
31021157, 31022858, 31029936, 31031955, 31032904, 31039627, 31039928
31051075, 31062010, 31066265, 31077365, 31084921, 31094688, 31100172
31106577, 31113089, 31118809, 31119057, 31134430, 31153120, 31156383
31172207, 31177193, 31182793, 31193936, 31200845, 31305624, 31306261
31335037, 31335142, 31338673, 31359215, 31383396, 31393600, 31414023
31414524

Version 19.0.0.0.ru-2020-04.rur-2020-04.r1

Version 19.0.0.0.ru-2020-04.rur-2020-04.r1 includes the following:

- Patch 30869156: Database Release Update 19.7.0.0.200414
- Patch 30805684: Oracle JVM Release Update 19.7.0.0.200414
- Patch 29997937: DSTV34 UPDATE for RDBMS (TZDATA2019B)
- Patch 29997959: DSTV34 UPDATE for OJVM (TZDATA2019B)
- PreUpgrade Jar: preupgrade_19_cbuild_5_lf.zip
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- Patch 30528704: 19C RMAN RECOVER DATABASE USE REDO LOG INSTEAD OF ARCHIVELOG AFTER APPLYING OCT DATABASE RU
- Support for [Purging the recycle bin \(p. 1062\)](#)
- Support for [Generating performance reports with Automatic Workload Repository \(AWR\) \(p. 1053\)](#) using the rdsadmin.rdsadmin_diagnostic_util package

Combined patches for version 19.0.0.0.ru-2020-04.rur-2020-04.r1, released April 2020

Bugs fixed:

30533132, 30312546, 29924181, 29549154, 30937410, 29970081, 8476681
14735102, 17428816, 19080742, 19697993, 20313356, 21374587, 21965541
23296836, 23606241, 24687075, 24971456, 25756945, 25806201, 25883179
25986062, 25997810, 26284288, 26476244, 26611353, 26668264, 26739322
26777814, 26872233, 27036163, 27044169, 27101798, 27126122, 27126938
27166935, 27195935, 27244999, 27254335, 27359766, 27369515, 27378053
27406105, 27411022, 27423500, 27439716, 27453490, 27458357, 27489107
27582210, 27666312, 27710072, 27729678, 27801144, 27846298, 27880025
27929509, 27934711, 27935464, 27941110, 27957203, 27967484, 28064977
28072567, 28109326, 28125947, 28129791, 28181021, 28189466, 28204262
28205555, 28209985, 28210681, 28263142, 28271258, 28271693, 28276054
28279456, 28294563, 28313275, 28319114, 28326928, 28350595, 28371123
28373960, 28375383, 28379065, 28381939, 28386259, 28390273, 28395302
28397317, 28402823, 28410431, 28431445, 28435333, 28454215, 28463226
28475242, 28484299, 28489419, 28492006, 28498976, 28502773, 28513333
28521330, 28530171, 28534475, 28535127, 28538439, 28542455, 28546290
28547068, 28547926, 28558645, 28561704, 28564479, 28565296, 28567417
28567819, 28569897, 28572407, 28572533, 28572544, 28572667, 28572834

28578945, 28587723, 28589509, 28593682, 28594086, 28597221, 28601957
28605066, 28606598, 28608211, 28612239, 28620697, 28622202, 28625862
28627033, 28628592, 28632796, 28636532, 28639299, 28643654, 28643718
28644549, 28645570, 28646200, 28646939, 28649388, 28655209, 28663782
28672457, 28673945, 28681153, 28692103, 28692275, 28694639, 28694872
28696373, 28705231, 28707931, 28708400, 28709063, 28710385, 28710469
28710734, 28714461, 28715727, 28718469, 28719348, 28720204, 28720418
28721497, 28722229, 28730079, 28734355, 28740708, 28742555, 28749853
28752923, 28755011, 28758722, 28760206, 28765983, 28767240, 28769456
28771947, 28772390, 28774416, 28776811, 28777214, 28781754, 28785531
28789531, 28791852, 28795551, 28795734, 28802734, 28804517, 28808656
28810381, 28811560, 28815123, 28815355, 28817449, 28819640, 28820669
28821847, 28824482, 28827682, 28831971, 28833912, 28835937, 28836716
28838385, 28844738, 28845346, 28846759, 28849776, 28854004, 28855520
28855922, 28857552, 28862532, 28863263, 28863432, 28863487, 28865569
28867992, 28872645, 28873575, 28875089, 28876253, 28876639, 28876926
28878865, 28882784, 28884931, 28887305, 28888327, 28889730, 28892794
28897512, 28899663, 28901126, 28905457, 28905615, 28907196, 28912691
28915561, 28917080, 28918429, 28919145, 28922227, 28922532, 28922608
28925634, 28925880, 28927452, 28932914, 28933158, 28935293, 28936114
28937717, 28938422, 28938698, 28940179, 28940281, 28941901, 28942455
28945421, 28945994, 28946233, 28949888, 28950868, 28951533, 28952168
28954762, 28955606, 28955883, 28956908, 28957292, 28957723, 28958088
28959493, 28960863, 28962775, 28965084, 28965095, 28965231, 28965376
28966444, 28974083, 28977322, 28981871, 28983095, 28983486, 28985478
28986207, 28986231, 28986257, 28986326, 28986481, 28988482, 28989306
28993295, 28993353, 28994307, 28994542, 28996376, 29000000, 29001305
29001888, 29002488, 29002784, 29002927, 29003407, 29003738, 29006318
29006621, 29007321, 29007353, 29007775, 29008035, 29008669, 29009513
29010126, 29011936, 29012609, 29013475, 29013832, 29014076, 29015118
29017265, 29018655, 29019121, 29021063, 29021352, 29022986, 29024054
29024552, 29024732, 29026582, 29026606, 29027456, 29027694, 29027940
29031575, 29031600, 29032234, 29032276, 29032457, 29032607, 29033052
29033145, 29033200, 29033280, 29034587, 29037290, 29038528, 29038728
29039089, 29039510, 29040739, 29043554, 29043651, 29043725, 29044763
29044954, 29046482, 29047850, 29048178, 29048289, 29048498, 29048605
29050357, 29050560, 29050765, 29050886, 29051702, 29051953, 29052726
29053783, 29056024, 29056270, 29056560, 29056767, 29056894, 29059011
29060216, 29061016, 29061959, 29062692, 29062848, 29062860, 29062868
29110526, 29110783, 29110790, 29110797, 29110802, 29110805, 29111598
29113282, 29113305, 29115857, 29117526, 29117642, 29118543, 29119077
29120223, 29122224, 29122254, 29123297, 29123432, 29123482, 29124368
29125036, 29125374, 29125380, 29126345, 29127957, 29128693, 29128935
29129450, 29129497, 29129712, 29130219, 29131539, 29132869, 29132938
29133470, 29134447, 29135383, 29135649, 29136111, 29138641, 29139761
29139956, 29141316, 29141341, 29141685, 29142609, 29142667, 29143516
29144995, 29145214, 29145730, 29149829, 29150338, 29151520, 29152357
29154725, 29155099, 29157051, 29157389, 29158680, 29158899, 29159909
29159936, 29160174, 29161597, 29162095, 29163073, 29163156, 29163415
29163437, 29163524, 29163567, 29164376, 29167111, 29167342, 29167374
29167940, 29168137, 29168219, 29168433, 29169073, 29169215, 29170232
29171683, 29171942, 29172618, 29172826, 29173140, 29173373, 29173817
29174004, 29174753, 29176318, 29177466, 29177543, 29177886, 29178385
29179097, 29180313, 29180455, 29180559, 29180893, 29181153, 29181231
29181620, 29181743, 29181923, 29182019, 29182517, 29182901, 29183912
29184297, 29184666, 29185193, 29186456, 29189302, 29189307, 29189889
29190235, 29190474, 29190663, 29190740, 29191541, 29192419, 29192468
29192685, 29193207, 29194205, 29194367, 29194493, 29194827, 29194981
29195279, 29195337, 29195758, 29196725, 29198092, 29198913, 29199635
29199733, 29200316, 29200700, 29201494, 29201539, 29201787, 29202104
29202461, 29202850, 29203122, 29203166, 29203227, 29203425, 29203443
29203604, 29205281, 29205323, 29205419, 29205463, 29205767, 29205918
29206109, 29206605, 29207073, 29208260, 29208732, 29211457, 29211724
29212012, 29212433, 29212611, 29213320, 29213351, 29213613, 29213775
29213850, 29213879, 29214561, 29214960, 29216312, 29216723, 29216746
29216984, 29217294, 29217472, 29217828, 29217848, 29218570, 29219205

29219273, 29220079, 29221248, 29221891, 29221942, 29222031, 29222784
29223833, 29223859, 29223967, 29224065, 29224605, 29225076, 29225168
29225758, 29227602, 29228869, 29229164, 29229754, 29229844, 29229955
29230252, 29230565, 29231133, 29232117, 29232154, 29232449, 29233415
29233810, 29233953, 29234123, 29236573, 29237538, 29237575, 29237744
29240307, 29240759, 29241345, 29241651, 29242017, 29242884, 29243958
29245137, 29245160, 29246163, 29247415, 29247712, 29247906, 29248495
29248552, 29248835, 29248858, 29249289, 29249412, 29249991, 29250059
29250317, 29251259, 29253184, 29253871, 29254031, 29254930, 29255178
29255273, 29255431, 29255435, 29255973, 29256426, 29259119, 29259320
29260452, 29260956, 29261547, 29261548, 29261906, 29262512, 29262887
29265448, 29266248, 29266899, 29267292, 29268412, 29269171, 29269228
29269825, 29270585, 29273539, 29273570, 29273735, 29273812, 29273847
29274428, 29274564, 29274627, 29275461, 29276272, 29277317, 29278218
29278684, 29279658, 29279751, 29279854, 29281527, 29281691, 29281796
29282233, 29282898, 29285197, 29285503, 29285788, 29285956, 29286037
29286229, 29287130, 29287705, 29292837, 29293072, 29293574, 29297863
29297915, 29298220, 29299049, 29299082, 29299844, 29301463, 29301566
29302963, 29303918, 29304781, 29306226, 29306713, 29307638, 29311528
29311588, 29312310, 29312672, 29312734, 29312753, 29312889, 29313347
29313417, 29313525, 29314539, 29314636, 29317756, 29318410, 29319441
29321489, 29323946, 29324568, 29324735, 29325087, 29325105, 29325257
29325765, 29325993, 29327044, 29327892, 29329042, 29329087, 29329807
29330361, 29331066, 29331209, 29331380, 29331493, 29332292, 29332395
29332763, 29332771, 29333500, 29336843, 29337310, 29337742, 29338315
29338348, 29338453, 29338780, 29338913, 29339101, 29339155, 29341209
29343086, 29343156, 29343861, 29345937, 29346057, 29346211, 29346943
29347620, 29348176, 29350052, 29350762, 29351044, 29351386, 29351662
29351716, 29351735, 29351749, 29351771, 29352298, 29352724, 29352867
29352947, 29353271, 29353432, 29353821, 29353960, 29355654, 29356547
29356704, 29356711, 29356752, 29356782, 29358509, 29358828, 29360252
29360285, 29360672, 29360911, 29360950, 29361319, 29361472, 29361801
29363151, 29364171, 29364177, 29366940, 29367019, 29367561, 29368253
29368310, 29372541, 29373418, 29373588, 29374179, 29375355, 29375941
29375984, 29376346, 29377804, 29377986, 29378029, 29378287, 29378834
29378913, 29379750, 29379978, 29382641, 29382784, 29382815, 29383695
29384781, 29384854, 29384864, 29385339, 29385429, 29385652, 29386502
29386635, 29386660, 29387073, 29387274, 29387310, 29388020, 29388072
29388094, 29388524, 29388830, 29389889, 29390011, 29390435, 29390785
29391030, 29391237, 29391438, 29391849, 29391925, 29392966, 29393291
29394014, 29394140, 29394749, 29395657, 29397954, 29397996, 29398488
29398863, 29399046, 29399100, 29399121, 29399336, 29399938, 29402131
29404483, 29405012, 29405462, 29405651, 29405996, 29407488, 29407804
29408853, 29409149, 29409455, 29410311, 29410834, 29411037, 29411469
29412066, 29412269, 29413382, 29413517, 29413544, 29413634, 29416688
29416700, 29417084, 29417173, 29417719, 29417884, 29418165, 29420834
29421059, 29423003, 29423016, 29423156, 29423826, 29424999, 29426241
29426320, 29429017, 29429087, 29429264, 29429466, 29429566, 29430524
29430866, 29431192, 29431485, 29432176, 29434301, 29435474, 29435652
29436454, 29436514, 29436522, 29436727, 29437594, 29437712, 29438150
29438277, 29438736, 29439522, 29441196, 29443187, 29443250, 29444072
29444282, 29444602, 29446669, 29448498, 29449477, 29449845, 29449852
29450162, 29450193, 29450421, 29450812, 29450936, 29451386, 29452251
29452576, 29452936, 29452953, 29454978, 29455424, 29456538, 29456714
29457312, 29457370, 29457502, 29457807, 29457978, 29460252, 29461420
29461791, 29462594, 29462767, 29462957, 29463047, 29463528, 29463798
29464616, 29464779, 29465177, 29466674, 29467622, 29469565, 29470291
29471832, 29471860, 29472618, 29473708, 29476473, 29481584, 29483452
29483532, 29483626, 29483672, 29483685, 29483712, 29483723, 29483771
29485099, 29486181, 29486848, 29487189, 29488894, 29489436, 29489546
29490256, 29492127, 29492939, 29493122, 29494245, 29495057, 29495684
29497311, 29497588, 29497696, 29498198, 29500257, 29500826, 29500963
29502561, 29503543, 29503631, 29503827, 29504492, 29504682, 29505668
29507270, 29507616, 29508681, 29509777, 29510278, 29511611, 29511980
29512890, 29514479, 29515134, 29515240, 29515476, 29515766, 29515834
29516300, 29516727, 29516766, 29517168, 29517883, 29519131, 29521187

29521688, 29521748, 29521862, 29522358, 29522561, 29522662, 29523055
29523216, 29523511, 29524599, 29524985, 29525467, 29525886, 29526966
29527595, 29527610, 29528368, 29529147, 29530440, 29530515, 29530812
29530909, 29531654, 29531836, 29532532, 29536342, 29536445, 29538631
29541742, 29541769, 29542084, 29542449, 29542580, 29542643, 29543034
29543956, 29546817, 29547010, 29547867, 29548427, 29548592, 29548687
29548722, 29549040, 29549071, 29549104, 29549730, 29552773, 29553141
29554092, 29557144, 29557261, 29557336, 29557556, 29558238, 29558452
29558975, 29559187, 29559446, 29559908, 29559981, 29564592, 29564593
29565611, 29580394, 29580983, 29581771, 29584261, 29584693, 29586143
29587765, 29589544, 29591343, 29592215, 29597536, 29597754, 29598039
29598046, 29598233, 29599008, 29599300, 29601461, 29602831, 29603460
29603884, 29604002, 29604257, 29607136, 29607797, 29608000, 29610506
29611020, 29611991, 29615824, 29616244, 29616414, 29618074, 29618190
29620042, 29622936, 29625065, 29625804, 29625876, 29626154, 29626732
29628200, 29629430, 29629650, 29629681, 29629745, 29631749, 29632095
29632265, 29632611, 29633697, 29633753, 29633936, 29634643, 29635427
29635717, 29635990, 29637362, 29637526, 29638285, 29641736, 29643721
29644464, 29645349, 29647176, 29648928, 29651183, 29651520, 29653132
29653246, 29655668, 29656400, 29656819, 29656843, 29657399, 29657422
29657744, 29657960, 29661028, 29661065, 29661722, 29663191, 29663368
29663494, 29663601, 29664087, 29664161, 29665940, 29667527, 29667994
29668005, 29669413, 29670782, 29671363, 29672507, 29675446, 29676089
29677051, 29677173, 29677733, 29677927, 29679856, 29681987, 29683039
29683211, 29684518, 29685137, 29685276, 29687214, 29687220, 29687459
29687718, 29687763, 29689145, 29689255, 29692694, 29694869, 29695425
29695841, 29696310, 29700125, 29700460, 29700770, 29703932, 29707099
29707493, 29707896, 29708915, 29710188, 29710858, 29713810, 29715220
29716194, 29716227, 29716491, 29716871, 29717659, 29719146, 29720133
29721418, 29721576, 29724658, 29725476, 29725781, 29726695, 29738400
29739576, 29741976, 29742223, 29744225, 29746962, 29747493, 29747648
29747653, 29748285, 29748336, 29748513, 29749471, 29750673, 29751094
29753244, 29754196, 29754951, 29755821, 29756274, 29756444, 29757099
29757264, 29757651, 29758203, 29758217, 29758661, 29761678, 29761837
29761911, 29763158, 29765035, 29765393, 29766207, 29766503, 29766679
29768899, 29770750, 29771032, 29771242, 29773197, 29773842, 29775393
29779196, 29780140, 29782211, 29782823, 29782866, 29784106, 29785239
29787292, 29787766, 29791152, 29791880, 29792213, 29793318, 29794174
29794462, 29795712, 29795957, 29796378, 29797726, 29802695, 29804875
29805772, 29807964, 29809792, 29809837, 29812489, 29813503, 29813650
29813671, 29815713, 29817278, 29817784, 29821582, 29825525, 29827647
29827852, 29831196, 29834506, 29836096, 29838485, 29838773, 29839715
29840619, 29841267, 29841687, 29843277, 29843692, 29843831, 29844226
29845530, 29846126, 29846445, 29846688, 29848084, 29848849, 29849100
29850930, 29851733, 29853485, 29856859, 29858121, 29858376, 29859068
29860994, 29861075, 29864203, 29864261, 29865188, 29865590, 29865658
29869086, 29869404, 29869887, 29870065, 29871098, 29872401, 29872937
29872983, 29873665, 29875459, 29875565, 29876358, 29876989, 29877608
29878076, 29881050, 29881478, 29881643, 29881839, 29882427, 29882729
29884958, 29885890, 29887111, 29888621, 29889184, 29890740, 29891075
29891853, 29891916, 29892604, 29893132, 29896510, 29897418, 29897863
29900203, 29900824, 29902299, 29902327, 29902330, 29902659, 29903190
29903299, 29903454, 29904002, 29906678, 29907942, 29908389, 29908777
29909658, 29910402, 29912286, 29913966, 29914449, 29914544, 29915217
29915848, 29916975, 29919789, 29920025, 29920376, 29920804, 29921318
29923452, 29926466, 29927756, 29928210, 29928427, 29928564, 29932202
29932430, 29932780, 29934052, 29937565, 29938225, 29939400, 29939795
29940373, 29942096, 29942554, 29943670, 29944035, 29944660, 29944963
29945645, 29946388, 29947145, 29950220, 29951620, 29956016, 29956222
29957412, 29957493, 29961353, 29961609, 29962160, 29962248, 29962834
29962927, 29962939, 29965052, 29965603, 29965888, 29966768, 29967223
29968085, 29969557, 29970298, 29971027, 29971481, 29971888, 29971951
29972176, 29973012, 29989783, 29989845, 29991257, 29993717, 29997326
29997553, 30000664, 30001331, 30003187, 30006159, 30006985, 30007450
30007797, 30008125, 30008198, 30008214, 30009710, 30012181, 30015070
30017836, 30018017, 30019864, 30024618, 30025814, 30026016, 30028599

30029519, 30029806, 30031027, 30032376, 30033547, 30034456, 30035598
30038392, 30039800, 30039959, 30040157, 30041501, 30042490, 30043398
30043610, 30043930, 30044507, 30046497, 30047531, 30047702, 30047765
30047931, 30049966, 30051176, 30051783, 30051804, 30052928, 30053036
30053501, 30053748, 30054980, 30056058, 30057718, 30057799, 30058149
30058453, 30059106, 30059109, 30062364, 30064268, 30066352, 30067565
30071446, 30072905, 30073314, 30073744, 30074296, 30074349, 30074469
30074472, 30075037, 30076058, 30076197, 30076253, 30078675, 30078934
30079949, 30080266, 30081546, 30081580, 30082145, 30083100, 30083216
30083488, 30083807, 30084971, 30085897, 30086596, 30086992, 30090568
30092859, 30095591, 30095952, 30097092, 30097115, 30098251, 30099302
30099420, 30099454, 30100354, 30101186, 30103551, 30103553, 30104378
30104555, 30106748, 30109365, 30110224, 30110370, 30110518, 30114477
30114489, 30114534, 30116085, 30116854, 30117335, 30117593, 30118261
30118279, 30120608, 30122583, 30127522, 30127904, 30128047, 30131286
30131645, 30135396, 30135731, 30135942, 30136346, 30139392, 30142907
30143470, 30143593, 30147473, 30147928, 30149035, 30149658, 30150606
30153885, 30154633, 30155814, 30155837, 30159329, 30159511, 30159536
30159752, 30159760, 30163243, 30164714, 30165493, 30165503, 30165897
30169254, 30170104, 30172925, 30173113, 30173556, 30174401, 30175291
30177597, 30178250, 30178839, 30178990, 30179644, 30180208, 30181756
30183920, 30184102, 30185852, 30186319, 30186476, 30186706, 30187866
30189516, 30190090, 30191274, 30193165, 30193505, 30194612, 30194710
30194972, 30195667, 30195668, 30195684, 30196195, 30196358, 30198861
30198905, 30200034, 30200237, 30200758, 30202349, 30202388, 30204542
30206493, 30206675, 30207473, 30208327, 30209736, 30210884, 30213031
30213540, 30215130, 30215302, 30215351, 30217206, 30217982, 30218044
30218317, 30221237, 30222512, 30223712, 30223847, 30224650, 30224868
30224950, 30225265, 30225718, 30225844, 30228567, 30229683, 30232638
30235919, 30235979, 30236554, 30238211, 30239480, 30240010, 30241567
30242120, 30242724, 30244340, 30246053, 30246179, 30247305, 30249432
30252005, 30252098, 30252156, 30252458, 30252977, 30253035, 30253090
30253608, 30253835, 30254525, 30254726, 30255143, 30255528, 30256542
30257412, 30264405, 30265523, 30265608, 30265703, 30266791, 30267155
30269428, 30269748, 30270647, 30270744, 30271114, 30272329, 30274090
30274188, 30274324, 30275578, 30276243, 30277120, 30277451, 30282501
30282591, 30283296, 30283577, 30283579, 30283581, 30283932, 30284219
30284369, 30285457, 30285843, 30288343, 30288491, 30289458, 30294267
30294671, 30295110, 30295549, 30299367, 30299817, 30299934, 30300030
30300538, 30305264, 30305395, 30305568, 30305880, 30307814, 30307883
30308368, 30308624, 30308772, 30309098, 30309798, 30312094, 30313848
30314079, 30314837, 30316667, 30317209, 30318638, 30318943, 30319080
30322980, 30323658, 30323849, 30324180, 30327149, 30328168, 30330123
30334484, 30334563, 30335127, 30335832, 30335987, 30336032, 30336742
30339103, 30341713, 30342371, 30342878, 30345201, 30352581, 30352623
30355490, 30357463, 30362003, 30362850, 30363716, 30364329, 30364613
30365745, 30367193, 30368048, 30368534, 30371909, 30374345, 30374570
30374739, 30375109, 30381525, 30383286, 30384121, 30384152, 30389229
30389414, 30389507, 30392011, 30394738, 30394974, 30398257, 30398422
30399906, 30402386, 30403763, 30403902, 30403989, 30404117, 30408515
30408808, 30409207, 30409339, 30409590, 30412188, 30412921, 30413137
30414714, 30416034, 30421476, 30422487, 30424347, 30430921, 30431274
30431504, 30431698, 30431703, 30431717, 30441687, 30441959, 30442749
30442884, 30447060, 30448917, 30449194, 30453442, 30454090, 30458568
30458593, 30460922, 30463938, 30464250, 30469777, 30474167, 30474774
30475115, 30477588, 30477767, 30479715, 30485255, 30490014, 30493518
30495133, 30495483, 30496957, 30497057, 30500344, 30501574, 30503943
30505497, 30506794, 30507032, 30509277, 30510347, 30510527, 30513285
30513848, 30516868, 30517214, 30522285, 30523750, 30534351, 30534549
30534827, 30537405, 30537533, 30540109, 30544247, 30545281, 30549637
30549789, 30549881, 30564139, 30573703, 30576393, 30577071, 30579051
30582500, 30592859, 30598682, 30598746, 30599405, 30600184, 30602230
30609799, 30612199, 30613937, 30613971, 30619787, 30623142, 30629799
30633259, 30635302, 30641755, 30654558, 30657875, 30661939, 30662651
30671813, 30676209, 30708735, 30720736, 30730026, 30732711, 30741263
30761878, 30776416, 30783551, 30785101, 30790441, 30803210, 30808869

30815852, 30825391, 30825419, 30881407, 30886188, 30890971, 30919691
30922996, 30993198, 31016413, 29997959, 29997937, 28852325, 28730253
29213893, 30528704, 29540327, 29254623, 29445548, 29774362, 30134746
30160625, 29942275, 30534662, 29512125, 30855101, 27222128, 27572040
27604329, 27760043, 27877830, 28302580, 28470673, 28621543, 28642469
28699321, 28710663, 28755846, 28772816, 28785321, 28800508, 28808652
28815557, 28847541, 28847572, 28870496, 28871040, 28874416, 28877252
28881191, 28881848, 28888083, 28911140, 28925250, 28925460, 28935956
28940472, 3, 28942694, 28951332, 28963036, 28968779, 28980448, 28995287
29003207, 29003617, 29016294, 29018680, 29024876, 29026154, 29027933
29047127, 29052850, 29058476, 29111631, 29112455, 29117337, 29123444
29125708, 29125786, 29129476, 29131772, 29132456, 29139727, 29146157
29147849, 29149170, 29152603, 29152752, 29154631, 29154636, 29154829
29159216, 29159661, 29160462, 29161923, 29169540, 29169739, 29170717
29173618, 29181568, 29182920, 29183298, 29186091, 29191827, 29201143
29201695, 29209545, 29210577, 29210610, 29210624, 29210683, 29213641
29219627, 29224294, 29225861, 29229839, 29235934, 29242906, 29243749
29244495, 29244766, 29244968, 29248723, 29249583, 29251564, 29255616
29260224, 29261695, 29271019, 29273360, 29282090, 29282666, 29285453
29285621, 29290235, 29292232, 29293806, 29294753, 29299830, 29307090
29307109, 29311336, 29329675, 29330791, 29339299, 29357821, 29360467
29360775, 29367971, 29368725, 29379299, 29379381, 29380527, 29381000
29382296, 29391301, 29393649, 29402110, 29411931, 29413360, 29457319
29465047

Version 19.0.0.0.ru-2020-01.rur-2020-01.r1

Version 19.0.0.0.ru-2020-01.rur-2020-01.r1 includes the following:

- Patch 30557433: Database Release Update: 19.6.0.0.200114
- Patch 30484981: OJVM RELEASE UPDATE: 19.6.0.0.200114
- Patch 29997937: DSTV34 UPDATE for RDBMS (TZDATA2019B)
- Patch 29997959: DSTV34 UPDATE for OJVM (TZDATA2019B)
- PreUpgrade Jar: preupgrade_19_cbuild_5_lf.zip
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- Patch 30528704: 19C RMAN RECOVER DATABASE USE REDO LOG INSTEAD OF ARCHIVELOG AFTER APPLYING OCT DATABASE RU

Oracle release update 19.6.0.0.200114, released January 2020

Bugs fixed:

30545281, 8476681, 14735102, 17428816, 19080742, 19697993, 20313356
21374587, 21965541, 23296836, 23606241, 24687075, 25756945, 25806201
25883179, 25986062, 25997810, 26476244, 26611353, 26739322, 26777814
26872233, 27036163, 27044169, 27101798, 27126938, 27195935, 27244999
27254335, 27359766, 27369515, 27406105, 27411022, 27423500, 27439716
27453490, 27458357, 27489107, 27666312, 27710072, 27729678, 27846298
27880025, 27934711, 27935464, 27941110, 27957203, 27967484, 28064977
28072567, 28109326, 28125947, 28129791, 28181021, 28189466, 28204262
28205555, 28209985, 28210681, 28271258, 28271693, 28279456, 28294563
28313275, 28319114, 28326928, 28350595, 28371123, 28373960, 28375383
28379065, 28381939, 28386259, 28390273, 28395302, 28397317, 28402823
28410431, 28431445, 28435333, 28454215, 28463226, 28475242, 28484299
28489419, 28492006, 28498976, 28502773, 28513333, 28521330, 28530171

28534475, 28538439, 28542455, 28546290, 28547068, 28547926, 28558645
28561704, 28567417, 28567819, 28569897, 28572407, 28572533, 28572544
28572667, 28572834, 28578945, 28587723, 28589509, 28593682, 28594086
28597221, 28601957, 28605066, 28606598, 28612239, 28620697, 28625862
28627033, 28636532, 28639299, 28643718, 28644549, 28645570, 28646200
28646939, 28649388, 28655209, 28663782, 28672457, 28673945, 28692103
28692275, 28694872, 28696373, 28705231, 28710385, 28710734, 28714461
28715727, 28718469, 28719348, 28720204, 28720418, 28721497, 28722229
28730079, 28734355, 28740708, 28742555, 28749853, 28752923, 28755011
28758722, 28760206, 28765983, 28767240, 28769456, 28772390, 28774416
28776811, 28777214, 28781754, 28785531, 28789531, 28791852, 28795551
28795734, 28802734, 28804517, 28810381, 28811560, 28815123, 28815355
28817449, 28819640, 28820669, 28821847, 28824482, 28827682, 28831971
28833912, 28835937, 28836716, 28844738, 28849776, 28854004, 28855520
28855922, 28857552, 28862532, 28863432, 28863487, 28867992, 28873575
28875089, 28876253, 28876639, 28878865, 28882784, 28884931, 28887305
28888327, 28889730, 28892794, 28897512, 28899663, 28901126, 28905457
28905615, 28907196, 28912691, 28915561, 28917080, 28918429, 28919145
28922227, 28922532, 28922608, 28925634, 28925880, 28927452, 28932914
28933158, 28935293, 28936114, 28937717, 28938698, 28940179, 28940281
28941901, 28942455, 28945421, 28945994, 28949888, 28950868, 28951533
28952168, 28954762, 28955606, 28955883, 28956908, 28957292, 28957723
28958088, 28959493, 28960863, 28962775, 28965084, 28965095, 28965231
28965376, 28966444, 28974083, 28977322, 28983095, 28983486, 28985478
28986207, 28986231, 28986257, 28986326, 28986481, 28988482, 28989306
28993295, 28993353, 28994307, 28996376, 29000000, 29001305, 29001888
29002488, 29002784, 29002927, 29003407, 29003738, 29006318, 29006621
29007321, 29007353, 29007775, 29008035, 29008669, 29009513, 29010126
29011936, 29012609, 29013475, 29013832, 29014076, 29015118, 29017265
29018655, 29019121, 29021063, 29021352, 29022986, 29024054, 29024552
29024732, 29026582, 29026606, 29027456, 29027694, 29027940, 29031575
29031600, 29032234, 29032276, 29032457, 29032607, 29033052, 29033145
29033200, 29033280, 29034587, 29037290, 29038528, 29038728, 29039089
29039510, 29040739, 29043554, 29043651, 29043725, 29044763, 29044954
29046482, 29047850, 29048178, 29048289, 29048498, 29048605, 29050357
29050560, 29050765, 29050886, 29051702, 29051953, 29052726, 29053783
29056024, 29056270, 29056560, 29056767, 29056894, 29059011, 29060216
29061016, 29061959, 29062692, 29062848, 29062860, 29062868, 29110526
29110783, 29110790, 29110797, 29110802, 29110805, 29111598, 29113282
29113305, 29115857, 29117526, 29117642, 29118543, 29119077, 29120223
29122224, 29122254, 29123297, 29123432, 29123482, 29124368, 29125036
29125374, 29125380, 29126345, 29127957, 29128693, 29128935, 29129450
29129497, 29129712, 29130219, 29131539, 29132869, 29132938, 29133470
29134447, 29135383, 29135649, 29136111, 29138641, 29139956, 29141316
29141341, 29141685, 29142609, 29142667, 29143516, 29144995, 29145214
29145730, 29149829, 29150338, 29151520, 29152357, 29155099, 29157051
29157389, 29158680, 29158899, 29159909, 29159936, 29160174, 29162095
29163156, 29163415, 29163437, 29163524, 29163567, 29167111, 29167342
29167374, 29167940, 29168137, 29168219, 29168433, 29169073, 29169215
29170232, 29171683, 29171942, 29172618, 29172826, 29173140, 29173373
29173817, 29174004, 29176318, 29177466, 29177543, 29177886, 29178385
29180313, 29180455, 29180559, 29180893, 29181153, 29181231, 29181620
29181743, 29181923, 29182019, 29182517, 29182901, 29183912, 29184297
29184666, 29185193, 29186456, 29189302, 29189307, 29189889, 29190235
29190474, 29190663, 29190740, 29191541, 29192419, 29192468, 29192685
29193207, 29194205, 29194367, 29194493, 29194827, 29194981, 29195279
29195337, 29195758, 29196725, 29198092, 29198913, 29199635, 29199733
29200316, 29200700, 29201494, 29201539, 29201787, 29202104, 29202461
29202850, 29203122, 29203166, 29203425, 29203443, 29203604, 29205281
29205323, 29205419, 29205463, 29205767, 29205918, 29206109, 29206605
29207073, 29208260, 29208732, 29211457, 29211724, 29212012, 29212433
29212611, 29213320, 29213351, 29213613, 29213775, 29213850, 29213879
29214561, 29214960, 29216312, 29216723, 29216746, 29216984, 29217294
29217472, 29217828, 29217848, 29218570, 29219205, 29219273, 29220079
29221248, 29221891, 29221942, 29222031, 29222784, 29223833, 29223859
29223967, 29224065, 29224605, 29225076, 29225168, 29225758, 29227602

29228869, 29229164, 29229754, 29229844, 29229955, 29230252, 29230565
29231133, 29232117, 29232154, 29232449, 29233415, 29233810, 29233953
29234123, 29236573, 29237538, 29237575, 29237744, 29240307, 29240759
29241345, 29241651, 29242017, 29242884, 29243958, 29245137, 29245160
29246163, 29247415, 29247712, 29247906, 29248495, 29248552, 29248835
29248858, 29249412, 29249991, 29250059, 29250317, 29251259, 29253184
29253871, 29254031, 29254930, 29255178, 29255273, 29255431, 29255435
29256426, 29259119, 29259320, 29260452, 29260956, 29261547, 29261548
29261906, 29262512, 29262887, 29265448, 29266248, 29266899, 29267292
29268412, 29269171, 29269228, 29269825, 29270585, 29273539, 29273570
29273735, 29273812, 29273847, 29274428, 29274564, 29274627, 29275461
29276272, 29277317, 29278218, 29278684, 29279658, 29279751, 29279854
29281527, 29281691, 29281796, 29282233, 29282898, 29285197, 29285503
29285788, 29285956, 29286037, 29286229, 29287130, 29287705, 29292837
29293072, 29293574, 29297863, 29297915, 29298220, 29299049, 29299082
29299844, 29301463, 29301566, 29302963, 29303918, 29304781, 29306226
29306713, 29307638, 29311528, 29311588, 29312310, 29312672, 29312734
29312753, 29312889, 29313347, 29313417, 29313525, 29314539, 29314636
29317756, 29318410, 29319441, 29321489, 29323946, 29324568, 29324735
29325087, 29325105, 29325257, 29325765, 29325993, 29327044, 29329042
29329087, 29329807, 29330361, 29331066, 29331209, 29331380, 29331493
29332292, 29332395, 29332763, 29332771, 29333500, 29336843, 29337310
29337742, 29338315, 29338348, 29338453, 29338780, 29338913, 29339101
29339155, 29341209, 29343086, 29343861, 29345937, 29346057, 29346211
29346943, 29347620, 29348176, 29350052, 29350762, 29351386, 29351662
29351716, 29351735, 29351749, 29352298, 29352724, 29352867, 29352947
29353271, 29353432, 29353821, 29353960, 29355654, 29356547, 29356704
29356711, 29356752, 29358509, 29358828, 29360252, 29360285, 29360672
29360911, 29360950, 29361319, 29361472, 29361801, 29363151, 29364171
29364177, 29366940, 29367019, 29367561, 29368253, 29368310, 29372541
29373418, 29373588, 29374179, 29375355, 29375941, 29375984, 29376346
29377804, 29377986, 29378029, 29378834, 29378913, 29379978, 29382641
29382784, 29382815, 29383695, 29384781, 29384854, 29384864, 29385429
29385652, 29386502, 29386635, 29386660, 29387073, 29387274, 29388020
29388072, 29388094, 29388524, 29388830, 29389889, 29390011, 29390435
29390785, 29391030, 29391237, 29391849, 29391925, 29392966, 29393291
29394014, 29394140, 29394749, 29395657, 29397954, 29397996, 29398488
29398863, 29399046, 29399100, 29399121, 29399336, 29399938, 29402131
29404483, 29405012, 29405462, 29405651, 29405996, 29407804, 29408853
29409149, 29409455, 29410311, 29410834, 29411037, 29411469, 29412066
29412269, 29416688, 29417173, 29417719, 29417884, 29418165, 29420834
29421059, 29423003, 29423016, 29423156, 29423826, 29424999, 29426241
29429017, 29429264, 29429466, 29429566, 29430524, 29430866, 29431192
29431485, 29432176, 29434301, 29435474, 29435652, 29436454, 29436514
29436727, 29437594, 29437712, 29438277, 29438736, 29439522, 29441196
29443187, 29443250, 29444072, 29444282, 29444602, 29446669, 29448498
29449477, 29449845, 29449852, 29450193, 29450421, 29450812, 29450936
29451386, 29452251, 29452576, 29452936, 29452953, 29454978, 29455424
29456714, 29457312, 29457370, 29457502, 29457807, 29457978, 29460252
29461420, 29461791, 29462594, 29462767, 29462957, 29463047, 29463528
29464616, 29464779, 29465177, 29467622, 29469565, 29470291, 29471860
29472618, 29476473, 29481584, 29483452, 29483532, 29483626, 29483672
29483685, 29483712, 29483723, 29483771, 29485099, 29486181, 29488894
29489436, 29489546, 29490256, 29492127, 29492939, 29493122, 29494245
29495057, 29495684, 29497311, 29497588, 29497696, 29498198, 29500257
29500826, 29502561, 29503543, 29503631, 29503827, 29504492, 29504682
29505668, 29507270, 29507616, 29508681, 29509777, 29510278, 29511611
29514479, 29515134, 29515240, 29515476, 29515766, 29515834, 29516300
29516727, 29516766, 29517168, 29517883, 29521187, 29521688, 29521748
29521862, 29522358, 29522561, 29522662, 29523055, 29523511, 29524599
29525467, 29525886, 29526966, 29527595, 29527610, 29528368, 29529147
29530440, 29530515, 29530812, 29530909, 29531654, 29531836, 29532532
29536342, 29536445, 29538631, 29541742, 29541769, 29542084, 29542449
29542643, 29543034, 29543956, 29546817, 29547010, 29547867, 29548427
29548687, 29548722, 29549071, 29549104, 29549154, 29549730, 29552773
29553141, 29557144, 29557261, 29557336, 29557556, 29558238, 29558975

29559187, 29559446, 29559908, 29559981, 29564592, 29564593, 29565611
29580394, 29580983, 29581771, 29584261, 29584693, 29586143, 29587765
29597536, 29597754, 29598039, 29598046, 29598233, 29599008, 29599300
29601461, 29602831, 29603460, 29603884, 29604002, 29604257, 29607136
29607797, 29608000, 29610506, 29611020, 29611991, 29615824, 29616244
29616414, 29618074, 29618190, 29620042, 29622936, 29625065, 29625804
29625876, 29626154, 29626732, 29628200, 29629430, 29629650, 29629681
29629745, 29631749, 29632095, 29632265, 29632611, 29633697, 29633753
29633936, 29634643, 29635427, 29635717, 29635990, 29637362, 29637526
29638285, 29641736, 29643721, 29645349, 29648928, 29651183, 29651520
29653132, 29653246, 29655668, 29656819, 29657422, 29657960, 29661028
29661065, 29661722, 29663368, 29664087, 29664161, 29665940, 29667994
29668005, 29669413, 29670782, 29671363, 29672507, 29676089, 29677051
29677173, 29677733, 29677927, 29679856, 29681987, 29683039, 29687214
29687459, 29687718, 29687763, 29689145, 29689255, 29692694, 29694869
29695425, 29695841, 29696310, 29700125, 29700460, 29700770, 29703932
29707099, 29707896, 29708915, 29710188, 29710858, 29713810, 29715220
29716194, 29716491, 29717659, 29719146, 29720133, 29721418, 29725476
29725781, 29726695, 29739576, 29741976, 29742223, 29746962, 29747493
29747648, 29747653, 29748285, 29748336, 29748513, 29749471, 29750673
29751094, 29753244, 29754951, 29755821, 29756274, 29756444, 29757099
29757264, 29757651, 29758217, 29758661, 29761678, 29761837, 29761911
29765393, 29766207, 29766503, 29766679, 29768899, 29770750, 29771032
29771242, 29773197, 29773842, 29775393, 29779196, 29782211, 29782823
29782866, 29784106, 29785239, 29787292, 29787766, 29791152, 29791880
29793318, 29794462, 29795712, 29795957, 29796378, 29797726, 29802695
29804875, 29805772, 29809837, 29812489, 29813503, 29815713, 29817278
29821582, 29825525, 29827852, 29831196, 29834506, 29836096, 29838485
29838773, 29839715, 29840619, 29841267, 29841687, 29843277, 29843692
29844226, 29845530, 29846126, 29846645, 29848084, 29848849, 29849100
29850930, 29851733, 29853485, 29858121, 29858376, 29865188, 29865658
29869086, 29869404, 29869887, 29870065, 29871098, 29873665, 29875459
29875565, 29877608, 29878076, 29881478, 29881839, 29882729, 29884958
29887111, 29888621, 29890740, 29891916, 29893132, 29897418, 29897863
29900203, 29902327, 29902330, 29903299, 29903454, 29906678, 29907942
29909658, 29912286, 29914449, 29915217, 29915848, 29916975, 29920025
29921318, 29926466, 29927756, 29928210, 29937565, 29938225, 29940373
29942096, 29942554, 29943670, 29944035, 29944660, 29945645, 29946388
29951620, 29956016, 29957493, 29961609, 29962248, 29962927, 29962939
29966768, 29967223, 29968085, 29970298, 29971027, 29971888, 29989783
29989845, 29991257, 29997326, 30003187, 30006159, 30006985, 30007797
30008125, 30008214, 30009710, 30015070, 30019864, 30024618, 30029806
30032376, 30033547, 30034456, 30035598, 30038392, 30040157, 30042490
30043610, 30044507, 30047702, 30047765, 30051176, 30051783, 30053036
30058149, 30058453, 30059106, 30059109, 30066352, 30074349, 30074472
30075037, 30076197, 30078675, 30079949, 30080266, 30081580, 30083488
30084971, 30085897, 30086992, 30090568, 30092859, 30095591, 30095952
30097092, 30098251, 30099454, 30101186, 30104555, 30106748, 30110224
30110370, 30110518, 30114489, 30114534, 30127522, 30127904, 30131645
30135396, 30142907, 30149658, 30150606, 30154633, 30155837, 30159329
30164714, 30165493, 30165503, 30170104, 30174401, 30175291, 30177597
30178250, 30187866, 30189516, 30191274, 30193165, 30206493, 30218044
30223712, 30223847, 30224950, 30235919, 30246179, 30247305, 30252098
30252156, 30253608, 30255143, 30264405, 30266791, 30269428, 30274188
30282591, 30299817, 30312094, 30318638, 30324180, 30342878, 30365745
30389229, 30402386, 30408515, 30412188, 30453442, 30458593, 30474167
30474774, 30485255, 30534827, 30641755

Version 19.0.0.0.ru-2019-10.rur-2019-10.r1

Version 19.0.0.0.ru-2019-10.rur-2019-10.r1 includes the following:

- Patch 30125133: DATABASE RELEASE UPDATE 19.5.0.0.0

- Patch 30128191: OJVM RELEASE UPDATE 19.5.0.0.0
- Patch 29997937: DSTv34 UPDATE for RDBMS (TZDATA2019B)
- Patch 29997959: DSTV34 UPDATE for OJVM (TZDATA2019B)
- PreUpgrade Jar: preupgrade_19_cbuild_4_lf.zip
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- Patch 30528704: 19C RMAN RECOVER DATABASE USE REDO LOG INSTEAD OF ARCHIVELOG AFTER APPLYING OCT DATABASE RU
- Support for [Resizing the temporary tablespace in a read replica \(p. 1061\)](#)

Oracle release update 19.5.0.0.0, released October 2019

Bugs fixed:

```
14735102, 17428816, 19080742, 19697993, 20313356, 21374587, 21965541
23296836, 23606241, 24687075, 25756945,
25806201, 25883179, 25986062 25997810, 26476244, 26611353, 26739322,
26872233, 27126938, 27244999 27359766,
27369515, 27406105, 27411022, 27423500, 27453490, 27458357 27489107,
27666312, 27710072, 27846298, 27880025,
27934711, 27935464 27941110, 27957203, 27967484, 28064977, 28072567,
28125947, 28129791 28181021, 28210681,
28271693, 28279456, 28294563, 28313275, 28319114 28326928, 28350595,
28371123, 28379065, 28381939, 28386259,
28390273 28395302, 28410431, 28431445, 28435333, 28463226, 28475242,
28484299 28489419, 28498976, 28502773,
28513333, 28521330, 28534475, 28542455 28547068, 28558645, 28561704,
28567417, 28567819, 28569897, 28572533,
28572544, 28572667, 28572834, 28587723, 28593682, 28594086, 28597221
28601957, 28605066, 28606598, 28612239,
28625862, 28627033, 28636532 28643718, 28644549, 28645570, 28646200,
28646939, 28649388, 28655209 28663782,
28672457, 28673945, 28692103, 28692275, 28694872, 28696373 28705231,
28710385, 28710734, 28714461, 28715727,
28718469, 28719348 28720418, 28721497, 28722229, 28730079, 28740708,
28742555, 28749853 28752923, 28755011,
28758722, 28760206, 28765983, 28767240, 28769456 28772390, 28774416,
28776811, 28777214, 28781754, 28785531,
28789531 28791852, 28795551, 28802734, 28804517, 28810381, 28811560,
28815123 28815355, 28817449, 28819640,
28820669, 28824482, 28827682, 28831971 28833912, 28835937, 28836716,
28849776, 28854004, 28855922, 28862532
28863432, 28863487, 28867992, 28873575, 28875089, 28876253, 28876639
28878865, 28884931, 28888327, 28889730,
28892794, 28897512, 28899663 28901126, 28905457, 28907196, 28912691,
28915561, 28917080, 28918429 28919145,
28922227, 28922532, 28922608, 28925634, 28925880, 28933158 28936114,
28937717, 28938698, 28940179, 28940281,
28941901, 28942455 28945421, 28945994, 28950868, 28951533, 28952168,
28954762, 28955606 28956908, 28957292,
28957723, 28958088, 28960863, 28962775, 28965084 28965231, 28966444,
28974083, 28977322, 28983095, 28983486,
28985478 28986207, 28986231, 28986326, 28986481, 28988482, 28989306,
28993295 28993353, 28994307, 28996376,
29000000, 29001305, 29001888, 29002488 29002784, 29002927, 29003738,
29006318, 29006621, 29007321, 29007353,
29007775, 29008035, 29008669, 29009513, 29010126, 29011936, 29012609
29013475, 29014076, 29015118, 29017265,
```

29018655, 29019121, 29021063 29021352, 29024054, 29024552, 29024732,
29026582, 29026606, 29027456 29027694,
29027940, 29031575, 29031600, 29032234, 29032276, 29032457 29032607,
29033052, 29033145, 29033200, 29033280,
29034587, 29037290 29038528, 29039089, 29039510, 29040739, 29043554,
29043651, 29043725 29044763, 29044954,
29047850, 29048178, 29048289, 29048498, 29048605 29050357, 29050560,
29050765, 29050886, 29051702, 29051953,
29052726 29053783, 29056024, 29056270, 29056560, 29056767, 29059011,
29060216 29061959, 29062692, 29062848,
29062860, 29062868, 29110526, 29110783 29110790, 29110797, 29110802,
29110805, 29111598, 29113282, 29113305
29117526, 29117642, 29119077, 29120223, 29122224, 29122254, 29123297
29123432, 29123482, 29124368, 29125036,
29125374, 29125380, 29126345 29127957, 29128693, 29128935, 29129450,
29129497, 29129712, 29130219 29131539,
29132869, 29132938, 29134447, 29135383, 29135649, 29136111 29138641,
29139956, 29141316, 29141341, 29141685,
29142609, 29142667 29143516, 29144995, 29145214, 29145730, 29149829,
29150338, 29151520 29152357, 29155099,
29157051, 29157389, 29158680, 29158899, 29159909 29159936, 29160174,
29162095, 29163156, 29163415, 29163437,
29163524 29163567, 29167111, 29167342, 29167374, 29167940, 29168137,
29168219 29168433, 29169073, 29169215,
29170232, 29171683, 29171942, 29172618 29172826, 29173140, 29173373,
29173817, 29174004, 29176318, 29177466,
29177543, 29177886, 29178385, 29180313, 29180455, 29180559, 29180893
29181153, 29181231, 29181620, 29181743,
29181923, 29182019, 29183912 29184297, 29184666, 29185193, 29186456,
29189302, 29189307, 29189889 29190235,
29190474, 29190740, 29191541, 29192419, 29192468, 29192685 29193207,
29194205, 29194367, 29194493, 29194827,
29194981, 29195279 29195337, 29195758, 29196725, 29198092, 29199635,
29199733, 29200316 29200700, 29201494,
29201539, 29202104, 29202850, 29203122, 29203166 29203425, 29203443,
29203604, 29205281, 29205323, 29205419,
29205463 29205767, 29205918, 29206109, 29206605, 29207073, 29208260,
29208732 29211457, 29211724, 29212012,
29212433, 29212611, 29213320, 29213351 29213613, 29213775, 29213850,
29213879, 29214561, 29214960, 29216746,
29216984, 29217294, 29217472, 29217828, 29217848, 29218570, 29219205
29219273, 29220079, 29221248, 29221891,
29222031, 29222784, 29223833 29223859, 29223967, 29224065, 29224605,
29225076, 29225168, 29225758 29227602,
29228869, 29229164, 29229754, 29229844, 29229955, 29230252 29230565,
29231133, 29232117, 29232154, 29232449,
29233415, 29234123 29236573, 29237538, 29237744, 29240307, 29241345,
29241651, 29242017 29242884, 29243958,
29245137, 29245160, 29246163, 29247415, 29247712 29247906, 29248495,
29248552, 29248835, 29248858, 29249991,
29250059 29250317, 29251259, 29253184, 29253871, 29254031, 29254930,
29255178 29255273, 29255431, 29255435,
29256426, 29259119, 29259320, 29260452 29261547, 29261906, 29262512,
29262887, 29265448, 29266248, 29266899,
29267292, 29268412, 29269171, 29269228, 29269825, 29270585, 29273539
29273570, 29273735, 29273812, 29273847,
29274428, 29274564, 29274627 29275461, 29276272, 29277317, 29278218,
29279658, 29279751, 29279854 29281527,
29281691, 29281796, 29282233, 29282898, 29285503, 29285788 29285956,
29286037, 29287130, 29287705, 29292837,
29293072, 29293574 29297863, 29297915, 29298220, 29299049, 29299082,
29299844, 29301463 29301566, 29302963,
29303918, 29304781, 29306226, 29306713, 29311528 29311588, 29312310,
29312672, 29312734, 29312753, 29313347,
29313417 29313525, 29314539, 29317756, 29318410, 29319441, 29321489,
29323946 29324568, 29324735, 29325087,

29325105, 29325257, 29325765, 29325993 29327044, 29329042, 29329087,
29329807, 29330361, 29331066, 29331209
29331380, 29331493, 29332292, 29332395, 29332771, 29333500, 29336843
29337310, 29337742, 29338315, 29338453,
29338780, 29338913, 29339101 29339155, 29341209, 29343086, 29345937,
29346057, 29346211, 29346943 29347620,
29348176, 29350052, 29350762, 29351386, 29351662, 29351716 29351735,
29351749, 29352298, 29352724, 29352867,
29352947, 29353271 29353432, 29353821, 29353960, 29355654, 29356547,
29356704, 29356711 29356752, 29358509,
29358828, 29360252, 29360285, 29360672, 29360911 29360950, 29361319,
29361472, 29361801, 29363151, 29364171,
29364177 29366940, 29367019, 29367561, 29368253, 29368310, 29372541,
29373418 29373588, 29374179, 29375941,
29376346, 29377986, 29378029, 29378834 29378913, 29379978, 29382784,
29382815, 29384781, 29384854, 29384864,
29385429, 29385652, 29386502, 29386635, 29386660, 29387073, 29387274
29388020, 29388072, 29388094, 29388524,
29388830, 29389889, 29390011 29390435, 29390785, 29391030, 29391237,
29391849, 29393291, 29394014 29394140,
29394749, 29395657, 29397954, 29397996, 29398488, 29398863 29399100,
29399121, 29399336, 29399938, 29402131,
29404483, 29405012 29405462, 29405651, 29405996, 29407804, 29408853,
29409149, 29409455 29410311, 29410834,
29411037, 29411469, 29412066, 29412269, 29417173 29417719, 29417884,
29420834, 29421059, 29423016, 29423156,
29423826 29424999, 29426241, 29429017, 29429264, 29429466, 29429566,
29430524 29430866, 29431192, 29431485,
29434301, 29435474, 29435652, 29436454 29436514, 29436727, 29437594,
29437712, 29438277, 29438736, 29439522,
29441196, 29443187, 29443250, 29444072, 29444282, 29444602, 29446669
29449477, 29449845, 29450193, 29450421,
29450812, 29450936, 29451386 29452576, 29452936, 29452953, 29454978,
29455424, 29457312, 29457370 29457807,
29460252, 29461420, 29461791, 29462594, 29462767, 29462957 29463047,
29463528, 29464616, 29464779, 29465177,
29467622, 29469565 29470291, 29471860, 29476473, 29481584, 29483532,
29483672, 29483685 29483712, 29483723,
29485099, 29486181, 29488894, 29489546, 29490256 29492127, 29492939,
29493122, 29494245, 29495057, 29495684,
29497311 29497696, 29498198, 29500257, 29500826, 29502561, 29503543,
29503631 29503827, 29504492, 29504682,
29505668, 29507270, 29507616, 29510278 29511611, 29514479, 29515134,
29515240, 29515476, 29515766, 29515834,
29516300, 29516766, 29517168, 29517883, 29521187, 29521748, 29521862
29522358, 29522561, 29522662, 29523511,
29524599, 29525467, 29525886 29527595, 29527610, 29529147, 29530440,
29530515, 29530812, 29531654 29532532,
29541742, 29541769, 29542449, 29542643, 29543034, 29543956 29546817,
29547867, 29548427, 29548687, 29548722,
29549071, 29549104 29549154, 29549730, 29552773, 29553141, 29557144,
29557261, 29557336 29558238, 29558975,
29559187, 29559446, 29559908, 29559981, 29565611 29580394, 29580983,
29581771, 29584261, 29584693, 29586143,
29587765 29597536, 29597754, 29598039, 29598046, 29598233, 29599008,
29599300 29601461, 29603460, 29604002,
29604257, 29607136, 29608000, 29611020 29611991, 29616244, 29616414,
29618074, 29618190, 29620042, 29622936,
29625065, 29625804, 29625876, 29626154, 29626732, 29628200, 29629430
29629650, 29629681, 29631749, 29632095,
29632265, 29633697, 29633753 29633936, 29634643, 29635427, 29635717,
29635990, 29637362, 29637526 29638285,
29641736, 29645349, 29651183, 29651520, 29653132, 29653246 29655668,
29656819, 29657422, 29657960, 29661028,
29661065, 29661722 29663368, 29664087, 29664161, 29667994, 29668005,
29669413, 29670782 29671363, 29672507,

29676089, 29677051, 29677733, 29677927, 29679856 29683039, 29687214,
29687459, 29687718, 29687763, 29689145,
29692694 29695425, 29695841, 29696310, 29700125, 29700460, 29703932,
29707099 29707896, 29708915, 29715220,
29720133, 29721418, 29725781, 29741976 29742223, 29747493, 29747648,
29749471, 29750673, 29751094, 29753244,
29754951, 29755821, 29756274, 29756444, 29757264, 29757651, 29758661
29761911, 29766207, 29766503, 29766679,
29771032, 29773842, 29775393 29779196, 29782211, 29782823, 29782866,
29785239, 29787766, 29791880 29795712,
29795957, 29802695, 29809837, 29815713, 29836096, 29838773 29845530,
29848849, 29850930, 29851733, 29858376,
29869086, 29869404 29869887, 29875459, 29875565, 29881478, 29881839,
29887111, 29888621 29893132, 29900203,
29902327, 29903299, 29903454, 29921318, 29926466 29943670, 29997326,
30006159, 30015070, 30019864, 30024618,
30033547 30034456, 30053036, 30075037, 30076197, 30092859, 30095591,
30114534 30142907, 30155837, 30174401

Version 19.0.0.0.ru-2019-07.rur-2019-07.r1

Version 19.0.0.0.ru-2019-07.rur-2019-07.r1 includes the following:

- Patch 29834717: Database Release Update: 19.4.0.0.190716
- Patch 29774421: OJVM RELEASE UPDATE: 19.4.0.0.190716
- Patch 28852325: RDBMS - DSTV33 UPDATE - TZDATA2018G
- Patch 28852334: DSTV33 UPDATE - TZDATA2018G - NEED OJVM FIX
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- PreUpgrade Jar: preupgrade_19_cbuild_4_lf.zip

Oracle release update 19.4.0.0.190716, released July 2019

Bugs fixed:

29850930, 29225758, 29276272, 14735102, 17428816, 19080742, 19697993
20313356, 21374587, 21965541, 23296836, 23606241, 25756945, 25806201
25883179, 25986062, 25997810, 26476244, 26611353, 26739322, 26872233
27126938, 27244999, 27359766, 27369515, 27423500, 27453490, 27666312
27710072, 27846298, 27957203, 28064977, 28072567, 28125947, 28129791
28181021, 28210681, 28271693, 28279456, 28313275, 28326928, 28350595
28371123, 28379065, 28381939, 28390273, 28431445, 28463226, 28475242
28489419, 28502773, 28513333, 28534475, 28547068, 28561704, 28569897
28572533, 28572544, 28572667, 28572834, 28587723, 28593682, 28594086
28597221, 28601957, 28605066, 28606598, 28612239, 28625862, 28627033
28636532, 28643718, 28644549, 28645570, 28646200, 28646939, 28649388
28655209, 28663782, 28673945, 28692275, 28694872, 28696373, 28705231
28710385, 28710734, 28714461, 28715727, 28718469, 28719348, 28720418
28722229, 28730079, 28740708, 28742555, 28749853, 28755011, 28758722
28760206, 28767240, 28772390, 28774416, 28777214, 28781754, 28785531
28789531, 28791852, 28795551, 28802734, 28804517, 28810381, 28811560
28815123, 28815355, 28819640, 28824482, 28827682, 28833912, 28835937
28849776, 28854004, 28862532, 28863432, 28867992, 28873575, 28876253
28876639, 28884931, 28888327, 28889730, 28892794, 28897512, 28899663
28901126, 28905457, 28907196, 28912691, 28915561, 28917080, 28918429
28919145, 28922227, 28922532, 28922608, 28925634, 28925880, 28933158

28936114, 28937717, 28938698, 28940179, 28940281, 28941901, 28942455
28945421, 28945994, 28950868, 28951533, 28952168, 28954762, 28955606
28956908, 28957292, 28957723, 28962775, 28965231, 28966444, 28974083
28977322, 28983095, 28983486, 28986207, 28986231, 28986326, 28986481
28988482, 28989306, 28993295, 28993353, 28994307, 28996376, 29000000
29001305, 29001888, 29002784, 29002927, 29003738, 29006318, 29006621
29007321, 29007775, 29008035, 29008669, 29009513, 29011936, 29012609
29013475, 29014076, 29015118, 29017265, 29018655, 29019121, 29021063
29021352, 29024054, 29024552, 29024732, 29026582, 29026606, 29027456
29027694, 29027940, 29031575, 29031600, 29032234, 29032276, 29032457
29032607, 29033052, 29033145, 29033200, 29033280, 29034587, 29037290
29038528, 29039089, 29039510, 29040739, 29043554, 29043651, 29043725
29044763, 29044954, 29047850, 29048178, 29048289, 29048498, 29048605
29050357, 29050560, 29050765, 29051702, 29051953, 29052726, 29053783
29056024, 29056270, 29056560, 29056767, 29059011, 29061959, 29062692
29062848, 29062860, 29062868, 29110526, 29110783, 29110790, 29110797
29110802, 29110805, 29111598, 29113282, 29113305, 29117526, 29117642
29119077, 29120223, 29122224, 29122254, 29123297, 29123432, 29123482
29124368, 29125036, 29125374, 29125380, 29126345, 29127957, 29128693
29128935, 29129450, 29129497, 29129712, 29130219, 29131539, 29132938
29134447, 29135383, 29135649, 29136111, 29138641, 29141316, 29141341
29141685, 29142609, 29142667, 29144995, 29145214, 29145730, 29149829
29150338, 29151520, 29152357, 29155099, 29157389, 29158680, 29158899
29159909, 29159936, 29160174, 29162095, 29163156, 29163415, 29163437
29163524, 29163567, 29167111, 29167342, 29167374, 29167940, 29168219
29168433, 29169073, 29169215, 29171683, 29171942, 29172618, 29172826
29173140, 29173373, 29173817, 29174004, 29176318, 29177466, 29177543
29177886, 29178385, 29180313, 29180455, 29180559, 29180893, 29181153
29181231, 29181620, 29181743, 29181923, 29182019, 29183912, 29184297
29184666, 29185193, 29186456, 29189302, 29189307, 29189889, 29190235
29190474, 29190740, 29191541, 29192419, 29192468, 29192685, 29193207
29194205, 29194367, 29194493, 29194827, 29194981, 29195279, 29195337
29195758, 29196725, 29199635, 29199733, 29200316, 29200700, 29201494
29201539, 29202104, 29202850, 29203122, 29203166, 29203425, 29203443
29203604, 29205281, 29205323, 29205419, 29205463, 29205767, 29205918
29206109, 29206605, 29207073, 29208260, 29208732, 29211457, 29211724
29212012, 29212433, 29212611, 29213351, 29213775, 29213850, 29213879
29214561, 29214960, 29216746, 29216984, 29217294, 29217472, 29217828
29217848, 29218570, 29219205, 29219273, 29220079, 29221248, 29221891
29222031, 29222784, 29223833, 29223859, 29223967, 29224065, 29224605
29225076, 29225168, 29227602, 29228869, 29229164, 29229754, 29229844
29229955, 29230252, 29230565, 29231133, 29232117, 29232154, 29233415
29234123, 29237538, 29240307, 29241345, 29241651, 29242017, 29242884
29243958, 29245137, 29245160, 29246163, 29247415, 29247712, 29247906
29248495, 29248552, 29248835, 29248858, 29249991, 29250059, 29250317
29251259, 29253184, 29253871, 29254031, 29254930, 29255178, 29255273
29255431, 29255435, 29256426, 29259119, 29259320, 29260452, 29261547
29261906, 29262512, 29262887, 29265448, 29266248, 29266899, 29267292
29268412, 29269171, 29269228, 29270585, 29273539, 29273570, 29273735
29273812, 29273847, 29274428, 29274564, 29274627, 29275461, 29277317
29278218, 29279658, 29279751, 29279854, 29281527, 29281691, 29281796
29282233, 29282898, 29285503, 29285788, 29285956, 29286037, 29287130
29287705, 29292837, 29293072, 29293574, 29297863, 29297915, 29298220
29299049, 29299082, 29299844, 29301463, 29301566, 29302963, 29303918
29304781, 29306226, 29306713, 29311588, 29312310, 29312672, 29312734
29312753, 29313347, 29313417, 29313525, 29314539, 29317756, 29318410
29319441, 29321489, 29323946, 29324568, 29324735, 29325087, 29325105
29325257, 29325765, 29325993, 29327044, 29329042, 29329087, 29329807
29330361, 29331066, 29331209, 29331380, 29331493, 29332292, 29332395
29332771, 29333500, 29336843, 29337310, 29338315, 29338453, 29338780
29338913, 29339101, 29339155, 29341209, 29343086, 29345937, 29346057
29346211, 29346943, 29347620, 29348176, 29350052, 29351386, 29351716
29351735, 29351749, 29352298, 29352724, 29352867, 29352947, 29353271
29353432, 29353960, 29355654, 29356547, 29356704, 29356711, 29356752
29358509, 29358828, 29360285, 29360672, 29360911, 29360950, 29361472
29361801, 29363151, 29364171, 29364177, 29366940, 29367019, 29367561

29368253, 29372541, 29373418, 29373588, 29374179, 29375941, 29376346
29377986, 29378029, 2937834, 29378913, 29379978, 29382784, 29382815
29384781, 29384854, 29384864, 29385429, 29385652, 29386502, 29386635
29386660, 29387073, 29387274, 29388020, 29388072, 29388094, 29388524
29388830, 29389889, 29390011, 29390435, 29390785, 29391030, 29394014
29394140, 29394749, 29395657, 29397954, 29397996, 29398488, 29398863
29399100, 29399121, 29399938, 29402131, 29404483, 29405012, 29405462
29405651, 29405996, 29407804, 29409149, 29410311, 29410834, 29411037
29412066, 29412269, 29417719, 29417884, 29420834, 29421059, 29423826
29424999, 29426241, 29429017, 29429264, 29429566, 29430524, 29431192
29431485, 29434301, 29435474, 29435652, 29436454, 29436514, 29437594
29437712, 29438277, 29438736, 29439522, 29441196, 29443187, 29443250
29444072, 29444282, 29444602, 29446669, 29449477, 29450421, 29451386
29452576, 29452936, 29452953, 29455424, 29457312, 29457370, 29457807
29460252, 29461791, 29462594, 29462767, 29462957, 29464779, 29465177
29467622, 29476473, 29483532, 29483672, 29483685, 29483712, 29486181
29488894, 29489546, 29490256, 29492127, 29492939, 29494245, 29495057
29495684, 29497311, 29500826, 29502561, 29503543, 29503631, 29503827
29504492, 29504682, 29505668, 29507616, 29510278, 29511611, 29514479
29515134, 29515240, 29515766, 29515834, 29516300, 29517168, 29521187
29521748, 29522358, 29522561, 29522662, 29523511, 29525467, 29525886
29527595, 29529147, 29530440, 29530515, 29530812, 29531654, 29541769
29542449, 29543034, 29546817, 29547867, 29548687, 29548722, 29549154
29549730, 29557336, 29558975, 29559187, 29559446, 29559908, 29559981
29565611, 29580983, 29581771, 29584261, 29586143, 29597536, 29597754
29598039, 29598233, 29599300, 29601461, 29604002, 29608000, 29611020
29611991, 29616244, 29616414, 29618074, 29618190, 29622936, 29626732
29628200, 29629650, 29629681, 29631749, 29632095, 29633697, 29635427
29635717, 29637362, 29638285, 29641736, 29653246, 29656819, 29657422
29664087, 29664161, 29670782, 29676089, 29677051, 29677733, 29679856
29687459, 29687763, 29692694, 29695841, 29703932, 29707099, 29742223
29747648, 29751094, 29753244, 29754951, 29756274, 29757651, 29766207
29766503, 29766679, 29775393, 29779196, 29795957, 29838773

Database engine: 18.0.0.0

The following versions are available for Oracle database engine 18.0.0.0:

- [Version 18.0.0.0.ru-2021-01.rur-2021-01.r1 \(p. 1293\)](#)
- [Version 18.0.0.0.ru-2020-10.rur-2020-10.r1 \(p. 1298\)](#)
- [Version 18.0.0.0.ru-2020-07.rur-2020-07.r1 \(p. 1303\)](#)
- [Version 18.0.0.0.ru-2020-04.rur-2020-04.r1 \(p. 1307\)](#)
- [Version 18.0.0.0.ru-2020-01.rur-2020-01.r1 \(p. 1311\)](#)
- [Version 18.0.0.0.ru-2019-10.rur-2019-10.r1 \(p. 1315\)](#)
- [Version 18.0.0.0.ru-2019-07.rur-2019-07.r1 \(p. 1318\)](#)

Note

Oracle Database 18c, Version 18.0.0.0 is on a deprecation path. Oracle Corporation will no longer provide patches for 18c after the end-of-support date. For more information, see [Preparing for the automatic upgrade of Oracle Database 18c \(p. 1214\)](#).

Version 18.0.0.0.ru-2021-01.rur-2021-01.r1

Version 18.0.0.0.ru-2021-01.rur-2021-01.r1 includes the following:

- Patch 32204699: DATABASE RELEASE UPDATE 18.13.0.0.210119

- Patch 32119939: OJVM RELEASE UPDATE 18.13.0.0.210119
- Patch 31335037: RDBMS - DSTV35 UPDATE - TZDATA2020A
- Patch 31335142: DSTV35 UPDATE - TZDATA2020A - NEED OJVM FIX
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 27539475: ORA-3816 - MISSING MESSAGE INFORMATION FOR 3816 ERROR
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER TABLE
- Patch 29374604: Golden Gate Integrated Extract not starting 18c/19c Standard Edition
- Patch 29782284: ORA-06508:"MDSYS.MDPRVT_IDX" WHILE UPGRADING DATABASE TO 18.3
- PreUpgrade Jar: preupgrade_181_cbuild_10_if.zip
- Java Cryptography Extension (JCE): Unlimited Strength Jurisdiction Policy Files for JVM version 8
- Support for [Setting parameters for advisor tasks \(p. 1099\)](#) using procedures in the rdsadmin.rdsadmin_util package

Combined patches for version 18.0.0.0.ru-2021-01.rur-2021-01.r1, released January 2021

Bugs fixed:

```
7391838, 8932139, 9062315, 12816839, 13554903, 14221306, 17468475
17958365, 18701017, 18986501, 20436508, 20549013, 20917487, 21095391
21223848, 21374587, 21547051, 21744603, 21766220, 21806121, 21935698
22174392, 22282748, 22363790, 22645496, 22729345, 22734786, 22820798
23003564, 23061453, 23109325, 23310101, 23698980, 23761724, 23763462
23840305, 24489904, 24596874, 24669730, 24687075, 24689376, 24737581
24763196, 24798481, 24841671, 24844841, 24903291, 24925863, 24971597
25031502, 25035594, 25035599, 25060506, 25092651, 25210690, 25287072
25293659, 25303284, 25309116, 25348956, 25404117, 25405687, 25416731
25487146, 25560538, 25573623, 25576115, 25591292, 25607397, 25634405
25644811, 25663488, 25686739, 25696520, 25709124, 25724089, 25726981
25736428, 25740844, 25743479, 25756945, 25809524, 25824236, 25882883
25890002, 25908728, 25911069, 25929650, 25943740, 25958554, 25986062
25997810, 26001677, 26083489, 26115103, 26164661, 26226953, 26237338
26281476, 26284722, 26297826, 26299684, 26313403, 26336101, 26362155
26375330, 26399691, 26399839, 26405036, 26410240, 26422277, 26423085
26427905, 26433972, 26440142, 26440169, 26441345, 26450454, 26476244
26521043, 26527054, 26536320, 26586174, 26587652, 26595088, 26598422
26615291, 26617804, 26646549, 26647619, 26654411, 26694735, 26716835
26724511, 26731697, 26745002, 26785169, 26790514, 26790923, 26792891
26798411, 26818960, 26822620, 26827699, 26843558, 26843664, 26846077
26860285, 26871815, 26882126, 26882316, 26883456, 26894737, 26895149
26898279, 26910716, 26914402, 26927998, 26928317, 26933599, 26943004
26943660, 26956033, 26960097, 26961415, 26966120, 26966916, 26970175
26976568, 26985002, 26986173, 26992964, 26996813, 27000158, 27005278
27006120, 27006664, 27012915, 27013566, 27016033, 27018734, 27026401
27028251, 27030974, 27032726, 27033520, 27034318, 27034688, 27035653
27036163, 27036408, 27037839, 27038986, 27040560, 27041253, 27044169
27044575, 27047831, 27053044, 27054231, 27058530, 27060167, 27060859
27061736, 27066451, 27066519, 27072923, 27073066, 27075854, 27080748
27080874, 27086406, 27086821, 27090765, 27092991, 27093423, 27098733
27100800, 27101105, 27101527, 27101652, 27105900, 27106301, 27106915
27110878, 27111780, 27112686, 27114112, 27115422, 27119621, 27119861
27121566, 27122162, 27125872, 27126666, 27128580, 27133637, 27135647
27142120, 27142529, 27143756, 27143882, 27144533, 27144928, 27147979
27150500, 27151826, 27152892, 27153641, 27153755, 27155549, 27156355
27160360, 27160922, 27163928, 27164352, 27165231, 27166354, 27166715
```

27169796, 27169888, 27170305, 27174938, 27174948, 27177551, 27177852
27179264, 27181521, 27181537, 27181897, 27182006, 27182064, 27184253
27185188, 27187440, 27189611, 27190851, 27193810, 27195935, 27197334
27199245, 27200959, 27202015, 27203055, 27204476, 27207634, 27208795
27208953, 27210038, 27210263, 27210872, 27212837, 27213140, 27214085
27214204, 27215007, 27216046, 27216224, 27217412, 27220610, 27220937
27221900, 27222121, 27222423, 27222626, 27222938, 27223075, 27224987
27226913, 27228786, 27229389, 27231051, 27232983, 27233563, 27234962
27236052, 27236110, 27236722, 27238077, 27238258, 27240246, 27240570
27241221, 27241247, 27242226, 27242616, 27244337, 27244785, 27244999
27249215, 27249531, 27249544, 27250547, 27251690, 27252023, 27254335
27254851, 27255377, 27256000, 27256488, 27256534, 27256584, 27257509
27258578, 27259307, 27259386, 27259983, 27260704, 27262601, 27262650
27262798, 27262945, 27262991, 27263276, 27263677, 27263996, 27264464
27265816, 27266245, 27267992, 27270197, 27271876, 27274143, 27274456
27274536, 27275136, 27275533, 27275776, 27276231, 27282707, 27283029
27283960, 27284375, 27284499, 27285244, 27285557, 27288230, 27288638
27288894, 27292213, 27293599, 27294480, 27299455, 27300007, 27301308
27301568, 27302415, 27302594, 27302632, 27302681, 27302695, 27302711
27302714, 27302730, 27302777, 27302800, 27302960, 27303287, 27303785
27303938, 27304131, 27304410, 27304906, 27304936, 27305318, 27307868
27308088, 27309182, 27310092, 27313687, 27314206, 27314390, 27314512
27314697, 27315159, 27318117, 27318869, 27320576, 27320985, 27321179
27321834, 27326204, 27329812, 27330158, 27330161, 27333658, 27333664
27333693, 27333731, 27334316, 27334353, 27334648, 27335682, 27338838
27338912, 27338946, 27339115, 27339396, 27339483, 27339495, 27341036
27343844, 27345190, 27345231, 27345450, 27345498, 27346329, 27346644
27346709, 27346949, 27346984, 27347126, 27348081, 27348707, 27349393
27350267, 27351628, 27352600, 27354783, 27356373, 27357773, 27358232
27358241, 27359178, 27359368, 27360126, 27362190, 27364854, 27364891
27364916, 27364947, 27365014, 27365139, 27365702, 27365993, 27367194
27368850, 27369515, 27370933, 27372756, 27375260, 27375542, 27376871
27377219, 27378103, 27378959, 27379233, 27379846, 27379956, 27381383
27381417, 27381498, 27381656, 27383281, 27384222, 27386467, 27389352
27392187, 27392968, 27393421, 27393570, 27394086, 27395404, 27395416
27395794, 27396357, 27396365, 27396377, 27396624, 27396666, 27396672
27396720, 27396794, 27396813, 27397048, 27398080, 27398223, 27398660
27399499, 27399762, 27399985, 27400416, 27400598, 27401618, 27401637
27403244, 27404017, 27404573, 27404599, 27404668, 27405242, 27405645
27405696, 27406105, 27410279, 27410300, 27410595, 27412805, 27416327
27416997, 27417186, 27420715, 27421101, 27421733, 27422874, 27423251
27424405, 27425507, 27425622, 27426277, 27426363, 27427805, 27428790
27430219, 27430254, 27430802, 27432062, 27432338, 27432355, 27432826
27433163, 27433385, 27433870, 27434050, 27434193, 27434486, 27434974
27435537, 27439835, 27441326, 27441980, 27442041, 27444727, 27445330
27445452, 27445462, 27445727, 27447452, 27447687, 27448162, 27449814
27450355, 27450400, 27450783, 27451049, 27451182, 27451187, 27451531
27452046, 27452760, 27452897, 27453225, 27454722, 27457666, 27457891
27458164, 27458829, 27459593, 27459909, 27459948, 27460675, 27461740
27462994, 27465480, 27466597, 27467543, 27468303, 27469245, 27469329
27471876, 27472969, 27473800, 27475272, 27479358, 27480784, 27481406
27481765, 27483974, 27484556, 27486253, 27486805, 27487309, 27487795
27487919, 27489107, 27489719, 27492916, 27493674, 27494663, 27496224
27496308, 27496424, 27496806, 27497950, 27498477, 27501327, 27501413
27501465, 27502420, 27503318, 27503413, 27504190, 27504770, 27505229
27505603, 27506774, 27507968, 27508936, 27508984, 27508985, 27510959
27511196, 27512439, 27513114, 27517818, 27518227, 27518310, 27519708
27520070, 27520900, 27522245, 27523368, 27523800, 27525909, 27526362
27526744, 27528204, 27529661, 27532009, 27532375, 27533780, 27533819
27534289, 27534509, 27537472, 27539475, 27539757, 27539876, 27540613
27541286, 27541468, 27542824, 27544030, 27544973, 27545630, 27547732
27550341, 27551855, 27554074, 27555481, 27558557, 27558559, 27558861
27560562, 27560602, 27560702, 27560735, 27562488, 27563629, 27563767
27565906, 27567477, 27570318, 27573154, 27573408, 27574335, 27576342
27576354, 27577122, 27577758, 27578007, 27579353, 27579969, 27580996
27581484, 27585755, 27585800, 27586810, 27586895, 27587672, 27587905

27588271, 27589260, 27591842, 27592466, 27593389, 27593501, 27593585
27593587, 27595096, 27595801, 27595973, 27599689, 27599927, 27600706
27601118, 27601441, 27602091, 27602488, 27603841, 27604293, 27605482
27607563, 27607805, 27608669, 27609819, 27610269, 27613080, 27613247
27613530, 27613554, 27615608, 27615649, 27616657, 27617522, 27617978
27620808, 27623159, 27623844, 27625010, 27625050, 27625274, 27625620
27627992, 27629756, 27629928, 27631506, 27632114, 27634676, 27634991
27635508, 27636900, 27642235, 27644757, 27645231, 27645940, 27649707
27652302, 27654039, 27654521, 27655217, 27657467, 27657712, 27657920
27658186, 27658205, 27662528, 27663370, 27664702, 27666312, 27668379
27671633, 27679488, 27679664, 27679793, 27679806, 27679961, 27680162
27680509, 27680669, 27682151, 27682288, 27686599, 27688036, 27688099
27688692, 27690513, 27690578, 27691717, 27691809, 27691920, 27691939
27692215, 27693416, 27693713, 27694261, 27695063, 27697092, 27698953
27700466, 27701795, 27702244, 27703242, 27704237, 27705761, 27707544
27708711, 27709046, 27710072, 27714373, 27717210, 27718914, 27719187
27723002, 27723151, 27725967, 27726269, 27726780, 27729678, 27731346
27732323, 27733415, 27734470, 27735534, 27739006, 27739957, 27740424
27740844, 27740854, 27744211, 27745220, 27745728, 27747407, 27747869
27748321, 27748954, 27751006, 27751755, 27753336, 27756900, 27757567
27757794, 27757888, 27757979, 27758544, 27758653, 27758972, 27759077
27759457, 27761402, 27766324, 27766679, 27767081, 27768034, 27769361
27772093, 27772815, 27773602, 27774320, 27774539, 27778433, 27779886
27780562, 27780683, 27782339, 27782464, 27783059, 27783289, 27786669
27786699, 27786772, 27791223, 27793533, 27797290, 27801337, 27801774
27803665, 27807441, 27810967, 27811439, 27812560, 27812593, 27813267
27815347, 27818389, 27818871, 27819881, 27824540, 27824543, 27825241
27828794, 27828892, 27829295, 27832643, 27833369, 27833672, 27834551
27834569, 27834984, 27835925, 27837219, 27839353, 27839616, 27839732
27840386, 27843646, 27846298, 27846499, 27847259, 27849825, 27850112
27850736, 27851757, 27856471, 27861226, 27861452, 27861909, 27862636
27864737, 27865439, 27869075, 27869339, 27873364, 27873412, 27873643
27876671, 27882176, 27886087, 27889841, 27892488, 27896388, 27896443
27896458, 27897639, 27897759, 27898015, 27900663, 27902561, 27906509
27908396, 27908644, 27909478, 27912301, 27917669, 27918832, 27920184
27924147, 27926113, 27927431, 27929287, 27929509, 27930478, 27931299
27931506, 27934468, 27935348, 27935464, 27935493, 27935826, 27936676
27938736, 27940876, 27941110, 27941514, 27941896, 27945870, 27948050
27948153, 27950708, 27952586, 27952762, 27957892, 27959594, 27960021
27961746, 27964051, 27964513, 27965400, 27965830, 27966472, 27967484
27970265, 27971503, 27971575, 27972265, 27975778, 27977039, 27978668
27983174, 27984028, 27984314, 27986817, 27989556, 27989849, 27991970
27993289, 27993298, 27994325, 27994333, 27995215, 27995248, 27997875
27998003, 27999073, 27999597, 27999638, 28000269, 28004853, 28006704
28007516, 28018962, 28019283, 28019592, 28021205, 28022101, 28022847
28023081, 28023399, 28023410, 28023482, 28024347, 28024793, 28025398
28025414, 28026866, 28032758, 28033429, 28036487, 28039471, 28039953
28043157, 28045209, 28045903, 28057267, 28058612, 28059199, 28066655
28067846, 28071549, 28072130, 28072383, 28072464, 28072567, 28073470
28074713, 28079127, 28085865, 28088762, 28089440, 28090453, 28091981
28092783, 28098040, 28098160, 28098865, 28099592, 28103600, 28103869
28104176, 28104361, 28104409, 28106402, 28108003, 28108898, 28109326
28109698, 28111583, 28120036, 28120951, 28124631, 28125601, 28125947
28127569, 28129791, 28131767, 28132287, 28135648, 28157786, 28164480
28165439, 28165545, 28169711, 28174827, 28174926, 28174951, 28175445
28180464, 28181021, 28182503, 28184554, 28184800, 28187706, 28188330
28189466, 28194173, 28199085, 28201419, 28204262, 28204423, 28204443
28209341, 28209985, 28210192, 28211734, 28214943, 28215510, 28218832
28220398, 28223871, 28226179, 28227512, 28229360, 28236305, 28238264
28240153, 28242712, 28250929, 28256164, 28257335, 28258608, 28264172
28271107, 28271119, 28271693, 28276054, 28278547, 28278640, 28279837
28281094, 28282606, 28285766, 28290434, 28294563, 28302049, 28304709
28305001, 28305362, 28305607, 28309182, 28309406, 28312508, 28315031
28315995, 28319114, 28319623, 28320117, 28320399, 28321446, 28323201
28328895, 28329450, 28330714, 28330971, 28333072, 28338399, 28338999
28344964, 28350595, 28354603, 28357401, 28361083, 28361221, 28361787

28365111, 28369092, 28371123, 28373960, 28375383, 28378446, 28379065
28384353, 28385102, 28386259, 28388910, 28389153, 28390273, 28391210
28391582, 28392168, 28392251, 28393678, 28394726, 28396445, 28397317
28401116, 28402823, 28403295, 28413955, 28420042, 28420457, 28423598
28432129, 28434028, 28435825, 28437849, 28439086, 28445741, 28448314
28454215, 28455212, 28468312, 28468493, 28475164, 28478676, 28481149
28481679, 28483184, 28489150, 28492362, 28493478, 28498976, 28501075
28502098, 28502403, 28502773, 28503038, 28503484, 28504545, 28507324
28508053, 28508296, 28508557, 28512336, 28512761, 28513333, 28514693
28521330, 28527416, 28528349, 28530171, 28535127, 28535272, 28538439
28542455, 28544633, 28545134, 28545687, 28546290, 28547068, 28547478
28553468, 28558645, 28564479, 28565296, 28571483, 28572407, 28572834
28578164, 28578945, 28580528, 28584193, 28584217, 28584444, 28585411
28587723, 28589509, 28600233, 28601874, 28602253, 28606598, 28608211
28611037, 28612674, 28614072, 28617631, 28617959, 28621470, 28622202
28627255, 28627686, 28632559, 28636676, 28639299, 28642273, 28642899
28644549, 28646200, 28670445, 28673203, 28678804, 28679454, 28680029
28685371, 28689483, 28690694, 28692103, 28692275, 28695694, 28697526
28697806, 28702188, 28703812, 28708023, 28709063, 28710469, 28710734
28710827, 28713840, 28714058, 28714988, 28715655, 28728040, 28728272
28730044, 28730076, 28730253, 28734355, 28740708, 28742555, 28745367
28747182, 28749289, 28752599, 28755011, 28757758, 28758090, 28758722
28761812, 28767240, 28770146, 28774416, 28776431, 28776811, 28777174
28777214, 28777332, 28781754, 28785022, 28785531, 28791725, 28793062
28794230, 28797711, 28803345, 28805612, 28805695, 28808314, 28809909
28817449, 28819640, 28820669, 28821847, 28827682, 28830691, 28831971
28835937, 28836716, 28838066, 28844866, 28847136, 28849751, 28852325
28852691, 28855922, 28856060, 28856172, 28863263, 28863487, 28865569
28867992, 28876639, 28878525, 28881723, 28887305, 28887509, 28889389
28889730, 28891984, 28900506, 28905390, 28905457, 28910498, 28910586
28915870, 28919145, 28925880, 28927452, 28938924, 28940179, 28945922
28948554, 28949888, 28950868, 28951014, 28951382, 28956908, 28959493
28960211, 28965084, 28965095, 28973650, 28986231, 28986257, 28987454
28993295, 28993353, 28993590, 28994890, 29000190, 29002488, 29006527
29007321, 29007353, 29009513, 29013832, 29015118, 29015706, 29024054
29026309, 29026582, 29027694, 29027940, 29032276, 29033896, 29036278
29037290, 29039510, 29040739, 29044086, 29044954, 29048498, 29048728
29050886, 29051702, 29055644, 29056270, 29056767, 29060216, 29061016
29115857, 29123482, 29125374, 29136111, 29139070, 29139591, 29154725
29158680, 29163567, 29165682, 29170232, 29171683, 29173817, 29177886
29179097, 29182517, 29182901, 29189889, 29190663, 29198092, 29200700
29202461, 29203604, 29205918, 29212433, 29213320, 29213351, 29213893
29224605, 29224710, 29225076, 29230252, 29230565, 29233415, 29237575
29241345, 29242017, 29247712, 29247906, 29249289, 29250317, 29254623
29255273, 29260956, 29261548, 29278684, 29281112, 29285503, 29296257
29301463, 29307638, 29311927, 29312672, 29312889, 29314539, 29331209
29331493, 29332763, 29337294, 29338348, 29339155, 29342099, 29343086
29343156, 29343861, 29344541, 29346057, 29347981, 29350868, 29351662
29351771, 29353821, 29356752, 29361472, 29362596, 29364171, 29366406
29372069, 29372460, 29374604, 29375355, 29375984, 29376346, 29378913
29379978, 29382784, 29383695, 29386635, 29388020, 29388952, 29391849
29394749, 29395657, 29396481, 29398488, 29399046, 29399336, 29404483
29405462, 29407804, 29408853, 29409149, 29409455, 29412269, 29417719
29418165, 29420254, 29426241, 29428230, 29429264, 29430524, 29434301
29436454, 29437712, 29439522, 29442936, 29445548, 29448498, 29450812
29452251, 29454978, 29457978, 29463047, 29464779, 29465177, 29472618
29477015, 29483626, 29483672, 29483723, 29483771, 29489436, 29493122
29500257, 29500963, 29501218, 29504682, 29506942, 29511611, 29515766
29521862, 29524599, 29524985, 29525886, 29530515, 29531541, 29536342
29538631, 29541742, 29542449, 29542580, 29548413, 29548592, 29549071
29557261, 29558238, 29559395, 29564592, 29579919, 29580394, 29591343
29604257, 29607136, 29608023, 29614098, 29614987, 29616244, 29625065
29626154, 29629430, 29629745, 29632265, 29633753, 29637526, 29637560
29643721, 29645167, 29645349, 29651520, 29656843, 29667994, 29668005
29670713, 29676089, 29685137, 29687220, 29687459, 29688867, 29703195
29705793, 29707896, 29717901, 29719146, 29720133, 29722167, 29724041

29726695, 29739576, 29741319, 29766435, 29769901, 29773197, 29774362
29780140, 29782211, 29782284, 29789911, 29791152, 29794174, 29794462
29796916, 29807964, 29809792, 29813494, 29814995, 29815341, 29817278
29822714, 29825525, 29827852, 29841687, 29844131, 29846645, 29850930
29853485, 29865188, 29869404, 29869906, 29875459, 29876358, 29881050
29881575, 29884958, 29891916, 29893132, 29896510, 29902299, 29914449
29922225, 29930457, 29932310, 29941062, 29942554, 29944035, 29944159
29944660, 29951620, 29951759, 29961353, 29962927, 29962939, 29965888
29991257, 29997326, 29997937, 30008125, 30014200, 30018017, 30018903
30031027, 30034456, 30039959, 30064268, 30068871, 30073422, 30073744
30074349, 30076253, 30078675, 30078934, 30085980, 30088912, 30092280
30098251, 30099302, 30114477, 30116203, 30117469, 30120608, 30125944
30128047, 30131286, 30139392, 30147928, 30149035, 30160625, 30163243
30164714, 30173113, 30177597, 30179644, 30186706, 30189023, 30193165
30193736, 30194710, 30196358, 30200680, 30200758, 30215130, 30218044
30218317, 30223712, 30225443, 30232638, 30239480, 30240547, 30241567
30246179, 30247305, 30252098, 30252156, 30253255, 30259008, 30265523
30265615, 30272329, 30282501, 30283932, 30293345, 30305880, 30312094
30312559, 30316897, 30320029, 30325407, 30331356, 30342878, 30345926
30350543, 30352623, 30355490, 30357897, 30361635, 30364613, 30365745
30368482, 30368668, 30372081, 30374739, 30376986, 30377692, 30381207
30384121, 30384152, 30387666, 30391272, 30396120, 30397100, 30402386
30403763, 30408515, 30409339, 30412188, 30413137, 30416034, 30421204
30431274, 30441687, 30443393, 30450787, 30453442, 30458593, 30460922
30464250, 30464655, 30473634, 30474774, 30475115, 30476768, 30485255
30496957, 30497057, 30498824, 30501574, 30503943, 30509277, 30510527
30517516, 30522998, 30528547, 30528704, 30532811, 30533172, 30534662
30544595, 30571306, 30578221, 30581448, 30582500, 30599407, 30602230
30606345, 30613937, 30619525, 30623138, 30624864, 30635302, 30652853
30654454, 30657365, 30662736, 30668407, 30671813, 30679595, 30679771
30681462, 30691604, 30698289, 30741263, 30749644, 30751639, 30755348
30758943, 30773164, 30783551, 30803210, 30809087, 30814266, 30814285
30815852, 30816938, 30826474, 30829779, 30841241, 30855101, 30856358
30860803, 30866988, 30870439, 30881588, 30887501, 30891760, 30896620
30904672, 30914674, 30919804, 30922819, 30930149, 30931311, 30937340
30957739, 30964194, 30968737, 30978554, 30985027, 30987088, 30994996
30998759, 31001455, 31004719, 31013127, 31019767, 31022858, 31028986
31029936, 31046619, 31058548, 31061482, 31100172, 31104809, 31106577
31109506, 31115502, 31156383, 31172207, 31182793, 31192039, 31194264
31200845, 31201001, 31204878, 31215438, 31228670, 31233170, 31254535
31258101, 31298871, 31302462, 31306248, 31306261, 31309867, 31315876
31326608, 31331354, 31335037, 31335142, 31343752, 31348018, 31377487
31393600, 31408636, 31430722, 31454972, 31476736, 31501139, 31525783
31544097, 31570161, 31574244, 31600023, 31628311, 31637680, 31658464
31658943, 31663788, 31668061, 31668872, 31674731, 31696577, 31711889
31718134, 31748000, 31749759, 31758083, 31769373, 31783451, 31786838
31799775, 31816631, 31867037, 31883124, 31886547, 31905033, 31909295
31921267, 31927930, 31986836, 31997805, 32032887, 32079739, 32089820
32097882, 32105135, 32165759, 32234161, 32290399, 32296941

Version 18.0.0.0.ru-2020-10.rur-2020-10.r1

Version 18.0.0.0.ru-2020-10.rur-2020-10.r1 includes the following:

- Patch 31730250: Database Release Update: 18.12.0.0.201020 (31730250)
- Patch 31668892: OJVM RELEASE UPDATE: 18.12.0.0.201020 (31668892)
- Patch 31335037: RDBMS - DSTV35 UPDATE - TZDATA2020A
- Patch 31335142: DSTV35 UPDATE - TZDATA2020A - NEED OJVM FIX
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE

- Patch 29374604: IE not starting against 18c Oracle RDBMS Standard Edition.
- PreUpgrade Jar: preupgrade_181_cbuild_10_lf.zip
- Support for [Setting and unsetting system diagnostic events \(p. 1046\)](#) using procedures in the rdsadmin.rdsadmin_util package
- Support for the procedure rdsadmin_util.truncate_apply\$_cdr_info described in [Integrated REPLICAT slow due to query on sys."_DBA_APPLY_CDR_INFO" \(p. 1236\)](#)

Combined patches for version 18.0.0.0.ru-2020-10.rur-2020-10.r1, released October 2020

Bugs fixed:

7391838, 8932139, 9062315, 12816839, 13554903, 14221306, 17468475
17958365, 18701017, 18986501, 20436508, 20549013, 20917487, 21095391
21223848, 21374587, 21547051, 21744603, 21766220, 21806121, 21935698
22174392, 22282748, 22363790, 22645496, 22729345, 22734786, 23003564
23061453, 23109325, 23310101, 23698980, 23761724, 23763462, 23840305
24489904, 24596874, 24669730, 24687075, 24689376, 24737581, 24763196
24798481, 24841671, 24844841, 24903291, 24925863, 24971597, 25031502
25035594, 25035599, 25060506, 25092651, 25210690, 25287072, 25293659
25303284, 25309116, 25348956, 25404117, 25405687, 25416731, 25487146
25560538, 25573623, 25576115, 25591292, 25607397, 25634405, 25644811
25663488, 25686739, 25696520, 25709124, 25724089, 25726981, 25736428
25740844, 25743479, 25756945, 25809524, 25824236, 25882883, 25890002
25908728, 25911069, 25929650, 25943740, 25958554, 25986062, 25997810
26001677, 26083489, 26115103, 26164661, 26226953, 26237338, 26281476
26284722, 26297826, 26299684, 26313403, 26336101, 26362155, 26375330
26399691, 26399839, 26405036, 26410240, 26422277, 26423085, 26427905
26433972, 26440142, 26440169, 26441345, 26450454, 26476244, 26521043
26527054, 26536320, 26586174, 26587652, 26595088, 26598422, 26615291
26617804, 26646549, 26647619, 26654411, 26694735, 26716835, 26724511
26731697, 26745002, 26785169, 26790514, 26790923, 26792891, 26798411
26818960, 26822620, 26827699, 26843558, 26843664, 26846077, 26860285
26871815, 26882126, 26882316, 26883456, 26894737, 26895149, 26898279
26910716, 26914402, 26927998, 26928317, 26933599, 26943004, 26943660
26956033, 26960097, 26961415, 26966120, 26966916, 26970175, 26976568
26985002, 26986173, 26992964, 26996813, 27000158, 27005278, 27006120
27006664, 27012915, 27013566, 27016033, 27018734, 27026401, 27028251
27030974, 27032726, 27033520, 27034318, 27034688, 27035653, 27036163
27036408, 27037839, 27038986, 27040560, 27041253, 27044169, 27044575
27047831, 27053044, 27054231, 27058530, 27060167, 27060859, 27061736
27066451, 27066519, 27072923, 27073066, 27075854, 27080748, 27080874
27086406, 27086821, 27090765, 27092991, 27093423, 27098733, 27100800
27101105, 27101527, 27101652, 27105900, 27106301, 27106915, 27110878
27111780, 27112686, 27114112, 27115422, 27119621, 27119861, 27121566
27122162, 27125872, 27126666, 27128580, 27133637, 27135647, 27142120
27142529, 27143756, 27143882, 27144533, 27144928, 27147979, 27150500
27151826, 27152892, 27153641, 27153755, 27155549, 27156355, 27160360
27160922, 27163928, 27164352, 27165231, 27166354, 27166715, 27169796
27169888, 27170305, 27174938, 27174948, 27177551, 27177852, 27179264
27181521, 27181537, 27181897, 27182006, 27182064, 27184253, 27185188
27187440, 27189611, 27190851, 27193810, 27195935, 27197334, 27199245
27200959, 27202015, 27203055, 27204476, 27207634, 27208795, 27208953
27210038, 27210263, 27210872, 27212837, 27213140, 27214085, 27214204
27215007, 27216046, 27216224, 27217412, 27220610, 27220937, 27221900
27222121, 27222423, 27222626, 27222938, 27223075, 27224987, 27226913
27228786, 27229389, 27231051, 27232983, 27233563, 27234962, 27236052
27236110, 27236722, 27238077, 27238258, 27240246, 27240570, 27241221
27241247, 27242226, 27242616, 27244337, 27244785, 27244999, 27249215
27249531, 27249544, 27250547, 27251690, 27252023, 27254335, 27254851

27255377, 27256000, 27256488, 27256534, 27256584, 27257509, 27258578
27259307, 27259386, 27259983, 27262601, 27262650, 27262798, 27262945
27262991, 27263276, 27263677, 27263996, 27264464, 27265816, 27266245
27267992, 27270197, 27271876, 27274143, 27274456, 27274536, 27275136
27275533, 27275776, 27276231, 27282707, 27283029, 27283960, 27284375
27284499, 27285244, 27285557, 27288230, 27288638, 27288894, 27292213
27293599, 27294480, 27299455, 27300007, 27301308, 27301568, 27302415
27302594, 27302632, 27302681, 27302695, 27302711, 27302714, 27302730
27302777, 27302800, 27302960, 27303287, 27303785, 27303938, 27304131
27304410, 27304906, 27304936, 27305318, 27307868, 27308088, 27309182
27310092, 27313687, 27314206, 27314390, 27314512, 27314697, 27315159
27318117, 27318869, 27320576, 27320985, 27321179, 27321834, 27326204
27329812, 27330158, 27330161, 27333658, 27333664, 27333693, 27333731
27334316, 27334353, 27334648, 27335682, 27338838, 27338912, 27338946
27339115, 27339396, 27339483, 27339495, 27341036, 27343844, 27345190
27345231, 27345450, 27345498, 27346329, 27346644, 27346709, 27346949
27346984, 27347126, 27348081, 27348707, 27349393, 27350267, 27351628
27352600, 27354783, 27356373, 27357773, 27358232, 27358241, 27359178
27359368, 27360126, 27362190, 27364854, 27364891, 27364916, 27364947
27365014, 27365139, 27365702, 27365993, 27367194, 27368850, 27369515
27370933, 27372756, 27375260, 27375542, 27376871, 27377219, 27378103
27378959, 27379233, 27379846, 27379956, 27381383, 27381417, 27381498
27381656, 27383281, 27384222, 27386467, 27389352, 27392187, 27392968
27393421, 27393570, 27394086, 27395404, 27395416, 27395794, 27396357
27396365, 27396377, 27396624, 27396666, 27396672, 27396720, 27396794
27396813, 27397048, 27398080, 27398223, 27398660, 27399499, 27399762
27399985, 27400416, 27400598, 27401618, 27401637, 27403244, 27404573
27404599, 27404668, 27405242, 27405645, 27405696, 27406105, 27410279
27410300, 27410595, 27412805, 27416327, 27416997, 27417186, 27420715
27421101, 27421733, 27422874, 27423251, 27424405, 27425507, 27425622
27426277, 27426363, 27427805, 27428790, 27430219, 27430254, 27430802
27432062, 27432338, 27432355, 27432826, 27433163, 27433385, 27433870
27434050, 27434193, 27434486, 27434974, 27435537, 27439835, 27441326
27441980, 27442041, 27444727, 27445330, 27445462, 27445727, 27447452
27447687, 27448162, 27449814, 27450355, 27450400, 27450783, 27451049
27451182, 27451187, 27451531, 27452046, 27452760, 27452897, 27453225
27454722, 27457666, 27457891, 27458164, 27458829, 27459593, 27459909
27459948, 27460675, 27461740, 27462994, 27465480, 27466597, 27467543
27468303, 27469245, 27469329, 27471876, 27472969, 27473800, 27475272
27479358, 27480784, 27481406, 27481765, 27483974, 27484556, 27486253
27486805, 27487309, 27487795, 27487919, 27489107, 27489719, 27492916
27493674, 27494663, 27496224, 27496308, 27496424, 27496806, 27497950
27498477, 27501327, 27501413, 27501465, 27502420, 27503318, 27503413
27504190, 27504770, 27505229, 27505603, 27506774, 27507968, 27508936
27508984, 27508985, 27510959, 27511196, 27512439, 27513114, 27517818
27518227, 27518310, 27519708, 27520070, 27520900, 27522245, 27523368
27523800, 27525909, 27526362, 27526744, 27528204, 27529661, 27532009
27532375, 27533780, 27533819, 27534289, 27534509, 27537472, 27539757
27539876, 27540613, 27541286, 27541468, 27542824, 27544030, 27544973
27545630, 27547732, 27550341, 27551855, 27554074, 27555481, 27558557
27558559, 27558861, 27560562, 27560602, 27560702, 27560735, 27562488
27563629, 27563767, 27565906, 27567477, 27570318, 27573154, 27573408
27574335, 27576342, 27576354, 27577122, 27577758, 27578007, 27579353
27579969, 27580996, 27581484, 27585755, 27585800, 27586810, 27586895
27587672, 27587905, 27588271, 27589260, 27591842, 27592466, 27593389
27593501, 27593585, 27593587, 27595096, 27595801, 27595973, 27599689
27599927, 27600706, 27601118, 27601441, 27602091, 27602488, 27603841
27604293, 27605482, 27607563, 27607805, 27608669, 27609819, 27610269
27613080, 27613247, 27613530, 27613554, 27615608, 27615649, 27616657
27617522, 27617978, 27620808, 27623159, 27623844, 27625010, 27625050
27625274, 27625620, 27627992, 27629756, 27629928, 27631506, 27632114
27634676, 27634991, 27635508, 27636900, 27642235, 27644757, 27645231
27645940, 27649707, 27652302, 27654039, 27654521, 27655217, 27657467
27657712, 27657920, 27658186, 27658205, 27662528, 27663370, 27664702
27666312, 27668379, 27671633, 27679488, 27679664, 27679793, 27679806
27679961, 27680162, 27680509, 27680669, 27682151, 27682288, 27686599

27688036, 27688099, 27688692, 27690513, 27690578, 27691717, 27691809
27691920, 27691939, 27692215, 27693416, 27693713, 27694261, 27695063
27697092, 27698953, 27700466, 27701795, 27702244, 27703242, 27704237
27705761, 27707544, 27708711, 27709046, 27710072, 27714373, 27717210
27718914, 27719187, 27723002, 27723151, 27725967, 27726269, 27726780
27729678, 27731346, 27732323, 27733415, 27734470, 27735534, 27739006
27739957, 27740424, 27740844, 27740854, 27744211, 27745220, 27745728
27747407, 27747869, 27748321, 27748954, 27751006, 27751755, 27753336
27756900, 27757567, 27757794, 27757888, 27757979, 27758544, 27758653
27758972, 27759077, 27759457, 27761402, 27766324, 27766679, 27767081
27768034, 27769361, 27772093, 27772815, 27773602, 27774320, 27774539
27778433, 27779886, 27780562, 27780683, 27782339, 27782464, 27783059
27783289, 27786669, 27786699, 27786772, 27791223, 27793533, 27797290
27801337, 27801774, 27803665, 27807441, 27810967, 27811439, 27812560
27812593, 27813267, 27815347, 27818389, 27818871, 27819881, 27824540
27824543, 27825241, 27828794, 27828892, 27829295, 27832643, 27833369
27833672, 27834551, 27834569, 27834984, 27835925, 27837219, 27839353
27839616, 27839732, 27840386, 27843646, 27846298, 27846499, 27847259
27849825, 27850112, 27850736, 27851757, 27856471, 27861226, 27861452
27861909, 27862636, 27864737, 27865439, 27869075, 27869339, 27873412
27873643, 27876671, 27882176, 27886087, 27889841, 27892488, 27896388
27896443, 27896458, 27897639, 27897759, 27898015, 27900663, 27902561
27906509, 27908396, 27908644, 27909478, 27912301, 27917669, 27918832
27920184, 27924147, 27926113, 27927431, 27929287, 27929509, 27930478
27931299, 27931506, 27934468, 27935348, 27935464, 27935493, 27935826
27936676, 27938736, 27940876, 27941110, 27941514, 27941896, 27945870
27948050, 27948153, 27950708, 27952586, 27952762, 27957892, 27959594
27960021, 27961746, 27964051, 27964513, 27965400, 27965830, 27966472
27967484, 27970265, 27971503, 27971575, 27972265, 27975778, 27977039
27978668, 27983174, 27984028, 27984314, 27986817, 27989556, 27989849
27991970, 27993289, 27993298, 27994325, 27994333, 27995215, 27995248
27997875, 27998003, 27999073, 27999597, 27999638, 28000269, 28004853
28006704, 28018962, 28019283, 28019592, 28021205, 28022101, 28022847
28023081, 28023399, 28023410, 28023482, 28024347, 28024793, 28025398
28025414, 28026866, 28032758, 28033429, 28036487, 28039471, 28039953
28043157, 28045209, 28045903, 28057267, 28058612, 28059199, 28066655
28067846, 28071549, 28072130, 28072383, 28072464, 28072567, 28073470
28074713, 28079127, 28085865, 28088762, 28089440, 28090453, 28091981
28092783, 28098040, 28098160, 28098865, 28099592, 28103600, 28103869
28104176, 28104361, 28104409, 28106402, 28108003, 28108898, 28109326
28109698, 28111583, 28120036, 28120951, 28124631, 28125601, 28125947
28129791, 28131767, 28132287, 28135648, 28157786, 28164480, 28165439
28165545, 28169711, 28174827, 28174926, 28174951, 28175445, 28180464
28181021, 28182503, 28184554, 28184800, 28187706, 28188330, 28189466
28194173, 28199085, 28201419, 28204262, 28204423, 28204443, 28209341
28209985, 28210192, 28211734, 28214943, 28215510, 28218832, 28220398
28223871, 28226179, 28227512, 28229360, 28236305, 28238264, 28240153
28242712, 28250929, 28256164, 28258608, 28264172, 28271107, 28271119
28271693, 28276054, 28278547, 28278640, 28279837, 28281094, 28282606
28285766, 28290434, 28294563, 28302049, 28304709, 28305001, 28305362
28305607, 28309182, 28309406, 28312508, 28315031, 28315995, 28319114
28319623, 28320117, 28320399, 28321446, 28323201, 28328895, 28329450
28330714, 28330971, 28333072, 28338399, 28338999, 28344964, 28350595
28354603, 28357401, 28361083, 28361221, 28361787, 28365111, 28369092
28371123, 28373960, 28375383, 28378446, 28379065, 28384353, 28385102
28386259, 28388910, 28389153, 28390273, 28391210, 28391582, 28392168
28392251, 28393678, 28394726, 28396445, 28397317, 28401116, 28402823
28403295, 28413955, 28420042, 28420457, 28423598, 28432129, 28434028
28435825, 28437849, 28439086, 28445741, 28448314, 28454215, 28455212
28468312, 28468493, 28475164, 28478676, 28481149, 28481679, 28483184
28489150, 28492362, 28493478, 28498976, 28501075, 28502098, 28502403
28502773, 28503038, 28503484, 28504545, 28507324, 28508053, 28508296
28508557, 28512336, 28512761, 28513333, 28514693, 28521330, 28527416
28528349, 28530171, 28535127, 28535272, 28538439, 28542455, 28544633
28545134, 28545687, 28546290, 28547068, 28547478, 28553468, 28558645
28564479, 28565296, 28571483, 28572407, 28572834, 28578164, 28578945

28580528, 28584193, 28584217, 28584444, 28585411, 28587723, 28589509
28600233, 28601874, 28602253, 28606598, 28608211, 28611037, 28612674
28614072, 28617631, 28617959, 28621470, 28622202, 28627255, 28627686
28632559, 28636676, 28639299, 28642273, 28642899, 28644549, 28646200
28670445, 28673203, 28678804, 28679454, 28680029, 28685371, 28689483
28690694, 28692103, 28692275, 28695694, 28697526, 28697806, 28702188
28703812, 28708023, 28709063, 28710469, 28710734, 28710827, 28713840
28714058, 28714988, 28715655, 28728040, 28728272, 28730044, 28730076
28730253, 28734355, 28740708, 28742555, 28745367, 28747182, 28749289
28752599, 28755011, 28757758, 28758090, 28758722, 28761812, 28767240
28770146, 28774416, 28776431, 28776811, 28777174, 28777214, 28777332
28781754, 28785022, 28785531, 28791725, 28793062, 28797711, 28803345
28805612, 28805695, 28808314, 28809909, 28817449, 28819640, 28820669
28821847, 28827682, 28830691, 28831971, 28835937, 28836716, 28838066
28844866, 28847136, 28849751, 28852325, 28852691, 28855922, 28856060
28856172, 28863263, 28863487, 28865569, 28867992, 28876639, 28878525
28881723, 28887305, 28887509, 28889730, 28891984, 28900506, 28905390
28905457, 28910498, 28910586, 28915870, 28919145, 28925880, 28927452
28938924, 28940179, 28945922, 28948554, 28949888, 28950868, 28951014
28951382, 28956908, 28959493, 28960211, 28965084, 28965095, 28986231
28986257, 28987454, 28993295, 28993353, 28993590, 28994890, 29000190
29002488, 29006527, 29007321, 29007353, 29009513, 29013832, 29015118
29015706, 29024054, 29026309, 29026582, 29027694, 29027940, 29032276
29033896, 29036278, 29037290, 29039510, 29040739, 29044086, 29044954
29048498, 29048728, 29050886, 29051702, 29055644, 29056270, 29056767
29060216, 29061016, 29115857, 29123482, 29125374, 29136111, 29139070
29139591, 29154725, 29158680, 29163567, 29165682, 29170232, 29171683
29173817, 29177886, 29179097, 29182517, 29182901, 29189889, 29190663
29198092, 29200700, 29202461, 29203604, 29205918, 29212433, 29213320
29213351, 29213893, 29224605, 29224710, 29225076, 29230252, 29230565
29233415, 29237575, 29241345, 29242017, 29247712, 29247906, 29249289
29250317, 29254623, 29255273, 29260956, 29261548, 29278684, 29285503
29296257, 29301463, 29307638, 29311927, 29312672, 29312889, 29314539
29331209, 29331493, 29332763, 29337294, 29338348, 29339155, 29343086
29343156, 29343861, 29344541, 29346057, 29347981, 29350868, 29351662
29351771, 29353821, 29356752, 29361472, 29362596, 29364171, 29366406
29372069, 29372460, 29374604, 29375355, 29375984, 29376346, 29378913
29379978, 29382784, 29383695, 29386635, 29388020, 29388952, 29391849
29394749, 29395657, 29398488, 29399046, 29399336, 29404483, 29405462
29407804, 29408853, 29409149, 29409455, 29412269, 29417719, 29418165
29420254, 29426241, 29428230, 29429264, 29430524, 29434301, 29436454
29437712, 29439522, 29442936, 29445548, 29448498, 29450812, 29452251
29454978, 29457978, 29463047, 29464779, 29465177, 29472618, 29477015
29483626, 29483672, 29483723, 29483771, 29489436, 29493122, 29500257
29500963, 29501218, 29504682, 29506942, 29511611, 29515766, 29521862
29524599, 29524985, 29525886, 29530515, 29531541, 29536342, 29538631
29541742, 29542449, 29542580, 29548413, 29548592, 29549071, 29557261
29558238, 29559395, 29564592, 29579919, 29580394, 29591343, 29604257
29607136, 29608023, 29614098, 29614987, 29616244, 29625065, 29626154
29629430, 29629745, 29632265, 29633753, 29637526, 29637560, 29643721
29645167, 29645349, 29651520, 29656843, 29667994, 29668005, 29676089
29685137, 29687220, 29687459, 29688867, 29703195, 29705793, 29707896
29717901, 29719146, 29720133, 29722167, 29724041, 29726695, 29739576
29741319, 29766435, 2976901, 29773197, 29774362, 29780140, 29782211
29789911, 29791152, 29794174, 29794462, 29807964, 29809792, 29813494
29815341, 29817278, 29822714, 29825525, 29827852, 29841687, 29844131
29846645, 29850930, 29853485, 29865188, 29869404, 29869906, 29875459
29876358, 29881050, 29881575, 29884958, 29891916, 29893132, 29896510
29902299, 29914449, 29922225, 29930457, 29941062, 29942554, 29944035
29944159, 29944660, 29951620, 29951759, 29961353, 29962927, 29962939
29965888, 29991257, 29997326, 29997937, 30008125, 30018017, 30018903
30031027, 30034456, 30039959, 30064268, 30068871, 30073744, 30074349
30076253, 30078934, 30085980, 30088912, 30092280, 30098251, 30099302
30114477, 30116203, 30120608, 30125944, 30128047, 30131286, 30139392
30147928, 30149035, 30160625, 30163243, 30164714, 30173113, 30177597
30179644, 30186706, 30189023, 30193165, 30193736, 30194710, 30196358

30200680, 30200758, 30215130, 30218044, 30218317, 30223712, 30225443
30232638, 30239480, 30240547, 30241567, 30246179, 30247305, 30252098
30252156, 30253255, 30259008, 30265523, 30265615, 30272329, 30282501
30283932, 30293345, 30305880, 30312094, 30312559, 30316897, 30320029
30325407, 30331356, 30342878, 30350543, 30352623, 30355490, 30357897
30364613, 30365745, 30368482, 30368668, 30372081, 30374739, 30376986
30381207, 30384121, 30384152, 30387666, 30391272, 30397100, 30402386
30403763, 30408515, 30409339, 30412188, 30413137, 30416034, 30421204
30431274, 30441687, 30443393, 30450787, 30453442, 30458593, 30460922
30464250, 30464655, 30473634, 30474774, 30475115, 30476768, 30485255
30496957, 30497057, 30498824, 30501574, 30503943, 30509277, 30510527
30517516, 30522998, 30528547, 30528704, 30532811, 30533172, 30534662
30544595, 30578221, 30581448, 30582500, 30599407, 30602230, 30606345
30613937, 30623138, 30624864, 30635302, 30652853, 30654454, 30662736
30668407, 30671813, 30679595, 30679771, 30681462, 30691604, 30698289
30741263, 30749644, 30755348, 30758943, 30773164, 30783551, 30803210
30814266, 30814285, 30815852, 30816938, 30829779, 30855101, 30856358
30866988, 30870439, 30881588, 30887501, 30904672, 30914674, 30919804
30922819, 30937340, 30957739, 30964194, 30968737, 30985027, 30987088
30994996, 30998759, 31001455, 31004719, 31013127, 31019767, 31022858
31028986, 31029936, 31061482, 31100172, 31104809, 31106577, 31109506
31115502, 31156383, 31172207, 31182793, 31192039, 31194264, 31200845
31201001, 31215438, 31228670, 31254535, 31258101, 31302462, 31306248
31306261, 31309867, 31315876, 31326608, 31331354, 31335037, 31335142
31343752, 31348018, 31393600, 31430722, 31544097, 31570161, 31600023
31658464, 31668061, 31668872, 31674731, 31718134, 31799775, 31867037
31886547, 31905033

Version 18.0.0.0.ru-2020-07.rur-2020-07.r1

Version 18.0.0.0.ru-2020-07.rur-2020-07.r1 includes the following:

- Patch 31308624: Database Release Update 18.11.0.0.200714
- Patch 31219909: OJVM RELEASE UPDATE: 18.11.0.0.200714
- Patch 31335037: DSTV35 for RDBMS (TZDATA2020A)
- Patch 31335142: DSTV35 for OJVM (TZDATA2020A)
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 29374604: IE not starting against 18c Oracle RDBMS Standard Edition
- PreUpgrade Jar: preupgrade_181_cbuild_10_lf.zip

Combined patches for version 18.0.0.0.ru-2020-07.rur-2020-07.r1, released July 2020

Bugs fixed:

7391838, 8932139, 9062315, 12816839, 13554903, 14221306, 18701017
18986501, 20436508, 20549013, 20917487, 21095391, 21223848, 21374587
21547051, 21744603, 21766220, 21806121, 21935698, 22174392, 22282748
22363790, 22645496, 22729345, 22734786, 23003564, 23061453, 23109325
23310101, 23698980, 23761724, 23840305, 24489904, 24596874, 24669730
24687075, 24689376, 24737581, 24798481, 24841671, 24844841, 24903291
24925863, 24971597, 25031502, 25035594, 25035599, 25060506, 25092651
25210690, 25287072, 25293659, 25303284, 25309116, 25348956, 25405687

25416731, 25487146, 25560538, 25573623, 25576115, 25591292, 25607397
25634405, 25644811, 25663488, 25686739, 25696520, 25709124, 25724089
25726981, 25736428, 25740844, 25743479, 25756945, 25809524, 25824236
25882883, 25890002, 25908728, 25911069, 25929650, 25943740, 25958554
25986062, 25997810, 26083489, 26115103, 26164661, 26226953, 26237338
26281476, 26284722, 26297826, 26299684, 26313403, 26336101, 26362155
26375330, 26399691, 26405036, 26410240, 26422277, 26423085, 26427905
26433972, 26440142, 26440169, 26441345, 26450454, 26476244, 26521043
26527054, 26536320, 26586174, 26587652, 26595088, 26598422, 26615291
26617804, 26646549, 26647619, 26654411, 26694735, 26724511, 26731697
26745002, 26785169, 26790514, 26790923, 26792891, 26798411, 26818960
26822620, 26827699, 26843558, 26843664, 26846077, 26860285, 26871815
26882126, 26882316, 26883456, 26894737, 26895149, 26898279, 26910716
26914402, 26927998, 26928317, 26933599, 26943004, 26943660, 26956033
26960097, 26961415, 26966120, 26966916, 26970175, 26976568, 26985002
26986173, 26992964, 26996813, 27000158, 27005278, 27006120, 27006664
27012915, 27013566, 27016033, 27018734, 27026401, 27028251, 27030974
27032726, 27033520, 27034318, 27034688, 27035653, 27036163, 27036408
27037839, 27038986, 27040560, 27041253, 27044169, 27044575, 27047831
27053044, 27054231, 27058530, 27060167, 27060859, 27061736, 27066451
27066519, 27072923, 27073066, 27075854, 27080748, 27080874, 27086406
27086821, 27090765, 27092991, 27093423, 27098733, 27100800, 27101105
27101527, 27101652, 27105900, 27106301, 27106915, 27110878, 27111780
27112686, 27114112, 27115422, 27119621, 27119861, 27121566, 27122162
27125872, 27126666, 27128580, 27133637, 27135647, 27142120, 27142529
27143756, 27143882, 27144533, 27144928, 27147979, 27150500, 27151826
27152892, 27153641, 27153755, 27155549, 27156355, 27160360, 27160922
27163928, 27164352, 27165231, 27166354, 27166715, 27169796, 27169888
27170305, 27174938, 27174948, 27177551, 27177852, 27179264, 27181521
27181537, 27181897, 27182006, 27182064, 27184253, 27185188, 27187440
27189611, 27190851, 27193810, 27195935, 27197334, 27199245, 27200959
27202015, 27203055, 27204476, 27207634, 27208795, 27208953, 27210038
27210263, 27210872, 27212837, 27213140, 27214085, 27214204, 27215007
27216046, 27216224, 27217412, 27220610, 27220937, 27221900, 27222121
27222423, 27222626, 27222938, 27223075, 27224987, 27226913, 27228786
27229389, 27231051, 27232983, 27233563, 27234962, 27236052, 27236110
27236722, 27238077, 27238258, 27240246, 27240570, 27241221, 27241247
27242226, 27242616, 27244337, 27244785, 27244999, 27249215, 27249531
27249544, 27250547, 27251690, 27252023, 27254335, 27254851, 27255377
27256000, 27256488, 27256534, 27256584, 27257509, 27258578, 27259307
27259386, 27259983, 27262601, 27262650, 27262798, 27262945, 27262991
27263276, 27263677, 27263996, 27264464, 27265816, 27266245, 27267992
27270197, 27271876, 27274143, 27274456, 27274536, 27275136, 27275533
27275776, 27276231, 27282707, 27283029, 27283960, 27284375, 27284499
27285244, 27285557, 27288230, 27288638, 27288894, 27292213, 27293599
27294480, 27299455, 27300007, 27301308, 27301568, 27302415, 27302594
27302632, 27302681, 27302695, 27302711, 27302714, 27302730, 27302777
27302800, 27302960, 27303287, 27303785, 27303938, 27304131, 27304410
27304906, 27304936, 27305318, 27307868, 27308088, 27309182, 27310092
27313687, 27314206, 27314390, 27314512, 27314697, 27315159, 27318117
27318869, 27320576, 27320985, 27321179, 27321834, 27326204, 27329812
27330158, 27330161, 27333658, 27333664, 27333693, 27333731, 27334316
27334353, 27334648, 27335682, 27338838, 27338912, 27338946, 27339115
27339396, 27339483, 27339495, 27341036, 27343844, 27345190, 27345231
27345450, 27345498, 27346329, 27346644, 27346709, 27346949, 27346984
27347126, 27348081, 27348707, 27349393, 27350267, 27351628, 27352600
27354783, 27356373, 27357773, 27358232, 27358241, 27359178, 27359368
27360126, 27362190, 27364854, 27364891, 27364916, 27364947, 27365014
27365139, 27365702, 27365993, 27367194, 27368850, 27369515, 27370933
27372756, 27375260, 27375542, 27376871, 27377219, 27378103, 27378959
27379233, 27379846, 27379956, 27381383, 27381417, 27381498, 27381656
27383281, 27384222, 27386467, 27389352, 27392187, 27392968, 27393421
27393570, 27394086, 27395404, 27395416, 27395794, 27396357, 27396365
27396377, 27396624, 27396666, 27396672, 27396720, 27396794, 27396813
27397048, 27398080, 27398223, 27398660, 27399499, 27399762, 27399985
27400416, 27400598, 27401618, 27401637, 27403244, 27404573, 27404599

27404668, 27405242, 27405645, 27405696, 27406105, 27410279, 27410300
27410595, 27412805, 27416327, 27416997, 27417186, 27420715, 27421101
27421733, 27422874, 27423251, 27424405, 27425507, 27425622, 27426277
27426363, 27427805, 27428790, 27430219, 27430254, 27430802, 27432062
27432338, 27432355, 27432826, 27433163, 27433385, 27433870, 27434050
27434193, 27434486, 27434974, 27435537, 27439835, 27441326, 27441980
27442041, 27444727, 27445330, 27445462, 27445727, 27447452, 27447687
27448162, 27449814, 27450355, 27450400, 27450783, 27451049, 27451182
27451187, 27451531, 27452046, 27452760, 27452897, 27453225, 27454722
27457666, 27457891, 27458164, 27458829, 27459593, 27459909, 27459948
27460675, 27461740, 27462994, 27465480, 27466597, 27467543, 27468303
27469245, 27469329, 27471876, 27472969, 27473800, 27475272, 27479358
27480784, 27481406, 27481765, 27483974, 27484556, 27486253, 27486805
27487309, 27487795, 27487919, 27489107, 27489719, 27492916, 27493674
27494663, 27496224, 27496308, 27496424, 27496806, 27497950, 27498477
27501327, 27501413, 27501465, 27502420, 27503318, 27503413, 27504190
27504770, 27505229, 27505603, 27506774, 27507968, 27508936, 27508984
27508985, 27510959, 27511196, 27512439, 27513114, 27517818, 27518227
27518310, 27519708, 27520070, 27520900, 27522245, 27523368, 27523800
27525909, 27526362, 27526744, 27528204, 27529661, 27532009, 27532375
27533780, 27533819, 27534289, 27534509, 27537472, 27539757, 27539876
27540613, 27541286, 27541468, 27542824, 27544030, 27544973, 27545630
27547732, 27550341, 27551855, 27554074, 27555481, 27558557, 27558559
27558861, 27560562, 27560602, 27560702, 27560735, 27562488, 27563629
27563767, 27565906, 27567477, 27570318, 27573154, 27573408, 27574335
27576342, 27576354, 27577122, 27577758, 27578007, 27579353, 27579969
27580996, 27581484, 27585755, 27585800, 27586810, 27586895, 27587672
27587905, 27588271, 27589260, 27591842, 27592466, 27593389, 27593501
27593585, 27593587, 27595096, 27595801, 27595973, 27599689, 27599927
27600706, 27601118, 27601441, 27602091, 27602488, 27603841, 27604293
27605482, 27607563, 27607805, 27608669, 27609819, 27610269, 27613080
27613247, 27613530, 27613554, 27615608, 27615649, 27616657, 27617522
27617978, 27620808, 27623159, 27623844, 27625010, 27625050, 27625274
27625620, 27627992, 27629756, 27629928, 27631506, 27632114, 27634676
27634991, 27635508, 27636900, 27642235, 27644757, 27645231, 27645940
27649707, 27652302, 27654039, 27654521, 27655217, 27657467, 27657712
27657920, 27658186, 27658205, 27662528, 27663370, 27664702, 27666312
27668379, 27671633, 27679488, 27679664, 27679793, 27679806, 27679961
27680162, 27680509, 27680669, 27682151, 27682288, 27686599, 27688036
27688099, 27688692, 27690513, 27690578, 27691717, 27691809, 27691920
27691939, 27692215, 27693416, 27693713, 27694261, 27695063, 27697092
27698953, 27700466, 27701795, 27702244, 27703242, 27704237, 27705761
27707544, 27708711, 27709046, 27710072, 27714373, 27718914, 27719187
27723002, 27723151, 27725967, 27726269, 27726780, 27729678, 27731346
27732323, 27733415, 27734470, 27735534, 27739006, 27739957, 27740424
27740844, 27740854, 27744211, 27745220, 27747407, 27747869, 27748321
27748954, 27751006, 27751755, 27753336, 27756900, 27757567, 27757794
27757888, 27757979, 27758544, 27758653, 27758972, 27759077, 27759457
27761402, 27766324, 27766679, 27767081, 27768034, 27769361, 27772093
27772815, 27773602, 27774320, 27774539, 27778433, 27779886, 27780562
27780683, 27782339, 27782464, 27783059, 27783289, 27786669, 27786699
27786772, 27791223, 27793533, 27797290, 27801337, 27801774, 27803665
27807441, 27810967, 27811439, 27812560, 27812593, 27813267, 27815347
27818389, 27818871, 27819881, 27824540, 27824543, 27825241, 27828794
27828892, 27829295, 27832643, 27833369, 27833672, 27834551, 27834569
27834984, 27835925, 27839353, 27839616, 27839732, 27840386, 27843646
27846298, 27846499, 27847259, 27849825, 27850112, 27850736, 27851757
27856471, 27861226, 27861452, 27861909, 27862636, 27864737, 27865439
27869075, 27869339, 27873412, 27873643, 27876671, 27882176, 27886087
27889841, 27892488, 27896388, 27896443, 27896458, 27897639, 27897759
27898015, 27900663, 27902561, 27906509, 27908396, 27908644, 27909478
27912301, 27917669, 27918832, 27920184, 27924147, 27926113, 27927431
27929287, 27929509, 27930478, 27931299, 27931506, 27934468, 27935348
27935464, 27935493, 27935826, 27936676, 27938736, 27940876, 27941110
27941514, 27941896, 27945870, 27948050, 27948153, 27950708, 27952586
27952762, 27957892, 27959594, 27960021, 27961746, 27964051, 27964513

27965400, 27965830, 27966472, 27967484, 27970265, 27971503, 27971575
27972265, 27975778, 27977039, 27978668, 27983174, 27984028, 27984314
27986817, 27989556, 27989849, 27991970, 27993289, 27993298, 27994325
27994333, 27995215, 27995248, 27997875, 27998003, 27999073, 27999597
27999638, 28000269, 28004853, 28006704, 28018962, 28019283, 28019592
28021205, 28022101, 28022847, 28023081, 28023399, 28023410, 28023482
28024347, 28024793, 28025398, 28025414, 28026866, 28032758, 28033429
28036487, 28039471, 28039953, 28043157, 28045209, 28045903, 28057267
28058612, 28059199, 28067846, 28071549, 28072130, 28072383, 28072464
28072567, 28073470, 28074713, 28079127, 28085865, 28088762, 28089440
28090453, 28091981, 28092783, 28098040, 28098160, 28098865, 28099592
28103600, 28103869, 28104176, 28104361, 28104409, 28106402, 28108003
28108898, 28109326, 28109698, 28111583, 28120036, 28120951, 28124631
28125601, 28125947, 28129791, 28131767, 28132287, 28135648, 28157786
28164480, 28165439, 28165545, 28169711, 28174827, 28174926, 28174951
28175445, 28180464, 28181021, 28182503, 28184554, 28184800, 28187706
28188330, 28189466, 28194173, 28199085, 28201419, 28204262, 28204423
28204443, 28209341, 28209985, 28210192, 28211734, 28214943, 28215510
28218832, 28220398, 28223871, 28226179, 28227512, 28229360, 28236305
28238264, 28240153, 28242712, 28250929, 28256164, 28258608, 28264172
28271107, 28271119, 28271693, 28276054, 28278547, 28278640, 28279837
28281094, 28282606, 28285766, 28290434, 28294563, 28302049, 28304709
28305001, 28305362, 28305607, 28309182, 28309406, 28312508, 28315031
28315995, 28319114, 28319623, 28320117, 28320399, 28321446, 28323201
28328895, 28329450, 28330714, 28330971, 28333072, 28338399, 28338999
28344964, 28350595, 28354603, 28357401, 28361083, 28361221, 28361787
28365111, 28369092, 28371123, 28373960, 28375383, 28378446, 28379065
28384353, 28385102, 28386259, 28388910, 28390273, 28391210, 28391582
28392168, 28392251, 28393678, 28396445, 28397317, 28401116, 28402823
28403295, 28413955, 28420042, 28420457, 28423598, 28432129, 28434028
28435825, 28437849, 28439086, 28445741, 28448314, 28454215, 28455212
28468312, 28468493, 28475164, 28478676, 28481149, 28481679, 28483184
28489150, 28492362, 28493478, 28498976, 28501075, 28502098, 28502403
28502773, 28503038, 28503484, 28504545, 28507324, 28508053, 28508296
28508557, 28512336, 28512761, 28513333, 28514693, 28521330, 28527416
28528349, 28530171, 28535127, 28535272, 28538439, 28542455, 28544633
28545134, 28545687, 28546290, 28547068, 28547478, 28553468, 28558645
28564479, 28565296, 28571483, 28572407, 28572834, 28578164, 28578945
28580528, 28584193, 28584217, 28584444, 28585411, 28587723, 28589509
28600233, 28601874, 28606598, 28608211, 28611037, 28612674, 28614072
28617631, 28617959, 28621470, 28622202, 28627255, 28627686, 28632559
28636676, 28639299, 28642273, 28642899, 28644549, 28646200, 28670445
28673203, 28678804, 28679454, 28680029, 28685371, 28689483, 28692103
28692275, 28695694, 28697806, 28702188, 28703812, 28708023, 28709063
28710469, 28710827, 28713840, 28714058, 28714988, 28715655, 28728040
28728272, 28730076, 28730253, 28734355, 28740708, 28742555, 28745367
28747182, 28749289, 28752599, 28755011, 28757758, 28758090, 28758722
28761812, 28767240, 28770146, 28774416, 28776431, 28776811, 28777174
28777214, 28777332, 28781754, 28785022, 28785531, 28791725, 28793062
28797711, 28803345, 28805612, 28805695, 28808314, 28809909, 28817449
28819640, 28820669, 28821847, 28827682, 28830691, 28831971, 28835937
28836716, 28838066, 28844866, 28847136, 28849751, 28852325, 28852691
28855922, 28856060, 28856172, 28863263, 28863487, 28865569, 28867992
28876639, 28878525, 28881723, 28887305, 28887509, 28889730, 28891984
28900506, 28905390, 28905457, 28910498, 28915870, 28919145, 28925880
28927452, 28938924, 28940179, 28949888, 28950868, 28951014, 28951382
28956908, 28959493, 28960211, 28965084, 28965095, 28986231, 28986257
28987454, 28993295, 28993353, 28993590, 28994890, 29000190, 29002488
29006527, 29007321, 29007353, 29009513, 29013832, 29015118, 29015706
29024054, 29026582, 29027694, 29027940, 29032276, 29033896, 29036278
29037290, 29039510, 29040739, 29044954, 29048498, 29048728, 29050886
29051702, 29055644, 29056270, 29056767, 29060216, 29061016, 29115857
29123482, 29125374, 29136111, 29139070, 29139591, 29158680, 29163567
29165682, 29170232, 29171683, 29173817, 29177886, 29179097, 29182517
29182901, 29189889, 29190663, 29198092, 29200700, 29202461, 29203604
29205918, 29212433, 29213320, 29213351, 29213893, 29224605, 29225076

29230252, 29230565, 29233415, 29237575, 29241345, 29242017, 29247712
29247906, 29249289, 29250317, 29254623, 29255273, 29260956, 29261548
29278684, 29285503, 29296257, 29301463, 29307638, 29311927, 29312672
29312889, 29314539, 29331209, 29331493, 29332763, 29339155, 29343086
29343156, 29343861, 29344541, 29346057, 29347981, 29350868, 29351662
29351771, 29353821, 29356752, 29361472, 29362596, 29364171, 29366406
29372069, 29372460, 29374604, 29375355, 29375984, 29376346, 29378913
29379978, 29382784, 29383695, 29386635, 29388020, 29388952, 29391849
29394749, 29395657, 29398488, 29399046, 29399336, 29404483, 29405462
29407804, 29408853, 29409149, 29409455, 29412269, 29417719, 29418165
29420254, 29426241, 29428230, 29429264, 29430524, 29434301, 29436454
29437712, 29439522, 29442936, 29445548, 29448498, 29450812, 29452251
29454978, 29457978, 29464779, 29465177, 29483626, 29483672, 29483723
29483771, 29489436, 29493122, 29500257, 29500963, 29501218, 29504682
29506942, 29511611, 29515766, 29521862, 29524599, 29524985, 29525886
29530515, 29531541, 29536342, 29538631, 29541742, 29542449, 29542580
29548413, 29548592, 29549071, 29557261, 29558238, 29559395, 29564592
29579919, 29580394, 29591343, 29604257, 29607136, 29608023, 29614098
29614987, 29616244, 29625065, 29626154, 29629430, 29629745, 29632265
29633753, 29637526, 29637560, 29643721, 29645167, 29645349, 29651520
29656843, 29667994, 29668005, 29676089, 29685137, 29687220, 29687459
29688867, 29703195, 29705793, 29707896, 29717901, 29719146, 29720133
29722167, 29724041, 29726695, 29739576, 29766435, 29773197, 29774362
29780140, 29782211, 29789911, 29791152, 29794462, 29807964, 29809792
29813494, 29815341, 29817278, 29822714, 29825525, 29827852, 29841687
29846645, 29853485, 29865188, 29869404, 29869906, 29875459, 29876358
29881050, 29881575, 29884958, 29891916, 29893132, 29896510, 29902299
29914449, 29922225, 29930457, 29944035, 29944660, 29951620, 29951759
29961353, 29962927, 29962939, 29965888, 29991257, 29997326, 29997937
30008125, 30018017, 30018903, 30031027, 30034456, 30039959, 30064268
30068871, 30073744, 30074349, 30076253, 30078934, 30085980, 30088912
30092280, 30098251, 30099302, 30114477, 30116203, 30120608, 30125944
30128047, 30131286, 30139392, 30147928, 30149035, 30160625, 30163243
30164714, 30173113, 30177597, 30179644, 30186706, 30189023, 30193165
30193736, 30194710, 30196358, 30200680, 30200758, 30215130, 30218044
30218317, 30223712, 30225443, 30232638, 30239480, 30240547, 30241567
30246179, 30247305, 30252098, 30252156, 30253255, 30259008, 30265523
30272329, 30282501, 30283932, 30293345, 30305880, 30312094, 30312559
30316897, 30325407, 30342878, 30352623, 30355490, 30357897, 30364613
30365745, 30368482, 30368668, 30372081, 30374739, 30376986, 30381207
30384121, 30384152, 30391272, 30397100, 30402386, 30403763, 30408515
30409339, 30412188, 30413137, 30416034, 30431274, 30441687, 30443393
30450787, 30453442, 30458593, 30460922, 30464655, 30474774, 30475115
30476768, 30485255, 30496957, 30497057, 30501574, 30503943, 30509277
30510527, 30522998, 30528547, 30528704, 30533172, 30534662, 30544595
30581448, 30582500, 30599407, 30602230, 30613937, 30623138, 30624864
30635302, 30652853, 30654454, 30668407, 30671813, 30681462, 30691604
30698289, 30741263, 30758943, 30773164, 30783551, 30803210, 30814285
30815852, 30816938, 30829779, 30855101, 30881588, 30887501, 30904672
30922819, 30957739, 30964194, 30968737, 30987088, 30998759, 31001455
31004719, 31013127, 31019767, 31022858, 31029936, 31100172, 31106577
31156383, 31172207, 31182793, 31200845, 31306261, 31335037, 31335142
31393600

Version 18.0.0.0.ru-2020-04.rur-2020-04.r1

Version 18.0.0.0.ru-2020-04.rur-2020-04.r1 includes the following:

- Patch 30872794: Database Release Update 18.10.0.0.200414
- Patch 30805598: Oracle JVM Release Update 18.10.0.0.200414
- Patch 29997937: DSTv34 for RDBMS (TZDATA2019B)
- Patch 29997959: DSTv34 for OJVM (TZDATA2019G)

- PreUpgrade Jar: preupgrade_181_cbuild_9_lf.zip
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR is included in DB PATCH 30138470
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- Support for [Purging the recycle bin \(p. 1062\)](#)
- Support for [Generating performance reports with Automatic Workload Repository \(AWR\) \(p. 1053\)](#) using the rdsadmin.rdsadmin_diagnostic_util package

Combined patches for version 18.0.0.0.ru-2020-04.rur-2020-04.r1, released April 2020

Bugs fixed:

8932139, 9062315, 13554903, 14221306, 20436508, 20549013, 21095391
21223848, 21374587, 21547051, 21744603, 21766220, 21806121, 21935698
22174392, 22282748, 22363790, 22645496, 23003564, 23061453, 23109325
23310101, 23761724, 24489904, 24687075, 24689376, 24737581, 24841671
24844841, 24903291, 24925863, 24971597, 25031502, 25035594, 25035599
25060506, 25210690, 25287072, 25293659, 25303284, 25309116, 25348956
25405687, 25416731, 25487146, 25573623, 25591292, 25607397, 25634405
25644811, 25663488, 25686739, 25696520, 25726981, 25736428, 25740844
25743479, 25756945, 25809524, 25824236, 25882883, 25890002, 25908728
25911069, 25929650, 25943740, 25958554, 25986062, 25997810, 26083489
26115103, 26164661, 26226953, 26237338, 26281476, 26284722, 26297826
26336101, 26362155, 26399691, 26405036, 26410240, 26422277, 26423085
26427905, 26440169, 26450454, 26476244, 26521043, 26536320, 26595088
26598422, 26615291, 26617804, 26646549, 26654411, 26694735, 26724511
26731697, 26745002, 26785169, 26790514, 26790923, 26792891, 26798411
26818960, 26822620, 26843558, 26843664, 26846077, 26871815, 26883456
26894737, 26895149, 26898279, 26910716, 26927998, 26928317, 26933599
26943004, 26956033, 26960097, 26961415, 26966120, 26966916, 26970175
26976568, 26985002, 26986173, 26992964, 27000158, 27005278, 27006120
27006664, 27013566, 27016033, 27026401, 27028251, 27030974, 27033520
27034688, 27035653, 27036163, 27036408, 27037839, 27038986, 27041253
27044169, 27044575, 27047831, 27053044, 27054231, 27058530, 27060167
27060859, 27061736, 27066451, 27066519, 27072923, 27073066, 27075854
27080874, 27086821, 27090765, 27093423, 27100800, 27101105, 27101527
27101652, 27105900, 27106301, 27110878, 27111780, 27112686, 27115422
27119621, 27119861, 27122162, 27125872, 27126666, 27128580, 27135647
27142120, 27142529, 27143756, 27143882, 27144928, 27147979, 27150500
27151826, 27152892, 27153641, 27155549, 27156355, 27160360, 27160922
27163928, 27164352, 27165231, 27166354, 27169796, 27169888, 27170305
27179264, 27181521, 27181537, 27181897, 27185188, 27187440, 27189611
27190851, 27193810, 27195935, 27197334, 27199245, 27200959, 27202015
27203055, 27208795, 27208953, 27210038, 27210263, 27210872, 27214085
27214204, 27215007, 27216046, 27216224, 27217412, 27220937, 27221900
27222121, 27222626, 27223075, 27224987, 27226913, 27228786, 27229389
27231051, 27232983, 27233563, 27234962, 27236052, 27236110, 27236722
27240246, 27240570, 27241221, 27241247, 27242226, 27242616, 27244337
27244785, 27244999, 27249215, 27249531, 27250547, 27251690, 27254851
27255377, 27256000, 27256488, 27256534, 27256584, 27258578, 27259307
27259386, 27259983, 27262601, 27262650, 27262798, 27262945, 27262991
27263276, 27263996, 27264464, 27266245, 27270197, 27274456, 27274536
27275136, 27275533, 27275776, 27276231, 27282707, 27283029, 27283960
27284375, 27284499, 27285244, 27288230, 27288638, 27288894, 27292213
27293599, 27294480, 27301308, 27301568, 27302594, 27302632, 27302681
27302695, 27302711, 27302714, 27302730, 27302777, 27302800, 27302960
27303287, 27303785, 27303938, 27304410, 27304906, 27304936, 27305318
27307868, 27308088, 27310092, 27313687, 27314206, 27314390, 27314697

27318117, 27318869, 27320576, 27321179, 27321834, 27326204, 27329812
27330158, 27330161, 27333658, 27333664, 27333693, 27333731, 27334316
27334648, 27335682, 27338912, 27338946, 27339115, 27339396, 27339483
27339495, 27341036, 27343844, 27345190, 27345231, 27345450, 27345498
27346329, 27346644, 27346709, 27346949, 27347126, 27348081, 27348707
27349393, 27350267, 27351628, 27352600, 27354783, 27356373, 27357773
27358241, 27359178, 27359368, 27360126, 27364854, 27364891, 27364916
27364947, 27365139, 27365702, 27365993, 27367194, 27368850, 27369515
27372756, 27375260, 27375542, 27376871, 27378103, 27379233, 27381383
27381417, 27381498, 27381656, 27383281, 27384222, 27386467, 27389352
27392187, 27393570, 27394086, 27395404, 27395416, 27395794, 27396357
27396365, 27396377, 27396624, 27396666, 27396672, 27396720, 27396794
27396813, 27397048, 27398080, 27398660, 27400416, 27400598, 27401637
27404573, 27404668, 27405242, 27405645, 27405696, 27406105, 27410279
27410300, 27410595, 27412805, 27416327, 27416997, 27417186, 27420715
27421101, 27421733, 27422874, 27423251, 27424405, 27425507, 27425622
27426363, 27427805, 27430802, 27432062, 27432338, 27432355, 27432826
27433385, 27433870, 27434050, 27434193, 27434486, 27434974, 27435537
27439835, 27441326, 27441980, 27442041, 27444727, 27445330, 27445462
27445727, 27447452, 27447687, 27448162, 27449814, 27450355, 27450400
27450783, 27451049, 27451182, 27451187, 27451531, 27452046, 27452760
27453225, 27454722, 27457666, 27457891, 27458164, 27459593, 27459909
27459948, 27460675, 27462994, 27466597, 27467543, 27468303, 27469245
27469329, 27471876, 27472969, 27473800, 27479358, 27480784, 27483974
27484556, 27486253, 27486805, 27487309, 27487795, 27487919, 27489107
27489719, 27493674, 27494663, 27496224, 27496308, 27496424, 27497950
27498477, 27501327, 27501413, 27501465, 27502420, 27504190, 27504770
27505229, 27505603, 27506774, 27507968, 27508985, 27510959, 27511196
27512439, 27517818, 27518227, 27518310, 27520070, 27520900, 27522245
27523368, 27523800, 27525909, 27526744, 27529661, 27532375, 27533780
27533819, 27534509, 27537472, 27539757, 27540613, 27541286, 27541468
27542824, 27544030, 27544973, 27545630, 27547732, 27550341, 27551855
27554074, 27555481, 27558557, 27558559, 27558861, 27560602, 27560702
27562488, 27563629, 27563767, 27565906, 27567477, 27570318, 27576342
27576354, 27577758, 27578007, 27579353, 27580996, 27585755, 27585800
27586810, 27586895, 27587672, 27587905, 27588271, 27591842, 27592466
27593389, 27593501, 27593585, 27595096, 27595973, 27599689, 27599927
27601118, 27601441, 27602091, 27602488, 27603841, 27604293, 27607563
27607805, 27608669, 27610269, 27613080, 27613247, 27613530, 27615608
27616657, 27617522, 27617978, 27620808, 27623159, 27623844, 27625274
27625620, 27629756, 27631506, 27632114, 27634676, 27634991, 27635508
27644757, 27645231, 27645940, 27649707, 27652302, 27654521, 27655217
27657712, 27658186, 27658205, 27662528, 27663370, 27664702, 27666312
27671633, 27679488, 27679664, 27679793, 27679806, 27679961, 27680162
27680509, 27680669, 27682151, 27686599, 27688036, 27688099, 27688692
27690513, 27690578, 27691809, 27691920, 27691939, 27692215, 27693416
27693713, 27694261, 27695063, 27697092, 27698953, 27700466, 27701795
27704237, 27705761, 27707544, 27709046, 27710072, 27718914, 27719187
27723002, 27723151, 27726269, 27726780, 27729678, 27732323, 27733415
27739006, 27740424, 27740844, 27744211, 27745220, 27747869, 27748954
27751006, 27751755, 27753336, 27756900, 27757567, 27757794, 27757888
27758544, 27758653, 27758972, 27759077, 27759457, 27761402, 27766324
27767081, 27769361, 27772093, 27772815, 27773602, 27774320, 27774539
27779886, 27780562, 27780683, 27782339, 27783289, 27786772, 27791223
27793533, 27797290, 27801337, 27803665, 27807441, 27810967, 27812560
27812593, 27813267, 27815347, 27818389, 27818871, 27819881, 27824540
27824543, 27825241, 27828794, 27829295, 27832643, 27833369, 27833672
27834551, 27834569, 27834984, 27835925, 27839353, 27840386, 27843646
27846298, 27846499, 27847259, 27849825, 27850112, 27851757, 27856471
27861226, 27861452, 27861909, 27869075, 27869339, 27873412, 27873643
27876671, 27882176, 27886087, 27892488, 27896443, 27896458, 27897759
27898015, 27900663, 27902561, 27908644, 27909478, 27912301, 27917669
27918832, 27920184, 27924147, 27926113, 27927431, 27929287, 27930478
27931299, 27934468, 27935348, 27935464, 27935493, 27938736, 27940876
27941110, 27941896, 27945870, 27948050, 27948153, 27950708, 27952762
27959594, 27960021, 27961746, 27964051, 27965400, 27965830, 27966472

27967484, 27970265, 27971503, 27971575, 27972265, 27975778, 27977039
27983174, 27984028, 27986817, 27989556, 27989849, 27991970, 27993289
27994325, 27994333, 27995215, 27995248, 27997875, 27998003, 27999073
27999597, 27999638, 28000269, 28004853, 28006704, 28018962, 28019283
28021205, 28022101, 28022847, 28023081, 28023399, 28023482, 28024347
28024793, 28025414, 28026866, 28033429, 28036487, 28043157, 28045903
28057267, 28058612, 28059199, 28067846, 28071549, 28072130, 28072383
28072464, 28072567, 28074713, 28079127, 28085865, 28088762, 28089440
28090453, 28091981, 28098160, 28098865, 28103600, 28103869, 28104361
28104409, 28106402, 28108003, 28108898, 28109326, 28111583, 28120036
28120951, 28124631, 28125947, 28129791, 28132287, 28135648, 28157786
28164480, 28165439, 28165545, 28169711, 28174827, 28174951, 28175445
28180464, 28181021, 28184554, 28184800, 28187706, 28188330, 28189466
28194173, 28199085, 28201419, 28204262, 28204443, 28209341, 28209985
28210192, 28211734, 28214943, 28215510, 28218832, 28220398, 28223871
28226179, 28227512, 28229360, 28236305, 28238264, 28242712, 28250929
28256164, 28258608, 28264172, 28271107, 28271119, 28271693, 28276054
28279837, 28281094, 28282606, 28285766, 28290434, 28294563, 28302049
28304709, 28305001, 28305362, 28309182, 28309406, 28312508, 28315031
28315995, 28319114, 28319623, 28320117, 28320399, 28321446, 28323201
28328895, 28329450, 28330714, 28330971, 28333072, 28338399, 28338999
28344964, 28350595, 28354603, 28357401, 28361083, 28361221, 28361787
28365111, 28369092, 28371123, 28373960, 28375383, 28378446, 28379065
28384353, 28385102, 28386259, 28388910, 28390273, 28391210, 28391582
28392168, 28392251, 28393678, 28396445, 28397317, 28401116, 28402823
28403295, 28413955, 28420042, 28420457, 28423598, 28432129, 28434028
28435825, 28437849, 28439086, 28445741, 28448314, 28454215, 28455212
28468312, 28468493, 28475164, 28478676, 28481149, 28481679, 28483184
28489150, 28492362, 28493478, 28498976, 28501075, 28502403, 28502773
28503038, 28503484, 28504545, 28507324, 28508053, 28508296, 28508557
28512336, 28512761, 28513333, 28514693, 28521330, 28527416, 28528349
28530171, 28535127, 28535272, 28538439, 28542455, 28544633, 28545134
28545687, 28546290, 28547068, 28553468, 28558645, 28564479, 28571483
28572407, 28572834, 28578164, 28578945, 28580528, 28584193, 28584217
28584444, 28585411, 28587723, 28589509, 28600233, 28608211, 28611037
28612674, 28614072, 28617631, 28617959, 28621470, 28622202, 28627255
28627686, 28632559, 28636676, 28639299, 28644549, 28646200, 28670445
28673203, 28678804, 28679454, 28680029, 28685371, 28692103, 28692275
28695694, 28697806, 28702188, 28708023, 28709063, 28710469, 28710827
28713840, 28714058, 28714988, 28728040, 28728272, 28730076, 28734355
28740708, 28742555, 28747182, 28749289, 28752599, 28755011, 28758090
28758722, 28761812, 28767240, 28770146, 28774416, 28776811, 28777174
28777214, 28777332, 28781754, 28785022, 28785531, 28791725, 28797711
28803345, 28805612, 28805695, 28809909, 28817449, 28819640, 28820669
28821847, 28830691, 28831971, 28835937, 28836716, 28838066, 28844866
28849751, 28852691, 28855922, 28856060, 28856172, 28863263, 28863487
28865569, 28867992, 28876639, 28878525, 28881723, 28887305, 28887509
28889730, 28891984, 28905457, 28919145, 28925880, 28927452, 28938924
28940179, 28949888, 28950868, 28951014, 28951382, 28956908, 28959493
28960211, 28965084, 28965095, 28986231, 28986257, 28987454, 28993295
28993353, 28993590, 29000190, 29002488, 29006527, 29007321, 29007353
29009513, 29013832, 29015118, 29015706, 29024054, 29026582, 29027694
29027940, 29032276, 29033896, 29037290, 29040739, 29048498, 29050886
29051702, 29056270, 29056767, 29060216, 29061016, 29115857, 29123482
29125374, 29136111, 29139591, 29158680, 29163567, 29170232, 29171683
29173817, 29177886, 29179097, 29182517, 29182901, 29189889, 29190663
29198092, 29200700, 29202461, 29203604, 29205918, 29212433, 29213320
29213351, 29224605, 29225076, 29230252, 29230565, 29233415, 29237575
29241345, 29242017, 29247712, 29247906, 29249289, 29250317, 29255273
29260956, 29261548, 29278684, 29285503, 29301463, 29312672, 29314539
29331209, 29331493, 29332763, 29339155, 29343086, 29343861, 29347981
29351662, 29351771, 29353821, 29356752, 29361472, 29364171, 29372460
29375355, 29375984, 29376346, 29378913, 29379978, 29382784, 29383695
29386635, 29388020, 29388952, 29391849, 29394749, 29395657, 29398488
29399046, 29399336, 29404483, 29405462, 29407804, 29408853, 29409149
29409455, 29412269, 29417719, 29426241, 29429264, 29430524, 29434301

29436454, 29437712, 29439522, 29448498, 29450812, 29452251, 29454978
29457978, 29464779, 29465177, 29483626, 29483672, 29483723, 29483771
29489436, 29493122, 29500257, 29500963, 29504682, 29511611, 29515766
29521862, 29524599, 29524985, 29525886, 29530515, 29531541, 29536342
29538631, 29541742, 29542449, 29542580, 29548413, 29548592, 29549071
29557261, 29558238, 29580394, 29591343, 29604257, 29607136, 29614098
29616244, 29625065, 29626154, 29629430, 29629745, 29632265, 29633753
29637526, 29643721, 29645349, 29651520, 29656843, 29667994, 29668005
29676089, 29685137, 29687220, 29687459, 29703195, 29707896, 29719146
29720133, 29724041, 29726695, 29739576, 29773197, 29780140, 29782211
29791152, 29794462, 29807964, 29813494, 29817278, 29825525, 29827852
29841687, 29846645, 29853485, 29865188, 29869404, 29875459, 29876358
29881050, 29884958, 29891916, 29893132, 29896510, 29902299, 29914449
29944035, 29944660, 29951620, 29961353, 29962927, 29962939, 29991257
29997326, 30018017, 30031027, 30034456, 30064268, 30073744, 30074349
30076253, 30078934, 30088912, 30098251, 30099302, 30114477, 30120608
30125944, 30128047, 30131286, 30147928, 30149035, 30163243, 30164714
30173113, 30177597, 30179644, 30189023, 30193165, 30194710, 30196358
30200758, 30215130, 30218044, 30218317, 30223712, 30232638, 30239480
30241567, 30246179, 30247305, 30252098, 30252156, 30253255, 30259008
30265523, 30272329, 30282501, 30283932, 30305880, 30312094, 30342878
30352623, 30355490, 30364613, 30365745, 30374739, 30384152, 30402386
30403763, 30408515, 30409339, 30412188, 30413137, 30416034, 30431274
30441687, 30453442, 30458593, 30460922, 30475115, 30485255, 30496957
30497057, 30501574, 30503943, 30509277, 30510527, 30582500, 30613937
30635302, 30654454, 30671813, 30741263, 30783551, 30803210, 30815852
30881588, 30533172, 30312559, 30085980, 29997959, 29997937, 28852325
28125601, 29213893, 28730253, 27304131, 27539876, 27952586, 27642235
27636900, 27461740, 28278547, 28278640, 27936676, 28502098, 28915870
28601874, 29445548, 29254623, 29774362, 30160625, 30534662, 26914402
30855101, 12816839, 18701017, 22734786, 23698980, 23840305, 25709124
25724089, 26299684, 26313403, 26433972, 26527054, 26586174, 26587652
26647619, 26827699, 26860285, 26882126, 26882316, 26943660, 26996813
27012915, 27018734, 27032726, 27034318, 27040560, 27080748, 27086406
27092991, 27098733, 27106915, 27114112, 27121566, 27133637, 27144533
27153755, 27166715, 27174938, 27174948, 27177551, 27177852, 27182006
27182064, 27184253, 27204476, 27212837, 27213140, 27220610, 27222423
27222938, 27238077, 27238258, 27249544, 27252023, 27257509, 27263677
27265816, 27267992, 27271876, 27274143, 27285557, 27299455, 27300007
27302415, 27309182, 27314512, 27315159, 27320985, 27334353, 27338838
27346984, 27358232, 27362190, 27370933, 27377219, 27378959, 27379846
27379956, 27393421, 27398223, 27399499, 27399762, 27399985, 27401618
27403244, 27404599, 27426277, 27428790, 27430219, 27430254, 27433163
27452897, 27458829, 27465480, 27475272, 27481406, 27481765, 27492916
27496806, 27503318, 27503413, 27508936, 27508984, 27513114, 27519708
27526362, 27528204, 27532009, 27534289, 27560562, 27560735, 27573154
27573408, 27574335, 27577122, 27579969, 27581484, 27593587, 27595801
27600706, 27609819, 27625010, 27625050, 27627992, 27654039, 27657467
27657920, 27668379, 27682288, 27691717, 27702244, 27703242, 27708711
27714373, 27725967, 27731346, 27734470, 27735534, 27739957, 27740854
27747407, 27748321, 27757979, 27766679, 27768034, 27778433, 27782464
27783059, 27786669, 27786699, 27801774, 27811439, 27839732, 27850736
27862636, 27864737, 27865439, 27889841, 27896388, 27897639, 27906509
27931506, 27935826, 27941514, 27957892, 27978668, 27984314, 27993298
28023410, 28025398, 28032758, 28039471, 28039953, 28045209, 28099592
28109698, 28174926, 28182503, 28204423, 28240153

Version 18.0.0.0.ru-2020-01.rur-2020-01.r1

Version 18.0.0.0.ru-2020-01.rur-2020-01.r1 includes the following:

- Patch 30480385: Database Release Update: 18.9.0.0.200114
- Patch 30501926: OJVM RELEASE UPDATE: 18.9.0.0.200114

- Patch 29997937: DSTv34 for RDBMS (TZDATA2019B)
- Patch 29997959: DSTv34 for OJVM (TZDATA2019G)
- PreUpgrade Jar: preupgrade_181_cbuild_9_lf.zip
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR is included in DB PATCH 30138470
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE

Oracle release update 18.9.0.0.200114, released January 2020

Bugs fixed:

```
8932139, 9062315, 13554903, 14221306, 20436508, 20549013, 21095391
21223848, 21374587, 21547051, 21744603, 21766220, 21806121, 21935698
22174392, 22282748, 22363790, 22645496, 23003564, 23061453, 23109325
23310101, 23761724, 24489904, 24687075, 24689376, 24737581, 24841671
24844841, 24903291, 24925863, 24971597, 25035594, 25035599, 25060506
25210690, 25287072, 25293659, 25303284, 25309116, 25348956, 25405687
25487146, 25573623, 25591292, 25607397, 25634405, 25644811, 25663488
25686739, 25696520, 25726981, 25736428, 25743479, 25756945, 25824236
25882883, 25890002, 25908728, 25911069, 25929650, 25943740, 25958554
25986062, 25997810, 26083489, 26115103, 26164661, 26226953, 26237338
26281476, 26284722, 26297826, 26336101, 26362155, 26399691, 26405036
26410240, 26422277, 26423085, 26427905, 26440169, 26450454, 26476244
26521043, 26536320, 26595088, 26598422, 26615291, 26617804, 26646549
26654411, 26694735, 26724511, 26731697, 26745002, 26785169, 26790514
26792891, 26798411, 26818960, 26822620, 26843558, 26843664, 26846077
26871815, 26883456, 26894737, 26895149, 26898279, 26910716, 26927998
26928317, 26933599, 26943004, 26956033, 26960097, 26961415, 26966120
26966916, 26970175, 26976568, 26985002, 26986173, 26992964, 27000158
27005278, 27006120, 27006664, 27016033, 27026401, 27028251, 27030974
27033520, 27034688, 27035653, 27036163, 27036408, 27037839, 27038986
27041253, 27044169, 27044575, 27047831, 27053044, 27054231, 27058530
27060167, 27060859, 27061736, 27066451, 27066519, 27072923, 27073066
27075854, 27080874, 27086821, 27090765, 27093423, 27101105, 27101527
27101652, 27105900, 27106301, 27110878, 27111780, 27112686, 27115422
27119621, 27119861, 27122162, 27125872, 27126666, 27128580, 27135647
27142529, 27143756, 27143882, 27144928, 27147979, 27150500, 27151826
27152892, 27153641, 27155549, 27156355, 27160360, 27160922, 27163928
27164352, 27165231, 27166354, 27169796, 27169888, 27170305, 27179264
27181521, 27181537, 27181897, 27185188, 27187440, 27189611, 27190851
27193810, 27195935, 27197334, 27199245, 27200959, 27202015, 27203055
27208795, 27208953, 27210038, 27210263, 27210872, 27214085, 27214204
27215007, 27216046, 27216224, 27220937, 27221900, 27222121, 27222626
27223075, 27224987, 27226913, 27228786, 27229389, 27231051, 27232983
27233563, 27234962, 27236052, 27236110, 27236722, 27240246, 27240570
27241221, 27241247, 27242226, 27242616, 27244337, 27244785, 27244999
27249215, 27249531, 27250547, 27251690, 27254851, 27255377, 27256000
27256488, 27256534, 27256584, 27258578, 27259307, 27259386, 27259983
27262601, 27262650, 27262798, 27262945, 27262991, 27263276, 27263996
27264464, 27266245, 27270197, 27274456, 27274536, 27275136, 27275533
27275776, 27276231, 27282707, 27283029, 27283960, 27284375, 27284499
27285244, 27288230, 27288638, 27288894, 27292213, 27293599, 27294480
27301308, 27301568, 27302594, 27302632, 27302681, 27302695, 27302711
27302714, 27302730, 27302777, 27302800, 27302960, 27303287, 27303785
27303938, 27304410, 27304906, 27304936, 27305318, 27307868, 27308088
27310092, 27313687, 27314206, 27314390, 27318117, 27318869, 27320576
27321179, 27321834, 27326204, 27329812, 27330158, 27330161, 27333658
27333664, 27333693, 27333731, 27334316, 27334648, 27335682, 27338912
27338946, 27339115, 27339396, 27339483, 27339495, 27341036, 27345190
```

27345231, 27345450, 27345498, 27346329, 27346644, 27346709, 27346949
27347126, 27348081, 27348707, 27349393, 27350267, 27351628, 27352600
27354783, 27356373, 27357773, 27358241, 27359178, 27359368, 27360126
27364854, 27364891, 27364916, 27364947, 27365139, 27365702, 27365993
27367194, 27368850, 27369515, 27372756, 27375260, 27375542, 27376871
27378103, 27379233, 27381383, 27381417, 27381498, 27381656, 27383281
27384222, 27386467, 27389352, 27392187, 27393570, 27394086, 27395404
27395416, 27395794, 27396357, 27396365, 27396377, 27396624, 27396666
27396672, 27396720, 27396813, 27397048, 27398080, 27398660, 27400416
27400598, 27401637, 27404573, 27404668, 27405242, 27405645, 27405696
27406105, 27410279, 27410300, 27410595, 27412805, 27416327, 27416997
27417186, 27420715, 27421101, 27421733, 27422874, 27423251, 27424405
27425507, 27425622, 27426363, 27427805, 27430802, 27432062, 27432338
27432355, 27432826, 27433385, 27433870, 27434050, 27434193, 27434486
27434974, 27435537, 27439835, 27441326, 27441980, 27442041, 27444727
27445330, 27445462, 27445727, 27447452, 27447687, 27448162, 27449814
27450355, 27450400, 27450783, 27451049, 27451182, 27451187, 27451531
27452046, 27452760, 27453225, 27454722, 27457666, 27457891, 27458164
27459909, 27459948, 27460675, 27462994, 27466597, 27467543, 27468303
27469245, 27469329, 27471876, 27472969, 27473800, 27479358, 27480784
27483974, 27484556, 27486253, 27486805, 27487309, 27487795, 27487919
27489107, 27489719, 27493674, 27496224, 27496308, 27496424, 27497950
27498477, 27501327, 27501413, 27501465, 27502420, 27504190, 27504770
27505229, 27505603, 27506774, 27507968, 27508985, 27510959, 27511196
27512439, 27517818, 27518227, 27518310, 27520070, 27520900, 27522245
27523368, 27523800, 27525909, 27526744, 27529661, 27532375, 27533780
27533819, 27534509, 27537472, 27539757, 27540613, 27541286, 27541468
27542824, 27544030, 27544973, 27545630, 27547732, 27550341, 27551855
27554074, 27555481, 27558557, 27558559, 27558861, 27560602, 27560702
27562488, 27563629, 27563767, 27565906, 27567477, 27570318, 27576342
27576354, 27577758, 27578007, 27579353, 27580996, 27585755, 27585800
27586810, 27586895, 27587672, 27587905, 27588271, 27591842, 27592466
27593389, 27593501, 27593585, 27595096, 27595973, 27599689, 27599927
27601118, 27601441, 27602091, 27602488, 27603841, 27604293, 27607563
27607805, 27608669, 27610269, 27613080, 27613247, 27613530, 27615608
27616657, 27617522, 27617978, 27620808, 27623159, 27623844, 27625274
27625620, 27629756, 27631506, 27632114, 27634676, 27634991, 27635508
27644757, 27645940, 27649707, 27652302, 27654521, 27655217, 27657712
27658186, 27658205, 27662528, 27663370, 27664702, 27666312, 27671633
27679488, 27679664, 27679793, 27679806, 27679961, 27680162, 27680509
27680669, 27682151, 27686599, 27688036, 27688099, 27688692, 27690513
27690578, 27691809, 27691920, 27691939, 27692215, 27693416, 27693713
27694261, 27695063, 27697092, 27698953, 27700466, 27701795, 27704237
27705761, 27707544, 27709046, 27710072, 27718914, 27719187, 27723002
27723151, 27726269, 27726780, 27729678, 27732323, 27733415, 27739006
27740424, 27740844, 27744211, 27745220, 27747869, 27748954, 27751006
27751755, 27753336, 27756900, 27757567, 27757794, 27757888, 27758544
27758653, 27758972, 27759077, 27759457, 27761402, 27766324, 27767081
27769361, 27772093, 27772815, 27773602, 27774320, 27774539, 27779886
27780562, 27780683, 27782339, 27783289, 27786772, 27791223, 27793533
27797290, 27801337, 27803665, 27807441, 27810967, 27812560, 27812593
27813267, 27815347, 27818389, 27818871, 27819881, 27824540, 27824543
27825241, 27828794, 27829295, 27832643, 27833369, 27833672, 27834551
27834569, 27834984, 27835925, 27839353, 27840386, 27843646, 27846298
27846499, 27847259, 27849825, 27850112, 27851757, 27856471, 27861226
27861452, 27861909, 27869075, 27869339, 27873412, 27873643, 27876671
27882176, 27886087, 27892488, 27896443, 27896458, 27897759, 27898015
27900663, 27902561, 27908644, 27909478, 27912301, 27917669, 27918832
27920184, 27924147, 27926113, 27927431, 27929287, 27930478, 27931299
27934468, 27935348, 27935493, 27938736, 27940876, 27941110, 27941896
27945870, 27948050, 27948153, 27950708, 27952762, 27959594, 27960021
27961746, 27964051, 27965400, 27965830, 27966472, 27967484, 27970265
27971503, 27971575, 27972265, 27975778, 27977039, 27983174, 27984028
27986817, 27989556, 27989849, 27991970, 27993289, 27994325, 27994333
27995215, 27995248, 27997875, 27998003, 27999073, 27999597, 27999638
28000269, 28004853, 28006704, 28018962, 28019283, 28021205, 28022101

28022847, 28023081, 28023399, 28023482, 28024347, 28026866, 28033429
28036487, 28043157, 28045903, 28057267, 28058612, 28059199, 28067846
28071549, 28072130, 28072464, 28072567, 28074713, 28079127, 28085865
28088762, 28089440, 28090453, 28091981, 28098160, 28098865, 28103600
28103869, 28104361, 28104409, 28106402, 28108003, 28108898, 28109326
28111583, 28120036, 28120951, 28124631, 28125947, 28129791, 28132287
28135648, 28164480, 28165439, 28165545, 28169711, 28174827, 28174951
28175445, 28180464, 28181021, 28184554, 28184800, 28187706, 28188330
28189466, 28194173, 28199085, 28201419, 28204262, 28204443, 28209341
28209985, 28210192, 28211734, 28214943, 28215510, 28218832, 28220398
28223871, 28226179, 28227512, 28229360, 28236305, 28238264, 28242712
28256164, 28258608, 28264172, 28271119, 28271693, 28279837, 28281094
28282606, 28285766, 28290434, 28294563, 28302049, 28304709, 28305001
28305362, 28309182, 28309406, 28312508, 28315031, 28315995, 28319114
28319623, 28320117, 28320399, 28321446, 28323201, 28328895, 28329450
28330714, 28330971, 28333072, 28338399, 28338999, 28344964, 28350595
28354603, 28357401, 28361083, 28361221, 28361787, 28365111, 28369092
28371123, 28373960, 28375383, 28378446, 28379065, 28384353, 28385102
28386259, 28388910, 28390273, 28391210, 28391582, 28392168, 28392251
28393678, 28396445, 28397317, 28401116, 28402823, 28403295, 28413955
28420042, 28420457, 28423598, 28432129, 28434028, 28435825, 28437849
28445741, 28448314, 28454215, 28455212, 28468312, 28468493, 28475164
28478676, 28481149, 28481679, 28483184, 28489150, 28492362, 28493478
28498976, 28501075, 28502403, 28502773, 28503484, 28504545, 28507324
28508053, 28508296, 28508557, 28512336, 28512761, 28513333, 28514693
28521330, 28527416, 28528349, 28530171, 28535272, 28538439, 28542455
28544633, 28545134, 28545687, 28546290, 28547068, 28553468, 28558645
28571483, 28572407, 28572834, 28578164, 28578945, 28580528, 28584193
28584217, 28584444, 28585411, 28587723, 28589509, 28600233, 28612674
28614072, 28617631, 28617959, 28621470, 28627255, 28627686, 28632559
28636676, 28639299, 28644549, 28646200, 28670445, 28673203, 28678804
28679454, 28680029, 28685371, 28692103, 28692275, 28695694, 28697806
28702188, 28708023, 28710827, 28713840, 28714058, 28714988, 28728040
28728272, 28730076, 28734355, 28742555, 28747182, 28749289, 28752599
28755011, 28758090, 28758722, 28761812, 28767240, 28770146, 28774416
28776811, 28777174, 28777214, 28777332, 28781754, 28785022, 28785531
28791725, 28797711, 28803345, 28805612, 28805695, 28809909, 28817449
28819640, 28820669, 28821847, 28830691, 28831971, 28836716, 28838066
28844866, 28849751, 28852691, 28855922, 28856060, 28856172, 28863487
28867992, 28876639, 28878525, 28881723, 28887305, 28887509, 28889730
28891984, 28905457, 28919145, 28925880, 28927452, 28938924, 28940179
28949888, 28951014, 28951382, 28956908, 28959493, 28960211, 28965084
28965095, 28986231, 28986257, 28987454, 28993353, 28993590, 29000190
29002488, 29006527, 29007321, 29007353, 29009513, 29013832, 29015118
29015706, 29024054, 29027694, 29027940, 29032276, 29033896, 29037290
29040739, 29050886, 29051702, 29056270, 29056767, 29060216, 29061016
29115857, 29123482, 29125374, 29136111, 29139591, 29158680, 29163567
29170232, 29171683, 29173817, 29177886, 29182901, 29189889, 29190663
29198092, 29200700, 29202461, 29203604, 29205918, 29212433, 29213320
29213351, 29224605, 29225076, 29230252, 29230565, 29233415, 29237575
29241345, 29242017, 29247906, 29250317, 29255273, 29260956, 29261548
29278684, 29285503, 29301463, 29312672, 29314539, 29331209, 29331493
29339155, 29347981, 29351662, 29353821, 29356752, 29364171, 29375355
29375984, 29376346, 29378913, 29379978, 29382784, 29383695, 29386635
29388020, 29388952, 29391849, 29394749, 29395657, 29398488, 29399336
29404483, 29405462, 29407804, 29408853, 29409149, 29412269, 29417719
29429264, 29430524, 29434301, 29436454, 29437712, 29439522, 29448498
29450812, 29452251, 29454978, 29457978, 29464779, 29465177, 29483626
29483672, 29483723, 29483771, 29489436, 29493122, 29500257, 29504682
29511611, 29515766, 29524599, 29525886, 29530515, 29531541, 29536342
29538631, 29541742, 29542449, 29548413, 29549071, 29557261, 29558238
29604257, 29607136, 29614098, 29616244, 29626154, 29629430, 29629745
29632265, 29633753, 29637526, 29643721, 29645349, 29651520, 29667994
29668005, 29676089, 29687459, 29703195, 29707896, 29719146, 29720133
29724041, 29726695, 29739576, 29773197, 29782211, 29791152, 29794462
29813494, 29817278, 29825525, 29841687, 29846645, 29853485, 29865188

29869404, 29875459, 29884958, 29893132, 29914449, 29944035, 29944660
29951620, 29962927, 29962939, 29991257, 29997326, 30034456, 30074349
30088912, 30098251, 30125944, 30164714, 30189023, 30193165, 30218044
30223712, 30252098, 30252156, 30253255, 30259008, 30342878, 30365745
30402386, 30408515, 30458593, 30485255

Version 18.0.0.0.ru-2019-10.rur-2019-10.r1

Version 18.0.0.0.ru-2019-10.rur-2019-10.r1 includes the following:

- Patch 30112122: DATABASE OCT 2019 RELEASE UPDATE 18.8.0.0.191015
- Patch 30133625: OJVM RELEASE UPDATE 12.2.0.1.191015
- Patch 29997937: DSTv34 for RDBMS (TZDATA2019G)
- Patch 29997959: DSTv34 for OJVM (TZDATA2019G)
- PreUpgrade Jar: preupgrade_181_cbuild_8_lf.zip
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR is included in DB PATCH 30138470
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- Support for [Resizing the temporary tablespace in a read replica \(p. 1061\)](#)

Oracle release update 18.8.0.0.0, released October 2019

Bugs fixed:

30259008, 30253255, 8932139, 9062315, 13554903, 14221306, 20436508
21095391, 21223848, 21374587, 21547051, 21744603, 21766220, 21806121
21935698, 23003564, 23061453, 23310101, 23761724, 24489904, 24687075
24689376, 24737581, 24841671, 24844841, 24903291, 24925863, 24971597
25035594, 25035599, 25060506, 25287072, 25293659, 25303284, 25309116
25348956, 25405687, 25487146, 25591292, 25607397, 25634405, 25644811
25663488, 25686739, 25696520, 25726981, 25736428, 25743479, 25756945
25824236, 25882883, 25890002, 25908728, 25911069, 25929650, 25943740
25958554, 25986062, 25997810, 26083489, 26115103, 26164661, 26226953
26237338, 26281476, 26284722, 26297826, 26336101, 26362155, 26399691
26422277, 26423085, 26427905, 26440169, 26450454, 26476244, 26521043
26536320, 26595088, 26598422, 26615291, 26617804, 26646549, 26654411
26694735, 26724511, 26731697, 26745002, 26785169, 26790514, 26792891
26798411, 26818960, 26822620, 26843558, 26843664, 26846077, 26871815
26883456, 26894737, 26895149, 26898279, 26910716, 26927998, 26928317
26933599, 26943004, 26956033, 26960097, 26961415, 26966120, 26966916
26970175, 26976568, 26985002, 26986173, 26992964, 27000158, 27005278
27006120, 27006664, 27016033, 27026401, 27028251, 27030974, 27033520
27034688, 27035653, 27036408, 27037839, 27038986, 27041253, 27044575
27047831, 27053044, 27054231, 27058530, 27060167, 27060859, 27061736
27066451, 27066519, 27072923, 27073066, 27075854, 27080874, 27086821
27090765, 27093423, 27101105, 27101527, 27101652, 27105900, 27106301
27110878, 27111780, 27112686, 27115422, 27119621, 27122162, 27125872
27126666, 27128580, 27135647, 27142529, 27143756, 27143882, 27144928
27147979, 27150500, 27151826, 27152892, 27153641, 27155549, 27156355
27160360, 27160922, 27163928, 27164352, 27165231, 27166354, 27169796
27169888, 27170305, 27179264, 27181521, 27181537, 27181897, 27185188
27187440, 27189611, 27190851, 27193810, 27197334, 27199245, 27200959
27202015, 27208795, 27208953, 27210038, 27210263, 27210872, 27214085
27214204, 27215007, 27216046, 27216224, 27220937, 27221900, 27222121
27222626, 27223075, 27224987, 27226913, 27228786, 27229389, 27231051

27232983, 27233563, 27236052, 27236110, 27236722, 27240246, 27240570
27241221, 27241247, 27242226, 27242616, 27244337, 27244785, 27249215
27249531, 27250547, 27251690, 27254851, 27255377, 27256000, 27256488
27256534, 27256584, 27258578, 27259307, 27259386, 27259983, 27262601
27262650, 27262798, 27262945, 27262991, 27263276, 27263996, 27264464
27266245, 27270197, 27274456, 27274536, 27275136, 27275533, 27275776
27276231, 27282707, 27283029, 27283960, 27284375, 27284499, 27285244
27288830, 27288838, 27288894, 27292213, 27293599, 27294480, 27301308
27301568, 27302594, 27302632, 27302681, 27302695, 27302711, 27302714
27302730, 27302777, 27302800, 27302960, 27303287, 27303785, 27303938
27304410, 27304906, 27304936, 27305318, 27307868, 27308088, 27310092
27313687, 27314206, 27314390, 27318117, 27318869, 27320576, 27321179
27321834, 27326204, 27329812, 27330158, 27330161, 27333658, 27333664
27333693, 27333731, 27334316, 27334648, 27335682, 27338912, 27338946
27339115, 27339396, 27339483, 27339495, 27341036, 27345190, 27345231
27345450, 27345498, 27346329, 27346644, 27346709, 27346949, 27347126
27348081, 27348707, 27349393, 27350267, 27351628, 27352600, 27354783
27356373, 27357773, 27358241, 27359178, 27359368, 27360126, 27364854
27364891, 27364916, 27364947, 27365139, 27365702, 27365993, 27367194
27368850, 27369515, 27372756, 27375260, 27375542, 27376871, 27378103
27379233, 27381383, 27381417, 27381498, 27381656, 27383281, 27384222
27386467, 27389352, 27392187, 27393570, 27394086, 27395404, 27395416
27395794, 27396357, 27396365, 27396377, 27396624, 27396666, 27396672
27396813, 27397048, 27398080, 27398660, 27400416, 27400598, 27401637
27404573, 27404668, 27405242, 27405645, 27405696, 27406105, 27410279
27410300, 27410595, 27412805, 27416327, 27416997, 27417186, 27420715
27421101, 27421733, 27422874, 27423251, 27424405, 27425507, 27425622
27426363, 27427805, 27430802, 27432062, 27432338, 27432355, 27432826
27433385, 27433870, 27434050, 27434193, 27434486, 27434974, 27435537
27439835, 27441326, 27441980, 27442041, 27444727, 27445330, 27445462
27445727, 27447452, 27447687, 27448162, 27449814, 27450355, 27450400
27450783, 27451049, 27451182, 27451187, 27451531, 27452046, 27452760
27453225, 27454722, 27457666, 27457891, 27458164, 27459909, 27459948
27460675, 27462994, 27466597, 27467543, 27468303, 27469245, 27469329
27471876, 27472969, 27473800, 27479358, 27480784, 27483974, 27484556
27486253, 27487309, 27487795, 27487919, 27489719, 27493674, 27496224
27496308, 27496424, 27497950, 27498477, 27501327, 27501413, 27501465
27502420, 27504190, 27504770, 27505229, 27505603, 27506774, 27507968
27508985, 27510959, 27511196, 27512439, 27517818, 27518227, 27518310
27520070, 27520900, 27522245, 27523368, 27523800, 27525909, 27526744
27532375, 27533780, 27533819, 27534509, 27537472, 27539757, 27540613
27541286, 27541468, 27542824, 27544030, 27544973, 27545630, 27547732
27550341, 27551855, 27554074, 27555481, 27558557, 27558559, 27558861
27560602, 27560702, 27562488, 27563629, 27563767, 27565906, 27567477
27570318, 27576342, 27576354, 27577758, 27578007, 27579353, 27580996
27585755, 27585800, 27586810, 27586895, 27587672, 27588271, 27591842
27592466, 27593389, 27593501, 27593585, 27595973, 27599689, 27599927
27601118, 27601441, 27602091, 27602488, 27603841, 27604293, 27607563
27607805, 27608669, 27610269, 27613080, 27613247, 27613530, 27615608
27616657, 27617522, 27617978, 27620808, 27623159, 27623844, 27625274
27625620, 27629756, 27631506, 27632114, 27634676, 27634991, 27635508
27644757, 27645940, 27649707, 27652302, 27654521, 27655217, 27658186
27658205, 27662528, 27663370, 27664702, 27666312, 27671633, 27679488
27679664, 27679793, 27679806, 27679961, 27680162, 27680509, 27680669
27682151, 27686599, 27688036, 27688099, 27688692, 27690513, 27690578
27691809, 27691920, 27691939, 27692215, 27693416, 27693713, 27694261
27695063, 27697092, 27698953, 27700466, 27701795, 27704237, 27705761
27707544, 27709046, 27710072, 27718914, 27719187, 27723002, 27723151
27726269, 27726780, 27729678, 27732323, 27733415, 27739006, 27740424
27740844, 27744211, 27745220, 27747869, 27748954, 27751006, 27751755
27753336, 27756900, 27757567, 27757794, 27757888, 27758544, 27758653
27758972, 27759077, 27759457, 27761402, 27766324, 27767081, 27772093
27772815, 27773602, 27774320, 27774539, 27779886, 27780562, 27782339
27783289, 27786772, 27791223, 27793533, 27797290, 27801337, 27803665
27807441, 27810967, 27812560, 27812593, 27813267, 27815347, 27818871
27824540, 27824543, 27825241, 27829295, 27832643, 27833369, 27833672

27834551, 27834569, 27834984, 27835925, 27839353, 27840386, 27843646
27846298, 27846499, 27847259, 27849825, 27850112, 27851757, 27856471
27861226, 27861452, 27861909, 27869075, 27869339, 27873412, 27873643
27876671, 27882176, 27886087, 27892488, 27896443, 27896458, 27898015
27900663, 27902561, 27908644, 27912301, 27918832, 27920184, 27924147
27926113, 27930478, 27931299, 27934468, 27935348, 27935493, 27938736
27940876, 27941110, 27941896, 27945870, 27948050, 27948153, 27950708
27952762, 27959594, 27960021, 27961746, 27964051, 27965400, 27965830
27966472, 27967484, 27970265, 27971503, 27971575, 27972265, 27975778
27977039, 27983174, 27984028, 27986817, 27989556, 27989849, 27991970
27993289, 27994325, 27994333, 27995215, 27995248, 27997875, 27998003
27999073, 27999597, 27999638, 28000269, 28004853, 28006704, 28018962
28019283, 28021205, 28022101, 28022847, 28023081, 28023399, 28023482
28024347, 28026866, 28033429, 28036487, 28043157, 28045903, 28057267
28058612, 28059199, 28067846, 28072130, 28072464, 28072567, 28074713
28085865, 28088762, 28090453, 28091981, 28098865, 28103600, 28103869
28104361, 28104409, 28106402, 28108003, 28108898, 28111583, 28120036
28120951, 28124631, 28125947, 28129791, 28132287, 28135648, 28165439
28165545, 28169711, 28174827, 28174951, 28175445, 28180464, 28181021
28184554, 28184800, 28187706, 28188330, 28189466, 28194173, 28199085
28201419, 28204443, 28209341, 28210192, 28211734, 28214943, 28215510
28218832, 28220398, 28223871, 28226179, 28227512, 28229360, 28236305
28238264, 28242712, 28256164, 28258608, 28264172, 28271119, 28271693
28279837, 28281094, 28282606, 28285766, 28290434, 28294563, 28302049
28304709, 28305001, 28305362, 28309182, 28312508, 28315031, 28315995
28319114, 28319623, 28320117, 28320399, 28321446, 28323201, 28328895
28329450, 28330714, 28333072, 28338399, 28338999, 28344964, 28350595
28354603, 28357401, 28361083, 28361221, 28361787, 28365111, 28369092
28371123, 28378446, 28379065, 28384353, 28385102, 28390273, 28391210
28391582, 28392168, 28392251, 28393678, 28396445, 28401116, 28403295
28413955, 28420042, 28420457, 28423598, 28432129, 28434028, 28435825
28445741, 28448314, 28455212, 28468312, 28475164, 28478676, 28481149
28481679, 28483184, 28489150, 28492362, 28493478, 28498976, 28501075
28502403, 28502773, 28503484, 28504545, 28507324, 28508053, 28508296
28508557, 28512761, 28513333, 28514693, 28521330, 28527416, 28528349
28535272, 28542455, 28544633, 28545134, 28545687, 28547068, 28553468
28571483, 28572834, 28578164, 28580528, 28584193, 28584217, 28584444
28585411, 28587723, 28600233, 28612674, 28614072, 28617631, 28617959
28621470, 28627255, 28627686, 28632559, 28636676, 28644549, 28646200
28670445, 28673203, 28678804, 28679454, 28680029, 28685371, 28692103
28692275, 28695694, 28697806, 28702188, 28708023, 28710827, 28713840
28714058, 28714988, 28728040, 28728272, 28730076, 28742555, 28747182
28749289, 28752599, 28755011, 28758090, 28758722, 28761812, 28767240
28770146, 28774416, 28776811, 28777214, 28781754, 28785022, 28785531
28791725, 28797711, 28803345, 28805612, 28805695, 28809909, 28817449
28819640, 28820669, 28831971, 28849751, 28852691, 28855922, 28856060
28856172, 28867992, 28876639, 28878525, 28881723, 28887509, 28889730
28891984, 28905457, 28919145, 28925880, 28938924, 28940179, 28951014
28951382, 28956908, 28960211, 28965084, 28986231, 28987454, 28993353
28993590, 29000190, 29002488, 29006527, 29009513, 29015118, 29015706
29024054, 29027694, 29027940, 29032276, 29033896, 29037290, 29051702
29056270, 29056767, 29123482, 29125374, 29136111, 29139591, 29158680
29163567, 29171683, 29177886, 29189889, 29198092, 29200700, 29203604
29205918, 29212433, 29213320, 29213351, 29224605, 29225076, 29230252
29230565, 29233415, 29241345, 29242017, 29247906, 29250317, 29255273
29285503, 29301463, 29312672, 29314539, 29331209, 29331493, 29339155
29347981, 29351662, 29353821, 29356752, 29364171, 29376346, 29378913
29379978, 29382784, 29386635, 29388020, 29388952, 29391849, 29394749
29395657, 29398488, 29404483, 29405462, 29407804, 29409149, 29412269
29417719, 29429264, 29430524, 29436454, 29437712, 29439522, 29464779
29465177, 29483672, 29483723, 29493122, 29500257, 29504682, 29511611
29515766, 29524599, 29525886, 29530515, 29531541, 29541742, 29542449
29548413, 29557261, 29607136, 29614098, 29616244, 29632265, 29633753
29637526, 29645349, 29668005, 29676089, 29687459, 29707896, 29724041
29782211, 29813494, 29893132, 30034456, 30088912, 30189023

Version 18.0.0.0.ru-2019-07.rur-2019-07.r1

Version 18.0.0.0.ru-2019-07.rur-2019-07.r1 includes the following:

- Patch 29757256: Database Release Update: 18.7.0.0.190716
- Patch 29774410: OJVM RELEASE UPDATE: 18.7.0.0.190716
- Patch 27539475: "ORA-3816 - MISSING MESSAGE INFORMATION FOR 3816 ERROR."
- Patch 29213893: "DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER\$ TABLE"
- Patch 28125601: DSTv32 for RDBMS (TZDATA2018E)
- Patch 28852325: DSTv33 for RDBMS (TZDATA2018G)
- Patch 28852334: DSTv33 for OJVM (TZDATA2018G)
- PreUpgrade Jar: preupgrade_181_cbuild_7_lf.zip
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR

Oracle release update 18.7.0.0.190716, released July 2019

Bugs fixed:

```
8932139, 9062315, 13554903, 14221306, 20436508, 21095391, 21223848  
21547051, 21744603, 21766220, 21806121, 23003564, 23061453, 23310101  
23761724, 24489904, 24689376, 24737581, 24841671, 24844841, 24903291  
24925863, 24971597, 25035594, 25035599, 25287072, 25293659, 25303284  
25309116, 25348956, 25405687, 25487146, 25591292, 25607397, 25634405  
25644811, 25663488, 25686739, 25726981, 25736428, 25743479, 25756945  
25824236, 25882883, 25890002, 25908728, 25911069, 25929650, 25943740  
25958554, 25986062, 25997810, 26083489, 26115103, 26164661, 26226953  
26237338, 26281476, 26284722, 26297826, 26336101, 26362155, 26399691  
26422277, 26423085, 26427905, 26440169, 26450454, 26476244, 26521043  
26536320, 26595088, 26598422, 26615291, 26617804, 26646549, 26654411  
26694735, 26731697, 26745002, 26785169, 26790514, 26792891, 26818960  
26822620, 26843558, 26843664, 26846077, 26871815, 26883456, 26894737  
26895149, 26898279, 26927998, 26928317, 26933599, 26943004, 26956033  
26960097, 26961415, 26966120, 26966916, 26970175, 26985002, 26986173  
26992964, 27000158, 27005278, 27006120, 27006664, 27016033, 27026401  
27028251, 27030974, 27034688, 27035653, 27036408, 27037839, 27038986  
27041253, 27044575, 27047831, 27053044, 27054231, 27058530, 27060167  
27060859, 27061736, 27066451, 27066519, 27073066, 27075854, 27080874  
27086821, 27090765, 27093423, 27101105, 27101527, 27101652, 27105900  
27106301, 27110878, 27111780, 27112686, 27115422, 27119621, 27122162  
27125872, 27126666, 27128580, 27135647, 27142529, 27143756, 27143882  
27147979, 27150500, 27151826, 27152892, 27153641, 27155549, 27156355  
27160360, 27160922, 27163928, 27164352, 27165231, 27166354, 27169796  
27169888, 27170305, 27179264, 27181521, 27181537, 27181897, 27185188  
27187440, 27189611, 27190851, 27193810, 27197334, 27199245, 27200959  
27202015, 27208795, 27208953, 27210038, 27210263, 27210872, 27214085  
27215007, 27216046, 27216224, 27220937, 27221900, 27222121, 27222626  
27223075, 27224987, 27226913, 27228786, 27229389, 27231051, 27232983  
27233563, 27236052, 27236110, 27236722, 27240246, 27240570, 27241221  
27241247, 27242226, 27242616, 27244337, 27244785, 27249215, 27249531  
27250547, 27251690, 27254851, 27255377, 27256000, 27256488, 27256534  
27256584, 27258578, 27259307, 27259386, 27259983, 27262601, 27262650  
27262798, 27262945, 27262991, 27263276, 27263996, 27266245, 27270197  
27274456, 27274536, 27275136, 27275533, 27275776, 27276231, 27282707  
27283029, 27283960, 27284375, 27284499, 27285244, 27288230, 27288638  
27288894, 27292213, 27293599, 27294480, 27301308, 27301568, 27302594
```

27302632, 27302681, 27302695, 27302711, 27302714, 27302730, 27302777
27302800, 27302960, 27303287, 27303785, 27303938, 27304410, 27304936
27305318, 27307868, 27308088, 27310092, 27313687, 27314206, 27314390
27318117, 27318869, 27320576, 27321179, 27321834, 27326204, 27329812
27330158, 27330161, 27333658, 27333664, 27333693, 27333731, 27334316
27334648, 27335682, 27338912, 27338946, 27339115, 27339396, 27339483
27339495, 27341036, 27345190, 27345231, 27345450, 27345498, 27346329
27346644, 27346709, 27346949, 27347126, 27348081, 27348707, 27349393
27350267, 27351628, 27352600, 27354783, 27356373, 27357773, 27358241
27359178, 27359368, 27360126, 27364854, 27364891, 27364916, 27364947
27365139, 27365702, 27365993, 27367194, 27368850, 27372756, 27375260
27375542, 27376871, 27378103, 27379233, 27381383, 27381417, 27381498
27381656, 27384222, 27386467, 27389352, 27392187, 27394086, 27395404
27395416, 27395794, 27396357, 27396365, 27396377, 27396624, 27396666
27396672, 27396813, 27397048, 27398080, 27398660, 27400416, 27400598
27401637, 27404668, 27405242, 27405645, 27405696, 27410279, 27410300
27410595, 27412805, 27416327, 27416997, 27417186, 27420715, 27421101
27421733, 27422874, 27423251, 27424405, 27425507, 27425622, 27426363
27427805, 27430802, 27432062, 27432338, 27432355, 27432826, 27433870
27434050, 27434193, 27434486, 27434974, 27435537, 27439835, 27441326
27441980, 27442041, 27444727, 27445330, 27445462, 27445727, 27447452
27447687, 27448162, 27449814, 27450355, 27450400, 27450783, 27451049
27451182, 27451187, 27451531, 27452046, 27452760, 27453225, 27454722
27457666, 27457891, 27458164, 27459909, 27460675, 27462994, 27466597
27467543, 27468303, 27469245, 27469329, 27471876, 27472969, 27473800
27479358, 27480784, 27483974, 27484556, 27486253, 27487309, 27487795
27487919, 27489719, 27493674, 27496224, 27496308, 27496424, 27497950
27498477, 27501327, 27501413, 27501465, 27502420, 27504190, 27504770
27505229, 27505603, 27506774, 27507968, 27508985, 27510959, 27511196
27512439, 27517818, 27518227, 27518310, 27520070, 27520900, 27522245
27523368, 27523800, 27525909, 27526744, 27532375, 27533819, 27534509
27537472, 27539757, 27540613, 27541286, 27541468, 27542824, 27544030
27544973, 27545630, 27547732, 27550341, 27551855, 27554074, 27555481
27558557, 27558559, 27558861, 27560602, 27560702, 27562488, 27563629
27563767, 27565906, 27567477, 27570318, 27576342, 27577758, 27578007
27579353, 27580996, 27585755, 27585800, 27586810, 27586895, 27587672
27591842, 27592466, 27593389, 27593501, 27593585, 27595973, 27599689
27599927, 27601118, 27602091, 27602488, 27603841, 27604293, 27607563
27607805, 27608669, 27610269, 27613080, 27613247, 27613530, 27615608
27616657, 27617522, 27617978, 27620808, 27623844, 27625274, 27625620
27629756, 27631506, 27634676, 27634991, 27635508, 27644757, 27645940
27649707, 27652302, 27654521, 27655217, 27658186, 27658205, 27662528
27663370, 27664702, 27666312, 27671633, 27679488, 27679664, 27679793
27679806, 27679961, 27680162, 27680509, 27680669, 27682151, 27686599
27688036, 27688099, 27688692, 27690513, 27690578, 27691809, 27691920
27691939, 27692215, 27693416, 27693713, 27694261, 27695063, 27697092
27698953, 27700466, 27701795, 27704237, 27705761, 27707544, 27709046
27710072, 27718914, 27719187, 27723002, 27723151, 27726269, 27726780
27729678, 27732323, 27733415, 27739006, 27740424, 27740844, 27744211
27745220, 27747869, 27748954, 27751006, 27751755, 27753336, 27756900
27757567, 27757794, 27757888, 27758972, 27759077, 27759457, 27761402
277666324, 27767081, 27772093, 27772815, 27773602, 27774320, 27774539
27779886, 27780562, 27782339, 27783289, 27786772, 27791223, 27793533
27797290, 27801337, 27803665, 27807441, 27810967, 27812560, 27812593
27813267, 27815347, 27818871, 27824540, 27824543, 27825241, 27829295
27832643, 27833369, 27833672, 27834551, 27834984, 27835925, 27839353
27840386, 27843646, 27846298, 27846499, 27847259, 27849825, 27851757
27856471, 27861226, 27861452, 27861909, 27869075, 27869339, 27873643
27876671, 27882176, 27892488, 27896443, 27896458, 27898015, 27900663
27908644, 27912301, 27918832, 27920184, 27924147, 27926113, 27930478
27931299, 27934468, 27935348, 27938736, 27940876, 27941110, 27941896
27945870, 27948050, 27948153, 27950708, 27952762, 27959594, 27960021
27961746, 27964051, 27965830, 27966472, 27970265, 27971503, 27971575
27972265, 27975778, 27977039, 27983174, 27984028, 27986817, 27989556
27989849, 27991970, 27993289, 27994325, 27994333, 27995215, 27995248
27997875, 27998003, 27999073, 27999597, 27999638, 28000269, 28004853

28006704, 28018962, 28019283, 28021205, 28022101, 28022847, 28023081
28023399, 28023482, 28024347, 28026866, 28033429, 28036487, 28045903
28057267, 28058612, 28059199, 28067846, 28072130, 28072464, 28072567
28074713, 28085865, 28088762, 28090453, 28091981, 28098865, 28103600
28103869, 28104361, 28106402, 28108003, 28108898, 28111583, 28120036
28120951, 28124631, 28129791, 28132287, 28135648, 28165439, 28165545
28169711, 28174827, 28174951, 28175445, 28180464, 28181021, 28184554
28184800, 28187706, 28188330, 28189466, 28194173, 28199085, 28201419
28204443, 28209341, 28210192, 28211734, 28214943, 28215510, 28218832
28220398, 28223871, 28226179, 28227512, 28229360, 28236305, 28238264
28242712, 28258608, 28264172, 28271119, 28271693, 28279837, 28281094
28282606, 28285766, 28290434, 28302049, 28304709, 28305001, 28305362
28309182, 28312508, 28315031, 28315995, 28319623, 28320117, 28320399
28321446, 28323201, 28328895, 28329450, 28330714, 28333072, 28338399
28338999, 28344964, 28350595, 28354603, 28357401, 28361083, 28361221
28361787, 28365111, 28369092, 28371123, 28378446, 28379065, 28385102
28390273, 28391582, 28392168, 28392251, 28393678, 28396445, 28403295
28413955, 28420042, 28420457, 28423598, 28432129, 28434028, 28435825
28445741, 28448314, 28455212, 28468312, 28475164, 28478676, 28481149
28481679, 28483184, 28489150, 28492362, 28493478, 28501075, 28502403
28502773, 28503484, 28504545, 28507324, 28508053, 28508296, 28508557
28512761, 28513333, 28514693, 28521330, 28527416, 28528349, 28535272
28544633, 28545134, 28545687, 28547068, 28553468, 28571483, 28572834
28578164, 28580528, 28584193, 28584217, 28584444, 28587723, 28600233
28612674, 28614072, 28617631, 28617959, 28621470, 28627255, 28632559
28636676, 28644549, 28646200, 28670445, 28673203, 28679454, 28680029
28685371, 28692275, 28695694, 28702188, 28708023, 28710827, 28713840
28714058, 28728040, 28728272, 28730076, 28742555, 28747182, 28749289
28752599, 28755011, 28758090, 28758722, 28761812, 28767240, 28770146
28774416, 28777214, 28781754, 28785022, 28785531, 28803345, 28805612
28805695, 28809909, 28819640, 28849751, 28852691, 28856060, 28856172
28878525, 28881723, 28887509, 28891984, 28919145, 28925880, 28938924
28940179, 28951014, 28951382, 28956908, 28960211, 28986231, 28987454
28993353, 28993590, 29000190, 29006527, 29015118, 29015706, 29024054
29027694, 29032276, 29033896, 29037290, 29051702, 29056270, 29123482
29125374, 29136111, 29139591, 29158680, 29171683, 29177886, 29189889
29200700, 29203604, 29205918, 29224605, 29230565, 29242017, 29301463
29314539, 29331209, 29331493, 29339155, 29347981, 29356752, 29364171
29376346, 29378913, 29379978, 29382784, 29388020, 29388952, 29394749
29395657, 29405462, 29409149, 29412269, 29429264, 29430524, 29436454
29437712, 29439522, 29504682, 29511611, 29531541, 29542449, 29616244
29676089, 29813494

Database engine: 12.2.0.1

For Oracle Database 12c Release 2 (12.2.0.1), Oracle changed the way it releases Oracle Database updates. Instead of Patch Set Updates (PSUs), Oracle supplies Release Updates (RUs) and Release Updates Revisions (RURs). RUs contain optimizer changes, feature additions, and security fixes. RURs only contain security fixes for the two preceding quarterly patch cycles. With this new system, you have more control over the features that you install with each update.

The naming conventions have also changed for Oracle Database 12c version Release 2 (12.2.0.1) versions. In previous versions, Amazon RDS for Oracle used the PSU naming convention of *oracle-version.vpatch-version*. The *patch-version* corresponded with an Oracle PSU. For example, in Oracle for Amazon RDS version 12.1.0.2.v13, the v13 part of the version number corresponds with an Oracle PSU.

Oracle Database 12c Release 2 (12.2.0.1) naming conventions account for both RU and RUR updates. For example, the first Amazon RDS for Oracle version available is 12.2.0.1.ru-2018-10.rur-2018-10.r1. In this example, 12.2 is the major version, and 0.1 is the minor version. The revision version has the following parts:

- ru-2018-10 – the October RU
- rur-2018-10 – the October RUR for the October RU
- r1 – Internal Amazon RDS revision, which lets Amazon RDS differentiate between emergency patches of pre-existing RU/RURs

For more information about the new Oracle Database versioning system, see the posts [Differences between PSU / BP and RU / RUR at the Upgrade your Database – NOW! blog](#) and [RU and RUR patches for Oracle 12.2 at the Oracle–Help blog](#).

The following versions are available for Oracle database engine 12.2.0.1:

- [Version 12.2.0.1.ru-2021-01.rur-2021-01.r1 \(p. 1321\)](#)
- [Version 12.2.0.1.ru-2020-10.rur-2020-10.r1 \(p. 1326\)](#)
- [Version 12.2.0.1.ru-2020-07.rur-2020-07.r1 \(p. 1331\)](#)
- [Version 12.2.0.1.ru-2020-04.rur-2020-04.r1 \(p. 1336\)](#)
- [Version 12.2.0.1.ru-2020-01.rur-2020-01.r1 \(p. 1340\)](#)
- [Version 12.2.0.1.ru-2019-10.rur-2019-10.r1 \(p. 1344\)](#)
- [Version 12.2.0.1.ru-2019-07.rur-2019-07.r1 \(p. 1348\)](#)
- [Version 12.2.0.1.ru-2019-04.rur-2019-04.r1 \(p. 1351\)](#)
- [Version 12.2.0.1.ru-2019-01.rur-2019-01.r1 \(p. 1354\)](#)
- [Version 12.2.0.1.ru-2018-10.rur-2018-10.r1 \(p. 1357\)](#)

Version 12.2.0.1.ru-2021-01.rur-2021-01.r1

Version 12.2.0.1.ru-2021-01.rur-2021-01.r1 includes the following:

- Patch 32228578: DATABASE JAN 2021 RELEASE UPDATE 12.2.0.1.210119
- Patch 32119931: OJVM RELEASE UPDATE 12.2.0.1.210119
- Patch 31335037: RDBMS - DSTV35 UPDATE - TZDATA2020A
- Patch 31335142: DSTV35 UPDATE - TZDATA2020A - NEED OJVM FIX
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- PreUpgrade Jar: preupgrade_12201_cbuild_23_1f.zip
- Java Cryptography Extension (JCE): Unlimited Strength Jurisdiction Policy Files for JVM version 8
- Support for [Managing advisor tasks \(p. 1099\)](#) using procedures in the `rdsadmin.rdsadmin_util` package

Combined patches for version 12.2.0.1.ru-2021-01.rur-2021-01.r1, released January 2021

Bugs fixed:

```
7391838, 8480838, 8932139, 8975044, 12763598, 13554903, 14221306
14690846, 15931756, 16002385, 16438495, 16727454, 16942578, 17027695
17533661, 17947871, 17958365, 18308268, 18521691, 18594510, 18774543
18878420, 18986501, 19072655, 19211433, 19285025, 19327292, 19526548
19614243, 19647894, 19649997, 19702201, 19721304, 20003668, 20087519
20118035, 20120236, 20324049, 20436508, 20532077, 20549013, 20588486
20591151, 20617383, 20620169, 20736227, 20756305, 20866970, 20917487
```

20976443, 21070321, 21089435, 21095391, 21143725, 21147908, 21159907
21178363, 21186167, 21197098, 21216226, 21320338, 21355390, 21433452
21479706, 21517767, 21520266, 21547051, 21638600, 21744603, 21788462
21837606, 21882528, 21935698, 21962287, 21981529, 21985256, 22007324
22070226, 22070473, 22070853, 22072543, 22087683, 22104866, 22107360
22174392, 22179537, 22282748, 22310426, 22347493, 22363790, 22364044
22367053, 22379010, 22446455, 22454940, 22468255, 22495673, 22503283
22503297, 22504793, 22522515, 22530986, 22564336, 22568728, 22581771
22594071, 22599050, 22628825, 22645009, 22645496, 22654475, 22700845
22726044, 22729345, 22820798, 22826067, 22843979, 22845846, 22864303
22898198, 22921674, 22939829, 22950945, 22970869, 22981722, 23018676
23019710, 23026585, 23035249, 23055900, 23056058, 23061453, 23065002
23066146, 23080557, 23104033, 23105538, 23109325, 23110523, 23125560
23126545, 23127945, 23143074, 23151677, 23168363, 23169712, 23177923
23179662, 23184263, 23197730, 23234232, 23237091, 23249829, 23271203
23278750, 23281269, 23282973, 23300142, 23306590, 23308065, 23310101
23312077, 23328639, 23333567, 23336559, 23342170, 23481673, 23491861
23499004, 23499160, 23521523, 23527363, 23533647, 23548817, 23567857
23572982, 23581777, 23588722, 23599216, 23600861, 23602213, 23614158
23645516, 23665623, 23709062, 23715460, 23715518, 23730961, 23733981
23735292, 23738304, 23741944, 23743596, 23746128, 23749454, 23761724
23763462, 24006569, 24010030, 24289874, 24289895, 24294174, 24303148
24307571, 24308349, 24326444, 24326846, 24328857, 24330708, 24332831
24334708, 24336249, 24337882, 24341675, 24343905, 24345420, 24346821
24348685, 24350620, 24352981, 24355111, 24357348, 24368004, 24371491
24373528, 24373756, 24374976, 24376875, 24376878, 24383086, 24385983
24401351, 24403922, 24409977, 24415926, 24416451, 24421668, 24423416
24425056, 24425998, 24432875, 24435982, 24437162, 24440612, 24440648
24443539, 24445571, 24457597, 24460392, 24461826, 24467122, 24468470
24470606, 24471079, 24471473, 24473736, 24484749, 24485034, 24485161
24485174, 24485619, 24486059, 24486237, 24509056, 24516314, 24530364
24534401, 24554533, 24555417, 24556862, 24556967, 24560906, 24563422
24570214, 24570598, 24573817, 24578718, 24578797, 24588377, 24589081
24589590, 24591506, 24593740, 24595699, 24596874, 24600330, 24609592
24609996, 24611527, 24616637, 24617969, 24623975, 24624166, 24642495
24654629, 24655717, 24664211, 24668398, 24669189, 24669730, 24674197
24674955, 24676172, 24677696, 24680959, 24687075, 24689376, 24692973
24693010, 24693290, 24697323, 24699619, 24701840, 24710696, 24713381
24714096, 24717183, 24717859, 24718260, 24719799, 24735430, 24737064
24737403, 24737581, 24737954, 24739791, 24744383, 24744686, 24752618
24757934, 24759556, 24760407, 24763196, 24764085, 24766309, 24784414
24786669, 24791883, 24792678, 24793511, 24796092, 24797119, 24798481
24800423, 24801152, 24802934, 24808504, 24811725, 24812047, 24817447
24818566, 24827228, 24827654, 24831514, 24835919, 24841671, 24843188
24844549, 24844841, 24845157, 24848746, 24848923, 24850622, 24907917
24908063, 24908321, 24911709, 24912588, 24920582, 24921478, 24922704
24923080, 24923215, 24923338, 24923790, 24924667, 24926999, 24929210
24938784, 24940060, 24942749, 24953434, 24957555, 24960044, 24960809
24965426, 24966594, 24966788, 24967993, 24968162, 24976007, 24978100
25022574, 25027852, 25028996, 25029022, 25029423, 25031502, 25032818
25034396, 25036006, 25036474, 25042823, 25044977, 25045228, 25050160
25051465, 25051628, 25054064, 25057811, 25058080, 25060506, 25062592
25063971, 25065563, 25072986, 25077278, 25078611, 25086233, 25087436
25091141, 25092777, 25093872, 25095982, 25098160, 25099339, 25099497
25099758, 25100063, 25100579, 25103996, 25107662, 25110233, 25114561
25115178, 25120284, 25120668, 25120742, 25121089, 25123585, 25124363
25129925, 25130312, 25140197, 25145163, 25145215, 25150925, 25159176
25162645, 25164293, 25166187, 25171041, 25171084, 25173124, 25175723
25176408, 25178032, 25178101, 25178179, 25179774, 25182817, 25184453
25184555, 25186079, 25189723, 25191872, 25192044, 25192528, 25192729
25195901, 25199585, 25200101, 25201454, 25202355, 25203656, 25205368
25205954, 25206864, 25207410, 25209912, 25210268, 25210499, 25210690
25211628, 25219450, 25223839, 25224242, 25225795, 25226665, 25227381
25230870, 25230945, 25237577, 25240188, 25240590, 25241448, 25241625
25244807, 25248384, 25250109, 25251648, 25257085, 25259611, 25262869
25263960, 25265499, 25269133, 25283790, 25287072, 25293659, 25296876

25299227, 25299807, 25300427, 25303284, 25303756, 25305405, 25307368
25309116, 25313154, 25313411, 25316758, 25317989, 25320555, 25323525
25328093, 25328518, 25329664, 25335249, 25335360, 25335790, 25337332
25337640, 25348567, 25348956, 25353983, 25356118, 25357142, 25360661
25362958, 25367588, 25367721, 25382812, 25383204, 25384462, 25386748
25388573, 25388896, 25392535, 25393714, 25395696, 25397936, 25398306
25404117, 25404202, 25405100, 25405687, 25405813, 25410017, 25410180
25410802, 25410877, 25411036, 25415713, 25416731, 25417050, 25417056
25417958, 25425005, 25425451, 25425760, 25427662, 25429959, 25430120
25433696, 25435038, 25437699, 25440818, 25442559, 25444961, 25445168
25451531, 25452452, 25455795, 25457409, 25459958, 25462714, 25463844
25472112, 25472885, 25476125, 25476149, 25477657, 25478885, 25479164
25481087, 25482971, 25486384, 25489342, 25489367, 25489607, 25492379
25498930, 25498994, 25516250, 25524955, 25528838, 25530080, 25530814
25535668, 25536819, 25537470, 25539063, 25540738, 25546580, 25546608
25547901, 25551676, 25553616, 25554787, 25555252, 25557886, 25558986
25560487, 25560538, 25561296, 25569149, 25569504, 25570929, 25573623
25575348, 25575369, 25575628, 25576115, 25579458, 25579761, 25591394
25594901, 25597525, 25598473, 25599428, 25600342, 25600421, 25601999
25602488, 25603923, 25606091, 25607726, 25612095, 25614866, 25616268
25616359, 25616417, 25616465, 25631933, 25633101, 25634317, 25634348
25635149, 25638456, 25639019, 25643818, 25643889, 25643931, 25646373
25647325, 25648731, 25653109, 25654459, 25654936, 25655390, 25655966
25659655, 25660847, 25661819, 25662088, 25662101, 25662524, 25663488
25667973, 25669791, 25670786, 25671354, 25672640, 25674386, 25680221
25685152, 25686739, 25687460, 25691904, 25694206, 25695903, 25696520
25699321, 25700654, 25709368, 25710420, 25715167, 25717371, 25722055
25722608, 25722720, 25723097, 25723158, 25728085, 25729507, 25730014
25734963, 25736747, 25739065, 25740844, 25741955, 25743479, 25747569
25749273, 25752755, 25754606, 25756945, 25757697, 25757748, 25760195
25762221, 25764020, 25766822, 25768681, 25772669, 25774077, 25775213
25775444, 25780343, 25783447, 25784002, 25785331, 25785441, 25788879
25789041, 25789277, 25789579, 25790353, 25792911, 25795865, 25797092
25797124, 25797305, 25800464, 25802510, 25803364, 25803545, 25807997
25809524, 25810263, 25810704, 25811105, 25811650, 25812390, 25813931
25818707, 25822410, 25823532, 25823754, 25824372, 25825910, 25826740
25830492, 25832935, 25834581, 25835365, 25838361, 25838755, 25852885
25856821, 25858672, 25861398, 25865785, 25866948, 25870579, 25871177
25871639, 25871753, 25872127, 25872389, 25873336, 25874050, 25874678
25881255, 25882264, 25883438, 25885148, 25888073, 25888984, 25890002
25890046, 25890056, 25890673, 25890782, 25894239, 25895224, 25897615
25898228, 25904273, 25904490, 25905130, 25906117, 25906886, 25908728
25911724, 25914276, 25919622, 25932524, 25932728, 25933494, 25941836
25942868, 25943271, 25945130, 25947799, 25951571, 25953857, 25954022
25954054, 25957038, 25963024, 25964954, 25967544, 25967985, 25970731
25971286, 25972417, 25973152, 25975723, 25977302, 25980605, 25980770
25981498, 25982666, 25986062, 25990907, 25995938, 25997810, 26006257
26007010, 26019148, 26023042, 26024732, 26024784, 26025681, 26029075
26029777, 26029780, 26032573, 26034119, 26036748, 26037215, 26038086
26039623, 26040483, 26045732, 26051656, 26078437, 26078493, 26080410
26083298, 26087754, 26088426, 26088836, 26089669, 26090767, 26090893
26091640, 26091786, 26095327, 26095405, 26096382, 26108080, 26108337
26110259, 26110632, 26111842, 26112621, 26115103, 26121990, 26124078
26130486, 26137367, 26137416, 26138085, 26145560, 26149904, 26153372
26153977, 26168933, 26169341, 26169345, 26170659, 26170715, 26176002
26187943, 26189861, 26198757, 26198926, 26201113, 26203182, 26223039
26237338, 26237431, 26237773, 26238195, 26242031, 26242677, 26243698
26244115, 26245237, 26248143, 26249718, 26256131, 26257953, 26259265
26261327, 26263328, 26263721, 26268756, 26269790, 26271001, 26274660
26275023, 26275415, 26277439, 26281476, 26285062, 26285933, 26301540
26308650, 26309047, 26317991, 26318200, 26318627, 26323308, 26324206
26324769, 26325856, 26327418, 26327624, 26327775, 26330994, 26331743
26333141, 26334602, 26336977, 26338953, 26351334, 26351996, 26353617
26354844, 26356098, 26358670, 26359091, 26362155, 26362821, 26366517
26367012, 26367460, 26371725, 26373967, 26374791, 26375052, 26375250
26375330, 26380097, 26385189, 26386858, 26388538, 26396790, 26398675

26399626, 26399691, 26399839, 26405036, 26406387, 26407408, 26410240
26412540, 26418088, 26420561, 26421667, 26422277, 26423085, 26426526
26426967, 26430323, 26430737, 26434436, 26434999, 26435073, 26436168
26438612, 26439748, 26440142, 26440169, 26440749, 26441345, 26442308
26444601, 26444887, 26446098, 26451793, 26452606, 26474662, 26474703
26475419, 26476090, 26476244, 26478970, 26479173, 26482376, 26486365
26492666, 26492866, 26493289, 26498354, 26513067, 26513709, 26521043
26522439, 26523432, 26526726, 26526799, 26536320, 26537307, 26542135
26542236, 26542835, 26544823, 26545688, 26546070, 26546664, 26546754
26548363, 26556014, 26558437, 26569225, 26570134, 26575788, 26580633
26582460, 26584641, 26588069, 26597140, 26599395, 26608137, 26608238
26609942, 26615291, 26615690, 26617804, 26623652, 26626879, 26629381
26633355, 26633558, 26635897, 26637273, 26637824, 26639167, 26641610
26641852, 26650226, 26650540, 26654363, 26658759, 26659182, 26669550
26680105, 26712331, 26714486, 26714910, 26716835, 26717528, 26724511
26725687, 26727397, 26729494, 26729611, 26740700, 26743240, 26744595
26745002, 26751106, 26751171, 26755171, 26758193, 26764561, 26765212
26768025, 26775602, 26784509, 26790923, 26794786, 26797591, 26798411
26798514, 26802503, 26816582, 26820076, 26822314, 26822620
26824833, 26828994, 26829845, 26830694, 26832296, 26833932, 26837569
26837702, 26840654, 26844406, 26844870, 26849779, 26855855, 26871815
26875822, 26883456, 26895149, 26896659, 26898563, 26907236, 26907327
26908788, 26909100, 26909504, 26910716, 26911000, 26923777, 26939314
26943004, 26944190, 26947373, 26958896, 26963310, 26966616, 26966916
26967713, 26968670, 26969321, 26970175, 26970717, 26981902, 26983259
26985002, 26986173, 26992964, 26999139, 27000158, 27000702, 27006120
27006664, 27009164, 27013146, 27015449, 27028251, 27032785, 27033520
27033652, 27034890, 27036163, 27037839, 27038986, 27039712, 27044169
27044297, 27045634, 27052607, 27056711, 27058530, 27060167, 27060859
27061736, 27072923, 27073314, 27079140, 27079651, 27084613, 27087426
27090765, 27092508, 27093423, 27097854, 27100800, 27101105, 27105900
27106179, 27110878, 27115422, 27117822, 27119621, 27119861, 27122162
27124624, 27125872, 27133662, 27134734, 27135647, 27135993, 27138325
27142120, 27142373, 27142529, 27144928, 27151826, 27153641, 27160922
27161071, 27162390, 27162405, 27163928, 27165231, 27169796, 27170305
27181537, 27181897, 27185188, 27195935, 27199245, 27200959, 27202015
27203055, 27207110, 27207634, 27208795, 27213224, 27216046, 27217412
27223075, 27229389, 27231051, 27234962, 27236722, 27242226, 27244337
27244999, 27248917, 27249531, 27250547, 27251690, 27254335, 27255377
27256000, 27258578, 27259307, 27262945, 27264464, 27266245, 27274456
27274536, 27275533, 27276231, 27283960, 27284499, 27285244, 27288638
27292213, 27293599, 27302711, 27302730, 27303287, 27303938, 27304410
27304906, 27305039, 27308088, 27314206, 27314390, 27314697, 27320576
27321179, 27329612, 27333106, 27334316, 27338912, 27338946, 27339115
27345231, 27346709, 27348081, 27349393, 27350267, 27351628, 27359178
27364854, 27365014, 27367194, 27369515, 27370965, 27375542, 27381498
27383281, 27386467, 27392968, 27393570, 27394703, 27395416, 27396624
27396672, 27396813, 27397048, 27400416, 27400598, 27404573, 27404668
27405645, 27416997, 27423251, 27424405, 27426363, 27432062, 27432826
27433385, 27433870, 27434193, 27439835, 27441326, 27442041, 27445727
27452046, 27457891, 27459593, 27459948, 27461740, 27466597, 27468303
27486805, 27487919, 27489107, 27493674, 27494663, 27501373, 27501413
27502420, 27504770, 27505229, 27508985, 27510959, 27525909, 27529661
27533780, 27533819, 27534509, 27539876, 27540613, 27544973, 27548131
27554074, 27555481, 27558861, 27560602, 27562488, 27565906, 27567477
27576342, 27576354, 27587905, 27588271, 27589260, 27593501, 27595973
27601118, 27601441, 27607563, 27611612, 27613080, 27613530, 27613554
27615649, 27617978, 27620808, 27623159, 27629756, 27629928, 27632114
27634676, 27634991, 27642235, 27645231, 27657712, 27658186, 27666312
27671633, 27680669, 27686599, 27687880, 27688036, 27688099, 27688692
27691920, 27691939, 27693416, 27693713, 27695063, 27698953, 27700466
27704237, 27709046, 27710072, 27717210, 27719000, 27726780, 27729678
27739006, 27740424, 27745728, 27748954, 27751755, 27757567, 27757888
27758544, 27758653, 27758972, 27759077, 27769361, 27779886, 27793533
27799032, 27801337, 27818389, 27819881, 27824540, 27824543, 27825241
27828794, 27828892, 27829295, 27833672, 27834551, 27834569, 27835925

27837219, 27839353, 27839616, 27846298, 27846499, 27847259, 27850112
27855490, 27861226, 27873412, 27882176, 27886087, 27897759, 27898015
27902561, 27908396, 27909478, 27927431, 27929287, 27929509, 27931299
27935493, 27940876, 27945870, 27951817, 27952586, 27959048, 27959594
27964513, 27966472, 27967484, 27983174, 27986817, 27994325, 27995215
27995248, 27997875, 27998003, 27999073, 27999638, 28000269, 28007516
28019592, 28022101, 28023081, 28023399, 28023482, 28024793, 28025414
28026866, 28033429, 28040776, 28043157, 28045903, 28066655, 28067846
28071549, 28072383, 28072567, 28073470, 28074713, 28079127, 28090453
28092783, 28098040, 28098160, 28099662, 28104176, 28108003, 28111583
28120036, 28120951, 28124631, 28125601, 28125947, 28129791, 28140658
28157786, 28164480, 28165439, 28171079, 28174827, 28180464, 28181021
28184554, 28188330, 28190796, 28194173, 28199085, 28201419, 28204262
28209985, 28215510, 28218832, 28220398, 28223871, 28226179, 28229360
28236305, 28238264, 28242712, 28250929, 28256164, 28271119, 28276054
28279837, 28281094, 28282606, 28287484, 28290434, 28294563, 28302049
28305001, 28305362, 28305607, 28309406, 28319114, 28320399, 28330714
28330971, 28350595, 28354603, 28357401, 28361221, 28365111, 28369092
28371123, 28373960, 28375383, 28378446, 28384353, 28386259, 28388910
28389153, 28390273, 28391210, 28394726, 28396445, 28397317, 28401116
28402823, 28420042, 28420457, 28423598, 28432129, 28434028, 28435902
28437315, 28439086, 28454215, 28454242, 28468312, 28468493, 28481149
28483184, 28489150, 28501075, 28502098, 28502343, 28503038, 28507324
28508053, 28508557, 28512336, 28521330, 28522441, 28528349, 28530171
28535127, 28535272, 28537715, 28538439, 28542455, 28545134, 28546290
28547068, 28547478, 28564479, 28566241, 28571483, 28572407, 28572834
28578164, 28578945, 28585411, 28587723, 28589509, 28600233, 28602253
28606598, 28608211, 28612674, 28614372, 28617631, 28617959, 28621470
28622202, 28627255, 28636676, 28639299, 28642899, 28678804, 28690694
28691965, 28692103, 28692275, 28697526, 28697806, 28703812, 28708023
28709063, 28710469, 28710734, 28714988, 28715655, 28728272, 28730044
28734355, 28740708, 28742555, 28749289, 28749724, 28758090, 28758722
28774416, 28776431, 28777174, 28791725, 28797711, 28803345, 28808314
28817449, 28819640, 28820669, 28821847, 28827682, 28830691, 28831971
28835937, 28836716, 28838066, 28844866, 28847136, 28849751, 28852325
28852691, 28855922, 28856060, 28856172, 28863263, 28863487, 28867992
28887305, 28889730, 28891984, 28905390, 28910498, 28915870, 28927452
28945922, 28948554, 28949888, 28951026, 28951382, 28956908, 28959493
28960211, 28965095, 28965787, 28986231, 28986257, 28987439, 28991884
28993295, 28993590, 29002488, 29006527, 29007321, 29007353, 29009513
29013832, 29024054, 29026309, 29026582, 29027694, 29032276, 29039510
29040739, 29044086, 29044954, 29048498, 29048728, 29050886, 29060216
29061016, 29115857, 29125374, 29154725, 29158680, 29163567, 29170232
29173817, 29179097, 29182517, 29182901, 29189889, 29198092, 29200700
29203604, 29213320, 29213351, 29213893, 29224605, 29224710, 29237575
29247712, 29249289, 29250230, 29250317, 29254623, 29260956, 29278684
29296257, 29301463, 29307638, 29312889, 29337294, 29338348, 29339155
29343086, 29343156, 29343861, 29347943, 29353821, 29372069, 29372460
29375355, 29375984, 29376346, 29378913, 29379978, 29383695, 29388020
29398488, 29399336, 29405462, 29409149, 29409455, 29418165, 29420254
29426241, 29434301, 29436454, 29437712, 29450812, 29452251, 29454978
29463047, 29464779, 29472618, 29477015, 29483626, 29483672, 29483723
29483771, 29500257, 29500963, 29501218, 29504682, 29511611, 29524985
29530515, 29536342, 29538631, 29542449, 29542580, 29548592, 29549071
29559395, 29564592, 29580394, 29591343, 29608023, 29614575, 29614987
29621961, 29625065, 29626154, 29629430, 29633753, 29637526, 29637560
29645349, 29651520, 29656843, 29667994, 29676089, 29678163, 29685137
29687220, 29688867, 29690625, 29703195, 29705793, 29707896, 29717901
29719146, 29724063, 29726695, 29741319, 29766435, 29767177, 29769901
29774362, 29782211, 29791152, 29794174, 29794462, 29796916, 29807964
29813494, 29815341, 29817278, 29822714, 29825525, 29836659, 29841687
29844131, 29846645, 29853485, 29865188, 29869404, 29869906, 29875459
29876358, 29881050, 29881575, 29884958, 29893132, 29902299, 29902311
29914449, 29930457, 29941062, 29942554, 29944035, 29944159, 29944660
29951620, 29951759, 29961353, 29962927, 29962939, 29965888, 29991257
29997937, 30008125, 30018017, 30018903, 30031027, 30039959, 30064268

30068871, 30076253, 30078934, 30086166, 30088912, 30092280, 30098251
30099302, 30114477, 30116203, 30120608, 30125995, 30131286, 30139392
30147928, 30150731, 30160625, 30163243, 30164714, 30173113, 30177597
30179644, 30186706, 30189023, 30193736, 30196358, 30200680, 30200758
30215130, 30218044, 30218317, 30223712, 30225443, 30239480, 30241567
30244787, 30246179, 30247305, 30252098, 30252156, 30253255, 30265523
30265615, 30272329, 30281591, 30282501, 30283932, 30293345, 30305880
30312094, 30312568, 30316897, 30320029, 30325407, 30331356, 30342878
30345926, 30352623, 30355490, 30357897, 30364613, 30365745, 30368482
30368668, 30372081, 30374739, 30377692, 30381207, 30384121, 30384152
30387666, 30391272, 30397100, 30402386, 30403763, 30408515, 30413137
30416034, 30421204, 30431274, 30441687, 30443393, 30453442, 30458593
30460922, 30464250, 30464655, 30473634, 30474774, 30475115, 30476768
30485255, 30496957, 30497057, 30498824, 30501574, 30503943, 30509277
30510527, 30517516, 30522998, 30528547, 30528704, 30532811, 30533198
30534662, 30578221, 30581448, 30582500, 30606345, 30613937, 30623138
30624864, 30635302, 30652853, 30654409, 30662736, 30668407, 30671813
30679595, 30679771, 30681462, 30698289, 30741263, 30749644, 30755348
30758943, 30783551, 30803210, 30809087, 30814266, 30814285, 30815852
30816938, 30826474, 30855101, 30856358, 30866988, 30887501, 30904672
30905638, 30914674, 30919804, 30922870, 30937340, 30964194, 30968208
30980615, 30985027, 30987088, 30994996, 30998759, 31001455, 31004719
31013127, 31022858, 31028986, 31061482, 31100172, 31104809, 31106577
31109506, 31115502, 31156383, 31172207, 31182793, 31192039, 31194264
31200845, 31201001, 31215438, 31228670, 31254535, 31258101, 31302499
31306248, 31306261, 31309867, 31315876, 31326608, 31331354, 31335037
31335142, 31341859, 31343752, 31347532, 31377487, 31393600, 31408636
31430722, 31501139, 31508450, 31525783, 31544097, 31570161, 31600023
31637680, 31658464, 31668061, 31668872, 31711889, 31718134, 31749740
31771858, 31786838, 31816631, 31867037, 31905033, 31986836, 31997805
32089820, 32165759, 32186646, 32234161, 32296941

Version 12.2.0.1.ru-2020-10.rur-2020-10.r1

Version 12.2.0.1.ru-2020-10.rur-2020-10.r1 includes the following:

- Patch 31741641: Database Oct 2020 Release Update : 12.2.0.1.201020 (31741641)
- Patch 31668898: OJVM RELEASE UPDATE 12.2.0.1.201020 (31668898)
- Patch 31335037: RDBMS - DSTV35 UPDATE - TZDATA2020A
- Patch 31335142: DSTV35 UPDATE - TZDATA2020A - NEED OJVM FIX
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- PreUpgrade Jar: preupgrade_12201_cbuild_23_lf.zip
- Support for [Setting and unsetting system diagnostic events \(p. 1046\)](#) using procedures in the rdsadmin.rdsadmin_util package
- Support for the procedure rdsadmin_util.truncate_apply\$_cdr_info described in [Integrated REPLICAT slow due to query on sys._DBA_APPLY_CDR_INFO \(p. 1236\)](#)

Combined patches for version 12.2.0.1.ru-2020-10.rur-2020-10.r1, released October 2020

Bugs fixed:

7391838, 8480838, 8932139, 8975044, 12763598, 13554903, 14221306
14690846, 15931756, 16002385, 16438495, 16727454, 16942578, 17027695
17533661, 17947871, 17958365, 18308268, 18521691, 18594510, 18774543

18878420, 18986501, 19072655, 19211433, 19285025, 19327292, 19526548
19614243, 19647894, 19649997, 19702201, 19721304, 20003668, 20087519
20118035, 20120236, 20324049, 20436508, 20532077, 20549013, 20588486
20591151, 20617383, 20620169, 20736227, 20756305, 20866970, 20917487
20976443, 21070321, 21089435, 21095391, 21143725, 21147908, 21159907
21178363, 21186167, 21197098, 21216226, 21320338, 21355390, 21433452
21479706, 21517767, 21520266, 21547051, 21638600, 21744603, 21788462
21837606, 21882528, 21935698, 21962287, 21981529, 21985256, 22007324
22070226, 22070473, 22070853, 22072543, 22087683, 22104866, 22107360
22174392, 22179537, 22282748, 22310426, 22347493, 22363790, 22364044
22367053, 22379010, 22446455, 22454940, 22468255, 22495673, 22503283
22503297, 22504793, 22522515, 22530986, 22564336, 22568728, 22581771
22594071, 22599050, 22628825, 22645009, 22645496, 22654475, 22700845
22726044, 22729345, 22826067, 22843979, 22845846, 22864303, 22898198
22921674, 22939829, 22950945, 22970869, 22981722, 23018676, 23019710
23026585, 23035249, 23055900, 23056058, 23061453, 23065002, 23066146
23080557, 23104033, 23105538, 23109325, 23110523, 23125560, 23126545
23127945, 23143074, 23151677, 23168363, 23169712, 23177923, 23179662
23184263, 23197730, 23234232, 23237091, 23249829, 23271203, 23278750
23281269, 23282973, 23300142, 23306590, 23308065, 23310101, 23312077
23328639, 23333567, 23336559, 23342170, 23481673, 23491861, 23499004
23499160, 23521523, 23527363, 23533647, 23548817, 23567857, 23572982
23581777, 23588722, 23599216, 23600861, 23602213, 23614158, 23645516
23665623, 23709062, 23715460, 23715518, 23730961, 23733981, 23735292
23738304, 23741944, 23743596, 23746128, 23749454, 23761724, 23763462
24006569, 24010030, 24289874, 24289895, 24294174, 24303148, 24307571
24308349, 24326444, 24326846, 24328857, 24330708, 24332831, 24334708
24336249, 24337882, 24341675, 24343905, 24345420, 24346821, 24348685
24350620, 24352981, 24355111, 24357348, 24368004, 24371491, 24373528
24373756, 24374976, 24376875, 24376878, 24383086, 24385983, 24401351
24403922, 24409977, 24415926, 24416451, 24421668, 24423416, 24425056
24425998, 24432875, 24435982, 24437162, 24440612, 24440648, 24443539
24445571, 24457597, 24460392, 24461826, 24467122, 24468470, 24470606
24471079, 24471473, 24473736, 24484749, 24485034, 24485161, 24485174
24485619, 24486059, 24486237, 24509056, 24516314, 24530364, 24534401
24554533, 24555417, 24556862, 24556967, 24560906, 24563422, 24570214
24570598, 24573817, 24578718, 24578797, 24588377, 24589081, 24589590
24591506, 24593740, 24595699, 24596874, 24600330, 24609592, 24609996
24611527, 24616637, 24617969, 24623975, 24624166, 24642495, 24654629
24655717, 24664211, 24668398, 24669189, 24669730, 24674197, 24674955
24676172, 24677696, 24680959, 24687075, 24689376, 24692973, 24693010
24693290, 24697323, 24699619, 24701840, 24710696, 24713381, 24714096
24717183, 24717859, 24718260, 24719799, 24735430, 24737064, 24737403
24737581, 24737954, 24739791, 24744383, 24744686, 24752618, 24757934
24759556, 24760407, 24763196, 24764085, 24766309, 24784414, 24786669
24791883, 24792678, 24793511, 24796092, 24797119, 24798481, 24800423
24801152, 24802934, 24808504, 24811725, 24812047, 24817447, 24818566
24827228, 24827654, 24831514, 24835919, 24841671, 24843188, 24844549
24844841, 24845157, 24848746, 24848923, 24850622, 24907917, 24908063
24908321, 24911709, 24912588, 24920582, 24921478, 24922704, 24923080
24923215, 24923338, 24923790, 24924667, 24926999, 24929210, 24938784
24940060, 24942749, 24953434, 24957555, 24960044, 24960809, 24965426
24966594, 24966788, 24967993, 24968162, 24976007, 24978100, 25022574
25027852, 25028996, 25029022, 25029423, 25031502, 25032818, 25034396
25036006, 25036474, 25042823, 25044977, 25045228, 25050160, 25051465
25051628, 25054064, 25057811, 25058080, 25060506, 25062592, 25063971
25065563, 25072986, 25077278, 25078611, 25086233, 25087436, 25091141
25092777, 25093872, 25095982, 25098160, 25099339, 25099497, 25099758
25100063, 25100579, 25103996, 25107662, 25110233, 25114561, 25115178
25120284, 25120668, 25120742, 25121089, 25123585, 25124363, 25129925
25130312, 25140197, 25145163, 25145215, 25150925, 25159176, 25162645
25164293, 25166187, 25171041, 25171084, 25173124, 25175723, 25176408
25178032, 25178101, 25178179, 25179774, 25182817, 25184453, 25184555
25186079, 25189723, 25191872, 25192044, 25192528, 25192729, 25195901
25199585, 25200101, 25201454, 25202355, 25203656, 25205368, 25205954
25206864, 25207410, 25209912, 25210268, 25210499, 25210690, 25211628

25219450, 25223839, 25224242, 25225795, 25226665, 25227381, 25230870
25230945, 25237577, 25240188, 25240590, 25241448, 25241625, 25244807
25248384, 25250109, 25251648, 25257085, 25259611, 25262869, 25263960
25265499, 25269133, 25283790, 25287072, 25293659, 25296876, 25299227
25299807, 25300427, 25303284, 25303756, 25305405, 25307368, 25309116
25313154, 25313411, 25316758, 25317989, 25320555, 25323525, 25328093
25328518, 25329664, 25335249, 25335360, 25335790, 25337332, 25337640
25348567, 25348956, 25353983, 25356118, 25357142, 25360661, 25362958
25367588, 25367721, 25382812, 25383204, 25384462, 25386748, 25388573
25388896, 25392535, 25393714, 25395696, 25397936, 25398306, 25404117
25404202, 25405100, 25405687, 25405813, 25410017, 25410180, 25410802
25410877, 25411036, 25415713, 25416731, 25417050, 25417056, 25417958
25425005, 25425451, 25425760, 25427662, 25429959, 25430120, 25433696
25435038, 25437699, 25440818, 25442559, 25444961, 25445168, 25451531
25452452, 25455795, 25457409, 25459958, 25462714, 25463844, 25472112
25472885, 25476125, 25476149, 25477657, 25478885, 25479164, 25481087
25482971, 25486384, 25489342, 25489367, 25489607, 25492379, 25498930
25498994, 25516250, 25524955, 25528838, 25530080, 25530814, 25535668
25536819, 25537470, 25539063, 25540738, 25546580, 25546608, 25547901
25551676, 25553616, 25554787, 25555252, 25557886, 25558986, 25560487
25560538, 25561296, 25569149, 25569504, 25570929, 25573623, 25575348
25575369, 25575628, 25576115, 25579458, 25579761, 25591394, 25594901
25597525, 25598473, 25599425, 25600342, 25600421, 25601999, 25602488
25603923, 25606091, 25607726, 25612095, 25614866, 25616268, 25616359
25616417, 25616645, 25631933, 25633101, 25634317, 25634348, 25635149
25638456, 25639019, 25643818, 25643889, 25643931, 25646373, 25647325
25648731, 25653109, 25654459, 25654936, 25655390, 25655966, 25659655
25660847, 25661819, 25662088, 25662101, 25662524, 25663488, 25667973
25669791, 25670786, 25671354, 25672640, 25674386, 25680221, 25685152
2568739, 25687460, 25691904, 25694206, 25695903, 25696520, 25699321
25700654, 25709368, 25710420, 25715167, 25717371, 25722055, 25722608
25722720, 25723097, 25723158, 25728085, 25729507, 25730014, 25734963
25736747, 25739065, 25740844, 25741955, 25743479, 25747569, 25749273
25752755, 25754606, 25756945, 25757697, 25757748, 25760195, 25762221
25764020, 25766822, 25768681, 25772669, 25774077, 25775213, 25775444
25780343, 25783447, 25784002, 25785331, 25785441, 25788879, 25789041
25789277, 25789579, 25790353, 25792911, 25795865, 25797092, 25797124
25797305, 25800464, 25802510, 25803364, 25803545, 25807997, 25809524
25810263, 25810704, 25811105, 25811650, 25812390, 25813931, 25818707
25822410, 25823532, 25823754, 25824372, 25825910, 25826740, 25830492
25832935, 25834581, 25835365, 25838361, 25838755, 25852885, 25856821
25858672, 25861398, 25865785, 25866948, 25870579, 25871177, 25871639
25871753, 25872127, 25872389, 25873336, 25874050, 25874678, 25881255
25882264, 25883438, 25885148, 25888073, 25888984, 25890002, 25890046
25890056, 25890673, 25890782, 25894239, 25895224, 25897615, 25898228
25904273, 25904490, 25905130, 25906117, 25906886, 25908728, 25911724
25914276, 25919622, 25932524, 25932728, 25933494, 25941836, 25942868
25943271, 25945130, 25947799, 25951571, 25953857, 25954022, 25954054
25957038, 25963024, 25964954, 25967544, 25967985, 25970731, 25971286
25972417, 25973152, 25975723, 25977302, 25980605, 25980770, 25981498
25982666, 25986062, 25990907, 25995938, 25997810, 26006257, 26007010
26019148, 26023042, 26024732, 26024784, 26025681, 26029075, 26029777
26029780, 26032573, 26034119, 26036748, 26037215, 26038086, 26039623
26040483, 26045732, 26051656, 26078437, 26078493, 26080410, 26083298
26087754, 26088426, 26088836, 26089669, 26090767, 26090893, 26091640
26091786, 26095327, 26095405, 26096382, 26108080, 26108337, 26110259
26110632, 26111842, 26112621, 26115103, 26121990, 26124078, 26130486
26137367, 26137416, 26138085, 26145560, 26149904, 26153372, 26153977
26168933, 26169341, 26169345, 26170659, 26170715, 26176002, 26187943
26189861, 26198757, 26198926, 26201113, 26203182, 26223039, 26237338
26237431, 26237773, 26238195, 26242031, 26242677, 26243698, 26244115
26245237, 26248143, 26249718, 26256131, 26257953, 26259265, 26261327
26263328, 26263721, 26268756, 26269790, 26271001, 26274660, 26275023
26275415, 26277439, 26281476, 26285062, 26285933, 26301540, 26308650
26309047, 26317991, 26318200, 26318627, 26323308, 26324206, 26324769
26325856, 26327418, 26327624, 26327775, 26330994, 26331743, 26333141

26334602, 26336977, 26338953, 26351334, 26351996, 26353617, 26354844
26356098, 26358670, 26359091, 26362155, 26362821, 26366517, 26367012
26367460, 26371725, 26373967, 26374791, 26375052, 26375250, 26375330
26380097, 26385189, 26386858, 26388538, 26396790, 26398675, 26399626
26399691, 26399839, 26405036, 26406387, 26407408, 26410240, 26412540
26418088, 26420561, 26421667, 26422277, 26423085, 26426526, 26426967
26430323, 26430737, 26434436, 26434999, 26435073, 26436168, 26438612
26439748, 26440142, 26440169, 26440749, 26441345, 26442308, 26444601
26444887, 26446098, 26452606, 26474662, 26474703, 26475419, 26476090
26476244, 26478970, 26479173, 26482376, 26486365, 26492666, 26492866
26493289, 26498354, 26513067, 26513709, 26521043, 26522439, 26523432
26526726, 26526799, 26536320, 26537307, 26542135, 26542236, 26542835
26544823, 26545688, 26546070, 26546664, 26546754, 26548363, 26556014
26558437, 26569225, 26570134, 26575788, 26580633, 26582460, 26584641
26588069, 26597140, 26599395, 26608137, 26608238, 26609942, 26615291
26615690, 26617804, 26623652, 26626879, 26629381, 26633355, 26633558
26635897, 26637273, 26637824, 26639167, 26641610, 26641852, 26650226
26650540, 26654363, 26658759, 26659182, 26669550, 26680105, 26712331
26714486, 26714910, 26716835, 26717528, 26724511, 26725687, 26727397
26729494, 26729611, 26740700, 26743240, 26744595, 26745002, 26751106
26751171, 26755171, 26758193, 26764561, 26765212, 26768025, 26775602
26784509, 26790923, 26794786, 26797591, 26798411, 26798514, 26798516
26802503, 26816582, 26820076, 26822314, 26822620, 26824833, 26828994
26829845, 26830694, 26832296, 26833932, 26837569, 26837702, 26840654
26844406, 26844870, 26849779, 26855855, 26871815, 26875822, 26883456
26895149, 26896659, 26898563, 26907236, 26907327, 26908788, 26909100
26909504, 26910716, 26911000, 26923777, 26939314, 26943004, 26944190
26947373, 26958896, 26963310, 26966616, 26966916, 26967713, 26968670
26969321, 26970175, 26970717, 26981902, 26983259, 26985002, 26986173
26992964, 26999139, 27000158, 27000702, 27006120, 27006664, 27009164
27013146, 27015449, 27028251, 27032785, 27033520, 27033652, 27034890
27036163, 27037839, 27038986, 27039712, 27044169, 27044297, 27045634
27052607, 27056711, 27058530, 27060167, 27060859, 27061736, 27072923
27073314, 27079140, 27079651, 27084613, 27087426, 27090765, 27092508
27093423, 27097854, 27100800, 27101105, 27105900, 27106179, 27110878
27115422, 27117822, 27119621, 27119861, 27122162, 27124624, 27125872
27133662, 27134734, 27135647, 27135993, 27138325, 27142120, 27142373
27142529, 27144928, 27151826, 27153641, 27160922, 27161071, 27162390
27162405, 27163928, 27165231, 27169796, 27170305, 27181537, 27181897
27185188, 27195935, 27199245, 27200959, 27202015, 27203055, 27207110
27207634, 27208795, 27213224, 27216046, 27217412, 27223075, 27229389
27231051, 27234962, 27236722, 27242226, 27244337, 27244999, 27248917
27249531, 27250547, 27251690, 27254335, 27255377, 27256000, 27258578
27259307, 27262945, 27264464, 27266245, 27274456, 27274536, 27275533
27276231, 27283960, 27284499, 27285244, 27288638, 27292213, 27293599
27302711, 27302730, 27303287, 27303938, 27304410, 27304906, 27305039
27308088, 27314206, 27314390, 27314697, 27320576, 27321179, 27329612
27333106, 27334316, 27338912, 27338946, 27339115, 27345231, 27346709
27348081, 27349393, 27350267, 27351628, 27359178, 27364854, 27365014
27367194, 27369515, 27370965, 27375542, 27381498, 27383281, 27386467
27392968, 27393570, 27394703, 27395416, 27396624, 27396672, 27396813
27397048, 27400416, 27400598, 27404573, 27404668, 27405645, 27416997
27423251, 27424405, 27426363, 27432062, 27432826, 27433385, 27433870
27434193, 27439835, 27441326, 27442041, 27445727, 27452046, 27457891
27459593, 27459948, 27461740, 27466597, 27468303, 27486805, 27487919
27489107, 27493674, 27494663, 27501373, 27501413, 27502420, 27504770
27505229, 27508985, 27510959, 27525909, 27529661, 27533780, 27533819
27534509, 27539876, 27540613, 27544973, 27548131, 27554074, 27555481
27558861, 27560602, 27562488, 27565906, 27567477, 27576342, 27576354
27587905, 27588271, 27589260, 27593501, 27595973, 27601118, 27601441
27607563, 27611612, 27613080, 27613530, 27613554, 27615649, 27617978
27620808, 27623159, 27629756, 27629928, 27632114, 27634676, 27634991
27642235, 27645231, 27657712, 27658186, 27666312, 27671633, 27680669
27686599, 27687880, 27688036, 27688099, 27688692, 27691920, 27691939
27693416, 27693713, 27695063, 27698953, 27700466, 27704237, 27709046
27710072, 27717210, 27719000, 27726780, 27729678, 27739006, 27740424

27745728, 27748954, 27751755, 27757567, 27757888, 27758544, 27758653
27758972, 27759077, 27769361, 27779886, 27793533, 27799032, 27801337
27818389, 27819881, 27824540, 27824543, 27825241, 27828794, 27828892
27829295, 27833672, 27834551, 27834569, 27835925, 27837219, 27839353
27839616, 27846298, 27846499, 27847259, 27850112, 27855490, 27861226
27873412, 27882176, 27886087, 27897759, 27898015, 27902561, 27908396
27909478, 27927431, 27929287, 27929509, 27931299, 27935493, 27940876
27945870, 27951817, 27952586, 27959048, 27959594, 27964513, 27966472
27967484, 27983174, 27986817, 27994325, 27995215, 27995248, 27997875
27999003, 27999073, 27999638, 28000269, 28019592, 28022101, 28023081
28023399, 28023482, 28024793, 28025414, 28026866, 28033429, 28040776
28043157, 28045903, 28066655, 28067846, 28071549, 28072383, 28072567
28073470, 28074713, 28079127, 28090453, 28092783, 28098040, 28098160
28099662, 28104176, 28108003, 28111583, 28120036, 28120951, 28124631
28125601, 28125947, 28129791, 28140658, 28157786, 28164480, 28165439
28171079, 28174827, 28180464, 28181021, 28184554, 28188330, 28190796
28194173, 28199085, 28201419, 28204262, 28209985, 28215510, 28218832
28220398, 28223871, 28226179, 28229360, 28236305, 28238264, 28242712
28250929, 28256164, 28271119, 28276054, 28279837, 28281094, 28282606
28287484, 28290434, 28294563, 28302049, 28305001, 28305362, 28305607
28309406, 28319114, 28320399, 28330714, 28330971, 28350595, 28354603
28357401, 28361221, 28365111, 28369092, 28371123, 28373960, 28375383
28378446, 28384353, 28386259, 28388910, 28389153, 28390273, 28391210
28394726, 28396445, 28397317, 28401116, 28402823, 28420042, 28420457
28423598, 28432129, 28434028, 28435902, 28437315, 28439086, 28454215
28454242, 28468312, 28468493, 28481149, 28483184, 28489150, 28501075
28502098, 28502343, 28503038, 28507324, 28508053, 28508557, 28512336
28521330, 28522441, 28528349, 28530171, 28535127, 28535272, 28537715
28538439, 28542455, 28545134, 28546290, 28547068, 28547478, 28564479
28566241, 28571483, 28572407, 28572834, 28578164, 28578945, 28585411
28587723, 28589509, 28600233, 28602253, 28606598, 28608211, 28612674
28614372, 28617631, 28617959, 28621470, 28622202, 28627255, 28636676
28639299, 28642899, 28678804, 28690694, 28691965, 28692103, 28692275
28697526, 28697806, 28703812, 28708023, 28709063, 28710469, 28710734
28714988, 28715655, 28728272, 28730044, 28734355, 28740708, 28742555
28749289, 28749724, 28758090, 28758722, 28774416, 28776431, 28777174
28791725, 28797711, 28803345, 28808314, 28817449, 28819640, 28820669
28821847, 28827682, 28830691, 28831971, 28835937, 28836716, 28838066
28844866, 28847136, 28849751, 28852325, 28852691, 28855922, 28856060
28856172, 28863263, 28863487, 28867992, 28887305, 28889730, 28891984
28905390, 28910498, 28915870, 28927452, 28945922, 28948554, 28949888
28951026, 28951382, 28956908, 28959493, 28960211, 28965095, 28965787
28986231, 28986257, 28987439, 28991884, 28993295, 28993590, 29002488
29006527, 29007321, 29009513, 29013832, 29024054, 29026309
29026582, 29027694, 29032276, 29039510, 29040739, 29044086, 29044954
29048498, 29048728, 29050886, 29060216, 29061016, 29115857, 29125374
29154725, 29158680, 29163567, 29170232, 29173817, 29179097, 29182517
29182901, 29189889, 29198092, 29200700, 29203604, 29213320, 29213351
29213893, 29224605, 29224710, 29237575, 29247712, 29249289, 29250230
29250317, 29254623, 29260956, 29278684, 29296257, 29301463, 29307638
29312889, 29337294, 29338348, 29339155, 29343086, 29343156, 29343861
29347943, 29353821, 29372069, 29372460, 29375355, 29375984, 29376346
29378913, 29379978, 29383695, 29388020, 29398488, 29399336, 29405462
29409149, 29409455, 29418165, 29420254, 29426241, 29434301, 29436454
29437712, 29450812, 29452251, 29454978, 29463047, 29464779, 29472618
29477015, 29483626, 29483672, 29483723, 29483771, 29500257, 29500963
29501218, 29504682, 29511611, 29524985, 29530515, 29536342, 29538631
29542449, 29542580, 29548592, 29549071, 29559395, 29564592, 29580394
29591343, 29608023, 29614575, 29614987, 29621961, 29625065, 29626154
29629430, 29633753, 29637526, 29637560, 29645349, 29651520, 29656843
29667994, 29676089, 29678163, 29685137, 29687220, 29688867, 29690625
29703195, 29705793, 29707896, 29717901, 29719146, 29724063, 29726695
29741319, 29766435, 29767177, 29769901, 29774362, 29782211, 29791152
29794174, 29794462, 29807964, 29813494, 29815341, 29817278, 29822714
29825525, 29836659, 29841687, 29844131, 29846645, 29853485, 29865188
29869404, 29869906, 29875459, 29876358, 29881050, 29881575, 29884958

29893132, 29902299, 29902311, 29914449, 29930457, 29941062, 29942554
29944035, 29944159, 29944660, 29951620, 29951759, 29961353, 29962927
29962939, 29965888, 29991257, 29997937, 30008125, 30018017, 30018903
30031027, 30039959, 30064268, 30068871, 30076253, 30078934, 30086166
30088912, 30092280, 30098251, 30099302, 30114477, 30116203, 30120608
30125995, 30131286, 30139392, 30147928, 30150731, 30160625, 30163243
30164714, 30173113, 30177597, 30179644, 30186706, 30189023, 30193736
30196358, 30200680, 30200758, 30215130, 30218044, 30218317, 30223712
30225443, 30239480, 30241567, 30244787, 30246179, 30247305, 30252098
30252156, 30253255, 30265523, 30265615, 30272329, 30281591, 30282501
30283932, 30293345, 30305880, 30312094, 30312568, 30316897, 30320029
30325407, 30331356, 30342878, 30352623, 30355490, 30357897, 30364613
30365745, 30368482, 30368668, 30372081, 30374739, 30381207, 30384121
30384152, 30387666, 30391272, 30397100, 30402386, 30403763, 30408515
30413137, 30416034, 30421204, 30431274, 30441687, 30443393, 30453442
30458593, 30460922, 30464250, 30464655, 30473634, 30474774, 30475115
30476768, 30485255, 30496957, 30497057, 30498824, 30501574, 30503943
30509277, 30510527, 30517516, 30522998, 30528547, 30528704, 30532811
30533198, 30534662, 30578221, 30581448, 30582500, 30606345, 30613937
30623138, 30624864, 30635302, 30652853, 30654409, 30662736, 30668407
30671813, 30679595, 30679771, 30681462, 30698289, 30741263, 30749644
30755348, 30758943, 30783551, 30803210, 30814266, 30814285, 30815852
30816938, 30855101, 30856358, 30866988, 30887501, 30904672, 30905638
30914674, 30919804, 30922870, 30937340, 30964194, 30968208, 30980615
30985027, 30987088, 30994996, 30998759, 31001455, 31004719, 31013127
31022858, 31028986, 31061482, 31100172, 31104809, 31106577, 31109506
31115502, 31156383, 31172207, 31182793, 31192039, 31194264, 31200845
31201001, 31215438, 31228670, 31254535, 31258101, 31302499, 31306248
31306261, 31309867, 31315876, 31326608, 31331354, 31335037, 31335142
31341859, 31343752, 31347532, 31393600, 31430722, 31508450, 31544097
31570161, 31600023, 31658464, 31668061, 31668872, 31718134, 31771858
31867037, 31905033

Version 12.2.0.1.ru-2020-07.rur-2020-07.r1

Version 12.2.0.1.ru-2020-07.rur-2020-07.r1 includes the following:

- Patch 31312468: Database Jul 2020 Release Update 12.2.0.1.200714
- Patch 31219919: OJVM RELEASE UPDATE: 12.2.0.1.200714
- Patch 31335037: DSTV35 for RDBMS (TZDATA2020A)
- Patch 31335142: DSTV35 for OJVM (TZDATA2020A)
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- PreUpgrade Jar: preupgrade_12201_cbuild_23_lf.zip

Combined patches for version 12.2.0.1.ru-2020-07.rur-2020-07.r1, released July 2020

Bugs fixed:

7391838, 8480838, 8932139, 8975044, 12763598, 13554903, 14221306
14690846, 15931756, 16002385, 16438495, 16727454, 16942578, 17027695
17533661, 17947871, 18308268, 18521691, 18594510, 18774543, 18878420
18986501, 19072655, 19211433, 19285025, 19327292, 19526548, 19614243
19647894, 19649997, 19702201, 19721304, 20003668, 20087519, 20118035
20120236, 20324049, 20436508, 20532077, 20549013, 20588486, 20591151
20617383, 20620169, 20736227, 20756305, 20866970, 20976443, 21070321

21089435, 21095391, 21143725, 21147908, 21159907, 21178363, 21186167
21197098, 21216226, 21320338, 21433452, 21479706, 21517767, 21520266
21547051, 21638600, 21744603, 21788462, 21837606, 21882528, 21935698
21981529, 21985256, 22007324, 22070226, 22070473, 22070853, 22072543
22087683, 22104866, 22107360, 22174392, 22179537, 22282748, 22310426
22347493, 22363790, 22364044, 22367053, 22379010, 22446455, 22454940
22495673, 22503283, 22503297, 22504793, 22522515, 22530986, 22564336
22568728, 22581771, 22594071, 22599050, 22628825, 22645009, 22645496
22654475, 22700845, 22726044, 22729345, 22826067, 22843979, 22845846
22864303, 22898198, 22921674, 22939829, 22950945, 22970869, 22981722
23018676, 23019710, 23026585, 23035249, 23055900, 23056058, 23061453
23065002, 23066146, 23080557, 23104033, 23105538, 23109325, 23110523
23125560, 23126545, 23127945, 23143074, 23151677, 23168363, 23169712
23177923, 23179662, 23184263, 23197730, 23234232, 23237091, 23249829
23271203, 23278750, 23281269, 23300142, 23306590, 23308065, 23310101
23312077, 23328639, 23333567, 23336559, 23342170, 23481673, 23491861
23499004, 23499160, 23521523, 23527363, 23533647, 23548817, 23567857
23572982, 23581777, 23588722, 23599216, 23600861, 23602213, 23614158
23645516, 23665623, 23709062, 23715460, 23715518, 23730961, 23733981
23735292, 23738304, 23741944, 23743596, 23746128, 23749454, 23761724
24006569, 24010030, 24289874, 24289895, 24294174, 24303148, 24307571
24308349, 24326444, 24326846, 24328857, 24330708, 24332831, 24334708
24336249, 24337882, 24341675, 24343905, 24345420, 24346821, 24348685
24350620, 24352981, 24355111, 24357348, 24368004, 24371491, 24373528
24373756, 24374976, 24376875, 24376878, 24383086, 24385983, 24401351
24403922, 24409977, 24415926, 24416451, 24421668, 24423416, 24425056
24425998, 24435982, 24437162, 24440648, 24443539, 24457597, 24460392
24461826, 24467122, 24468470, 24470606, 24471079, 24471473, 24473736
24484749, 24485034, 24485161, 24485174, 24485619, 24486059, 24486237
24509056, 24516314, 24530364, 24534401, 24554533, 24555417, 24556862
24556967, 24560906, 24563422, 24570214, 24570598, 24573817, 24578718
24578797, 24588377, 24589081, 24589590, 24591506, 24593740, 24595699
24596874, 24600330, 24609592, 24609996, 24611527, 24616637, 24617969
24623975, 24624166, 24642495, 24654629, 24655717, 24664211, 24668398
24669189, 24669730, 24674197, 24674955, 24676172, 24677696, 24680959
24687075, 24689376, 24692973, 24693010, 24693290, 24697323, 24699619
24701840, 24710696, 24713381, 24714096, 24717183, 24717859, 24718260
24719799, 24735430, 24737064, 24737403, 24737581, 24737954, 24739791
24744383, 24744686, 24752618, 24757934, 24759556, 24760407, 24764085
24766309, 24784414, 24786669, 24791883, 24792678, 24793511, 24796092
24797119, 24798481, 24800423, 24801152, 24802934, 24808504, 24811725
24812047, 24818566, 24827228, 24827654, 24831514, 24835919, 24841671
24843188, 24844549, 24844841, 24845157, 24848746, 24848923, 24850622
24907917, 24908063, 24908321, 24911709, 24912588, 24920582, 24921478
24922704, 24923080, 24923215, 24923338, 24923790, 24924667, 24926999
24929210, 24938784, 24940060, 24942749, 24953434, 24957555, 24960044
24960809, 24965426, 24966594, 24966788, 24967993, 24968162, 24976007
24978100, 25022574, 25027852, 25028996, 25029022, 25029423, 25031502
25032818, 25034396, 25036006, 25036474, 25042823, 25044977, 25045228
25050160, 25051465, 25051628, 25054064, 25057811, 25058080, 25060506
25062592, 25063971, 25065563, 25072986, 25077278, 25078611, 25086233
25087436, 25091141, 25092777, 25093872, 25095982, 25098160, 25099339
25099497, 25099758, 25100063, 25100579, 25103996, 25107662, 25110233
25114561, 25120284, 25120668, 25120742, 25121089, 25123585, 25124363
25129925, 25130312, 25140197, 25145163, 25145215, 25150925, 25159176
25162645, 25164293, 25166187, 25171041, 25171084, 25173124, 25175723
25176408, 25178032, 25178101, 25178179, 25179774, 25182817, 25184453
25184555, 25186079, 25189723, 25191872, 25192044, 25192528, 25192729
25195901, 25199585, 25200101, 25201454, 25202355, 25203656, 25205954
25206864, 25207410, 25209912, 25210268, 25210499, 25210690, 25211628
25219450, 25223839, 25224242, 25225795, 25226665, 25227381, 25230870
25230945, 25237577, 25240188, 25240590, 25241448, 25241625, 25244807
25248384, 25250109, 25251648, 25257085, 25259611, 25262869, 25263960
25265499, 25269133, 25283790, 25287072, 25293659, 25296876, 25299227
25299807, 25300427, 25303284, 25303756, 25305405, 25307368, 25309116
25313154, 25313411, 25316758, 25317989, 25320555, 25323525, 25328093

25328518, 25329664, 25335249, 25335360, 25335790, 25337332, 25337640
25348956, 25353983, 25356118, 25357142, 25360661, 25362958, 25367588
25367721, 25382812, 25383204, 25384462, 25386748, 25388573, 25388896
25392535, 25393714, 25395696, 25397936, 25398306, 25404202, 25405100
25405687, 25405813, 25410017, 25410180, 25410802, 25410877, 25411036
25415713, 25416731, 25417050, 25417056, 25417958, 25425005, 25425451
25425760, 25427662, 25429959, 25430120, 25433696, 25435038, 25437699
25440818, 25442559, 25444961, 25445168, 25451531, 25452452, 25455795
25457409, 25459958, 25462714, 25463844, 25472112, 25472885, 25476125
25476149, 25477657, 25478885, 25479164, 25481087, 25482971, 25486384
25489342, 25489367, 25489607, 25492379, 25498930, 25498994, 25516250
25524955, 25528838, 25530080, 25530814, 25535668, 25536819, 25537470
25539063, 25540738, 25546580, 25546608, 25547901, 25551676, 25553616
25554787, 25555252, 25557886, 25558986, 25560487, 25560538, 25561296
25569149, 25570929, 25573623, 25575348, 25575369, 25575628, 25576115
25579458, 25579761, 25591394, 25594901, 25597525, 25598473, 25599425
25600342, 25600421, 25601999, 25602488, 25603923, 25606091, 25607726
25612095, 25614866, 25616268, 25616359, 25616417, 25616645, 25631933
25633101, 25634317, 25634348, 25635149, 25638456, 25639019, 25643818
25643889, 25643931, 25646373, 25647325, 25648731, 25653109, 25654459
25654936, 25655390, 25655966, 25659655, 25660847, 25661819, 25662088
25662101, 25662524, 25663488, 25667973, 25669791, 25670786, 25671354
25672640, 25674386, 25680221, 25685152, 25686739, 25687460, 25691904
25694206, 25695903, 25696520, 25699321, 25700654, 25709368, 25710420
25715167, 25717371, 25722055, 25722608, 25722720, 25723097, 25723158
25728085, 25729507, 25730014, 25734963, 25736747, 25739065, 25740844
25741955, 25743479, 25747569, 25749273, 25752755, 25754606, 25756945
25757697, 25757748, 25760195, 25762221, 25764020, 25766822, 25768681
25772669, 25774077, 25775213, 25775444, 25780343, 25783447, 25784002
25785331, 25785441, 25788879, 25789041, 25789277, 25789579, 25790353
25792911, 25795865, 25797092, 25797124, 25797305, 25800464, 25802510
25803364, 25803545, 25807997, 25809524, 25810263, 25810704, 25811105
25811650, 25812390, 25813931, 25818707, 25822410, 25823532, 25823754
25824372, 25825910, 25826740, 25830492, 25832935, 25834581, 25835365
25838361, 25838755, 25852885, 25856821, 25858672, 25861398, 25865785
25866948, 25870579, 25871177, 25871639, 25871753, 25872127, 25872389
25873336, 25874050, 25874678, 25881255, 25882264, 25883438, 25885148
25888073, 25888984, 25890002, 25890046, 25890056, 25890673, 25890782
25894239, 25895224, 25897615, 25898228, 25904273, 25904490, 25905130
25906117, 25906886, 25908728, 25911724, 25914276, 25919622, 25932524
25932728, 25933494, 25941836, 25942868, 25943271, 25945130, 25947799
25951571, 25953857, 25954022, 25954054, 25957038, 25963024, 25964954
25967544, 25967985, 25970731, 25971286, 25972417, 25973152, 25975723
25977302, 25980605, 25980770, 25981498, 25982666, 25986062, 25990907
25995938, 25997810, 26006257, 26007010, 26019148, 26023042, 26024732
26024784, 26025681, 26029075, 26029777, 26029780, 26032573, 26034119
26036748, 26037215, 26038086, 26039623, 26040483, 26045732, 26051656
26078437, 26078493, 26080410, 26083298, 26087754, 26088426, 26088836
26089669, 26090767, 26090893, 26091640, 26091786, 26095327, 26095405
26096382, 26108080, 26108337, 26110259, 26110632, 26111842, 26112621
26115103, 26121990, 26124078, 26130486, 26137367, 26137416, 26138085
26145560, 26149904, 26153372, 26153977, 26168933, 26169341, 26169345
26170659, 26170715, 26176002, 26187943, 26189861, 26198757, 26198926
26201113, 26203182, 26223039, 26237338, 26237431, 26237773, 26238195
26242031, 26242677, 26243698, 26244115, 26245237, 26248143, 26249718
26256131, 26257953, 26259265, 26261327, 26263328, 26263721, 26268756
26269790, 26271001, 26274660, 26275023, 26275415, 26277439, 26281476
26285062, 26285933, 26308650, 26309047, 26317991, 26318627, 26323308
26324206, 26324769, 26325856, 26327418, 26327624, 26327775, 26330994
26331743, 26333141, 26334602, 26336977, 26338953, 26351334, 26351996
26353617, 26354844, 26356098, 26358670, 26359091, 26362155, 26362821
26366517, 26367012, 26367460, 26371725, 26373967, 26374791, 26375052
26375250, 26375330, 26380097, 26385189, 26386858, 26388538, 26396790
26398675, 26399626, 26399691, 26405036, 26406387, 26407408, 26410240
26412540, 26418088, 26420561, 26421667, 26422277, 26423085, 26426526
26426967, 26430323, 26430737, 26434436, 26434999, 26435073, 26436168

26438612, 26439748, 26440142, 26440169, 26440749, 26441345, 26442308
26444601, 26444887, 26446098, 26452606, 26474662, 26474703, 26475419
26476090, 26476244, 26478970, 26479173, 26482376, 26486365, 26492866
26493289, 26498354, 26513067, 26513709, 26521043, 26522439, 26523432
26526726, 26526799, 26536320, 26537307, 26542135, 26542236, 26542835
26544823, 26545688, 26546070, 26546664, 26546754, 26548363, 26556014
26558437, 26569225, 26570134, 26575788, 26580633, 26582460, 26584641
26588069, 26597140, 26599395, 26608137, 26608238, 26609942, 26615291
26615690, 26617804, 26623652, 26626879, 26629381, 26633355, 26633558
26635897, 26637273, 26637824, 26639167, 26641610, 26641852, 26650226
26650540, 26654363, 26658759, 26659182, 26669550, 26680105, 26712331
26714486, 26714910, 26717528, 26724511, 26725687, 26727397, 26729494
26729611, 26740700, 26743240, 26744595, 26745002, 26751106, 26751171
26755171, 26758193, 26764561, 26765212, 26768025, 26775602, 26784509
26790923, 26794786, 26797591, 26798411, 26798514, 26798516, 26802503
26816582, 26820076, 26822314, 26822620, 26824833, 26828994, 26829845
26830694, 26832296, 26833932, 26837569, 26837702, 26840654, 26844406
26844870, 26849779, 26871815, 26875822, 26883456, 26895149, 26896659
26898563, 26907236, 26907327, 26908788, 26909100, 26909504, 26910716
26911000, 26923777, 26939314, 26943004, 26944190, 26947373, 26958896
26963310, 26966616, 26966916, 26967713, 26968670, 26969321, 26970175
26970717, 26981902, 26983259, 26985002, 26986173, 26992964, 26999139
27000158, 27000702, 27006120, 27006664, 27009164, 27013146, 27015449
27028251, 27032785, 27033520, 27033652, 27034890, 27036163, 27037839
27038986, 27039712, 27044169, 27044297, 27045634, 27052607, 27056711
27058530, 27060167, 27060859, 27061736, 27072923, 27073314, 27079140
27084613, 27087426, 27090765, 27092508, 27093423, 27097854, 27100800
27101105, 27105900, 27106179, 27110878, 27115422, 27117822, 27119621
27119861, 27122162, 27124624, 27125872, 27133662, 27134734, 27135647
27135993, 27138325, 27142120, 27142373, 27142529, 27144928, 27151826
27153641, 27160922, 27161071, 27162390, 27162405, 27163928, 27165231
27169796, 27170305, 27181537, 27181897, 27185188, 27195935, 27199245
27200959, 27202015, 27203055, 27207110, 27207634, 27208795, 27213224
27216046, 27217412, 27223075, 27229389, 27231051, 27234962, 27236722
27242226, 27244337, 27244999, 27248917, 27249531, 27250547, 27251690
27254335, 27255377, 27256000, 27258578, 27259307, 27262945, 27264464
27266245, 27274456, 27274536, 27275533, 27276231, 27283960, 27284499
27285244, 27288638, 27292213, 27293599, 27302711, 27302730, 27303287
27303938, 27304410, 27304906, 27305039, 27308088, 27314206, 27314390
27314697, 27320576, 27321179, 27329612, 27333106, 27334316, 27338912
27338946, 27339115, 27345231, 27346709, 27348081, 27349393, 27350267
27351628, 27359178, 27364854, 27365014, 27367194, 27369515, 27370965
27375542, 27381498, 27383281, 27386467, 27392968, 27393570, 27394703
27395416, 27396624, 27396672, 27396813, 27397048, 27400416, 27400598
27404573, 27404668, 27405645, 27416997, 27423251, 27424405, 27426363
27432062, 27432826, 27433385, 27433870, 27434193, 27439835, 27441326
27442041, 27445727, 27457891, 27459593, 27459948, 27461740, 27466597
27468303, 27486805, 27487919, 27489107, 27493674, 27494663, 27501373
27501413, 27502420, 27504770, 27505229, 27508985, 27510959, 27525909
27529661, 27533780, 27533819, 27534509, 27539876, 27540613, 27544973
27548131, 27554074, 27555481, 27558861, 27560602, 27562488, 27565906
27567477, 27576342, 27576354, 27587905, 27588271, 27589260, 27593501
27595973, 27601118, 27601441, 27607563, 27611612, 27613080, 27613530
27613554, 27615649, 27617978, 27620808, 27623159, 27629756, 27629928
27632114, 27634676, 27634991, 27642235, 27645231, 27657712, 27658186
27666312, 27671633, 27680669, 27686599, 27687880, 27688036, 27688099
27688692, 27691920, 27691939, 27693416, 27693713, 27695063, 27698953
27700466, 27704237, 27709046, 27710072, 27719000, 27726780, 27729678
27739006, 27740424, 27748954, 27751755, 27757567, 27757888, 27758544
27758653, 27758972, 27759077, 27769361, 27779886, 27793533, 27799032
27801337, 27818389, 27819881, 27824540, 27824543, 27825241, 27828794
27828892, 27829295, 27833672, 27834551, 27834569, 27835925, 27839353
27839616, 27846298, 27846499, 27847259, 27850112, 27855490, 27861226
27873412, 27882176, 27886087, 27897759, 27898015, 27902561, 27908396
27909478, 27927431, 27929287, 27929509, 27931299, 27935493, 27940876
27945870, 27951817, 27952586, 27959048, 27959594, 27964513, 27966472

27967484, 27983174, 27986817, 27994325, 27995215, 27995248, 27997875
27998003, 27999073, 27999638, 28000269, 28019592, 28022101, 28023081
28023399, 28023482, 28024793, 28025414, 28026866, 28033429, 28040776
28043157, 28045903, 28067846, 28071549, 28072383, 28072567, 28073470
28074713, 28079127, 28090453, 28092783, 28098040, 28098160, 28099662
28104176, 28108003, 28111583, 28120036, 28120951, 28124631, 28125601
28125947, 28129791, 28140658, 28157786, 28164480, 28165439, 28171079
28174827, 28180464, 28181021, 28184554, 28188330, 28190796, 28194173
28199085, 28201419, 28204262, 28209985, 28215510, 28218832, 28220398
28223871, 28226179, 28229360, 28236305, 28238264, 28242712, 28250929
28256164, 28271119, 28276054, 28279837, 28281094, 28282606, 28287484
28290434, 28294563, 28302049, 28305001, 28305362, 28305607, 28309406
28319114, 28320399, 28330714, 28330971, 28350595, 28354603, 28357401
28361221, 28365111, 28369092, 28371123, 28373960, 28375383, 28378446
28384353, 28386259, 28388910, 28390273, 28391210, 28396445, 28397317
28401116, 28402823, 28420042, 28420457, 28423598, 28432129, 28434028
28435902, 28437315, 28439086, 28454215, 28454242, 28468312, 28468493
28481149, 28483184, 28489150, 28501075, 28502098, 28502343, 28503038
28507324, 28508053, 28508557, 28512336, 28521330, 28522441, 28528349
28530171, 28535127, 28535272, 28537715, 28538439, 28542455, 28545134
28546290, 28547068, 28547478, 28564479, 28566241, 28571483, 28572407
28572834, 28578164, 28578945, 28585411, 28587723, 28589509, 28600233
28606598, 28608211, 28612674, 28614372, 28617631, 28617959, 28621470
28622202, 28627255, 28636676, 28639299, 28642899, 28678804, 28691965
28692103, 28692275, 28697806, 28703812, 28708023, 28709063, 28710469
28714988, 28715655, 28728272, 28734355, 28740708, 28742555, 28749289
28749724, 28758090, 28758722, 28774416, 28776431, 28777174, 28791725
28797711, 28803345, 28808314, 28817449, 28819640, 28820669, 28821847
28827682, 28830691, 28831971, 28835937, 28836716, 28838066, 28844866
28847136, 28849751, 28852325, 28852691, 28855922, 28856060, 28856172
28863263, 28863487, 28867992, 28887305, 28889730, 28891984, 28905390
28910498, 28915870, 28927452, 28949888, 28951026, 28951382, 28956908
28959493, 28960211, 28965095, 28965787, 28986231, 28986257, 28987439
28991884, 28993295, 28993590, 29002488, 29006527, 29007321, 29007353
29009513, 29013832, 29024054, 29026582, 29027694, 29032276, 29039510
29040739, 29044954, 29048498, 29048728, 29050886, 29060216, 29061016
29115857, 29125374, 29158680, 29163567, 29170232, 29173817, 29179097
29182517, 29182901, 29189889, 29198092, 29200700, 29203604, 29213320
29213893, 29224605, 29237575, 29247712, 29249289, 29250230, 29250317
29254623, 29260956, 29278684, 29296257, 29301463, 29307638, 29312889
29339155, 29343086, 29343156, 29343861, 29347943, 29353821, 29372069
29372460, 29375355, 29375984, 29376346, 29378913, 29379978, 29383695
29388020, 29398488, 29399336, 29405462, 29409149, 29409455, 29420254
29426241, 29434301, 29436454, 29437712, 29450812, 29452251, 29454978
29464779, 29483626, 29483672, 29483723, 29483771, 29500257, 29500963
29501218, 29504682, 29511611, 29524985, 29530515, 29536342, 29538631
29542449, 29542580, 29548592, 29549071, 29559395, 29564592, 29580394
29591343, 29608023, 29614575, 29614987, 29621961, 29625065, 29626154
29629430, 29633753, 29637526, 29637560, 29645349, 29651520, 29656843
29667994, 29676089, 29678163, 29685137, 29687220, 29688867, 29690625
29703195, 29705793, 29707896, 29717901, 29719146, 29724063, 29726695
29766435, 29767177, 29774362, 29782211, 29791152, 29794462, 29807964
29813494, 29815341, 29817278, 29822714, 29825525, 29836659, 29841687
29846645, 29853485, 29865188, 29869404, 29869906, 29875459, 29876358
29881050, 29881575, 29884958, 29893132, 29902299, 29902311, 29914449
29930457, 29944035, 29944660, 29951620, 29951759, 29961353, 29962927
29962939, 29965888, 29991257, 29997937, 30008125, 30018017, 30018903
30031027, 30039959, 30064268, 30068871, 30076253, 30078934, 30086166
30088912, 30092280, 30098251, 30099302, 30114477, 30116203, 30120608
30125995, 30131286, 30139392, 30147928, 30150731, 30160625, 30163243
30164714, 30173113, 30177597, 30179644, 30186706, 30189023, 30193736
30196358, 30200680, 30200758, 30215130, 30218044, 30218317, 30223712
30225443, 30239480, 30241567, 30244787, 30246179, 30247305, 30252098
30252156, 30253255, 30265523, 30272329, 30281591, 30282501, 30283932
30293345, 30305880, 30312094, 30312568, 30316897, 30325407, 30342878
30352623, 30355490, 30357897, 30364613, 30365745, 30368482, 30372081

30374739, 30381207, 30384121, 30384152, 30391272, 30397100, 30402386
30403763, 30408515, 30413137, 30416034, 30431274, 30441687, 30443393
30453442, 30458593, 30460922, 30464655, 30474774, 30475115, 30476768
30485255, 30496957, 30497057, 30501574, 30503943, 30509277, 30510527
30522998, 30528547, 30528704, 30533198, 30534662, 30581448, 30582500
30613937, 30623138, 30624864, 30635302, 30652853, 30654409, 30668407
30671813, 30681462, 30698289, 30741263, 30758943, 30803210, 30814285
30815852, 30816938, 30855101, 30887501, 30904672, 30922870, 30964194
30968208, 30980615, 30987088, 30998759, 31001455, 31004719, 31013127
31022858, 31100172, 31106577, 31156383, 31172207, 31182793, 31200845
31306261, 31335037, 31335142, 31341859, 31393600

Version 12.2.0.1.ru-2020-04.rur-2020-04.r1

Version 12.2.0.1.ru-2020-04.rur-2020-04.r1 includes the following:

- Patch 30886680: Database Apr 2020 Release Update 12.2.0.1.200414
- Patch 30805580: Oracle JVM Release Update 12.2.0.1.200414
- Patch 29997937: DSTv34 for RDBMS (TZDATA2019G)
- Patch 29997959: DSTv34 for OJVM (TZDATA2019G)
- PreUpgrade Jar: preupgrade_12201_cbuild_23_lf.zip
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR is included in DB PATCH 30138470
- Support for [Purging the recycle bin \(p. 1062\)](#)
- Support for [Generating performance reports with Automatic Workload Repository \(AWR\) \(p. 1053\)](#) using the `rdsadmin.rdsadmin_diagnostic_util` package

Combined patches for version 12.2.0.1.ru-2020-04.rur-2020-04.r1, released April 2020

Bugs fixed:

30533198, 30312568, 30086166, 28951026, 28435902, 8480838, 8932139
12763598, 13554903, 14221306, 14690846, 15931756, 16002385, 16438495
16727454, 16942578, 17027695, 17533661, 17947871, 18308268, 18521691
18594510, 18774543, 18878420, 19072655, 19211433, 19285025, 19327292
19526548, 19614243, 19647894, 19649997, 19702201, 19721304, 20003668
20087519, 20118035, 20120236, 20324049, 20436508, 20532077, 20549013
20588486, 20591151, 20617383, 20620169, 20736227, 20756305, 20866970
20976443, 21089435, 21095391, 21143725, 21147908, 21159907, 21178363
21186167, 21197098, 21216226, 21320338, 21433452, 21479706, 21517767
21520266, 21547051, 21638600, 21744603, 21788462, 21837606, 21882528
21935698, 21981529, 21985256, 22007324, 22070473, 22070853, 22072543
22087683, 22104866, 22107360, 22174392, 22179537, 22282748, 22310426
22347493, 22363790, 22364044, 22367053, 22379010, 22446455, 22454940
22495673, 22503283, 22503297, 22504793, 22522515, 22530986, 22564336
22568728, 22581771, 22594071, 22599050, 22628825, 22645009, 22645496
22654475, 22700845, 22726044, 22729345, 22826067, 22843979, 22845846
22864303, 22898198, 22921674, 22939829, 22950945, 22970869, 22981722
23018676, 23019710, 23026585, 23035249, 23055900, 23061453, 23065002
23066146, 23080557, 23104033, 23105538, 23109325, 23110523, 23125560
23126545, 23127945, 23143074, 23151677, 23168363, 23169712, 23177923
23179662, 23184263, 23197730, 23234232, 23237091, 23249829, 23271203
23278750, 23300142, 23306590, 23308065, 23310101, 23312077, 23328639
23333567, 23336559, 23342170, 23481673, 23491861, 23499004, 23499160
23521523, 23527363, 23533647, 23548817, 23567857, 23572982, 23581777

23588722, 23599216, 23600861, 23602213, 23645516, 23665623, 23709062
23715460, 23715518, 23730961, 23733981, 23735292, 23738304, 23741944
23743596, 23746128, 23749454, 23761724, 24006569, 24010030, 24289874
24289895, 24294174, 24303148, 24307571, 24308349, 24326444, 24326846
24328857, 24330708, 24332831, 24334708, 24336249, 24337882, 24341675
24343905, 24345420, 24346821, 24348685, 24350620, 24352981, 24355111
24357348, 24368004, 24371491, 24373528, 24373756, 24374976, 24376875
24376878, 24383086, 24385983, 24401351, 24403922, 24409977, 24415926
24416451, 24421668, 24423416, 24425056, 24425998, 24435982, 24437162
24440648, 24443539, 24457597, 24460392, 24461826, 24467122, 24468470
24470606, 24471079, 24471473, 24473736, 24485034, 24485161, 24485174
24486059, 24486237, 24509056, 24516314, 24530364, 24534401, 24554533
24555417, 24556862, 24556967, 24560906, 24563422, 24570214, 24570598
24573817, 24578718, 24578797, 24588377, 24589081, 24589590, 24591506
24593740, 24595699, 24600330, 24609592, 24609996, 24611527, 24616637
24617969, 24623975, 24624166, 24642495, 24654629, 24655717, 24664211
24668398, 24669189, 24674197, 24674955, 24676172, 24677696, 24680959
24687075, 24689376, 24692973, 24693010, 24693290, 24697323, 24699619
24710696, 24713381, 24714096, 24717183, 24717859, 24718260, 24719799
24735430, 24737064, 24737403, 24737581, 24737954, 24739791, 24744383
24744686, 24752618, 24757934, 24759556, 24760407, 24764085, 24766309
24784414, 24786669, 24791883, 24792678, 24793511, 24796092, 24797119
24800423, 24801152, 24802934, 24808504, 24811725, 24812047, 24818566
24827228, 24827654, 24831514, 24835919, 24841671, 24843188, 24844549
24844841, 24845157, 24848746, 24848923, 24850622, 24907917, 24908321
24911709, 24912588, 24920582, 24921478, 24922704, 24923080, 24923215
24923338, 24923790, 24924667, 24926999, 24929210, 24938784, 24940060
24942749, 24953434, 24957555, 24960044, 24960809, 24965426, 24966594
24966788, 24967993, 24968162, 24976007, 24978100, 25022574, 25027852
25028996, 25029022, 25029423, 25031502, 25032818, 25034396, 25036006
25036474, 25042823, 25044977, 25045228, 25050160, 25051465, 25051628
25054064, 25057811, 25058080, 25060506, 25062592, 25063971, 25065563
25072986, 25077278, 25078611, 25086233, 25087436, 25092777, 25093872
25095982, 25098160, 25099339, 25099497, 25099758, 25100063, 25100579
25103996, 25107662, 25110233, 25114561, 25120284, 25120668, 25120742
25121089, 25123585, 25124363, 25129925, 25130312, 25140197, 25145163
25145215, 25150925, 25159176, 25162645, 25164293, 25166187, 25171041
25171084, 25175723, 25176408, 25178032, 25178101, 25178179, 25179774
25182817, 25184453, 25184555, 25186079, 25191872, 25192044, 25192528
25192729, 25195901, 25199585, 25201454, 25202355, 25203656, 25205954
25206864, 25207410, 25209912, 25210268, 25210499, 25210690, 25211628
25219450, 25223839, 25224242, 25225795, 25226665, 25227381, 25230870
25230945, 25237577, 25240188, 25240590, 25241448, 25241625, 25244807
25248384, 25250109, 25251648, 25257085, 25259611, 25262869, 25263960
25265499, 25269133, 25283790, 25287072, 25293659, 25296876, 25299227
25299807, 25300427, 25303284, 25303756, 25305405, 25307368, 25309116
25313154, 25313411, 25316758, 25317989, 25320555, 25323525, 25328093
25328518, 25329664, 25335249, 25335360, 25335790, 25337332, 25337640
25348956, 25353983, 25356118, 25357142, 25360661, 25362958, 25367588
25367721, 25382812, 25383204, 25384462, 25386748, 25388896, 25392535
25393714, 25395696, 25397936, 25398306, 25404202, 25405100, 25405687
25405813, 25410017, 25410180, 25410802, 25410877, 25411036, 25415713
25416731, 25417050, 25417056, 25417958, 25425005, 25425451, 25425760
25427662, 25429959, 25430120, 25433696, 25435038, 25437699, 25440818
25442559, 25444961, 25445168, 25451531, 25455795, 25457409, 25459958
25462714, 25463844, 25472112, 25472885, 25476125, 25476149, 25477657
25478885, 25479164, 25481087, 25482971, 25486384, 25489342, 25489367
25489607, 25492379, 25498930, 25498994, 25516250, 25524955, 25528838
25530080, 25530814, 25535668, 25536819, 25537470, 25539063, 25540738
25546580, 25546608, 25547901, 25551676, 25553616, 25554787, 25555252
25557886, 25558986, 25560487, 25561296, 25569149, 25570929, 25573623
25575348, 25575369, 25575628, 25579458, 25579761, 25591394, 25594901
25597525, 25598473, 25599425, 25600342, 25600421, 25601999, 25602488
25603923, 25606091, 25607726, 25612095, 25614866, 25616268, 25616359
25616417, 25616645, 25631933, 25633101, 25634317, 25634348, 25635149
25638456, 25639019, 25643818, 25643889, 25643931, 25646373, 25647325

25648731, 25653109, 25654459, 25654936, 25655390, 25655966, 25659655
25660847, 25661819, 25662088, 25662101, 25662524, 25663488, 25667973
25669791, 25670786, 25671354, 25672640, 25674386, 25680221, 25685152
25686739, 25687460, 25691904, 25694206, 25695903, 25696520, 25699321
25700654, 25709368, 25710420, 25715167, 25717371, 25722055, 25722608
25722720, 25723097, 25723158, 25728085, 25729507, 25730014, 25734963
25736747, 25739065, 25740844, 25741955, 25743479, 25747569, 25749273
25752755, 25754606, 25756945, 25757697, 25757748, 25760195, 25762221
25764020, 25766822, 25768681, 25772669, 25774077, 25775213, 25775444
25780343, 25783447, 25784002, 25785331, 25785441, 25788879, 25789041
25789277, 25789579, 25790353, 25792911, 25795865, 25797092, 25797124
25797305, 25800464, 25802510, 25803364, 25803545, 25807997, 25809524
25810263, 25810704, 25811650, 25812390, 25813931, 25818707, 25822410
25823532, 25823754, 25825910, 25826740, 25830492, 25832935, 25834581
25835365, 25838361, 25838755, 25852885, 25856821, 25858672, 25861398
25865785, 25870579, 25871177, 25871639, 25871753, 25872127, 25872389
25873336, 25874050, 25874678, 25882264, 25883438, 25885148, 25888073
25888984, 25890056, 25890673, 25890782, 25894239, 25895224, 25897615
25898228, 25904273, 25904490, 25905130, 25906117, 25906886, 25908728
25911724, 25914276, 25919622, 25932524, 25932728, 25933494, 25941836
25943271, 25945130, 25947799, 25951571, 25953857, 25954022, 25954054
25957038, 25963024, 25964954, 25967544, 25967985, 25970731, 25971286
25972417, 25973152, 25975723, 25977302, 25980605, 25980770, 25981498
25982666, 25986062, 25990907, 25995938, 25997810, 26006257, 26007010
26019148, 26024732, 26024784, 26025681, 26029075, 26029777, 26029780
26032573, 26034119, 26036748, 26037215, 26038086, 26039623, 26040483
26045732, 26051656, 26078437, 26078493, 26080410, 26083298, 26087754
26088426, 26088836, 26090767, 26090893, 26091640, 26091786, 26095327
26095405, 26096382, 26108080, 26108337, 26110259, 26110632, 26111842
26112621, 26115103, 26121990, 26124078, 26130486, 26137367, 26137416
26138085, 26145560, 26149904, 26153372, 26153977, 26168933, 26169341
26169345, 26170659, 26170715, 26176002, 26187943, 26189861, 26198757
26198926, 26201113, 26203182, 26223039, 26237338, 26237431, 26237773
26238195, 26242031, 26242677, 26243698, 26244115, 26245237, 26248143
26249718, 26256131, 26257953, 26259265, 26261327, 26263328, 26263721
26268756, 26269790, 26271001, 26274660, 26275023, 26275415, 26277439
26281476, 26285062, 26285933, 26308650, 26309047, 26317991, 26318627
26323308, 26324206, 26324769, 26325856, 26327418, 26327624, 26327775
26330994, 26331743, 26333141, 26336977, 26338953, 26351334, 26351996
26353617, 26358670, 26359091, 26362155, 26362821, 26366517, 26367012
26367460, 26371725, 26373967, 26374791, 26375052, 26375250, 26380097
26385189, 26388538, 26396790, 26398675, 26399626, 26399691, 26405036
26406387, 26407408, 26410240, 26412540, 26418088, 26420561, 26421667
26422277, 26423085, 26426526, 26426967, 26430323, 26430737, 26434436
26434999, 26435073, 26436168, 26438612, 26439748, 26440169, 26440749
26442308, 26444601, 26444887, 26446098, 26452606, 26474662, 26474703
26475419, 26476244, 26478970, 26479173, 26482376, 26486365, 26492866
26493289, 26498354, 26513067, 26513709, 26521043, 26522439, 26523432
26526726, 26526799, 26536320, 26537307, 26542135, 26542236, 26542835
26544823, 26545688, 26546070, 26546664, 26546754, 26548363, 26556014
26558437, 26569225, 26575788, 26580633, 26582460, 26584641, 26588069
26597140, 26599395, 26608137, 26608238, 26609942, 26615291, 26615690
26617804, 26623652, 26626879, 26629381, 26633355, 26633558, 26635897
26637273, 26637824, 26639167, 26641610, 26650226, 26650540, 26654363
26658759, 26659182, 26669550, 26680105, 26712331, 26714486, 26714910
26717528, 26724511, 26725687, 26727397, 26729494, 26729611, 26740700
26744595, 26745002, 26751106, 26751171, 26755171, 26758193, 26764561
26765212, 26768025, 26775602, 26784509, 26790923, 26794786, 26797591
26798411, 26798516, 26802503, 26816582, 26820076, 26822314, 26822620
26824833, 26828994, 26829845, 26833932, 26837569, 26837702, 26840654
26844406, 26844870, 26849779, 26871815, 26875822, 26883456, 26895149
26896659, 26898563, 26907236, 26907327, 26908788, 26909100, 26909504
26910716, 26911000, 26939314, 26943004, 26944190, 26958896, 26963310
26966616, 26966916, 26967713, 26968670, 26969321, 26970175, 26970717
26981902, 26983259, 26985002, 26986173, 26992964, 27000158, 27006120
27006664, 27009164, 27013146, 27028251, 27032785, 27033520, 27033652

27034890, 27036163, 27037839, 27038986, 27039712, 27044169, 27044297
27045634, 27052607, 27056711, 27058530, 27060167, 27060859, 27061736
27072923, 27073314, 27079140, 27084613, 27087426, 27090765, 27092508
27093423, 27097854, 27100800, 27101105, 27105900, 27106179, 27110878
27115422, 27117822, 27119621, 27119861, 27122162, 27124624, 27125872
27133662, 27134734, 27135647, 27135993, 27138325, 27142120, 27142373
27142529, 27144928, 27151826, 27153641, 27160922, 27161071, 27162390
27162405, 27163928, 27165231, 27169796, 27170305, 27181537, 27181897
27185188, 27195935, 27199245, 27200959, 27202015, 27203055, 27207110
27208795, 27213224, 27216046, 27217412, 27223075, 27229389, 27231051
27234962, 27236722, 27242226, 27244337, 27244999, 27248917, 27249531
27250547, 27251690, 27255377, 27256000, 27258578, 27259307, 27262945
27264464, 27266245, 27274456, 27274536, 27275533, 27276231, 27283960
27284499, 27285244, 27288638, 27292213, 27293599, 27302711, 27302730
27303287, 27303938, 27304410, 27304906, 27305039, 27308088, 27314206
27314390, 27314697, 27320576, 27321179, 27329612, 27333106, 27334316
27338912, 27338946, 27339115, 27345231, 27346709, 27348081, 27349393
27350267, 27351628, 27359178, 27364854, 27367194, 27369515, 27370965
27375542, 27381498, 27383281, 27386467, 27393570, 27394703, 27395416
27396624, 27396672, 27396813, 27397048, 27400416, 27400598, 27404573
27404668, 27405645, 27416997, 27423251, 27424405, 27426363, 27432062
27432826, 27433385, 27433870, 27434193, 27439835, 27441326, 27442041
27445727, 27457891, 27459593, 27459948, 27466597, 27468303, 27486805
27487919, 27489107, 27493674, 27494663, 27501373, 27501413, 27502420
27504770, 27505229, 27508985, 27510959, 27525909, 27529661, 27533780
27533819, 27534509, 27540613, 27544973, 27548131, 27554074, 27555481
27558861, 27560602, 27562488, 27565906, 27567477, 27576342, 27576354
27587905, 27588271, 27593501, 27595973, 27601118, 27601441, 27607563
27611612, 27613080, 27613530, 27617978, 27620808, 27623159, 27629756
27632114, 27634676, 27634991, 27645231, 27657712, 27658186, 27666312
27671633, 27680669, 27686599, 27687880, 27688036, 27688099, 27688692
27691920, 27691939, 27693416, 27693713, 27695063, 27698953, 27700466
27704237, 27709046, 27710072, 27719000, 27726780, 27729678, 27739006
27740424, 27748954, 27751755, 27757567, 27757888, 27758544, 27758653
27758972, 27759077, 27769361, 27793533, 27799032, 27801337, 27818389
27819881, 27824540, 27824543, 27825241, 27828794, 27829295, 27833672
27834551, 27834569, 27835925, 27839353, 27846298, 27846499, 27847259
27850112, 27855490, 27861226, 27873412, 27882176, 27886087, 27897759
27898015, 27902561, 27909478, 27927431, 27929287, 27931299, 27935493
27940876, 27945870, 27951817, 27959048, 27959594, 27966472, 27967484
27983174, 27986817, 27994325, 27995215, 27995248, 27997875, 27998003
27999073, 27999638, 28000269, 28022101, 28023081, 28023399, 28023482
28024793, 28025414, 28026866, 28033429, 28040776, 28043157, 28045903
28067846, 28071549, 28072383, 28072567, 28074713, 28079127, 28090453
28098160, 28099662, 28108003, 28111583, 28120036, 28120951, 28124631
28125947, 28129791, 28140658, 28157786, 28164480, 28165439, 28171079
28174827, 28180464, 28181021, 28184554, 28188330, 28190796, 28194173
28199085, 28201419, 28209985, 28215510, 28218832, 28220398, 28223871
28226179, 28229360, 28236305, 28238264, 28242712, 28250929, 28256164
28271119, 28276054, 28279837, 28281094, 28282606, 28290434, 28294563
28302049, 28305001, 28305362, 28309406, 28319114, 28320399, 28330714
28330971, 28354603, 28357401, 28361221, 28365111, 28369092, 28371123
28373960, 28375383, 28378446, 28384353, 28386259, 28388910, 28390273
28391210, 28396445, 28397317, 28401116, 28402823, 28420042, 28420457
28423598, 28432129, 28434028, 28437315, 28439086, 28454215, 28454242
28468312, 28468493, 28481149, 28483184, 28489150, 28501075, 28502343
28503038, 28507324, 28508053, 28508557, 28512336, 28521330, 28522441
28528349, 28530171, 28535127, 28535272, 28537715, 28538439, 28542455
28545134, 28546290, 28547068, 28564479, 28571483, 28572407, 28572834
28578164, 28578945, 28585411, 28587723, 28589509, 28600233, 28608211
28612674, 28617631, 28617959, 28621470, 28622202, 28627255, 28636676
28639299, 28642899, 28678804, 28691965, 28692103, 28692275, 28697806
28708023, 28709063, 28710469, 28714988, 28728272, 28734355, 28740708
28742555, 28749289, 28749724, 28758090, 28758722, 28774416, 28777174
28791725, 28797711, 28803345, 28817449, 28819640, 28820669, 28821847
28830691, 28831971, 28835937, 28836716, 28838066, 28844866, 28849751

28852691, 28855922, 28856060, 28856172, 28863263, 28863487, 28867992
28887305, 28889730, 28891984, 28927452, 28949888, 28951382, 28956908
28959493, 28960211, 28965095, 28965787, 28986231, 28986257, 28987439
28991884, 28993295, 28993590, 29002488, 29006527, 29007321, 29007353
29009513, 29013832, 29024054, 29026582, 29027694, 29032276, 29040739
29048498, 29050886, 29060216, 29061016, 29115857, 29125374, 29158680
29163567, 29170232, 29173817, 29179097, 29182517, 29182901, 29189889
29198092, 29200700, 29203604, 29213320, 29224605, 29237575, 29247712
29249289, 29250230, 29250317, 29260956, 29278684, 29301463, 29339155
29343086, 29343861, 29347943, 29353821, 29372069, 29372460, 29375355
29375984, 29376346, 29378913, 29379978, 29383695, 29388020, 29398488
29399336, 29405462, 29409149, 29409455, 29426241, 29434301, 29436454
29437712, 29450812, 29452251, 29454978, 29464779, 29483626, 29483672
29483723, 29483771, 29500257, 29500963, 29504682, 29511611, 29524985
29530515, 29536342, 29538631, 29542449, 29542580, 29548592, 29549071
29580394, 29614575, 29621961, 29625065, 29626154, 29629430, 29633753
29637526, 29645349, 29651520, 29656843, 29667994, 29676089, 29678163
29685137, 29687220, 29690625, 29703195, 29707896, 29719146, 29724063
29726695, 29767177, 29782211, 29791152, 29794462, 29807964, 29813494
29817278, 29825525, 29836659, 29841687, 29846645, 29853485, 29865188
29869404, 29875459, 29876358, 29881050, 29884958, 29893132, 29902311
29914449, 29944035, 29944660, 29951620, 29961353, 29962927, 29962939
29991257, 30018017, 30031027, 30064268, 30076253, 30078934, 30088912
30098251, 30099302, 30114477, 30120608, 30125995, 30131286, 30147928
30150731, 30163243, 30164714, 30173113, 30177597, 30179644, 30189023
30196358, 30200758, 30215130, 30218044, 30218317, 30223712, 30239480
30241567, 30244787, 30246179, 30247305, 30252098, 30252156, 30253255
30265523, 30272329, 30281591, 30282501, 30283932, 30305880, 30312094
30342878, 30364613, 30365745, 30374739, 30384152, 30402386, 30403763
30408515, 30413137, 30416034, 30431274, 30441687, 30453442, 30458593
30460922, 30475115, 30485255, 30496957, 30497057, 30501574, 30503943
30509277, 30510527, 30582500, 30613937, 30635302, 30654409, 30671813
30741263, 30803210, 30815852, 30968208, 29997959, 29997937, 28852325
28125601, 27015449, 25881255, 25173124, 24701840, 23614158, 29213893
25811105, 25890046, 26023042, 26570134, 27000702, 27461740, 27952586
27642235, 27539876, 28502098, 28915870, 29254623, 29774362, 30160625
30534662, 30855101

Version 12.2.0.1.ru-2020-01.rur-2020-01.r1

Version 12.2.0.1.ru-2020-01.rur-2020-01.r1 includes the following:

- Patch 30593149: Database Jan 2020 Release Update: 12.2.0.1.200114
- Patch 30502018: OJVM RELEASE UPDATE 12.2.0.1.200114
- Patch 29997937: DSTv34 for RDBMS (TZDATA2019G)
- Patch 29997959: DSTv34 for OJVM (TZDATA2019G)
- PreUpgrade Jar: preupgrade_12201_cbuild_23_lf.zip
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR is included in DB PATCH 30138470

Oracle release update 12.2.0.1.200114, released January 2020

Bugs fixed:

30125995, 8480838, 8932139, 12763598, 13554903, 14221306, 14690846
15931756, 16002385, 16438495, 16727454, 16942578, 17027695, 17533661
17947871, 18308268, 18521691, 18594510, 18774543, 18878420, 19072655
19211433, 19285025, 19327292, 19526548, 19614243, 19647894, 19649997

19702201, 19721304, 20003668, 20087519, 20118035, 20120236, 20324049
20436508, 20532077, 20549013, 2058486, 20591151, 20617383, 20620169
20736227, 20756305, 20866970, 20976443, 21089435, 21095391, 21143725
21147908, 21159907, 21178363, 21186167, 21197098, 21216226, 21320338
21433452, 21479706, 21517767, 21520266, 21547051, 21638600, 21744603
21788462, 21837606, 21882528, 21935698, 21981529, 21985256, 22007324
22070473, 22070853, 22072543, 22087683, 22104866, 22107360, 22174392
22179537, 22282748, 22310426, 22347493, 22363790, 22364044, 22367053
22379010, 22446455, 22454940, 22495673, 22503283, 22503297, 22504793
22522515, 22530986, 22564336, 22568728, 22581771, 22594071, 22599050
22628825, 22645009, 22645496, 22654475, 22700845, 22726044, 22729345
22826067, 22843979, 22845846, 22864303, 22898198, 22921674, 22939829
22950945, 22970869, 22981722, 23019710, 23026585, 23035249, 23055900
23061453, 23065002, 23066146, 23080557, 23104033, 23105538, 23109325
23110523, 23125560, 23126545, 23127945, 23151677, 23169712, 23177923
23179662, 23184263, 23197730, 23234232, 23237091, 23249829, 23271203
23278750, 23300142, 23306590, 23308065, 23310101, 23312077, 23328639
23333567, 23336559, 23342170, 23481673, 23491861, 23499004, 23499160
23521523, 23527363, 23533647, 23548817, 23567857, 23572982, 23581777
23588722, 23599216, 23600861, 23602213, 23645516, 23665623, 23709062
23715460, 23715518, 23730961, 23733981, 23735292, 23738304, 23741944
23743596, 23746128, 23749454, 23761724, 24010030, 24289874, 24294174
24303148, 24307571, 24308349, 24326444, 24326846, 24328857, 24330708
24332831, 24334708, 24336249, 24337882, 24341675, 24343905, 24345420
24346821, 24348685, 24350620, 24352981, 24355111, 24357348, 24368004
24371491, 24373528, 24373756, 24374976, 24376875, 24376878, 24383086
24385983, 24401351, 24403922, 24409977, 24415926, 24416451, 24421668
24423416, 24425056, 24425998, 24435982, 24437162, 24440648, 24443539
24457597, 24460392, 24461826, 24467122, 24468470, 24470606, 24471079
24471473, 24473736, 24485034, 24485161, 24485174, 24486059, 24486237
24509056, 24516314, 24534401, 24554533, 24555417, 24556862, 24556967
24560906, 24563422, 24570214, 24570598, 24573817, 24578718, 24578797
24588377, 24589081, 24589590, 24591506, 24593740, 24595699, 24600330
24609592, 24609996, 24611527, 24616637, 24617969, 24623975, 24624166
24642495, 24654629, 24655717, 24664211, 24668398, 24669189, 24674197
24674955, 24676172, 24677696, 24680959, 24687075, 24689376, 24692973
24693010, 24693290, 24697323, 24699619, 24710696, 24713381, 24714096
24717183, 24717859, 24718260, 24719799, 24735430, 24737064, 24737403
24737581, 24737954, 24739791, 24744383, 24744686, 24752618, 24757934
24759556, 24760407, 24764085, 24766309, 24784414, 24786669, 24792678
24793511, 24796092, 24797119, 24800423, 24801152, 24802934, 24811725
24812047, 24818566, 24827228, 24827654, 24831514, 24835919, 24841671
24843188, 24844549, 24844841, 24845157, 24848746, 24848923, 24850622
24907917, 24908321, 24911709, 24912588, 24921478, 24922704, 24923080
24923215, 24923338, 24923790, 24924667, 24926999, 24929210, 24938784
24940060, 24942749, 24953434, 24957555, 24960044, 24960809, 24965426
24966594, 24966788, 24967993, 24968162, 24976007, 24978100, 25022574
25027852, 25028996, 25029022, 25029423, 25032818, 25034396, 25036006
25036474, 25042823, 25044977, 25045228, 25050160, 25051465, 25051628
25054064, 25057811, 25058080, 25060506, 25062592, 25063971, 25065563
25072986, 25077278, 25078611, 25086233, 25087436, 25092777, 25093872
25095982, 25098160, 25099339, 25099497, 25099758, 25100063, 25100579
25103996, 25107662, 25110233, 25114561, 25120284, 25120668, 25120742
25121089, 25123585, 25124363, 25129925, 25130312, 25140197, 25145163
25145215, 25150925, 25159176, 25162645, 25164293, 25166187, 25171041
25171084, 25175723, 25176408, 25178032, 25178101, 25178179, 25179774
25182817, 25184453, 25184555, 25186079, 25191872, 25192044, 25192528
25192729, 25195901, 25199585, 25201454, 25202355, 25203656, 25205954
25206864, 25207410, 25209912, 25210268, 25210499, 25210690, 25211628
25219450, 25223839, 25224242, 25225795, 25226665, 25227381, 25230870
25230945, 25237577, 25240188, 25240590, 25241448, 25241625, 25244807
25248384, 25250109, 25251648, 25257085, 25259611, 25262869, 25263960
25265499, 25269133, 25283790, 25287072, 25293659, 25296876, 25299227
25299807, 25300427, 25303284, 25303756, 25305405, 25307368, 25309116
25313154, 25313411, 25316758, 25317989, 25320555, 25323525, 25328518
25329664, 25335249, 25335360, 25335790, 25337332, 25337640, 25348956

25353983, 25356118, 25357142, 25360661, 25362958, 25367588, 25367721
25382812, 25383204, 25384462, 25386748, 25388896, 25392535, 25393714
25395696, 25397936, 25398306, 25404202, 25405100, 25405687, 25405813
25410017, 25410180, 25410802, 25410877, 25411036, 25415713, 25417050
25417056, 25417958, 25425451, 25425760, 25427662, 25429959, 25430120
25433696, 25435038, 25437699, 25440818, 25442559, 25444961, 25445168
25451531, 25455795, 25457409, 25459958, 25462714, 25463844, 25472112
25472885, 25476125, 25476149, 25477657, 25478885, 25479164, 25481087
25482971, 25489342, 25489367, 25489607, 25492379, 25498930, 25498994
25516250, 25524955, 25528838, 25530080, 25530814, 25535668, 25536819
25537470, 25539063, 25540738, 25546580, 25546608, 25547901, 25551676
25553616, 25554787, 25555252, 25557886, 25558986, 25560487, 25561296
25569149, 25570929, 25573623, 25575348, 25575369, 25575628, 25579458
25579761, 25591394, 25594901, 25597525, 25598473, 25599425, 25600342
25600421, 25602488, 25603923, 25606091, 25607726, 25612095, 25614866
25616268, 25616359, 25616417, 25616645, 25631933, 25633101, 25634317
25634348, 25635149, 25638456, 25639019, 25643818, 25643889, 25643931
25646373, 25647325, 25648731, 25653109, 25654459, 25654936, 25655390
25655966, 25659655, 25660847, 25661819, 25662088, 25662101, 25662524
25663488, 25669791, 25670786, 25671354, 25672640, 25674386, 25680221
25685152, 25686739, 25687460, 25691904, 25694206, 25695903, 25696520
25699321, 25700654, 25709368, 25710420, 25715167, 25717371, 25722055
25722608, 25722720, 25723158, 25728085, 25729507, 25730014, 25734963
25736747, 25739065, 25741955, 25743479, 25747569, 25749273, 25752755
25754606, 25756945, 25757748, 25760195, 25762221, 25764020, 25766822
25768681, 25772669, 25774077, 25775213, 25775444, 25780343, 25783447
25784002, 25785331, 25785441, 25788879, 25789041, 25789277, 25789579
25790353, 25792911, 25795865, 25797092, 25797124, 25797305, 25800464
25802510, 25803364, 25803545, 25807997, 25810263, 25810704, 25811650
25812390, 25813931, 25818707, 25822410, 25823754, 25825910, 25826740
25830492, 25832935, 25834581, 25835365, 25838361, 25838755, 25852885
25856821, 25858672, 25861398, 25865785, 25870579, 25871177, 25871639
25871753, 25872127, 25872389, 25873336, 25874050, 25874678, 25882264
25883438, 25885148, 25888073, 25888984, 25890056, 25890673, 25890782
25894239, 25895224, 25897615, 25898228, 25904273, 25904490, 25906117
25906886, 25908728, 25911724, 25914276, 25919622, 25932524, 25932728
25933494, 25941836, 25943271, 25945130, 25947799, 25951571, 25953857
25954022, 25954054, 25957038, 25963024, 25964954, 25967544, 25967985
25970731, 25971286, 25973152, 25975723, 25977302, 25980605, 25980770
25981498, 25982666, 25986062, 25990907, 25995938, 25997810, 26006257
26007010, 26019148, 26024732, 26024784, 26025681, 26029075, 26029777
26029780, 26032573, 26034119, 26036748, 26037215, 26038086, 26039623
26040483, 26045732, 26051656, 26078437, 26078493, 26080410, 26083298
26087754, 26088426, 26088836, 26090767, 26090893, 26091640, 26091786
26095327, 26095405, 26096382, 26108080, 26108337, 26110259, 26110632
26111842, 26112621, 26115103, 26121990, 26124078, 26130486, 26137367
26138085, 26145560, 26149904, 26153372, 26153977, 26168933, 26169341
26169345, 26170659, 26170715, 26176002, 26187943, 26189861, 26198757
26198926, 26201113, 26203182, 26223039, 26237338, 26237431, 26237773
26238195, 26242031, 26242677, 26243698, 26244115, 26245237, 26248143
26249718, 26256131, 26257953, 26259265, 26261327, 26263328, 26263721
26268756, 26269790, 26271001, 26274660, 26275023, 26275415, 26277439
26281476, 26285062, 26285933, 26308650, 26309047, 26317991, 26318627
26323308, 26324206, 26324769, 26325856, 26327418, 26327624, 26327775
26330994, 26331743, 26333141, 26336977, 26338953, 26351334, 26353617
26358670, 26359091, 26362155, 26362821, 26366517, 26367012, 26367460
26371725, 26373967, 26374791, 26375052, 26375250, 26380097, 26385189
26388538, 26396790, 26399626, 26399691, 26405036, 26406387, 26407408
26410240, 26412540, 26418088, 26420561, 26421667, 26422277, 26423085
26426526, 26426967, 26430323, 26430737, 26434436, 26434999, 26435073
26436168, 26438612, 26439748, 26440169, 26440749, 26442308, 26444601
26444887, 26446098, 26452606, 26474662, 26474703, 26475419, 26476244
26478970, 26479173, 26482376, 26486365, 26492866, 26493289, 26498354
26513067, 26513709, 26521043, 26522439, 26523432, 26526726, 26526799
26536320, 26537307, 26542135, 26542236, 26542835, 26544823, 26545688
26546070, 26546664, 26546754, 26548363, 26556014, 26558437, 26569225

26575788, 26580633, 26582460, 26584641, 26588069, 26597140, 26599395
26608137, 26608238, 26609942, 26615291, 26615690, 26617804, 26623652
26626879, 26629381, 26633355, 26633558, 26635897, 26637273, 26637824
26639167, 26641610, 26650226, 26650540, 26654363, 26658759, 26659182
26669550, 26680105, 26712331, 26714486, 26714910, 26717528, 26724511
26725687, 26727397, 26729494, 26729611, 26740700, 26744595, 26745002
26751106, 26751171, 26755171, 26758193, 26764561, 26765212, 26768025
26775602, 26784509, 26794786, 26797591, 26798411, 26798516, 26802503
26816582, 26820076, 26822620, 26824833, 26828994, 26829845, 26833932
26837702, 26840654, 26844406, 26844870, 26849779, 26871815, 26875822
26883456, 26895149, 26896659, 26898563, 26907236, 26907327, 26908788
26909100, 26909504, 26910716, 26911000, 26939314, 26943004, 26944190
26958896, 26963310, 26966616, 26966916, 26967713, 26968670, 26969321
26970175, 26970717, 26981902, 26983259, 26985002, 26986173, 26992964
27000158, 27006120, 27006664, 27009164, 27013146, 27028251, 27032785
27033520, 27034890, 27036163, 27037839, 27038986, 27039712, 27044169
27044297, 27045634, 27052607, 27056711, 27058530, 27060167, 27060859
27061736, 27072923, 27073314, 27079140, 27087426, 27090765, 27092508
27093423, 27097854, 27101105, 27105900, 27106179, 27110878, 27115422
27117822, 27119621, 27119861, 27122162, 27124624, 27125872, 27133662
27134734, 27135647, 27135993, 27138325, 27142373, 27142529, 27144928
27151826, 27153641, 27160922, 27161071, 27162390, 27162405, 27163928
27165231, 27169796, 27170305, 27181537, 27181897, 27185188, 27195935
27199245, 27200959, 27202015, 27203055, 27207110, 27208795, 27213224
27216046, 27223075, 27229389, 27231051, 27234962, 27236722, 27242226
27244337, 27244999, 27248917, 27249531, 27250547, 27251690, 27255377
27256000, 27258578, 27259307, 27262945, 27264464, 27266245, 27274456
27274536, 27275533, 27276231, 27283960, 27284499, 27285244, 27288638
27292213, 27293599, 27302711, 27302730, 27303287, 27304410, 27304906
27305039, 27308088, 27314206, 27314390, 27320576, 27321179, 27329612
27333106, 27334316, 27338912, 27338946, 27339115, 27345231, 27346709
27348081, 27349393, 27350267, 27351628, 27359178, 27364854, 27367194
27369515, 27370965, 27375542, 27381498, 27383281, 27386467, 27393570
27394703, 27395416, 27396624, 27396672, 27396813, 27397048, 27400416
27400598, 27404668, 27405645, 27416997, 27423251, 27424405, 27426363
27432062, 27432826, 27433385, 27433870, 27434193, 27439835, 27441326
27442041, 27445727, 27457891, 27459948, 27466597, 27468303, 27486805
27489107, 27493674, 27501373, 27501413, 27502420, 27504770, 27505229
27508985, 27510959, 27525909, 27529661, 27533780, 27533819, 27534509
27540613, 27544973, 27548131, 27554074, 27555481, 27558861, 27560602
27562488, 27565906, 27567477, 27576342, 27576354, 27587905, 27588271
27593501, 27595973, 27601118, 27601441, 27607563, 27611612, 27613080
27613530, 27617978, 27620808, 27623159, 27629756, 27632114, 27634676
27634991, 27657712, 27658186, 27666312, 27671633, 27680669, 27686599
27687880, 27688036, 27688099, 27688692, 27691920, 27691939, 27693416
27693713, 27695063, 27698953, 27700466, 27704237, 27709046, 27710072
27726780, 27729678, 27739006, 27740424, 27748954, 27751755, 27757567
27757888, 27758544, 27758653, 27758972, 27759077, 27769361, 27793533
27799032, 27801337, 27818389, 27819881, 27824540, 27824543, 27825241
27828794, 27829295, 27833672, 27834551, 27834569, 27835925, 27839353
27846298, 27846499, 27847259, 27850112, 27855490, 27861226, 27873412
27882176, 27886087, 27897759, 27898015, 27902561, 27909478, 27927431
27929287, 27931299, 27935493, 27940876, 27945870, 27951817, 27959048
27959594, 27966472, 27967484, 27983174, 27986817, 27994325, 27995215
27995248, 27997875, 27998003, 27999073, 27999638, 28000269, 28022101
28023081, 28023399, 28023482, 28026866, 28033429, 28040776, 28043157
28045903, 28067846, 28071549, 28072567, 28074713, 28079127, 28090453
28098160, 28099662, 28108003, 28111583, 28120036, 28120951, 28124631
28125947, 28129791, 28140658, 28164480, 28165439, 28171079, 28174827
28180464, 28181021, 28184554, 28188330, 28190796, 28194173, 28199085
28201419, 28209985, 28215510, 28218832, 28220398, 28223871, 28226179
28229360, 28236305, 28242712, 28256164, 28271119, 28279837, 28281094
28282606, 28290434, 28294563, 28302049, 28305001, 28305362, 28309406
28319114, 28320399, 28330971, 28354603, 28357401, 28361221, 28365111
28369092, 28371123, 28373960, 28375383, 28378446, 28384353, 28386259
28388910, 28390273, 28391210, 28396445, 28397317, 28401116, 28402823

28420042, 28420457, 28423598, 28432129, 28434028, 28435902, 28437315
28454215, 28454242, 28468312, 28468493, 28481149, 28483184, 28489150
28501075, 28502343, 28507324, 28508053, 28508557, 28512336, 28521330
28522441, 28528349, 28530171, 28535272, 28537715, 28538439, 28542455
28545134, 28546290, 28547068, 28571483, 28572407, 28572834, 28578164
28578945, 28585411, 28587723, 28589509, 28600233, 28612674, 28617631
28617959, 28621470, 28627255, 28636676, 28639299, 28642899, 28678804
28691965, 28692103, 28692275, 28697806, 28708023, 28714988, 28728272
28734355, 28742555, 28749289, 28749724, 28758090, 28758722, 28774416
28777174, 28791725, 28797711, 28803345, 28817449, 28819640, 28820669
28821847, 28830691, 28831971, 28838066, 28844866, 28849751, 28852691
28855922, 28856060, 28856172, 28863487, 28867992, 28887305, 28889730
28891984, 28927452, 28949888, 28951026, 28951382, 28956908, 28959493
28960211, 28965095, 28965787, 28986231, 28986257, 28987439, 28991884
28993590, 29002488, 29006527, 29007353, 29009513, 29013832, 29024054
29027694, 29032276, 29040739, 29050886, 29060216, 29061016, 29115857
29125374, 29158680, 29163567, 29170232, 29173817, 29182901, 29189889
29198092, 29200700, 29213320, 29224605, 29237575, 29250230, 29250317
29260956, 29278684, 29301463, 29339155, 29347943, 29353821, 29372069
29375355, 29375984, 29376346, 29378913, 29379978, 29383695, 29388020
29398488, 29399336, 29405462, 29409149, 29434301, 29436454, 29437712
29450812, 29452251, 29454978, 29464779, 29483626, 29483672, 29483723
29483771, 29500257, 29504682, 29511611, 29530515, 29536342, 29538631
29542449, 29549071, 29626154, 29629430, 29633753, 29637526, 29645349
29651520, 29667994, 29676089, 29678163, 29690625, 29703195, 29707896
29719146, 29724063, 29726695, 29767177, 29782211, 29791152, 29794462
29813494, 29817278, 29825525, 29836659, 29841687, 29846645, 29853485
29865188, 29869404, 29875459, 29884958, 29893132, 29902311, 29914449
29944035, 29944660, 29951620, 29962927, 29962939, 29991257, 30088912
30098251, 30150731, 30164714, 30189023, 30218044, 30223712, 30244787
30252098, 30252156, 30253255, 30281591, 30342878, 30365745, 30402386
30408515, 30453442, 30458593, 30485255

Version 12.2.0.1.ru-2019-10.rur-2019-10.r1

Version 12.2.0.1.ru-2019-10.rur-2019-10.r1 includes the following:

- Patch 30138470: DATABASE OCT 2019 RELEASE UPDATE 12.2.0.1.191015
- Patch 30133625: OJVM RELEASE UPDATE 12.2.0.1.191015
- Patch 29997937: DSTv34 for RDBMS (TZDATA2019G)
- Patch 29997959: DSTv34 for OJVM (TZDATA2019G)
- PreUpgrade Jar: preupgrade_12201_cbuild_23_if.zip
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR is included in DB PATCH 30138470
- Support for [Resizing the temporary tablespace in a read replica \(p. 1061\)](#)

Oracle release update 12.2.0.1.191015, released October 2019

Bugs fixed:

29013832, 30244787, 30253255, 8480838, 8932139, 12763598, 13554903
14221306, 14690846, 15931756, 16002385, 16438495, 16727454, 16942578
17027695, 17533661, 17947871, 18308268, 18521691, 18594510, 18774543
18878420, 19072655, 19211433, 19285025, 19327292, 19526548, 19614243
19647894, 19649997, 19702201, 19721304, 20003668, 20087519, 20118035
20120236, 20324049, 20436508, 20532077, 20588486, 20591151, 20617383
20620169, 20736227, 20756305, 20866970, 20976443, 21089435, 21095391

21143725, 21147908, 21159907, 21178363, 21186167, 21197098, 21216226
21320338, 21433452, 21479706, 21517767, 21520266, 21547051, 21638600
21744603, 21882528, 21935698, 21981529, 21985256, 22007324, 22070473
22070853, 22072543, 22087683, 22104866, 22107360, 22179537, 22310426
22347493, 22364044, 22367053, 22379010, 22446455, 22495673, 22503283
22503297, 22504793, 22522515, 22530986, 22564336, 22568728, 22581771
22594071, 22599050, 22628825, 22645009, 22654475, 22700845, 22726044
22729345, 22826067, 22843979, 22845846, 22864303, 22898198, 22921674
22939829, 22950945, 22970869, 22981722, 23019710, 23026585, 23035249
23055900, 23061453, 23065002, 23066146, 23080557, 23104033, 23105538
23110523, 23125560, 23126545, 23127945, 23151677, 23169712, 23177923
23179662, 23184263, 23197730, 23234232, 23237091, 23249829, 23271203
23278750, 23300142, 23306590, 23310101, 23312077, 23328639, 23336559
23342170, 23481673, 23491861, 23499004, 23499160, 23521523, 23527363
23533647, 23548817, 23567857, 23572982, 23581777, 23588722, 23599216
23600861, 23602213, 23645516, 23665623, 23709062, 23715460, 23715518
23730961, 23733981, 23735292, 23741944, 23743596, 23746128, 23749454
23761724, 24010030, 24289874, 24294174, 24303148, 24307571, 24308349
24326444, 24326846, 24328857, 24332831, 24334708, 24336249, 24337882
24341675, 24343905, 24345420, 24346821, 24348685, 24350620, 24352981
24355111, 24357348, 24368004, 24371491, 24373528, 24373756, 24374976
24376875, 24376878, 24383086, 24385983, 24401351, 24403922, 24409977
24415926, 24416451, 24421668, 24423416, 24425056, 24425998, 24435982
24437162, 24440648, 24443539, 24457597, 24460392, 24461826, 24467122
24468470, 24470606, 24471079, 24471473, 24473736, 24485034, 24485161
24485174, 24486059, 24486237, 24509056, 24534401, 24554533, 24555417
24556967, 24560906, 24563422, 24570214, 24570598, 24573817, 24578718
24578797, 24589081, 24589590, 24591506, 24593740, 24595699, 24600330
24609592, 24609996, 24611527, 24616637, 24617969, 24623975, 24624166
24642495, 24654629, 24655717, 24664211, 24668398, 24669189, 24674197
24674955, 24676172, 24677696, 24680959, 24687075, 24689376, 24692973
24693290, 24697323, 24699619, 24710696, 24713381, 24714096, 24717183
24717859, 24718260, 24719799, 24735430, 24737064, 24737403, 24737581
24737954, 24739791, 24744383, 24744686, 24757934, 24759556, 24760407
24764085, 24766309, 24786669, 24792678, 24793511, 24796092, 24797119
24800423, 24801152, 24802934, 24811725, 24812047, 24818566, 24827228
24827654, 24831514, 24835919, 24841671, 24843188, 24844549, 24844841
24845157, 24848746, 24848923, 24850622, 24907917, 24908321, 24911709
24912588, 24922704, 24923080, 24923215, 24923338, 24923790, 24924667
24926999, 24929210, 24938784, 24940060, 24942749, 24953434, 24957555
24960044, 24965426, 24966594, 24966788, 24967993, 24968162, 24976007
24978100, 25022574, 25027852, 25028996, 25029022, 25029423, 25032818
25034396, 25036006, 25036474, 25042823, 25044977, 25045228, 25050160
25051465, 25051628, 25057811, 25058080, 25060506, 25062592, 25063971
25065563, 25072986, 25078611, 25086233, 25087436, 25092777, 25093872
25095982, 25098160, 25099339, 25099497, 25099758, 25100063, 25100579
25103996, 25107662, 25110233, 25114561, 25120284, 25120668, 25120742
25121089, 25123585, 25124363, 25129925, 25130312, 25140197, 25145163
25145215, 25150925, 25159176, 25162645, 25164293, 25166187, 25171041
25171084, 25175723, 25176408, 25178032, 25178101, 25178179, 25179774
25182817, 25184453, 25184555, 25186079, 25191872, 25192044, 25192528
25192729, 25195901, 25199585, 25201454, 25202355, 25203656, 25206864
25207410, 25209912, 25210268, 25210499, 25211628, 25223839, 25224242
25225795, 25226665, 25227381, 25230870, 25230945, 25237577, 25240188
25240590, 25241448, 25241625, 25244807, 25248384, 25251648, 25257085
25259611, 25262869, 25263960, 25265499, 25283790, 25287072, 25293659
25296876, 25299227, 25299807, 25300427, 25303284, 25303756, 25305405
25307368, 25309116, 25313154, 25313411, 25316758, 25317989, 25320555
25323525, 25328518, 25329664, 25335249, 25335360, 25335790, 25337332
25337640, 25348956, 25353983, 25356118, 25357142, 25360661, 25362958
25367588, 25367721, 25382812, 25383204, 25384462, 25386748, 25388896
25392535, 25393714, 25395696, 25397936, 25398306, 25404202, 25405100
25405687, 25405813, 25410017, 25410180, 25410802, 25410877, 25411036
25415713, 25417050, 25417056, 25417958, 25425451, 25425760, 25427662
25429959, 25430120, 25433696, 25435038, 25437699, 25440818, 25442559
25444961, 25445168, 25451531, 25455795, 25457409, 25459958, 25462714

25463844, 25472112, 25476149, 25477657, 25478885, 25479164, 25482971
25489342, 25489367, 25489607, 25492379, 25498930, 25498994, 25516250
25524955, 25528838, 25530080, 25530814, 25535668, 25536819, 25537470
25539063, 25540738, 25546580, 25546608, 25547901, 25551676, 25553616
25554787, 25555252, 25557886, 25558986, 25560487, 25561296, 25569149
25570929, 25575348, 25575369, 25575628, 25579458, 25579761, 25594901
25597525, 25598473, 25599425, 25600342, 25600421, 25602488, 25603923
25606091, 25607726, 25612095, 25614866, 25616268, 25616359, 25616417
25616645, 25631933, 25633101, 25634317, 25634348, 25635149, 25638456
25639019, 25643818, 25643889, 25643931, 25646373, 25647325, 25648731
25653109, 25654459, 25654936, 25655390, 25655966, 25659655, 25660847
25661819, 25662088, 25662101, 25662524, 25663488, 25669791, 25670786
25671354, 25672640, 25674386, 25680221, 25685152, 25686739, 25687460
25691904, 25694206, 25695903, 25696520, 25699321, 25700654, 25709368
25710420, 25715167, 25717371, 25722055, 25722608, 25722720, 25723158
25728085, 25729507, 25734963, 25736747, 25739065, 25741955, 25743479
25747569, 25749273, 25752755, 25754606, 25756945, 25757748, 25760195
25762221, 25764020, 25766822, 25768681, 25772669, 25774077, 25775213
25780343, 25783447, 25784002, 25785331, 25785441, 25788879, 25789041
25789277, 25789579, 25790353, 25792911, 25795865, 25797092, 25797124
25797305, 25800464, 25802510, 25803545, 25807997, 25810263, 25810704
25811650, 25813931, 25818707, 25822410, 25823754, 25825910, 25826740
25830492, 25832935, 25834581, 25835365, 25838361, 25838755, 25852885
25856821, 25858672, 25861398, 25865785, 25870579, 25871177, 25871639
25871753, 25872127, 25872389, 25873336, 25874050, 25874678, 25882264
25883438, 25885148, 25888073, 25888984, 25890056, 25890673, 25890782
25894239, 25895224, 25897615, 25904273, 25904490, 25906117, 25906886
25908728, 25911724, 25914276, 25919622, 25932524, 25932728, 25933494
25941836, 25943271, 25945130, 25947799, 25951571, 25953857, 25954022
25954054, 25957038, 25963024, 25964954, 25967544, 25967985, 25970731
25971286, 25973152, 25975723, 25977302, 25980605, 25980770, 25981498
25982666, 25986062, 25990907, 25995938, 25997810, 26006257, 26007010
26019148, 26024732, 26024784, 26025681, 26029075, 26029777, 26029780
26032573, 26034119, 26036748, 26037215, 26038086, 26039623, 26040483
26045732, 26051656, 26078437, 26078493, 26080410, 26083298, 26088426
26088836, 26090767, 26091640, 26091786, 26095327, 26095405, 26096382
26108080, 26108337, 26110259, 26110632, 26111842, 26112621, 26115103
26121990, 26124078, 26130486, 26137367, 26138085, 26149904, 26153372
26153977, 26168933, 26169341, 26169345, 26170659, 26170715, 26176002
26187943, 26189861, 26198757, 26198926, 26201113, 26203182, 26223039
26237338, 26237431, 26237773, 26238195, 26242031, 26242677, 26243698
26244115, 26245237, 26248143, 26249718, 26256131, 26257953, 26259265
26261327, 26263328, 26263721, 26268756, 26269790, 26271001, 26275023
26275415, 26277439, 26281476, 26285062, 26285933, 26308650, 26309047
26317991, 26318627, 26323308, 26324206, 26324769, 26327624, 26330994
26331743, 26333141, 26336977, 26338953, 26351334, 26353617, 26358670
26359091, 26362155, 26362821, 26366517, 26367012, 26367460, 26371725
26374791, 26375052, 26375250, 26380097, 26385189, 26388538, 26396790
26399626, 26399691, 26406387, 26407408, 26412540, 26418088, 26420561
26421667, 26422277, 26423085, 26426526, 26426967, 26430323, 26430737
26434436, 26434999, 26435073, 26436168, 26438612, 26439748, 26440169
26440749, 26442308, 26444601, 26444887, 26446098, 26452606, 26474662
26474703, 26475419, 26476244, 26478970, 26479173, 26482376, 26486365
26492866, 26493289, 26498354, 26513067, 26513709, 26521043, 26522439
26523432, 26526726, 26526799, 26536320, 26537307, 26542135, 26542236
26544823, 26545688, 26546070, 26546664, 26546754, 26548363, 26556014
26558437, 26569225, 26575788, 26580633, 26582460, 26584641, 26597140
26599395, 26608137, 26608238, 26609942, 26615291, 26615690, 26617804
26623652, 26626879, 26629381, 26633355, 26633558, 26635897, 26637273
26637824, 26639167, 26641610, 26650226, 26654363, 26658759, 26659182
26680105, 26712331, 26714486, 26714910, 26717528, 26724511, 26725687
26727397, 26729494, 26729611, 26740700, 26744595, 26745002, 26751106
26751171, 26755171, 26758193, 26764561, 26765212, 26768025, 26775602
26784509, 26794786, 26797591, 26798411, 26798516, 26802503, 26816582
26820076, 26822620, 26828994, 26829845, 26833932, 26837702, 26840654
26844406, 26844870, 26849779, 26871815, 26875822, 26883456, 26895149

26896659, 26898563, 26907327, 26908788, 26909100, 26909504, 26910716
26911000, 26939314, 26943004, 26944190, 26958896, 26963310, 26966616
26966916, 26967713, 26969321, 26970175, 26970717, 26981902, 26983259
26985002, 26986173, 26992964, 27000158, 27006120, 27006664, 27009164
27013146, 27028251, 27032785, 27033520, 27034890, 27037839, 27038986
27039712, 27044297, 27052607, 27058530, 27060167, 27060859, 27061736
27072923, 27073314, 27079140, 27087426, 27090765, 27092508, 27093423
27097854, 27101105, 27105900, 27106179, 27110878, 27115422, 27117822
27119621, 27122162, 27124624, 27125872, 27133662, 27134734, 27135647
27135993, 27138325, 27142373, 27142529, 27144928, 27151826, 27153641
27160922, 27161071, 27162390, 27162405, 27163928, 27165231, 27169796
27170305, 27181537, 27181897, 27185188, 27199245, 27200959, 27202015
27207110, 27208795, 27213224, 27216046, 27223075, 27229389, 27231051
27236722, 27242226, 27244337, 27248917, 27249531, 27250547, 27251690
27255377, 27256000, 27258578, 27259307, 27262945, 27264464, 27266245
27274456, 27274536, 27275533, 27276231, 27283960, 27284499, 27285244
27288638, 27292213, 27293599, 27302711, 27302730, 27303287, 27304410
27304906, 27305039, 27308088, 27314206, 27314390, 27320576, 27321179
27329612, 27333106, 27334316, 27338912, 27338946, 27339115, 27345231
27346709, 27348081, 27349393, 27350267, 27351628, 27359178, 27364854
27367194, 27369515, 27370965, 27375542, 27381498, 27383281, 27386467
27393570, 27394703, 27395416, 27396624, 27396672, 27396813, 27397048
27400416, 27400598, 27404668, 27405645, 27416997, 27423251, 27424405
27426363, 27432062, 27432826, 27433385, 27433870, 27434193, 27439835
27441326, 27442041, 27445727, 27457891, 27459948, 27466597, 27468303
27493674, 27501373, 27501413, 27502420, 27504770, 27505229, 27508985
27510959, 27525909, 27533780, 27533819, 27534509, 27540613, 27544973
27548131, 27554074, 27555481, 27558861, 27560602, 27562488, 27565906
27567477, 27576342, 27576354, 27588271, 27593501, 27595973, 27601118
27601441, 27607563, 27611612, 27613080, 27613530, 27617978, 27620808
27623159, 27629756, 27632114, 27634676, 27634991, 27658186, 27666312
27671633, 27680669, 27686599, 27687880, 27688036, 27688099, 27688692
27691920, 27691939, 27693416, 27693713, 27695063, 27698953, 27700466
27704237, 27709046, 27710072, 27726780, 27740424, 27748954, 27751755
27757567, 27757888, 27758544, 27758653, 27758972, 27759077, 27793533
27799032, 27801337, 27824540, 27824543, 27825241, 27829295, 27833672
27834551, 27834569, 27835925, 27839353, 27846298, 27846499, 27847259
27850112, 27855490, 27861226, 27873412, 27882176, 27886087, 27898015
27902561, 27931299, 27935493, 27940876, 27945870, 27951817, 27959048
27959594, 27966472, 27967484, 27983174, 27986817, 27994325, 27995215
27995248, 27997875, 27998003, 27999073, 27999638, 28000269, 28022101
28023081, 28023399, 28023482, 28026866, 28033429, 28040776, 28043157
28045903, 28067846, 28072567, 28074713, 28090453, 28099662, 28108003
28111583, 28120036, 28120951, 28124631, 28125947, 28129791, 28140658
28165439, 28171079, 28174827, 28180464, 28181021, 28184554, 28188330
28194173, 28199085, 28201419, 28215510, 28218832, 28220398, 28223871
28226179, 28229360, 28236305, 28242712, 28256164, 28271119, 28279837
28281094, 28282606, 28290434, 28294563, 28302049, 28305001, 28305362
28319114, 28320399, 28354603, 28357401, 28361221, 28365111, 28369092
28371123, 28378446, 28384353, 28390273, 28391210, 28396445, 28401116
28420042, 28420457, 28423598, 28432129, 28434028, 28435902, 28437315
28454242, 28468312, 28481149, 28483184, 28489150, 28501075, 28502343
28507324, 28508053, 28508557, 28521330, 28522441, 28528349, 28535272
28537715, 28542455, 28545134, 28547068, 28571483, 28572834, 28578164
28585411, 28587723, 28600233, 28612674, 28617631, 28617959, 28621470
28627255, 28636676, 28678804, 28691965, 28692103, 28692275, 28697806
28708023, 28714988, 28728272, 28742555, 28749289, 28749724, 28758722
28774416, 28791725, 28803345, 28817449, 28819640, 28820669, 28831971
28849751, 28852691, 28855922, 28856060, 28856172, 28867992, 28889730
28891984, 28951026, 28951382, 28956908, 28960211, 28965787, 28986231
28987439, 28991884, 28993590, 29002488, 29006527, 29009513, 29024054
29027694, 29032276, 29125374, 29158680, 29163567, 29189889, 29198092
29200700, 29213320, 29224605, 29250230, 29250317, 29301463, 29339155
29347943, 29353821, 29376346, 29378913, 29379978, 29388020, 29405462
29409149, 29436454, 29437712, 29483672, 29483723, 29500257, 29511611
29542449, 29633753, 29637526, 29645349, 29676089, 29690625, 29707896

29724063, 29767177, 29782211, 29813494, 29836659, 29893132, 29902311
30088912, 30189023

Version 12.2.0.1.ru-2019-07.rur-2019-07.r1

Version 12.2.0.1.ru-2019-07.rur-2019-07.r1 includes the following:

- Patch 29757449: DATABASE JUL 2019 RELEASE UPDATE 12.2.0.1.190716
- Patch 29774415: OJVM RELEASE UPDATE 12.2.0.1.190716
- Patch 28125601: DSTv33 for RDBMS (TZDATA2018G)
- Patch 28127287: DSTv33 for OJVM (TZDATA2018G)
- PreUpgrade Jar: preupgrade_12201_cbuild_22_lf.zip
- Patch 29213893: DBMS_STATS FAILING WITH ERROR ORA-01422 WHEN GATHERING STATS FOR USER \$ TABLE
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR

Oracle release update 12.2.0.1.190716, released July 2019

Bugs fixed:

8480838, 8932139, 12763598, 13554903, 14221306, 14690846, 15931756
16002385, 16438495, 16727454, 16942578, 17027695, 17533661, 17947871
18308268, 18521691, 18594510, 18774543, 18878420, 19072655, 19211433
19285025, 19327292, 19526548, 19614243, 19647894, 19649997, 19702201
19721304, 20003668, 20087519, 20118035, 20120236, 20324049, 20436508
20532077, 20588486, 20591151, 20617383, 20620169, 20736227, 20756305
20866970, 20976443, 21089435, 21095391, 21143725, 21147908, 21159907
21178363, 21186167, 21197098, 21216226, 21320338, 21433452, 21479706
21517767, 21520266, 21547051, 21638600, 21744603, 21882528, 21981529
21985256, 22007324, 22070473, 22070853, 22072543, 22087683, 22104866
22179537, 22310426, 22347493, 22364044, 22367053, 22379010, 22446455
22495673, 22503283, 22503297, 22504793, 22522515, 22530986, 22564336
22568728, 22581771, 22594071, 22599050, 22628825, 22645009, 22654475
22700845, 22726044, 22729345, 22826067, 22843979, 22845846, 22864303
22898198, 22950945, 22970869, 22981722, 23019710, 23026585, 23035249
23055900, 23061453, 23065002, 23066146, 23080557, 23104033, 23105538
23110523, 23125560, 23126545, 23127945, 23151677, 23169712, 23177923
23179662, 23184263, 23197730, 23234232, 23237091, 23249829, 23271203
23278750, 23300142, 23306590, 23310101, 23312077, 23328639, 23336559
23481673, 23491861, 23499004, 23499160, 23521523, 23527363, 23533647
23548817, 23567857, 23572982, 23581777, 23588722, 23599216, 23600861
23602213, 23645516, 23665623, 23709062, 23715460, 23715518, 23730961
23733981, 23735292, 23741944, 23746128, 23749454, 23761724, 24010030
24289874, 24294174, 24303148, 24307571, 24308349, 24326444, 24326846
24328857, 24332831, 24334708, 24336249, 24337882, 24341675, 24343905
24345420, 24346821, 24348685, 24350620, 24352981, 24355111, 24357348
24368004, 24371491, 24373528, 24373756, 24374976, 24376875, 24376878
24385983, 24401351, 24403922, 24409977, 24415926, 24416451, 24421668
24423416, 24425056, 24425998, 24435982, 24437162, 24443539, 24457597
24460392, 24461826, 24467122, 24468470, 24470606, 24471079, 24471473
24473736, 24485034, 24485161, 24485174, 24486059, 24486237, 24509056
24534401, 24554533, 24555417, 24556967, 24560906, 24563422, 24570214
24570598, 24573817, 24578718, 24578797, 24589081, 24589590, 24591506
24593740, 24595699, 24600330, 24609592, 24609996, 24611527, 24616637
24617969, 24623975, 24624166, 24642495, 24654629, 24655717, 24664211
24668398, 24669189, 24674197, 24674955, 24676172, 24677696, 24680959
24689376, 24692973, 24693290, 24697323, 24699619, 24710696, 24713381

24714096, 24717183, 24717859, 24718260, 24719799, 24735430, 24737064
24737403, 24737581, 24744383, 24744686, 24757934, 24759556, 24760407
24764085, 24766309, 24786669, 24792678, 24793511, 24796092, 24797119
24800423, 24801152, 24802934, 24811725, 24812047, 24827228, 24827654
24831514, 24835919, 24841671, 24843188, 24844549, 24844841, 24845157
24848746, 24848923, 24850622, 24907917, 24908321, 24911709, 24912588
24922704, 24923080, 24923215, 24923338, 24923790, 24924667, 24926999
24929210, 24938784, 24940060, 24942749, 24953434, 24957555, 24960044
24965426, 24966594, 24966788, 24967993, 24968162, 24976007, 24978100
25022574, 25027852, 25028996, 25029022, 25029423, 25032818, 25034396
25036474, 25042823, 25044977, 25045228, 25050160, 25051465, 25051628
25057811, 25058080, 25062592, 25063971, 25065563, 25072986, 25078611
25086233, 25087436, 25092777, 25093872, 25098160, 25099339, 25099497
25099758, 25100063, 25100579, 25103996, 25107662, 25110233, 25114561
25120284, 25120668, 25120742, 25121089, 25123585, 25124363, 25129925
25140197, 25145163, 25145215, 25150925, 25159176, 25162645, 25164293
25166187, 25171041, 25171084, 25175723, 25176408, 25178032, 25178101
25178179, 25179774, 25182817, 25184555, 25186079, 25191872, 25192044
25192528, 25192729, 25199585, 25201454, 25202355, 25203656, 25206864
25207410, 25209912, 25210268, 25210499, 25211628, 25223839, 25224242
25225795, 25226665, 25227381, 25230870, 25230945, 25237577, 25240188
25240590, 25241448, 25241625, 25244807, 25248384, 25251648, 25257085
25259611, 25262869, 25263960, 25265499, 25283790, 25287072, 25293659
25296876, 25299227, 25299807, 25300427, 25303756, 25305405, 25307368
25309116, 25313154, 25313411, 25316758, 25317989, 25320555, 25323525
25328518, 25329664, 25335249, 25335360, 25335790, 25337332, 25337640
25348956, 25353983, 25356118, 25357142, 25360661, 25362958, 25367588
25367721, 25382812, 25383204, 25384462, 25386748, 25388896, 25392535
25395696, 25397936, 25398306, 25404202, 25405100, 25405687, 25405813
25410017, 25410180, 25410802, 25410877, 25411036, 25415713, 25417050
25417056, 25417958, 25425451, 25425760, 25427662, 25429959, 25430120
25433696, 25435038, 25437699, 25440818, 25442559, 25444961, 25445168
25451531, 25455795, 25457409, 25459958, 25462714, 25463844, 25472112
25476149, 25477657, 25478885, 25479164, 25482971, 25489342, 25489367
25489607, 25492379, 25498930, 25498994, 25516250, 25524955, 25528838
25530080, 25530814, 25535668, 25536819, 25537470, 25539063, 25540738
25546580, 25546608, 25547901, 25551676, 25553616, 25554787, 25555252
25557886, 25558986, 25560487, 25561296, 25569149, 25570929, 25575348
25575369, 25575628, 25579458, 25579761, 25594901, 25597525, 25598473
25599425, 25600342, 25600421, 25602488, 25603923, 25606091, 25607726
25612095, 25614866, 25616268, 25616359, 25616417, 25616645, 25631933
25633101, 25634317, 25634348, 25635149, 25638456, 25639019, 25643818
25643889, 25643931, 25646373, 25647325, 25648731, 25653109, 25654459
25654936, 25655390, 25655966, 25659655, 25660847, 25661819, 25662088
25662101, 25662524, 25663488, 25669791, 25670786, 25671354, 25672640
25674386, 25680221, 25685152, 25686739, 25687460, 25691904, 25694206
25695903, 25699321, 25700654, 25709368, 25710420, 25715167, 25717371
25722055, 25722608, 25722720, 25723158, 25728085, 25729507, 25734963
25736747, 25739065, 25741955, 25743479, 25747569, 25749273, 25752755
25754606, 25757748, 25760195, 25762221, 25764020, 25766822, 25768681
25772669, 25774077, 25775213, 25780343, 25783447, 25784002, 25785331
25785441, 25788879, 25789041, 25789277, 25789579, 25790353, 25792911
25795865, 25797092, 25797124, 25797305, 25800464, 25802510, 25803545
25807997, 25810263, 25810704, 25811650, 25813931, 25818707, 25822410
25823754, 25825910, 25826740, 25830492, 25832935, 25834581, 25835365
25838361, 25838755, 25852885, 25856821, 25858672, 25861398, 25865785
25870579, 25871177, 25871639, 25871753, 25872127, 25872389, 25873336
25874050, 25874678, 25882264, 25883438, 25885148, 25888073, 25888984
25890056, 25890673, 25894239, 25895224, 25897615, 25904273, 25904490
25906117, 25906886, 25908728, 25911724, 25914276, 25919622, 25932524
25932728, 25933494, 25941836, 25943271, 25945130, 25947799, 25951571
25953857, 25954022, 25954054, 25957038, 25963024, 25964954, 25967544
25967985, 25970731, 25971286, 25973152, 25975723, 25977302, 25980605
25980770, 25981498, 25982666, 25986062, 25990907, 25995938, 25997810
26006257, 26007010, 26019148, 26024732, 26024784, 26025681, 26029075
26029777, 26029780, 26032573, 26034119, 26036748, 26037215, 26038086

26039623, 26040483, 26045732, 26051656, 26078437, 26078493, 26080410
26083298, 26088426, 26088836, 26090767, 26091640, 26091786, 26095327
26095405, 26096382, 26108080, 26108337, 26110259, 26110632, 26111842
26112621, 26115103, 26121990, 26124078, 26137367, 26138085, 26149904
26153977, 26168933, 26169341, 26169345, 26170659, 26170715, 26176002
26187943, 26189861, 26198757, 26198926, 26201113, 26203182, 26223039
26237338, 26237431, 26237773, 26238195, 26242031, 26243698, 26244115
26245237, 26248143, 26249718, 26256131, 26257953, 26259265, 26261327
26263328, 26263721, 26268756, 26269790, 26271001, 26275023, 26275415
26277439, 26281476, 26285062, 26285933, 26308650, 26309047, 26317991
26318627, 26323308, 26324206, 26324769, 26327624, 26330994, 26331743
26333141, 26336977, 26338953, 26351334, 26353617, 26358670, 26359091
26362155, 26362821, 26366517, 26367012, 26367460, 26371725, 26374791
26375052, 26375250, 26380097, 26385189, 26388538, 26396790, 26399626
26399691, 26406387, 26407408, 26412540, 26418088, 26420561, 26421667
26422277, 26423085, 26426526, 26426967, 26430323, 26430737, 26434436
26434999, 26435073, 26436168, 26438612, 26439748, 26440169, 26440749
26442308, 26444601, 26444887, 26446098, 26452606, 26474662, 26474703
26475419, 26476244, 26478970, 26479173, 26482376, 26486365, 26492866
26493289, 26498354, 26513067, 26513709, 26521043, 26522439, 26523432
26526726, 26526799, 26536320, 26537307, 26542135, 26542236, 26544823
26545688, 26546070, 26546664, 26546754, 26548363, 26556014, 26569225
26575788, 26580633, 26582460, 26584641, 26597140, 26599395, 26608137
26608238, 26609942, 26615291, 26615690, 26617804, 26623652, 26626879
26629381, 26633355, 26633558, 26635897, 26637273, 26637824, 26639167
26641610, 26650226, 26654363, 26658759, 26659182, 26680105, 26712331
26714486, 26714910, 26717528, 26725687, 26727397, 26729494, 26729611
26740700, 26744595, 26745002, 26751106, 26751171, 26755171, 26758193
26764561, 26765212, 26775602, 26784509, 26794786, 26797591, 26798516
26802503, 26816582, 26820076, 26822620, 26828994, 26829845, 26833932
26837702, 26840654, 26844406, 26844870, 26849779, 26875822, 26883456
26895149, 26896659, 26898563, 26907327, 26908788, 26909100, 26909504
26911000, 26939314, 26943004, 26944190, 26958896, 26963310, 26966616
26966916, 26967713, 26969321, 26970175, 26970717, 26981902, 26983259
26985002, 26986173, 26992964, 27006120, 27006664, 27009164, 27013146
27028251, 27032785, 27034890, 27037839, 27038986, 27039712, 27044297
27052607, 27058530, 27060167, 27060859, 27061736, 27073314, 27079140
27087426, 27090765, 27092508, 27093423, 27097854, 27101105, 27105900
27106179, 27110878, 27115422, 27117822, 27119621, 27122162, 27124624
27125872, 27133662, 27134734, 27135647, 27135993, 27138325, 27142373
27142529, 27151826, 27153641, 27161071, 27162390, 27162405, 27163928
27165231, 27169796, 27170305, 27181537, 27181897, 27199245, 27200959
27207110, 27208795, 27213224, 27216046, 27223075, 27229389, 27231051
27236722, 27242226, 27244337, 27248917, 27249531, 27250547, 27251690
27255377, 27256000, 27258578, 27259307, 27262945, 27266245, 27274456
27274536, 27275533, 27276231, 27283960, 27284499, 27285244, 27288638
27292213, 27293599, 27302711, 27302730, 27303287, 27304410, 27305039
27308088, 27314206, 27314390, 27320576, 27321179, 27329612, 27333106
27334316, 27338912, 27338946, 27339115, 27345231, 27346709, 27348081
27349393, 27350267, 27351628, 27359178, 27364854, 27367194, 27370965
27375542, 27381498, 27386467, 27394703, 27395416, 27396624, 27396672
27396813, 27397048, 27400416, 27400598, 27404668, 27405645, 27416997
27423251, 27424405, 27426363, 27432062, 27433870, 27434193, 27439835
27441326, 27442041, 27445727, 27457891, 27466597, 27468303, 27493674
27501373, 27501413, 27502420, 27504770, 27505229, 27508985, 27510959
27525909, 27533819, 27534509, 27540613, 27544973, 27548131, 27554074
27555481, 27558861, 27560602, 27562488, 27565906, 27567477, 27576342
27593501, 27595973, 27607563, 27611612, 27613080, 27613530, 27617978
27620808, 27629756, 27634676, 27634991, 27658186, 27666312, 27671633
27680669, 27686599, 27687880, 27688036, 27688099, 27688692, 27691920
27691939, 27693416, 27693713, 27695063, 27698953, 27700466, 27704237
27709046, 27710072, 27726780, 27740424, 27748954, 27751755, 27757567
27757888, 27758972, 27759077, 27793533, 27799032, 27801337, 27824540
27824543, 27825241, 27829295, 27833672, 27834551, 27835925, 27846298
27846499, 27847259, 27855490, 27861226, 27882176, 27898015, 27931299
27940876, 27945870, 27951817, 27959048, 27959594, 27966472, 27986817

27994325, 27995215, 27995248, 27997875, 27998003, 27999073, 27999638
28000269, 28022101, 28023081, 28023399, 28023482, 28026866, 28033429
28040776, 28045903, 28067846, 28072567, 28074713, 28090453, 28099662
28108003, 28111583, 28120036, 28120951, 28124631, 28129791, 28140658
28165439, 28171079, 28174827, 28180464, 28181021, 28184554, 28188330
28194173, 28199085, 28201419, 28218832, 28220398, 28223871, 28226179
28229360, 28236305, 28271119, 28279837, 28282606, 28290434, 28302049
28305001, 28305362, 28320399, 28354603, 28357401, 28361221, 28365111
28378446, 28390273, 28396445, 28420042, 28420457, 28423598, 28432129
28434028, 28437315, 28454242, 28468312, 28483184, 28489150, 28501075
28502343, 28507324, 28508053, 28508557, 28522441, 28528349, 28535272
28545134, 28547068, 28571483, 28572834, 28578164, 28587723, 28600233
28612674, 28617631, 28617959, 28621470, 28627255, 28636676, 28691965
28692275, 28708023, 28728272, 28742555, 28749289, 28758722, 28774416
28803345, 28819640, 28849751, 28852691, 28856060, 28891984, 28951382
28956908, 28960211, 28965787, 28986231, 28987439, 28991884, 28993590
29006527, 29024054, 29027694, 29032276, 29125374, 29158680, 29189889
29200700, 29224605, 29250230, 29301463, 29339155, 29347943, 29376346
29378913, 29379978, 29388020, 29405462, 29436454, 29437712, 29511611
29542449, 29676089, 29690625, 29813494, 29836659

Version 12.2.0.1.ru-2019-04.rur-2019-04.r1

Version 12.2.0.1.ru-2019-04.rur-2019-04.r1 includes the following:

- Patch 29314339: Database Apr 2019 Release Update: 12.2.0.1.190416
- Patch 29249637: OJVM RELEASE UPDATE: 12.2.0.1.190416
- Patch 28852325: DSTv33 for RDBMS (TZDATA2018G)
- Patch 28852334: DSTv33 for OJVM (TZDATA2018G)
- PreUpgrade Jar: preupgrade_12201_cbuild_21_lf.zip
- Patch 28423598: GOLDENGATE AUTH CAUSES ACTIVE DG TO BE UNUSABLE UNTIL BOUNCE
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Support for the package `rdsadmin_dbms_goldengate_auth`, which provides GRANT privileges needed by a GoldenGate administrator account (see [Grant account privileges on the source DB \(p. 1230\)](#))

Oracle release update 12.2.0.1.190416, released April 2019

Bugs fixed:

26362155, 28023399, 25741955, 25873336, 26966616, 27097854, 28617631
28742555, 29006527, 8480838, 8932139, 12763598, 13554903, 14221306
14690846, 15931756, 16002385, 16438495, 16727454, 16942578, 17027695
17533661, 17947871, 18308268, 18521691, 18594510, 18774543, 18878420
19072655, 19211433, 19285025, 19327292, 19526548, 19614243, 19647894
19649997, 19702201, 19721304, 20003668, 20087519, 20118035, 20120236
20324049, 20436508, 20532077, 20591151, 20617383, 20620169, 20736227
20756305, 20866970, 20976443, 21089435, 21095391, 21143725, 21147908
21159907, 21178363, 21186167, 21197098, 21216226, 21320338, 21433452
21479706, 21520266, 21547051, 21638600, 21744603, 21882528, 21981529
21985256, 22007324, 22070853, 22072543, 22087683, 22104866, 22179537
22347493, 22364044, 22367053, 22379010, 22446455, 22495673, 22503283
22503297, 22504793, 22530986, 22564336, 22568728, 22581771, 22594071
22599050, 22628825, 22645009, 22654475, 22700845, 22726044, 22729345
22826067, 22843979, 22845846, 22864303, 22898198, 22950945, 22970869
22981722, 23019710, 23026585, 23035249, 23055900, 23061453, 23065002

23066146, 23080557, 23104033, 23105538, 23110523, 23125560, 23126545
23127945, 23151677, 23179662, 23184263, 23197730, 23234232, 23237091
23249829, 23271203, 23278750, 23300142, 23310101, 23312077, 23328639
23336559, 23481673, 23491861, 23499004, 23499160, 23521523, 23527363
23533647, 23548817, 23567857, 23572982, 23581777, 23588722, 23599216
23600861, 23602213, 23645516, 23665623, 23709062, 23715460, 23715518
23730961, 23733981, 23735292, 23741944, 23746128, 23749454, 23761724
24010030, 24289874, 24294174, 24303148, 24307571, 24308349, 24326444
24326846, 24328857, 24332831, 24334708, 24336249, 24337882, 24341675
24343905, 24345420, 24346821, 24348685, 24350620, 24352981, 24368004
24371491, 24373756, 24374976, 24376875, 24376878, 24385983, 24401351
24403922, 24415926, 24421668, 24423416, 24425056, 24425998, 24435982
24437162, 24443539, 24457597, 24460392, 24461826, 24467122, 24468470
24470606, 24471473, 24473736, 24485034, 24485161, 24485174, 24486059
24486237, 24509056, 24534401, 24554533, 24555417, 24556967, 24560906
24563422, 24570214, 24570598, 24573817, 24578718, 24578797, 24589081
24589590, 24593740, 24595699, 24600330, 24609592, 24609996, 24611527
24616637, 24617969, 24623975, 24624166, 24642495, 24654629, 24655717
24664211, 24668398, 24669189, 24674197, 24674955, 24676172, 24677696
24680959, 24689376, 24692973, 24693290, 24697323, 24699619, 24710696
24713381, 24714096, 24717183, 24717859, 24718260, 24719799, 24735430
24737064, 24737403, 24737581, 24744383, 24744686, 24757934, 24759556
24760407, 24764085, 24766309, 24786669, 24792678, 24793511, 24796092
24797119, 24800423, 24801152, 24802934, 24811725, 24812047, 24827228
24827654, 24831514, 24835919, 24843188, 24844549, 24845157, 24848746
24848923, 24850622, 24907917, 24908321, 24911709, 24912588, 24922704
24923080, 24923215, 24923338, 24923790, 24924667, 24926999, 24929210
24938784, 24940060, 24942749, 24953434, 24957555, 24960044, 24966594
24966788, 24968162, 24976007, 24978100, 25022574, 25027852, 25028996
25029022, 25029423, 25032818, 25034396, 25036474, 25042823, 25044977
25045228, 25050160, 25051628, 25057811, 25058080, 25062592, 25063971
25065563, 25072986, 25078611, 25086233, 25087436, 25093872, 25098160
25099339, 25099497, 25099758, 25100063, 25100579, 25103996, 25107662
25110233, 25114561, 25120284, 25120668, 25120742, 25121089, 25123585
25124363, 25129925, 25140197, 25145163, 25145215, 25150925, 25159176
25162645, 25164293, 25166187, 25171084, 25175723, 25176408, 25178032
25178101, 25178179, 25179774, 25182817, 25184555, 25186079, 25191872
25192044, 25192729, 25199585, 25201454, 25202355, 25203656, 25206864
25207410, 25209912, 25210268, 25210499, 25211628, 25223839, 25224242
25225795, 25226665, 25227381, 25230870, 25230945, 25237577, 25240188
25240590, 25241448, 25241625, 25244807, 25248384, 25251648, 25257085
25259611, 25262869, 25263960, 25265499, 25283790, 25287072, 25293659
25296876, 25299227, 25299807, 25300427, 25303756, 25305405, 25307368
25309116, 25313154, 25313411, 25316758, 25317989, 25320555, 25323525
25328518, 25329664, 25335249, 25335360, 25335790, 25337332, 25337640
25348956, 25353983, 25356118, 25357142, 25362958, 25367588, 25367721
25382812, 25383204, 25384462, 25386748, 25388896, 25392535, 25395696
25397936, 25398306, 25404202, 25405100, 25405687, 25405813, 25410017
25410180, 25410802, 25410877, 25411036, 25417050, 25417056, 25417958
25425451, 25425760, 25427662, 25429959, 25430120, 25433696, 25435038
25437699, 25440818, 25442559, 25444961, 25445168, 25451531, 25455795
25457409, 25459958, 25462714, 25463844, 25472112, 25476149, 25477657
25478885, 25479164, 25489342, 25489367, 25489607, 25492379, 25498930
25498994, 25516250, 25524955, 25528838, 25530080, 25530814, 25535668
25536819, 25537470, 25539063, 25540738, 25546580, 25546608, 25547901
25551676, 25553616, 25554787, 25555252, 25557886, 25558986, 25560487
25561296, 25569149, 25570929, 25575348, 25575628, 25579458, 25579761
25594901, 25597525, 25598473, 25599425, 25600342, 25600421, 25602488
25603923, 25606091, 25607726, 25612095, 25614866, 25616268, 25616359
25616417, 25616645, 25631933, 25633101, 25634317, 25634348, 25635149
25638456, 25639019, 25643818, 25643931, 25646373, 25647325, 25648731
25653109, 25654459, 25654936, 25655390, 25655966, 25659655, 25660847
25661819, 25662088, 25662101, 25662524, 25663488, 25669791, 25670786
25671354, 25672640, 25674386, 25680221, 25685152, 25686739, 25687460
25691904, 25694206, 25695903, 25700654, 25710420, 25715167, 25717371
25722055, 25722608, 25722720, 25728085, 25729507, 25734963, 25736747

25739065, 25743479, 25754606, 25757748, 25760195, 25762221, 25764020
25766822, 25768681, 25772669, 25774077, 25775213, 25780343, 25783447
25784002, 25785331, 25785441, 25788879, 25789041, 25789277, 25789579
25790353, 25795865, 25797092, 25797124, 25797305, 25800464, 25802510
25803545, 25807997, 25810704, 25813931, 25818707, 25822410, 25823754
25825910, 25826740, 25830492, 25832935, 25834581, 25838361, 25852885
25856821, 25858672, 25861398, 25865785, 25870579, 25871177, 25871639
25871753, 25872127, 25872389, 25874050, 25874678, 25882264, 25883438
25885148, 25888073, 25890056, 25890673, 25894239, 25895224, 25897615
25904273, 25904490, 25906117, 25908728, 25911724, 25914276, 25919622
25932524, 25932728, 25933494, 25941836, 25943271, 25945130, 25947799
25953857, 25954022, 25954054, 25957038, 25963024, 25964954, 25967544
25967985, 25970731, 25971286, 25973152, 25975723, 25977302, 25980605
25980770, 25981498, 25982666, 25986062, 25990907, 25995938, 26006257
26007010, 26019148, 26024732, 26025681, 26029075, 26029777, 26029780
26032573, 26034119, 26036748, 26037215, 26038086, 26039623, 26040483
26045732, 26051656, 26078437, 26078493, 26080410, 26083298, 26088426
26088836, 26090767, 26091640, 26091786, 26095327, 26095405, 26096382
26108080, 26108337, 26110259, 26110632, 26111842, 26112621, 26115103
26121990, 26124078, 26137367, 26138085, 26149904, 26153977, 26169341
26169345, 26170715, 26176002, 26187943, 26189861, 26198757, 26198926
26201113, 26203182, 26223039, 26237431, 26237773, 26238195, 26242031
26243698, 26244115, 26245237, 26248143, 26249718, 26256131, 26259265
26261327, 26263328, 26263721, 26269790, 26271001, 26277439, 26285062
26285933, 26308650, 26309047, 26318627, 26323308, 26324206, 26324769
26327624, 26330994, 26331743, 26333141, 26336977, 26338953, 26351334
26353617, 26358670, 26359091, 26362821, 26366517, 26367012, 26367460
26371725, 26374791, 26375250, 26380097, 26385189, 26388538, 26396790
26399626, 26399691, 26406387, 26407408, 26412540, 26418088, 26420561
26421667, 26422277, 26423085, 26426526, 26426967, 26430323, 26430737
26434436, 26434999, 26435073, 26436168, 26438612, 26439748, 26440169
26440749, 26442308, 26444601, 26444887, 26446098, 26452606, 26474703
26475419, 26476244, 26478970, 26479173, 26486365, 26492866, 26493289
26498354, 26513709, 26521043, 26522439, 26523432, 26526726, 26526799
26536320, 26537307, 26542135, 26544823, 26545688, 26546070, 26546664
26546754, 26548363, 26556014, 26569225, 26575788, 26580633, 26582460
26584641, 26597140, 26599395, 26608137, 26608238, 26609942, 26615291
26615690, 26617804, 26623652, 26626879, 26629381, 26633355, 26633558
26635897, 26637273, 26637824, 26639167, 26641610, 26650226, 26654363
26658759, 26659182, 26680105, 26712331, 26714486, 26714910, 26717528
26725687, 26727397, 26729494, 26729611, 26740700, 26744595, 26745002
26751106, 26751171, 26755171, 26758193, 26764561, 26765212, 26775602
26784509, 26794786, 26797591, 26798516, 26802503, 26816582, 26820076
26822620, 26828994, 26829845, 26833932, 26837702, 26840654, 26844406
26844870, 26849779, 26875822, 26883456, 26895149, 26896659, 26898563
26907327, 26908788, 26909100, 26909504, 26911000, 26939314, 26944190
26958896, 26963310, 26966916, 26967713, 26969321, 26970175, 26970717
26981902, 26983259, 26986173, 26992964, 27006120, 27006664, 27009164
27013146, 27028251, 27034890, 27037839, 27038986, 27039712, 27044297
27052607, 27058530, 27060167, 27060859, 27061736, 27073314, 27079140
27087426, 27090765, 27093423, 27106179, 27110878, 27115422, 27117822
27119621, 27122162, 27124624, 27125872, 27133662, 27135647, 27135993
27138325, 27142373, 27142529, 27151826, 27153641, 27161071, 27162390
27162405, 27163928, 27165231, 27169796, 27170305, 27181537, 27181897
27199245, 27200959, 27207110, 27208795, 27213224, 27216046, 27223075
27229389, 27231051, 27236722, 27242226, 27244337, 27248917, 27250547
27251690, 27255377, 27256000, 27258578, 27259307, 27262945, 27266245
27274456, 27274536, 27275533, 27276231, 27283960, 27284499, 27285244
27288638, 27292213, 27293599, 27302711, 27302730, 27304410, 27305039
27314206, 27314390, 27320576, 27321179, 27329612, 27333106, 27334316
27338912, 27338946, 27339115, 27345231, 27346709, 27348081, 27349393
27350267, 27351628, 27359178, 27364854, 27367194, 27370965, 27375542
27381498, 27386467, 27394703, 27395416, 27396624, 27396672, 27396813
27397048, 27400416, 27400598, 27404668, 27405645, 27416997, 27423251
27424405, 27426363, 27432062, 27433870, 27434193, 27439835, 27441326
27442041, 27445727, 27457891, 27466597, 27493674, 27501373, 27501413

27502420, 27504770, 27505229, 27508985, 27510959, 27525909, 27534509
27540613, 27544973, 27548131, 27554074, 27555481, 27558861, 27560602
27562488, 27565906, 27567477, 27593501, 27595973, 27607563, 27611612
27613080, 27613530, 27617978, 27620808, 27634676, 27634991, 27658186
27666312, 27671633, 27680669, 27686599, 27687880, 27688036, 27688099
27688692, 27691920, 27691939, 27693713, 27695063, 27698953, 27700466
27704237, 27709046, 27726780, 27740424, 27748954, 27751755, 27757567
27757888, 27758972, 27759077, 27793533, 27799032, 27801337, 27824543
27825241, 27829295, 27833672, 27834551, 27835925, 27847259, 27855490
27882176, 27898015, 27931299, 27940876, 27945870, 27951817, 27959048
27966472, 27986817, 27994325, 27995215, 27995248, 27997875, 27998003
27999638, 28000269, 28022101, 28023081, 28023482, 28033429, 28040776
28045903, 28067846, 28072567, 28074713, 28090453, 28099662, 28108003
28111583, 28120036, 28120951, 28140658, 28171079, 28174827, 28180464
28184554, 28188330, 28194173, 28199085, 28218832, 28220398, 28223871
28226179, 28229360, 28279837, 28282606, 28290434, 28302049, 28305001
28305362, 28320399, 28354603, 28361221, 28365111, 28378446, 28390273
28396445, 28420042, 28420457, 28432129, 28434028, 28437315, 28454242
28468312, 28483184, 28489150, 28502343, 28507324, 28508053, 28508557
28522441, 28528349, 28535272, 28545134, 28571483, 28578164, 28587723
28600233, 28617959, 28621470, 28627255, 28636676, 28691965, 28708023
28728272, 28749289, 28803345, 28849751, 28852691, 28856060, 28891984
28951382, 28960211, 28987439, 28991884, 28993590, 29027694, 29189889
29250230

Version 12.2.0.1.ru-2019-01.rur-2019-01.r1

Version 12.2.0.1.ru-2019-01.rur-2019-01.r1 includes the following:

- Patch 28822515: Database Jan 2019 Release Update: 12.2.0.1.190115
- Patch 28790651: OJVM RELEASE UPDATE: 12.2.0.1.190115
- Patch 28125601: DSTv32 for RDBMS (TZDATA2018E)
- Patch 28127287: DSTv32 for OJVM (TZDATA2018E)
- PreUpgrade Jar: preupgrade_12201_cbuild_19_lf.zip

Oracle release update 12.2.0.1.190115, released January 2019

Bugs fixed:

26285062, 8480838, 8932139, 13554903, 14221306, 14690846, 15931756
16002385, 16438495, 16727454, 16942578, 17027695, 17533661, 17947871
18308268, 18521691, 18594510, 18774543, 19072655, 19211433, 19285025
19327292, 19526548, 19614243, 19647894, 19649997, 19721304, 20003668
20087519, 20118035, 20120236, 20324049, 20436508, 20532077, 20591151
20620169, 20736227, 20756305, 2086970, 20976443, 21095391, 21143725
21147908, 21159907, 21178363, 21186167, 21197098, 21216226, 21320338
21433452, 21479706, 21520266, 21547051, 21744603, 21882528, 21981529
21985256, 22007324, 22070853, 22072543, 22087683, 22104866, 22179537
22347493, 22364044, 22367053, 22379010, 22446455, 22495673, 22503283
22503297, 22504793, 22530986, 22564336, 22568728, 22581771, 22594071
22599050, 22628825, 22645009, 22654475, 22700845, 22729345, 22826067
22843979, 22845846, 22864303, 22898198, 22950945, 22970869, 22981722
23019710, 23026585, 23035249, 23055900, 23061453, 23065002, 23066146
23080557, 23104033, 23105538, 23110523, 23125560, 23126545, 23127945
23151677, 23179662, 23184263, 23197730, 23234232, 23249829, 23271203
23300142, 23310101, 23312077, 23481673, 23491861, 23499004, 23499160
23521523, 23527363, 23533647, 23548817, 23567857, 23572982, 23581777
23588722, 23599216, 23600861, 23602213, 23645516, 23665623, 23709062

23715460, 23730961, 23733981, 23735292, 23741944, 23746128, 23749454
23761724, 24010030, 24289874, 24294174, 24303148, 24307571, 24308349
24326444, 24326846, 24332831, 24334708, 24336249, 24337882, 24341675
24343905, 24345420, 24346821, 24348685, 24350620, 24352981, 24368004
24371491, 24373756, 24374976, 24376875, 24376878, 24385983, 24401351
24403922, 24415926, 24421668, 24423416, 24425056, 24425998, 24435982
24437162, 24443539, 24457597, 24460392, 24461826, 24467122, 24468470
24470606, 24471473, 24473736, 24485034, 24485161, 24485174, 24486059
24486237, 24509056, 24534401, 24554533, 24555417, 24556967, 24560906
24563422, 24570214, 24570598, 24573817, 24578718, 24578797, 24589081
24589590, 24593740, 24595699, 24600330, 24609592, 24609996, 24616637
24617969, 24623975, 24624166, 24642495, 24654629, 24655717, 24664211
24668398, 24669189, 24674197, 24674955, 24676172, 24677696, 24680959
24689376, 24692973, 24693290, 24697323, 24699619, 24710696, 24713381
24714096, 24717183, 24717859, 24718260, 24719799, 24735430, 24737064
24737403, 24737581, 24744383, 24744686, 24757934, 24759556, 24760407
24764085, 24766309, 24786669, 24792678, 24793511, 24796092, 24797119
24800423, 24801152, 24802934, 24811725, 24812047, 24827228, 24827654
24831514, 24835919, 24843188, 24844549, 24845157, 24848746, 24848923
24850622, 24907917, 24908321, 24911709, 24912588, 24922704, 24923080
24923215, 24923338, 24923790, 24929210, 24938784, 24940060, 24942749
24953434, 24957555, 24960044, 24966594, 24966788, 24968162, 24976007
24978100, 25022574, 25027852, 25028996, 25029022, 25029423, 25034396
25036474, 25044977, 25045228, 25050160, 25051628, 25057811, 25058080
25062592, 25063971, 25065563, 25072986, 25078611, 25086233, 25087436
25093872, 25098160, 25099339, 25099497, 25099758, 25100063, 25100579
25103996, 25107662, 25110233, 25114561, 25120284, 25120668, 25120742
25121089, 25123585, 25124363, 25129925, 25140197, 25145163, 25145215
25150925, 25159176, 25162645, 25164293, 25166187, 25171084, 25175723
25176408, 25178032, 25178101, 25178179, 25179774, 25182817, 25184555
25186079, 25191872, 25192044, 25192729, 25199585, 25201454, 25202355
25203656, 25206864, 25207410, 25209912, 25210268, 25210499, 25211628
25223839, 25224242, 25225795, 25226665, 25227381, 25230870, 25230945
25237577, 25240188, 25240590, 25241448, 25241625, 25244807, 25248384
25251648, 25257085, 25259611, 25262869, 25263960, 25265499, 25283790
25287072, 25296876, 25299227, 25299807, 25300427, 25305405, 25307368
25309116, 25313154, 25313411, 25316758, 25317989, 25320555, 25323525
25328518, 25329664, 25335249, 25335360, 25335790, 25337332, 25337640
25348956, 25353983, 25356118, 25357142, 25362958, 25382812, 25383204
25384462, 25386748, 25388896, 25392535, 25395696, 25397936, 25405813
25410017, 25410180, 25410802, 25410877, 25411036, 25417050, 25417056
25417958, 25425451, 25425760, 25427662, 25429959, 25430120, 25433696
25435038, 25437699, 25440818, 25444961, 25445168, 25451531, 25455795
25457409, 25459958, 25462714, 25463844, 25472112, 25476149, 25478885
25479164, 25489342, 25489367, 25489607, 25492379, 25498930, 25498994
25516250, 25524955, 25528383, 25530080, 25530814, 25535668, 25536819
25537470, 25539063, 25540738, 25546580, 25546608, 25547901, 25551676
25553616, 25554787, 25555252, 25557886, 25558986, 25560487, 25561296
25569149, 25570929, 25575348, 25575628, 25579458, 25579761, 25594901
25597525, 25598473, 25599425, 25600342, 25600421, 25602488, 25603923
25606091, 25607726, 25612095, 25614866, 25616268, 25616359, 25616417
25616645, 25631933, 25633101, 25634317, 25634348, 25635149, 25638456
25639019, 25643818, 25643931, 25646373, 25647325, 25648731, 25653109
25654459, 25654936, 25655390, 25655966, 25659655, 25660847, 25661819
25662088, 25662101, 25662524, 25669791, 25670786, 25671354, 25672640
25674386, 25680221, 25685152, 25686739, 25687460, 25691904, 25694206
25695903, 25700654, 25710420, 25715167, 25717371, 25722055, 25722608
25722720, 25728085, 25729507, 25734963, 25736747, 25739065, 25754606
25757748, 25760195, 25762221, 25764020, 25766822, 25768681, 25772669
25774077, 25775213, 25780343, 25784002, 25785331, 25785441, 25788879
25789041, 25789277, 25789579, 25790353, 25797092, 25797124, 25797305
25800464, 25802510, 25803545, 25807997, 25810704, 25813931, 25818707
25822410, 25823754, 25825910, 25826740, 25830492, 25832935, 25834581
25838361, 25852885, 25856821, 25858672, 25861398, 25865785, 25870579
25871177, 25871639, 25871753, 25872127, 25872389, 25874050, 25874678
25882264, 25885148, 25888073, 25890056, 25890673, 25894239, 25895224

25897615, 25904273, 25904490, 25906117, 25911724, 25914276, 25919622
25932524, 25932728, 25933494, 25941836, 25943271, 25945130, 25947799
25953857, 25954022, 25954054, 25957038, 25963024, 25964954, 25967544
25967985, 25970731, 25973152, 25975723, 25977302, 25980605, 25980770
25981498, 25982666, 25990907, 25995938, 26006257, 26007010, 26019148
26024732, 26025681, 26029780, 26032573, 26034119, 26036748, 26037215
26038086, 26039623, 26040483, 26045732, 26078437, 26078493, 26080410
26083298, 26088426, 26088836, 26090767, 26091640, 26091786, 26095327
26095405, 26096382, 26108080, 26108337, 26110632, 26111842, 26112621
26115103, 26121990, 26124078, 26137367, 26138085, 26149904, 26153977
26169341, 26169345, 26170715, 26176002, 26187943, 26189861, 26198757
26198926, 26201113, 26203182, 26223039, 26237431, 26237773, 26238195
26242031, 26243698, 26244115, 26245237, 26248143, 26249718, 26256131
26259265, 26261327, 26263328, 26263721, 26269790, 26271001, 26277439
26285933, 26308650, 26309047, 26318627, 26323308, 26324769, 26327624
26330994, 26331743, 26333141, 26336977, 26338953, 26351334, 26353617
26358670, 26359091, 26362821, 26366517, 26367012, 26371725, 26374791
26375250, 26380097, 26385189, 26388538, 26396790, 26399626, 26399691
26406387, 26407408, 26412540, 26418088, 26420561, 26421667, 26422277
26423085, 26426526, 26426967, 26430737, 26434436, 26434999, 26435073
26436168, 26438612, 26440749, 26442308, 26444601, 26444887, 26446098
26452606, 26474703, 26475419, 26476244, 26478970, 26479173, 26486365
26492866, 26493289, 26498354, 26513709, 26521043, 26522439, 26523432
26526726, 26536320, 26537307, 26542135, 26544823, 26545688, 26546070
26546664, 26546754, 26548363, 26556014, 26569225, 26575788, 26580633
26582460, 26584641, 26597140, 26599395, 26608137, 26609942, 26615291
26615690, 26623652, 26626879, 26629381, 26633355, 26633558, 26635897
26637273, 26637824, 26639167, 26641610, 26650226, 26658759, 26659182
26680105, 26712331, 26714486, 26714910, 26717528, 26725687, 26727397
26729494, 26729611, 26740700, 26744595, 26745002, 26751106, 26751171
26755171, 26758193, 26764561, 26765212, 26775602, 26784509, 26794786
26797591, 26798516, 26802503, 26820076, 26822620, 26828994, 26840654
26844406, 26844870, 26849779, 26875822, 26883456, 26895149, 26896659
26898563, 26907327, 26908788, 26909100, 26909504, 26911000, 26939314
26944190, 26958896, 26963310, 26966916, 26967713, 26969321, 26970717
26981902, 26983259, 26986173, 26992964, 27006664, 27009164, 27013146
27028251, 27034890, 27037839, 27038986, 27039712, 27044297, 27052607
27058530, 27060167, 27060859, 27073314, 27079140, 27087426, 27090765
27093423, 27110878, 27117822, 27119621, 27122162, 27124624, 27125872
27133662, 27135647, 27135993, 27138325, 27142373, 27142529, 27151826
27153641, 27161071, 27162405, 27163928, 27165231, 27169796, 27170305
27181537, 27181897, 27199245, 27200959, 27207110, 27213224, 27223075
27229389, 27236722, 27244337, 27248917, 27250547, 27251690, 27255377
27256000, 27258578, 27259307, 27262945, 27266245, 27274456, 27274536
27276231, 27283960, 27285244, 27292213, 27293599, 27302711, 27302730
27304410, 27305039, 27314206, 27314390, 27320576, 27321179, 27329612
27333106, 27334316, 27338912, 27338946, 27339115, 27345231, 27346709
27348081, 27349393, 27351628, 27359178, 27367194, 27370965, 27375542
27381498, 27386467, 27394703, 27395416, 27396624, 27396813, 27397048
27400416, 27400598, 27404668, 27405645, 27416997, 27433870, 27434193
27439835, 27441326, 27442041, 27445727, 27466597, 27493674, 27501373
27501413, 27502420, 27504770, 27505229, 27508985, 27510959, 27525909
27534509, 27540613, 27544973, 27548131, 27555481, 27558861, 27560602
27567477, 27593501, 27595973, 27607563, 27611612, 27613080, 27613530
27617978, 27620808, 27634676, 27658186, 27687880, 27688036, 27688099
27688692, 27691920, 27691939, 27693713, 27695063, 27698953, 27700466
27709046, 27726780, 27740424, 27748954, 27757888, 27759077, 27793533
27799032, 27801337, 27835925, 27847259, 27855490, 27882176, 27898015
27931299, 27940876, 27945870, 27951817, 27959048, 27994325, 27995248
27997875, 27998003, 27999638, 28000269, 28022101, 28023081, 28023482
28033429, 28040776, 28067846, 28072567, 28074713, 28090453, 28099662
28111583, 28120951, 28140658, 28171079, 28174827, 28180464, 28184554
28188330, 28194173, 28218832, 28220398, 28226179, 28229360, 28282606
28290434, 28302049, 28305001, 28305362, 28320399, 28354603, 28361221
28365111, 28390273, 28396445, 28420042, 28420457, 28434028, 28437315
28454242, 28483184, 28489150, 28502343, 28508053, 28508557, 28522441

28528349, 28535272, 28571483, 28617959, 28621470, 28627255, 28636676
28691965, 28708023, 28728272, 28749289, 28960211, 28993590

Version 12.2.0.1.ru-2018-10.rur-2018-10.r1

Version 12.2.0.1.ru-2018-10.rur-2018-10.r1 includes the following:

- October 2018 Release Update: 12.2.0.1.181016 (28662603)

Oracle release update 12.2.0.1.181016, released October 2018

Bugs fixed:

28390273, 28571483, 28483184, 8480838, 13554903, 14221306, 14690846
15931756, 16002385, 16438495, 16727454, 16942578, 17027695, 17533661
17947871, 18308268, 18521691, 18594510, 18774543, 19072655, 19211433
19285025, 19327292, 19526548, 19614243, 19647894, 19649997, 19721304
20003668, 20087519, 20118035, 20120236, 20324049, 20436508, 20532077
20591151, 20620169, 20736227, 20756305, 20866970, 20976443, 21143725
21147908, 21159907, 21178363, 21186167, 21216226, 21320338, 21433452
21479706, 21520266, 21547051, 21744603, 21882528, 21981529, 21985256
22007324, 22070853, 22072543, 22087683, 22104866, 22179537, 22347493
22364044, 22367053, 22379010, 22446455, 22495673, 22503283, 22503297
22504793, 22530986, 22564336, 22568728, 22581771, 22594071, 22599050
22628825, 22645009, 22654475, 22700845, 22729345, 22826067, 22843979
22845846, 22864303, 22898198, 22950945, 22970869, 22981722, 23019710
23026585, 23035249, 23055900, 23061453, 23065002, 23066146, 23080557
23105538, 23110523, 23125560, 23126545, 23127945, 23151677, 23179662
23184263, 23197730, 23234232, 23249829, 23271203, 23300142, 23310101
23312077, 23481673, 23491861, 23499160, 23521523, 23527363, 23533647
23548817, 23567857, 23572982, 23581777, 23588722, 23599216, 23600861
23602213, 23645516, 23665623, 23709062, 23715460, 23730961, 23733981
23735292, 23741944, 23746128, 23749454, 24010030, 24289874, 24294174
24303148, 24307571, 24308349, 24326444, 24326846, 24332831, 24334708

24336249, 24337882, 24341675, 24343905, 24345420, 24346821, 24348685
24350620, 24368004, 24371491, 24373756, 24374976, 24376875, 24376878
24385983, 24401351, 24403922, 24415926, 24421668, 24423416, 24425056
24425998, 24435982, 24437162, 24443539, 24457597, 24461826, 24467122
24468470, 24470606, 24473736, 24485034, 24485161, 24485174, 24486059
24486237, 24509056, 24534401, 24554533, 24555417, 24556967, 24560906
24563422, 24570598, 24573817, 24578718, 24578797, 24589081, 24589590
24593740, 24595699, 24600330, 24609592, 24609996, 24616637, 24617969
24623975, 24624166, 24642495, 24654629, 24655717, 24664211, 24668398
24674197, 24674955, 24676172, 24677696, 24680959, 24689376, 24692973
24693290, 24699619, 24710696, 24713381, 24714096, 24717183, 24717859
24718260, 24719799, 24735430, 24737064, 24737403, 24737581, 24744383
24744686, 24757934, 24759556, 24760407, 24766309, 24786669, 24792678
24793511, 24796092, 24797119, 24800423, 24801152, 24802934, 24811725
24812047, 24827228, 24827654, 24831514, 24835919, 24843188, 24844549
24845157, 24848746, 24848923, 24850622, 24907917, 24908321, 24911709
24912588, 24922704, 24923080, 24923215, 24923338, 24923790, 24929210
24938784, 24940060, 24942749, 24953434, 24957555, 24960044, 24966594
24966788, 24968162, 24976007, 24978100, 25027852, 25029022, 25029423
25034396, 25036474, 25044977, 25045228, 25050160, 25051628, 25057811
25058080, 25062592, 25063971, 25065563, 25072986, 25078611, 25086233
25087436, 25093872, 25098160, 25099339, 25099497, 25099758, 25100063
25100579, 25103996, 25107662, 25110233, 25120284, 25120742, 25121089
25123585, 25124363, 25129925, 25140197, 25145163, 25145215, 25150925
25159176, 25162645, 25164293, 25166187, 25171084, 25175723, 25176408

25178032, 25178101, 25178179, 25179774, 25182817, 25184555, 25186079
25191872, 25192044, 25192729, 25199585, 25201454, 25202355, 25203656
25206864, 25207410, 25209912, 25210268, 25210499, 25211628, 25223839
25224242, 25225795, 25226665, 25227381, 25230945, 25237577, 25240590
25241448, 25241625, 25244807, 25248384, 25251648, 25257085, 25259611
25262869, 25263960, 25265499, 25283790, 25287072, 25296876, 25299227
25299807, 25300427, 25305405, 25307368, 25309116, 25313154, 25313411
25316758, 25317989, 25320555, 25323525, 25328518, 25329664, 25335249
25335360, 25335790, 25337332, 25337640, 25348956, 25353983, 25357142
25362958, 25382812, 25383204, 25384462, 25386748, 25388896, 25392535
25395696, 25397936, 25405813, 25410017, 25410180, 25410802, 25410877
25411036, 25417050, 25417056, 25417958, 25425451, 25425760, 25427662
25429959, 25430120, 25433696, 25435038, 25437699, 25440818, 25444961
25451531, 25455795, 25457409, 25459958, 25462714, 25463844, 25472112
25476149, 25478885, 25479164, 25489342, 25489367, 25489607, 25492379
25498930, 25498994, 25516250, 25524955, 25528838, 25530080, 25530814

25535668, 25536819, 25537470, 25539063, 25540738, 25546580, 25546608
25547901, 25551676, 25553616, 25554787, 25555252, 25557886, 25558986
25560487, 25561296, 25569149, 25570929, 25575348, 25575628, 25579458
25579761, 25594901, 25597525, 25598473, 25600342, 25600421, 25602488
25603923, 25606091, 25607726, 25612095, 25614866, 25616268, 25616359
25616417, 25616645, 25631933, 25633101, 25634317, 25634348, 25635149
25638456, 25639019, 25643818, 25643931, 25646373, 25647325, 25648731
25653109, 25654459, 25654936, 25655390, 25655966, 25659655, 25660847
25661819, 25662088, 25662101, 25662524, 25669791, 25670786, 25672640
25674386, 25680221, 25685152, 25686739, 25687460, 25691904, 25694206
25695903, 25700654, 25710420, 25715167, 25717371, 25722055, 25722608
25722720, 25728085, 25729507, 25734963, 25736747, 25739065, 25754606
25757748, 25760195, 25762221, 25764020, 25766822, 25768681, 25772669
25774077, 25775213, 25780343, 25784002, 25785331, 25785441, 25788879
25789041, 25789277, 25789579, 25790353, 25797092, 25797124, 25797305
25800464, 25803545, 25807997, 25810704, 25813931, 25818707, 25822410
25823754, 25825910, 25826740, 25830492, 25832935, 25834581, 25838361
25852885, 25856821, 25858672, 25861398, 25865785, 25870579, 25871177
25871639, 25871753, 25872127, 25872389, 25874050, 25874678, 25882264
25885148, 25888073, 25890056, 25890673, 25894239, 25895224, 25897615
25904273, 25904490, 25906117, 25911724, 25914276, 25919622, 25932524
25932728, 25933494, 25941836, 25943271, 25945130, 25947799, 25953857
25954022, 25954054, 25957038, 25963024, 25964954, 25967544, 25967985
25970731, 25973152, 25975723, 25977302, 25980605, 25980770, 25981498
25982666, 25990907, 25995938, 26006257, 26007010, 26019148, 26024732
26025681, 26029780, 26032573, 26036748, 26037215, 26038086, 26039623
26040483, 26045732, 26078437, 26078493, 26080410, 26083298, 26088426
26088836, 26090767, 26091640, 26091786, 26095327, 26095405, 26096382
26108080, 26110632, 26111842, 26121990, 26137367, 26138085, 26149904
26153977, 26169341, 26169345, 26170715, 26176002, 26187943, 26189861
26198757, 26198926, 26201113, 26223039, 26237431, 26237773, 26242031
26243698, 26244115, 26245237, 26249718, 26256131, 26259265, 26261327
26263328, 26263721, 26269790, 26271001, 26277439, 26285933, 26308650
26309047, 26318627, 26323308, 26324769, 26327624, 26330994, 26331743
26333141, 26338953, 26351334, 26353617, 26358670, 26362821, 26366517
26367012, 26374791, 26375250, 26380097, 26385189, 26388538, 26396790
26399626, 26399691, 26406387, 26412540, 26418088, 26420561, 26421667
26422277, 26426526, 26430737, 26434999, 26435073, 26436168, 26438612
26440749, 26442308, 26444601, 26444887, 26446098, 26452606, 26475419
26476244, 26478970, 26479173, 26486365, 26492866, 26493289, 26498354
26513709, 26521043, 26522439, 26523432, 26526726, 26536320, 26537307
26542135, 26544823, 26545688, 26546070, 26546664, 26546754, 26548363
26556014, 26569225, 26575788, 26582460, 26584641, 26597140, 26599395
26608137, 26609942, 26615291, 26615690, 26623652, 26626879, 26629381
26633355, 26633558, 26635897, 26637273, 26637824, 26639167, 26641610
26650226, 26658759, 26659182, 26680105, 26712331, 26714910, 26717528
26727397, 26729494, 26729611, 26740700, 26744595, 26751106, 26751171
26758193, 26764561, 26765212, 26775602, 26784509, 26794786, 26797591
26802503, 26820076, 26822620, 26828994, 26840654, 26844870, 26849779

26875822, 26883456, 26896659, 26898563, 26907327, 26908788, 26909100
26909504, 26911000, 26939314, 26944190, 26963310, 26966916, 26967713
26969321, 26970717, 26981902, 26983259, 26986173, 26992964, 27006664
27009164, 27013146, 27028251, 27034890, 27038986, 27039712, 27044297
27052607, 27060167, 27060859, 27073314, 27079140, 27087426, 27090765
27093423, 27110878, 27117822, 27119621, 27124624, 27125872, 27133662
27135647, 27135993, 27138325, 27142373, 27153641, 27161071, 27162405
27163928, 27165231, 27169796, 27170305, 27181537, 27199245, 27207110
27213224, 27229389, 27244337, 27248917, 27250547, 27251690, 27255377
27256000, 27259307, 27262945, 27274536, 27276231, 27285244, 27292213
27293599, 27302711, 27302730, 27304410, 27305039, 27314206, 27314390
27321179, 27329612, 27333106, 27334316, 27338912, 27338946, 27339115
27345231, 27346709, 27348081, 27349393, 27351628, 27359178, 27367194
27370965, 27375542, 27394703, 27395416, 27396624, 27396813, 27400598
27405645, 27416997, 27433870, 27434193, 27439835, 27441326, 27442041
27466597, 27493674, 27501373, 27501413, 27502420, 27504770, 27505229
27508985, 27510959, 27534509, 27544973, 27548131, 27555481, 27558861
27560602, 27567477, 27595973, 27607563, 27611612, 27613080, 27620808
27687880, 27688036, 27688692, 27691920, 27691939, 27698953, 27700466
27709046, 27726780, 27740424, 27748954, 27757888, 27799032, 27835925
27847259, 27882176, 27898015, 27940876, 27945870, 27951817, 27959048
27994325, 27997875, 27998003, 28000269, 28033429, 28040776, 28074713
28090453, 28099662, 28140658, 28171079, 28174827, 28184554, 28188330
28218832, 28226179, 28290434, 28305001, 28320399, 28354603, 28437315
28454242, 28508557, 28522441

Database engine: 12.1.0.2

The following versions are available for database engine 12.1.0.2:

- [Version 12.1.0.2.v23 \(p. 1360\)](#)
- [Version 12.1.0.2.v22 \(p. 1364\)](#)
- [Version 12.1.0.2.v21 \(p. 1368\)](#)
- [Version 12.1.0.2.v20 \(p. 1372\)](#)
- [Version 12.1.0.2.v19 \(p. 1375\)](#)
- [Version 12.1.0.2.v18 \(p. 1379\)](#)
- [Version 12.1.0.2.v17 \(p. 1382\)](#)
- [Version 12.1.0.2.v16 \(p. 1384\)](#)
- [Version 12.1.0.2.v15 \(p. 1387\)](#)
- [Version 12.1.0.2.v14 \(p. 1389\)](#)
- [Version 12.1.0.2.v13 \(p. 1392\)](#)
- [Version 12.1.0.2.v12 \(p. 1394\)](#)
- [Version 12.1.0.2.v11 \(p. 1396\)](#)
- [Version 12.1.0.2.v10 \(p. 1398\)](#)
- [Version 12.1.0.2.v9 \(p. 1400\)](#)
- [Version 12.1.0.2.v8 \(p. 1401\)](#)
- [Version 12.1.0.2.v7 \(p. 1403\)](#)
- [Version 12.1.0.2.v6 \(p. 1405\)](#)
- [Version 12.1.0.2.v5 \(p. 1406\)](#)
- [Version 12.1.0.2.v4 \(p. 1407\)](#)
- [Version 12.1.0.2.v3 \(p. 1408\)](#)
- [Version 12.1.0.2.v2 \(p. 1409\)](#)

- [Version 12.1.0.2.v1 \(p. 1410\)](#)

Version 12.1.0.2.v23

Version 12.1.0.2.v23 includes the following:

- Patch 31985579: DATABASE PATCH SET UPDATE 12.1.0.2.210119
- Patch 32119956: OJVM PATCH SET UPDATE 12.1.0.2.210119
- Patch 31335037: RDBMS - DSTV35 UPDATE - TZDATA2020A
- Patch 31335142: DSTV35 UPDATE - TZDATA2020A - NEED OJVM FIX
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 17969866: Oracle GoldenGate – 46719 ENH REPPLICATION SUPPORT FOR INSERTS / FULL UPDATES WITH LARGE VALUES
- Patch 20394750: Oracle GoldenGate – APPLY CDR RESOLUTION FAILING FOR LOBS, XML, LONG, AND OBJECTS
- Patch 24835919: Oracle GoldenGate – IR EXECUTING DEPENDENT TRANSACTIONS OUT OF ORDER WITH PARALLELISM GREATER THAN
- Patch 23262847: Oracle GoldenGate - MALFORMED REDO CAUSED OGG REPPLICATION ABEND
- Patch 21171382: ADD CONTROL FOR AUTOMATIC CREATION OF STATS EXTENSIONS
- Patch 21091901: ONLINE MOVE OF HASH OR REF PARTITION CAN LEAVE LOCAL INDEXES INCONSISTENT
- Patch 25031502: MV QUERY REWRITE WORKLOAD HIT ORA 600 [KGL-HEAP-SIZE-EXCEEDED]
- Patch 23711335: CDB_UPG PDCDB CDB UPGRADE AS WHOLE TAKES 1 MORE HOUR THAN PREVIOUS LABELS IN MAY
- Patch 32327179: JSON Bundle Patch
- PreUpgrade Jar: preupgrade_12.1.0.2.0_18_crlf.zip
- Java Cryptography Extension (JCE): Unlimited Strength Jurisdiction Policy Files for JVM version 6

Combined patches for version 12.1.0.2.v23, released February 2021

Bugs fixed:

```
6194865, 6418158, 6599380, 13542050, 13787015, 13854364, 14283239
14643995, 14705949, 16090440, 16354467, 16359751, 16439813, 16619249
16756406, 16777441, 16799735, 16863642, 16870214, 16875041, 16887946
16923858, 16938780, 16941434, 17008068, 17210525, 17258582, 17274537
17319928, 17365043, 17409174, 17414008, 17432124, 17495022, 17532729
17532734, 17533661, 17551063, 17655240, 17722075, 17760068, 17835294
17867700, 17890099, 17969866, 18007682, 18043064, 18051556, 18090142
18110491, 18122373, 18191823, 18202441, 18250893, 18254023, 18272672
18288842, 18306996, 18307021, 18308268, 18324100, 18354830, 18371441
18373438, 18382302, 18411216, 18417036, 18419520, 18427406, 18436647
18440095, 18456643, 18475439, 18492302, 18499088, 18510194, 18542562
18548246, 18548433, 18549238, 18604493, 18604692, 18607546, 18610915
18618122, 18628388, 18648816, 18662619, 18673090, 18674024, 18674047
18681056, 18693124, 18700762, 18705806, 18727933, 18733351, 18740837
18742258, 18743542, 18758877, 18759211, 18774543, 18775971, 18778801
18791688, 18797519, 18798250, 18799063, 18799993, 18801391, 18803726
18810904, 18818069, 18819908, 18840932, 18841764, 18845653, 18849537
18849970, 18851894, 18856106, 18856999, 18866977, 18868646, 18885870
18886413, 18893947, 18895170, 18899974, 18900107, 18904062, 18909599
18913440, 18914624, 18921743, 18940497, 18948177, 18952766, 18952989
18964939, 18964978, 18966843, 18967382, 18973548, 18974476, 18988834
```

18990023, 18990693, 18999568, 19001359, 19001390, 19012119, 19013183
19016730, 19017309, 19018206, 19018447, 19022470, 19023822, 19024808
19028800, 19032777, 19035573, 19044962, 19048007, 19050649, 19052488
19054077, 19058490, 19060015, 19065556, 19065677, 19067244, 19068380
19068610, 19068970, 19074147, 19075256, 19076343, 19077215, 19079752
19081128, 19124336, 19124589, 19130152, 19131386, 19131607, 19134173
19141838, 19143550, 19146474, 19149990, 19153980, 19154375, 19155797
19157754, 19165673, 19168167, 19171086, 19174430, 19174521, 19174942
19176223, 19176326, 19176885, 19178851, 19180394, 19180770, 19183343
19185876, 19188385, 19188927, 19189317, 19189525, 19195895, 19197175
19201867, 19207117, 19211433, 19213447, 19223010, 19231857, 19238590
19243521, 19245191, 19248279, 19248799, 19258504, 19272708, 19279273
19280225, 19284031, 19285025, 19289642, 19291380, 19297917, 19303936
19304354, 19306797, 19307662, 19308965, 19309466, 19313563, 19315668
19315691, 19317646, 19326908, 19327391, 19329654, 19330795, 19332396
19333670, 19335438, 19339555, 19347458, 19354335, 19354794, 19358317
19363645, 19364502, 19366375, 19370504, 19371175, 19373893, 19375649
19382851, 19383839, 19385656, 19390567, 19390620, 19393542, 19396455
19399918, 19402853, 19404068, 19409212, 19430401, 19433930, 19434529
19439759, 19440520, 19440586, 19445860, 19448499, 19450116, 19450314
19452434, 19461270, 19461428, 19468347, 19468612, 19468991, 19469538
19475971, 19487147, 19490948, 19501299, 19503821, 19512341, 19516448
19518079, 19520602, 19523462, 19524158, 19524384, 19529868, 19532017
19534363, 19536415, 19543384, 19547370, 19547774, 19548064, 19550902
19561643, 19562381, 19566592, 19571055, 19571082, 19571367, 19577410
19578247, 19578350, 19583624, 19587324, 19590877, 19591608, 19593445
19597439, 19597583, 19601762, 19604659, 19606174, 19617921, 19619732
19623450, 19627012, 19632912, 19637186, 19639483, 19644859, 19647503
19649152, 19658708, 19662635, 19663176, 19670108, 19676012, 19676905
19680796, 19684504, 19687159, 19689979, 19693090, 19699191, 19699946
19701015, 19703301, 19705781, 19706965, 19708342, 19708632, 19718981
19721304, 19723336, 19730508, 19769480, 19769625, 19777862, 19781326
19784751, 19790243, 19791273, 19791377, 19799847, 19805359, 19809171
19811709, 19817386, 19818513, 19824871, 19831647, 19835133, 19841800
19855285, 19859472, 19865345, 19869255, 19871910, 19873610, 19877336
19879746, 19880190, 19883092, 19886165, 19888853, 19889230, 19891090
19895326, 19895362, 19896336, 19902195, 19908836, 19909862, 19915271
19928926, 19930276, 19931367, 19931709, 19932634, 19933147, 19941352
19943771, 19952975, 19957298, 19978542, 19982584, 19988852, 19989009
19990037, 19995869, 20001168, 20001466, 20009569, 20009833, 20011515
20011646, 20011897, 20017509, 20023340, 20031873, 20043616, 20048359
20052269, 20061399, 20074391, 20076781, 20078186, 20087383, 20093776
20101006, 20117253, 20118035, 20122715, 20124446, 20134113, 20134339
20139391, 20144019, 20144308, 20165574, 20169408, 20171986, 20172151
20173897, 20175161, 20181030, 20212067, 20217801, 20228093, 20229001
20233181, 20235511, 20245930, 20250147, 20267166, 20273319, 20281121
20284155, 20294666, 20296619, 20298413, 20302006, 20308798, 20315311
20318889, 20322560, 20324049, 20328248, 20331945, 20347562, 20348653
20354900, 20356733, 20361671, 20368850, 20373598, 20374572, 20378086
20382309, 20387265, 20394750, 20397490, 20401975, 20402832, 20408829
20408866, 20413820, 20415564, 20424183, 20424899, 20425790, 20428621
20432873, 20437153, 20440930, 20441797, 20446883, 20447445, 20459944
20464614, 20466322, 20466628, 20468401, 20468490, 20470877, 20471920
20474192, 20475845, 20476175, 20480209, 20493163, 20505778, 20509482
20513399, 20524085, 20528052, 20539050, 20543011, 20544065, 20554364
20557786, 20558005, 20560611, 20562898, 20564072, 20565112, 20565133
20569094, 20577490, 20581111, 20582405, 20588486, 20588502, 20591183
20596234, 20598042, 20603378, 20603431, 20613079, 20618595, 20627866
20635353, 20641666, 20657411, 20657441, 20669434, 20671094, 20673810
20677396, 20677974, 20684983, 20686773, 20688221, 20703000, 20703629
20704450, 20705577, 20707932, 20708701, 20711718, 20717081, 20717091
20717359, 20725343, 20734332, 20736227, 20746251, 20757079, 20764012
20766180, 20768076, 20778986, 20794034, 20798891, 20800890, 20801783
20803014, 20825533, 20828947, 20830459, 20831538, 20832516, 20835241
20839705, 20842388, 20844426, 20848335, 20856766, 20859910, 20860659
20862087, 20868862, 20869721, 20875898, 20877664, 20878790, 20879709

20879889, 20880215, 20882568, 20890311, 20897759, 20898391, 20898997
20899461, 20904530, 20907061, 20914870, 20919320, 20920911, 20922010
20925795, 20926021, 20929771, 20936731, 20936905, 20938170, 20951038
20952966, 20958816, 20977794, 20978259, 21037905, 21037923, 21047766
21047803, 21052842, 21059919, 21060755, 21061354, 21063322, 21068507
21072646, 21080143, 21091431, 21091901, 21095391, 21097043, 21099555
21101873, 21106027, 21132297, 21133343, 21142837, 21147908, 21153266
21157728, 21159665, 21164318, 21171382, 21172913, 21174504, 21184223
21186167, 21188532, 21188537, 21188584, 21196809, 21197626, 21220620
21225209, 21239530, 21241052, 21241829, 21246723, 21260397, 21260431
21263635, 21266085, 21270823, 21273804, 21275255, 21281532, 21281607
21285458, 21291274, 21293600, 21294938, 21296029, 21297872, 21299490
21300341, 21308727, 21315084, 21322887, 21329301, 21354456, 21373076
21373473, 21380789, 21383171, 21385422, 21387128, 21387964, 21419850
21421886, 21422580, 21424824, 21425496, 21429602, 21442094, 21450666
21476308, 21479753, 21492036, 21502702, 21514877, 21516611, 21517440
21522582, 21526048, 21532755, 21534893, 21542577, 21555660, 21560152
21566639, 21566944, 21566993, 21575362, 21612959, 21620471, 21623164
21625179, 21626377, 21629064, 21632821, 21641414, 21641760, 21644640
21649497, 21656630, 21659726, 21665897, 21668627, 21675340, 21695575
21698350, 21744290, 21756661, 21756677, 21756699, 21764119, 21773465
21780146, 21785691, 21787056, 21794615, 21795111, 21797203, 21811517
21820934, 21821302, 21828126, 21834568, 21837606, 21842017, 21842740
21847223, 21856522, 21863727, 21868720, 21875360, 21889720, 21893235
21896069, 21899588, 21911701, 21913183, 21915719, 21917884, 21924131
21960504, 21967197, 21977186, 21977392, 22007324, 22018363, 22022760
22024071, 22037014, 22046677, 22062026, 22062517, 22068305, 22070866
22072818, 22075064, 22077517, 22083366, 22087683, 22092979, 22118835
22118851, 22139226, 22146062, 22148226, 22157363, 22160989, 22165897
22168163, 22173980, 22175564, 22176950, 22178855, 22179537, 22185234
22205263, 22214989, 22223463, 22228324, 22232606, 22233505, 22238921
22243719, 22243983, 22250006, 22256431, 22256560, 22258530, 22261050
22264489, 22268833, 22282748, 22294260, 22296366, 22301880, 22305887
22346829, 22347493, 22351572, 22353199, 22353346, 22359063, 22364044
22365117, 22366322, 22366558, 22374754, 22380919, 22454326, 22454940
22458049, 22465352, 22468255, 22468781, 22475617, 22492533, 22495062
22495673, 22496904, 22499356, 22501616, 22503297, 22507210, 22507234
22515353, 22517782, 22518784, 22519146, 22520320, 22528741, 22529728
22533631, 22536802, 22551446, 22568016, 22568177, 22568797, 22606521
22624709, 22645009, 22654475, 22657942, 22662332, 22670385, 22670413
22674709, 22675136, 22686674, 22690648, 22695831, 22707244, 22707866
22721409, 22729345, 22730454, 22733141, 22734547, 22750215, 22757364
22760595, 22760679, 22762046, 22782647, 22806698, 22808310, 22809871
22815955, 22816287, 22820579, 22820798, 22826718, 22836801, 22842151
22855193, 22862134, 22865673, 22873635, 22894101, 22894949, 22897344
22901797, 22905130, 22916353, 22922076, 22923409, 22950945, 22961508
22972770, 22977256, 23002524, 23003979, 23007241, 23008056, 23019710
23020270, 23025340, 23026585, 23028781, 23029562, 23035249, 23053606
23061453, 23061702, 23065323, 23066146, 23068169, 23080557, 23082876
23084507, 23088803, 23089357, 23096938, 23101501, 23104033, 23105538
23108128, 23115139, 23124895, 23125826, 23126410, 23140259, 23148260
23149541, 23151677, 23168363, 23170620, 23172924, 23177536, 23177923
23184013, 23184263, 23195445, 23197103, 23209741, 23220453, 23229229
23237313, 23240358, 23260854, 23262847, 23265914, 23265965, 23266217
23272045, 23294548, 23302839, 23314180, 23315153, 23315889, 23324000
23326313, 23328639, 23338911, 23342170, 23492665, 23501901, 23514710
23514911, 23521523, 23528412, 23533524, 23533807, 23543183, 23548817
23567857, 23571055, 23572982, 23584909, 23602213, 23614158, 23628685
23642282, 23709062, 23711335, 23713236, 23717151, 23725036, 23727148
23731896, 23746128, 23854396, 24285405, 24300640, 24303148, 24307571
24308635, 24315824, 24316947, 24321547, 24326444, 24341675, 24350620
24350831, 24365589, 24385625, 24385983, 24386767, 24393981, 24397438
24401351, 24411921, 24413809, 24415926, 24416451, 24421668, 24423416
24425998, 24437510, 24448240, 24448282, 24457597, 24461826, 24509056
24523374, 24530364, 24534298, 24555417, 24560906, 24563422, 24570598
24573817, 24577566, 24589081, 24597536, 24600330, 24624166, 24642295

24642495, 24652769, 24662775, 24674955, 24683149, 24690216, 24693382
24701840, 24713381, 24717859, 24718260, 24719736, 24737064, 24737403
24737581, 24737954, 24739928, 24752618, 24766121, 24784414, 24790914
24792678, 24796092, 24801152, 24802934, 24808595, 24812585, 24817447
24825843, 24831514, 24835538, 24835919, 24848928, 24907917, 24908321
24917972, 24920582, 24929210, 24966594, 25029423, 25031502, 25034396
25042823, 25047724, 25051465, 25056052, 25058080, 25060506, 25067795
25076732, 25076756, 25077278, 25079710, 25091141, 25093739, 25093872
25099339, 25107334, 25110233, 25115178, 25123585, 25150925, 25161298
25165496, 25178179, 25179774, 25192729, 25205368, 25210690, 25240188
25248384, 25264559, 25300427, 25307368, 25313154, 25328093, 25330273
25353983, 25357142, 25377044, 25392535, 25405687, 25415713, 25417056
25417958, 25423453, 25427662, 25429959, 25437695, 25437699, 25459958
25472885, 25475853, 25476125, 25477657, 25482971, 25483815, 25484507
25486384, 25489342, 25489367, 25489607, 25490238, 25492379, 25494379
25494413, 25495682, 25539063, 25546608, 25547060, 25551676, 25555252
25575628, 25579761, 25599425, 25600342, 25600421, 25602488, 25606091
25612095, 25616268, 25633101, 25634317, 25635149, 25639019, 25643818
25643931, 25649873, 25653109, 25654936, 25655390, 25669791, 25670786
25695903, 25699321, 25722055, 25733479, 25740844, 25743479, 25760195
25764020, 25766822, 25775213, 25780343, 25789277, 25790353, 25809524
25822410, 25823532, 25823754, 25856821, 25861398, 25879984, 25881255
25885148, 25890782, 25897615, 25904490, 25914276, 25919622, 25947799
25957038, 25982666, 25986062, 25997810, 26007010, 26023002, 26023025
26024732, 26027162, 26029780, 26039623, 26088426, 26089440, 26110259
26110632, 26111842, 26121990, 26126424, 26153977, 26169341, 26187943
26198757, 26198926, 26203182, 26243698, 26245237, 26248143, 26256131
26262953, 26263721, 26318200, 26318627, 26324206, 26325856, 26336977
26353617, 26366517, 26412540, 26430737, 26439748, 26444887, 26446098
26482376, 26513067, 26513709, 26544823, 26546664, 26546754, 26556014
26569225, 26570171, 26575788, 26633558, 26635845, 26637592, 26637824
26654363, 26658759, 26714910, 26716835, 26729494, 26758193, 26768025
26784509, 26822620, 26828994, 26832296, 26844406, 26875822, 26898563
26968670, 26999139, 27000663, 27000690, 27001733, 27012701, 27015449
27033652, 27034890, 27052607, 27060167, 27072923, 27086138, 27092508
27097854, 27101105, 27122162, 27133662, 27169796, 27185188, 27199245
27207110, 27207634, 27213224, 27217412, 27223075, 27229389, 27255377
27274536, 27276231, 27285244, 27303938, 27314206, 27314390, 27314697
27337759, 27348081, 27351628, 27370965, 27375542, 27397048, 27404573
27424405, 27433385, 27434193, 27441326, 27445727, 27452046, 27461789
27461842, 27468303, 27475603, 27487919, 27534509, 27548131, 27565906
27567477, 27611612, 27620950, 27623159, 27629756, 27634991, 27642235
27710072, 27726780, 27729678, 27745728, 27751755, 27829295, 27846298
27847259, 27897759, 27923320, 27929509, 27938623, 27952577, 27952584
27986817, 27995248, 27997875, 28000269, 28022101, 28023399, 28023482
28024793, 28025414, 28026866, 28043157, 28072383, 28079127, 28098160
28125601, 28164480, 28174827, 28199085, 28215510, 28250929, 28281094
28302049, 28305362, 28357401, 28369092, 28371123, 28384353, 28390273
28394726, 28420042, 28423598, 28432129, 28440711, 28501075, 28502113
28502128, 28507324, 28542455, 28566241, 28578164, 28587723, 28612674
28636676, 28639299, 28683167, 28708023, 28714988, 28730253, 28758090
28774416, 28790654, 28797711, 28821847, 28849751, 28852325, 28867992
28891741, 28891984, 28915933, 28950969, 28986231, 28993590, 29006527
29009513, 29027694, 29030780, 29061016, 29142109, 29163567, 29170232
29189889, 29200700, 29247712, 29250317, 29251241, 29254615, 29260956
29342099, 29343156, 29372460, 29378913, 29379978, 29388020, 29408136
29434301, 29437712, 29464779, 29477015, 29483626, 29483672, 29483723
29483771, 29500257, 29500963, 29511611, 29559723, 29621961, 29625065
29633753, 29637560, 29645349, 29707896, 29719146, 29726695, 29774367
29774383, 29782211, 29791152, 29796916, 29814995, 29817278, 29828111
29869404, 29869906, 29884958, 29893132, 29944660, 29961353, 29962927
29962939, 29965888, 29997937, 30018017, 30031027, 30116203, 30117469
30128197, 30160639, 30164714, 30179644, 30186245, 30196358, 30200680
30215130, 30218044, 30225443, 30252098, 30295478, 30305880, 30320029
30365745, 30387666, 30421204, 30497057, 30499600, 30502041, 30517516
30522998, 30534664, 30578221, 30624864, 30659882, 30668407, 30698289

30758943, 30803210, 30805558, 30816938, 30826474, 30855121, 30860803
30866988, 30889351, 30973003, 30987088, 31001455, 31013127, 31022858
31028986, 31031715, 31046619, 31106577, 31156383, 31172207, 31194264
31219939, 31228670, 31306274, 31335037, 31335142, 31501139, 31637680
31668061, 31668867, 31668915, 31675539, 31711889, 31786838, 31799139
31927930, 31985579, 32065792, 32097882, 32119956, 32165915, 32186646
32296941

Version 12.1.0.2.v22

Version 12.1.0.2.v22 includes the following:

- Patch 31550110: Database Patch Set Update : 12.1.0.2.201020 (31550110)
- Patch 31668915: Database PSU 12.1.0.2.201020, Oracle JavaVM Component (OCT2020)
- Patch 31335037: RDBMS - DSTV35 UPDATE - TZDATA2020A
- Patch 31335142: DSTV35 UPDATE - TZDATA2020A - NEED OJVM FIX
- Patch 17969866: Oracle GoldenGate – 46719 ENH REPLICATION SUPPORT FOR INSERTS / FULL UPDATES WITH LARGE VALUES
- Patch 20394750: Oracle GoldenGate – APPLY CDR RESOLUTION FAILING FOR LOBS, XML, LONG, AND OBJECTS
- Patch 24835919: Oracle GoldenGate – IR EXECUTING DEPENDENT TRANSACTIONS OUT OF ORDER WITH PARALLELISM GREATER THAN
- Patch 23262847: Oracle GoldenGate - MALFORMED REDO CAUSED OGG REPLICATION ABEND
- Patch 21091901: ONLINE MOVE OF HASH OR REF PARTITION CAN LEAVE LOCAL INDEXES INCONSISTENT
- Patch 21171382: ADD CONTROL FOR AUTOMATIC CREATION OF STATS EXTENSIONS
- Patch 23711335: CDB_UPG PDCDB CDB UPGRADE AS WHOLE TAKES 1 MORE HOUR THAN PREVIOUS LABELS IN MAY
- Patch 25031502: MV QUERY REWRITE WORKLOAD HIT ORA 600 [KGL-HEAP-SIZE-EXCEEDED]
- Patch 31911280: JSON Bundle Patch
- PreUpgrade Jar: preupgrade_12.1.0.2.0_18_crlf.zip
- Support for [Setting and unsetting system diagnostic events \(p. 1046\)](#) using procedures in the `rdsadmin.rdsadmin_util` package
- Support for the procedure `rdsadmin_util.truncate_apply$cdr_info` described in [Integrated REPLICAT slow due to query on sys."_DBA_APPLY_CDR_INFO" \(p. 1236\)](#)

Combined patches for version 12.1.0.2.v22, released October 2020

Bugs fixed:

6194865, 6418158, 6599380, 13542050, 13787015, 13854364, 14283239
14643995, 14705949, 16090440, 16354467, 16359751, 16439813, 16619249
16756406, 16777441, 16799735, 16863642, 16870214, 16875041, 16887946
16923858, 16938780, 16941434, 17008068, 17210525, 17258582, 17274537
17319928, 17365043, 17409174, 17414008, 17432124, 17495022, 17532729
17532734, 17533661, 17551063, 17655240, 17722075, 17760068, 17835294
17867700, 17890099, 17969866, 18007682, 18043064, 18051556, 18090142
18110491, 18122373, 18191823, 18202441, 18250893, 18254023, 18272672
18288842, 18306996, 18307021, 18308268, 18324100, 18354830, 18371441
18373438, 18382302, 18411216, 18417036, 18419520, 18427406, 18436647
18440095, 18456643, 18475439, 18492302, 18499088, 18510194, 18542562
18548246, 18548433, 18549238, 18604493, 18604692, 18607546, 18610915

18618122, 18628388, 18648816, 18662619, 18673090, 18674024, 18674047
18681056, 18693124, 18700762, 18705806, 18727933, 18733351, 18740837
18742258, 18743542, 18758877, 18759211, 18774543, 18775971, 18778801
18791688, 18797519, 18798250, 18799063, 18799993, 18801391, 18803726
18810904, 18818069, 18819908, 18840932, 18841764, 18845653, 18849537
18849970, 18851894, 18856106, 18856999, 18866977, 18868646, 18885870
18886413, 18893947, 18895170, 18899974, 18900107, 18904062, 18909599
18913440, 18914624, 18921743, 18940497, 18948177, 18952766, 18952989
18964939, 18964978, 18966843, 18967382, 18973548, 18974476, 18988834
18990023, 18990693, 18999568, 19001359, 19001390, 19012119, 19013183
19016730, 19017309, 19018206, 19018447, 19022470, 19023822, 19024808
19028800, 19032777, 19035573, 19044962, 19048007, 19050649, 19052488
19054077, 19058490, 19060015, 19065556, 19065677, 19067244, 19068380
19068610, 19068970, 19074147, 19075256, 19076343, 19077215, 19079752
19081128, 19124336, 19124589, 19130152, 19131386, 19131607, 19134173
19141838, 19143550, 19146474, 19149990, 19153980, 19154375, 19155797
19157754, 19165673, 19168167, 19171086, 19174430, 19174521, 19174942
19176223, 19176326, 19176885, 19178851, 19180394, 19180770, 19183343
19188576, 19188385, 19188927, 19189317, 19189525, 19195895, 19197175
19201867, 19207117, 19211433, 19213447, 19223010, 19231857, 19238590
19243521, 19245191, 19248279, 19248799, 19258504, 19272708, 19279273
19280225, 19284031, 19285025, 19289642, 19291380, 19297917, 19303936
19304354, 19306797, 19307662, 19308965, 19309466, 19313563, 19315668
19315691, 19317646, 19326908, 19327391, 19329654, 19330795, 19332396
19333670, 19335438, 19339555, 19347458, 19354335, 19354794, 19358317
19363645, 19364502, 19366375, 19370504, 19371175, 19373893, 19375649
19382851, 19383839, 19385656, 19390567, 19390620, 19393542, 19396455
19399918, 19402853, 19404068, 19409212, 19430401, 19433930, 19434529
19439759, 19440520, 19440586, 19445860, 19448499, 19450116, 19450314
19452434, 19461270, 19461428, 19468347, 19468612, 19468991, 19469538
19475971, 19487147, 19490948, 19501299, 19503821, 19512341, 19516448
19518079, 19520602, 19523462, 19524158, 19524384, 19529868, 19532017
19534363, 19536415, 19543384, 19547370, 19547774, 19548064, 19550902
19561643, 19562381, 19566592, 19571055, 19571082, 19571367, 19577410
19578247, 19578350, 19583624, 19587324, 19590877, 19591608, 19593445
19597439, 19597583, 19601762, 19604659, 19606174, 19617921, 19619732
19623450, 19627012, 19632912, 19637186, 19639483, 19644859, 19647503
19649152, 19658708, 19662635, 19663176, 19670108, 19676012, 19676905
19680796, 19684504, 19687159, 19689979, 19693090, 19699191, 19699946
19701015, 19703301, 19705781, 19706965, 19708342, 19708632, 19718981
19721304, 19723336, 19730508, 19769480, 19769625, 19777862, 19781326
19784751, 19790243, 19791273, 19791377, 19799847, 19805359, 19809171
19811709, 19817386, 19818513, 19824871, 19831647, 19835133, 19841800
19855285, 19859472, 19865345, 19869255, 19871910, 19873610, 19877336
19879746, 19880190, 19883092, 19886165, 19888853, 19889230, 19891090
19895326, 19895362, 19896336, 19902195, 19908836, 19909862, 19915271
19928926, 19930276, 19931367, 19931709, 19932634, 19933147, 19941352
19943771, 19952975, 19957298, 19978542, 19982584, 19988852, 19989009
19990037, 19995869, 20001168, 20001466, 20009569, 20009833, 20011515
20011646, 20011897, 20017509, 20023340, 20031873, 20043616, 20048359
20052269, 20061399, 20074391, 20076781, 20078186, 20087383, 20093776
20101006, 20117253, 20118035, 20122715, 20124446, 20134113, 20134339
20139391, 20144019, 20144308, 20165574, 20169408, 20171986, 20172151
20173897, 20175161, 20181030, 20212067, 20217801, 20228093, 20229001
20233181, 20235511, 20245930, 20250147, 20267166, 20273319, 20281121
20284155, 20294666, 20296619, 20298413, 20302006, 20308798, 20315311
20318889, 20322560, 20324049, 20328248, 20331945, 20347562, 20348653
20354900, 20356733, 20361671, 20368850, 20373598, 20374572, 20378086
20382309, 20387265, 20394750, 20397490, 20401975, 20402832, 20408829
20408866, 20413820, 20415564, 20424183, 20424899, 20425790, 20428621
20432873, 20437153, 20440930, 20441797, 20446883, 20447445, 20459944
20464614, 20466322, 20466628, 20468401, 20468490, 20470877, 20471920
20474192, 20475845, 20476175, 20480209, 20493163, 20505778, 20509482
20513399, 20524085, 20528052, 20539050, 20543011, 20544065, 20554364
20557786, 20558005, 20560611, 20562898, 20564072, 20565112, 20565133
20569094, 20577490, 20581111, 20582405, 20588486, 20588502, 20591183

20596234, 20598042, 20603378, 20603431, 20613079, 20618595, 20627866
20635353, 20641666, 20657411, 20657441, 20669434, 20671094, 20673810
20677396, 20677974, 20684983, 20686773, 20688221, 20703000, 20703629
20704450, 20705577, 20707932, 20708701, 20711718, 20717081, 20717091
20717359, 20725343, 20734332, 20736227, 20746251, 20757079, 20764012
20766180, 20768076, 20778986, 20794034, 20798891, 20800890, 20801783
20803014, 20825533, 20828947, 20830459, 20831538, 20832516, 20835241
20839705, 20842388, 20844426, 20848335, 20856766, 20859910, 20860659
20862087, 20868862, 20869721, 20875898, 20877664, 20878790, 20879709
20879889, 20880215, 20882568, 20890311, 20897759, 20898391, 20898997
20899461, 20904530, 20907061, 20914870, 20919320, 20920911, 20922010
20925795, 20926021, 20929771, 20936731, 20936905, 20938170, 20951038
20952966, 20958816, 20977794, 20978259, 21037905, 21037923, 21047766
21047803, 21052842, 21059919, 21060755, 21061354, 21063322, 21068507
21072646, 21080143, 21091431, 21091901, 21095391, 21097043, 21099555
21101873, 21106027, 21132297, 21133343, 21142837, 21147908, 21153266
21157728, 21159665, 21164318, 21171382, 21172913, 21174504, 21184223
21186167, 21188532, 21188537, 21188584, 21196809, 21197626, 21220620
21225209, 21239530, 21241052, 21241829, 21246723, 21260397, 21260431
21263635, 21266085, 21270823, 21273804, 21275255, 21281532, 21281607
21285458, 21291274, 21293600, 21294938, 21296029, 21297872, 21299490
21300341, 21308727, 21315084, 21322887, 21329301, 21354456, 21373076
21373473, 21380789, 21383171, 21385422, 21387128, 21387964, 21419850
21421886, 21422580, 21424824, 21425496, 21429602, 21442094, 21450666
21476308, 21479753, 21492036, 21502702, 21514877, 21516611, 21517440
21522582, 21526048, 21532755, 21534893, 21542577, 21555660, 21560152
21566639, 21566944, 21566993, 21575362, 21612959, 21620471, 21623164
21625179, 21626377, 21629064, 21632821, 21641414, 21641760, 21644640
21649497, 21656630, 21659726, 21665897, 21668627, 21675340, 21695575
21698350, 21744290, 21756661, 21756677, 21756699, 21764119, 21773465
21780146, 21785691, 21787056, 21794615, 21795111, 21797203, 21811517
21820934, 21821302, 21828126, 21837606, 21842017, 21842740, 21847223
21856522, 21863727, 21868720, 21875360, 21889720, 21893235, 21896069
21899588, 21911701, 21913183, 21915719, 21917884, 21924131, 21960504
21967197, 21977186, 21977392, 22007324, 22018363, 22022760, 22024071
22037014, 22046677, 22062026, 22062517, 22068305, 22070866, 22072818
22075064, 22077517, 22083366, 22087683, 22092979, 22118835, 22118851
22139226, 22146062, 22148226, 22157363, 22160989, 22165897, 22168163
22173980, 22175564, 22176950, 22178855, 22179537, 22185234, 22205263
22214989, 22223463, 22228324, 22232606, 22233505, 22238921, 22243719
22243983, 22250006, 22256431, 22256560, 22258530, 22261050, 22264489
22268833, 22282748, 22294260, 22296366, 22301880, 22305887, 22346829
22347493, 22351572, 22353199, 22353346, 22359063, 22364044, 22365117
22366322, 22366558, 22374754, 22380919, 22454326, 22454940, 22458049
22465352, 22468255, 22468781, 22475617, 22492533, 22495062, 22495673
22496904, 22499356, 22501616, 22503297, 22507210, 22507234, 22515353
22517782, 22518784, 22519146, 22520320, 22528741, 22529728, 22533631
22536802, 22551446, 22568016, 22568177, 22568797, 22606521, 22624709
22645009, 22654475, 22657942, 22670385, 22670413, 22674709, 22675136
22686674, 22690648, 22695831, 22707244, 22707866, 22721409, 22729345
22730454, 22733141, 22734547, 22750215, 22757364, 22760595, 22760679
22762046, 22782647, 22806698, 22808310, 22809871, 22815955, 22816287
22820579, 22826718, 22836801, 22842151, 22855193, 22862134, 22865673
22873635, 22894101, 22894949, 22897344, 22901797, 22905130, 22916353
22922076, 22923409, 22950945, 22961508, 22972770, 22977256, 23002524
23003979, 23007241, 23008056, 23019710, 23020270, 23025340, 23026585
23028781, 23029562, 23035249, 23053606, 23061453, 23061702, 23065323
23066146, 23068169, 23080557, 23082876, 23084507, 23088803, 23089357
23096938, 23101501, 23104033, 23105538, 23108128, 23115139, 23124895
23125826, 23126410, 23140259, 23148260, 23149541, 23151677, 23168363
23170620, 23172924, 23177536, 23177923, 23184013, 23184263, 23195445
23197103, 23209741, 23220453, 23229229, 23237313, 23240358, 23260854
23262847, 23265914, 23265965, 23266217, 23272045, 23294548, 23302839
23314180, 23315153, 23315889, 23324000, 23326313, 23328639, 23338911
23342170, 23492665, 23501901, 23514710, 23514911, 23521523, 23528412
23533524, 23533807, 23543183, 23548817, 23567857, 23571055, 23572982

23584909, 23602213, 23614158, 23628685, 23642282, 23709062, 23711335
23713236, 23717151, 23725036, 23727148, 23731896, 23746128, 23854396
24285405, 24300640, 24303148, 24307571, 24308635, 24315824, 24316947
24321547, 24326444, 24341675, 24350620, 24350831, 24365589, 24385625
24385983, 24386767, 24393981, 24397438, 24401351, 24411921, 24413809
24415926, 24416451, 24421668, 24423416, 24425998, 24437510, 24448240
24448282, 24457597, 24461826, 24509056, 24523374, 24530364, 24534298
24555417, 24560906, 24563422, 24570598, 24573817, 24577566, 24589081
24600330, 24624166, 24642295, 24642495, 24652769, 24662775, 24674955
24683149, 24690216, 24693382, 24701840, 24713381, 24717859, 24718260
24719736, 24737064, 24737403, 24737581, 24737954, 24739928, 24752618
24766121, 24790914, 24792678, 24796092, 24801152, 24802934, 24808595
24812585, 24817447, 24825843, 24831514, 24835538, 24835919, 24848928
24907917, 24908321, 24917972, 24920582, 24929210, 24966594, 25029423
25031502, 25034396, 25042823, 25047724, 25051465, 25056052, 25058080
25060506, 25067795, 25076732, 25076756, 25079710, 25091141, 25093739
25093872, 25099339, 25107334, 25110233, 25115178, 25123585, 25150925
25161298, 25165496, 25178179, 25179774, 25192729, 25205368, 25210690
25240188, 25248384, 25264559, 25300427, 25307368, 25313154, 25328093
25330273, 25353983, 25357142, 25377044, 25392535, 25405687, 25415713
25417056, 25417958, 25423453, 25427662, 25429959, 25437695, 25437699
25459958, 25472885, 25475853, 25476125, 25477657, 25482971, 25483815
25484507, 25486384, 25489342, 25489367, 25489607, 25490238, 25492379
25494379, 25494413, 25495682, 25539063, 25546608, 25547060, 25551676
25555252, 25575628, 25579761, 25599425, 25600342, 25600421, 25602488
25606091, 25612095, 25616268, 25633101, 25634317, 25635149, 25639019
25643818, 25643931, 25649873, 25653109, 25654936, 25655390, 25669791
25670786, 25695903, 25699321, 25722055, 25733479, 25740844, 25743479
25760195, 25764020, 25766822, 25775213, 25780343, 25789277, 25790353
25809524, 25822410, 25823532, 25823754, 25856821, 25861398, 25879984
25881255, 25885148, 25890782, 25897615, 25904490, 25914276, 25919622
25947799, 25957038, 25982666, 25986062, 25997810, 26007010, 26023002
26023025, 26024732, 26027162, 26029780, 26039623, 26088426, 26089440
26110259, 26110632, 26111842, 26121990, 26153977, 26169341, 26187943
26198757, 26198926, 26203182, 26243698, 26245237, 26248143, 26256131
26262953, 26263721, 26318200, 26318627, 26324206, 26325856, 26336977
26353617, 26366517, 26412540, 26430737, 26439748, 26444887, 26446098
26482376, 26513067, 26513709, 26544823, 26546664, 26546754, 26556014
26569225, 26570171, 26575788, 26633558, 26635845, 26637592, 26637824
26654363, 26658759, 26714910, 26716835, 26758193, 26768025, 26784509
26822620, 26828994, 26832296, 26844406, 26875822, 26898563, 26968670
26999139, 27000663, 27000690, 27001733, 27012701, 27015449, 27033652
27034890, 27052607, 27060167, 27072923, 27086138, 27092508, 27097854
27101105, 27122162, 27133662, 27169796, 27199245, 27207110, 27207634
27213224, 27217412, 27223075, 27229389, 27255377, 27274536, 27276231
27285244, 27303938, 27314206, 27314390, 27314697, 27337759, 27348081
27351628, 27370965, 27375542, 27397048, 27404573, 27424405, 27433385
27434193, 27441326, 27445727, 27452046, 27461789, 27461842, 27468303
27475603, 27487919, 27534509, 27548131, 27565906, 27567477, 27611612
27620950, 27623159, 27629756, 27634991, 27642235, 27710072, 27726780
27729678, 27745728, 27751755, 27829295, 27846298, 27847259, 27897759
27923320, 27929509, 27938623, 27952577, 27952584, 27986817, 27995248
27997875, 28000269, 28022101, 28023399, 28024793, 28025414, 28026866
28043157, 28072383, 28079127, 28098160, 28125601, 28164480, 28174827
28199085, 28215510, 28250929, 28281094, 28302049, 28305362, 28357401
28369092, 28371123, 28384353, 28390273, 28394726, 28420042, 28423598
28432129, 28440711, 28501075, 28502113, 28502128, 28507324, 28542455
28566241, 28578164, 28587723, 28612674, 28636676, 28639299, 28683167
28708023, 28714988, 28730253, 28758090, 28774416, 28790654, 28797711
28821847, 28849751, 28852325, 28867992, 28891741, 28891984, 28915933
28950969, 28986231, 28993590, 29006527, 29009513, 29027694, 29030780
29142109, 29163567, 29189889, 29200700, 29247712, 29250317, 29251241
29254615, 29260956, 29343156, 29372460, 29378913, 29379978, 29388020
29408136, 29434301, 29437712, 29464779, 29483626, 29483672, 29483723
29483771, 29500257, 29500963, 29511611, 29559723, 29621961, 29633753
29637560, 29645349, 29707896, 29719146, 29726695, 29774367, 29774383

29782211, 29791152, 29817278, 29828111, 29869404, 29869906, 29893132
29944660, 29961353, 29962927, 29962939, 29965888, 29997937, 30018017
30116203, 30128197, 30160639, 30164714, 30179644, 30196358, 30200680
30215130, 30218044, 30225443, 30252098, 30295478, 30305880, 30320029
30365745, 30387666, 30421204, 30497057, 30499600, 30502041, 30517516
30522998, 30534664, 30578221, 30624864, 30659882, 30668407, 30698289
30758943, 30803210, 30805558, 30816938, 30855121, 30889351, 30973003
30987088, 31001455, 31013127, 31022858, 31028986, 31031715, 31106577
31156383, 31172207, 31194264, 31219939, 31228670, 31306274, 31335037
31335142, 31550110, 31668061, 31668867, 31668915, 31799139

Version 12.1.0.2.v21

Version 12.1.0.2.v21 includes the following:

- Patch 31113348: Database Patch Set Update 12.1.0.2.200714
- Patch 31219939: Database PSU 12.1.0.2.200714, Oracle JavaVM Component (JUL2020)
- Patch 31335037: DSTV35 for RDBMS (TZDATA2020A)
- Patch 31335142: DSTV35 for OJVM (TZDATA2020A)
- Patch 17969866: Oracle GoldenGate - 46719: ENH: REPLICATION SUPPORT FOR INSERTS / FULL UPDATES WITH LARGE VALUES
- Patch 20394750: Oracle GoldenGate - APPLY CDR RESOLUTION FAILING FOR LOBS, XML, LONG, AND OBJECTS
- Patch 24835919: Oracle GoldenGate - IR EXECUTING DEPENDENT TRANSACTIONS OUT OF ORDER WITH PARALLELISM GREATER THAN
- Patch 20033733: PART :IMC: HIT ORA 600 [KGL-HEAP-SIZE-EXCEEDED]
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- Patch 23711335: CDB_UPG:PDCDB:CDB UPGRADE AS WHOLE TAKES 1 MORE HOUR THAN PREVIOUS LABELS IN MAY
- Patch 31579750: JSON bundle Patch
- Patch 21091901: ONLINE MOVE OF HASH OR REF PARTITION CAN LEAVE LOCAL INDEXES INCONSISTENT
- Patch 21171382: DBMS_STATS Patch AUTO DOP COMPUTES A HIGH DOP UNNECESSARILY
- Patch 23262847: Oracle GoldenGate - MALFORMED REDO CAUSED OGG REPLICATION ABEND
- PreUpgrade Jar: preupgrade_12.1.0.2.0_18_crlf.zip

Combined patches for version 12.1.0.2.v21, released July 2020

Bugs fixed:

6194865, 6418158, 6599380, 13542050, 13787015, 14283239, 14643995
14705949, 16090440, 16354467, 16359751, 16439813, 16619249, 16756406
16777441, 16799735, 16863642, 16870214, 16875041, 16887946, 16923858
16938780, 16941434, 17008068, 17210525, 17258582, 17274537, 17319928
17365043, 17409174, 17414008, 17432124, 17495022, 17532729, 17532734
17533661, 17551063, 17655240, 17722075, 17760068, 17835294, 17867700
17890099, 17969866, 18007682, 18043064, 18051556, 18090142, 18110491
18122373, 18191823, 18202441, 18250893, 18254023, 18272672, 18288842
18306996, 18307021, 18308268, 18354830, 18371441, 18373438, 18382302
18411216, 18417036, 18419520, 18436647, 18440095, 18456643, 18475439
18492302, 18499088, 18510194, 18542562, 18548246, 18548433, 18549238
18604493, 18604692, 18607546, 18610915, 18618122, 18628388, 18648816

18662619, 18673090, 18674024, 18674047, 18681056, 18693124, 18700762
18705806, 18727933, 18733351, 18740837, 18742258, 18743542, 18758877
18759211, 18774543, 18775971, 18778801, 18791688, 18797519, 18798250
18799063, 18799993, 18801391, 18803726, 18810904, 18818069, 18819908
18840932, 18841764, 18845653, 18849537, 18849970, 18851894, 18856106
18856999, 18866977, 18868646, 18885870, 18886413, 18893947, 18895170
18899974, 18900107, 18904062, 18909599, 18913440, 18914624, 18921743
18940497, 18948177, 18952766, 18952989, 18964939, 18964978, 18966843
18967382, 18973548, 18974476, 18988834, 18990023, 18990693, 18999568
19001359, 19001390, 19012119, 19013183, 19016730, 19017309, 19018206
19018447, 19022470, 19023822, 19024808, 19028800, 19032777, 19035573
19044962, 19048007, 19050649, 19052488, 19054077, 19058490, 19060015
19065556, 19065677, 19067244, 19068380, 19068610, 19068970, 19074147
19075256, 19076343, 19077215, 19079752, 19081128, 19124336, 19124589
19130152, 19131386, 19131607, 19134173, 19141838, 19143550, 19146474
19149990, 19153980, 19154375, 19155797, 19157754, 19165673, 19168167
19171086, 19174430, 19174521, 19174942, 19176223, 19176326, 19176885
19178851, 19180394, 19180770, 19183343, 19185876, 19188927, 19189317
19189525, 19195895, 19197175, 19201867, 19207117, 19211433, 19213447
19223010, 19231857, 19238590, 19243521, 19245191, 19248279, 19248799
19258504, 19272708, 19279273, 19280225, 19284031, 19285025, 19289642
19291380, 19297917, 19303936, 19304354, 19306797, 19307662, 19308965
19309466, 19313563, 19315668, 19315691, 19317646, 19326908, 19327391
19329654, 19330795, 19332396, 19333670, 19335438, 19339555, 19347458
19354335, 19354794, 19358317, 19363645, 19364502, 19366375, 19370504
19371175, 19373893, 19375649, 19382851, 19383839, 19385656, 19390567
19390620, 19393542, 19396455, 19399918, 19402853, 19404068, 19409212
19430401, 19433930, 19434529, 19439759, 19440520, 19440586, 19445860
19448499, 19450116, 19450314, 19452434, 19461270, 19461428, 19468347
19468612, 19468991, 19469538, 19475971, 19487147, 19490948, 19501299
19503821, 19512341, 19516448, 19518079, 19520602, 19523462, 19524158
19524384, 19529868, 19532017, 19534363, 19536415, 19543384, 19547370
19547774, 19548064, 19550902, 19561643, 19562381, 19566592, 19571055
19571082, 19571367, 19577410, 19578247, 19578350, 19583624, 19587324
19590877, 19591608, 19593445, 19597439, 19597583, 19601762, 19604659
19606174, 19617921, 19619732, 19623450, 19627012, 19632912, 19637186
19639483, 19644859, 19647503, 19649152, 19658708, 19662635, 19663176
19670108, 19676012, 19676905, 19680796, 19684504, 19687159, 19689979
19693090, 19699191, 19699946, 19701015, 19703301, 19705781, 19706965
19708342, 19708632, 19718981, 19721304, 19723336, 19730508, 19769480
19769625, 19777862, 19781326, 19784751, 19790243, 19791273, 19791377
19799847, 19805359, 19809171, 19811709, 19817386, 19818513, 19824871
19831647, 19835133, 19841800, 19855285, 19859472, 19865345, 19869255
19871910, 19873610, 19877336, 19879746, 19880190, 19883092, 19886165
19888853, 19889230, 19891090, 19895326, 19895362, 19896336, 19902195
19908836, 19909862, 19915271, 19928926, 19930276, 19931367, 19931709
19932634, 19933147, 19941352, 19943771, 19952975, 19957298, 19978542
19982584, 19988852, 19989009, 19990037, 19995869, 20001168, 20009569
20009833, 20011515, 20011646, 20011897, 20017509, 20023340, 20031873
20043616, 20048359, 20052269, 20061399, 20074391, 20076781, 20078186
20087383, 20093776, 20101006, 20117253, 20118035, 20122715, 20124446
20134113, 20134339, 20139391, 20144019, 20144308, 20165574, 20169408
20171986, 20172151, 20173897, 20175161, 20181030, 20212067, 20217801
20228093, 20229001, 20233181, 20235511, 20245930, 20250147, 20267166
20273319, 20281121, 20284155, 20294666, 20296619, 20298413, 20302006
20308798, 20315311, 20318889, 20322560, 20324049, 20328248, 20331945
20347562, 20348653, 20354900, 20356733, 20361671, 20368850, 20373598
20374572, 20378086, 20382309, 20387265, 20394750, 20397490, 20401975
20402832, 20408829, 20408866, 20413820, 20415564, 20424183, 20424899
20425790, 20428621, 20432873, 20437153, 20440930, 20441797, 20446883
20447445, 20459944, 20464614, 20466322, 20466628, 20468401, 20468490
20470877, 20471920, 20474192, 20475845, 20476175, 20480209, 20493163
20505778, 20509482, 20513399, 20524085, 20528052, 20539050, 20543011
20544065, 20554364, 20557786, 20558005, 20560611, 20562898, 20564072
20565112, 20565133, 20569094, 20577490, 20581111, 20582405, 20588486
20588502, 20591183, 20596234, 20598042, 20603378, 20603431, 20613079

20618595, 20627866, 20635353, 20641666, 20657411, 20657441, 20669434
20671094, 20673810, 20677396, 20677974, 20684983, 20686773, 20688221
20703000, 20703629, 20704450, 20705577, 20707932, 20708701, 20711718
20717081, 20717091, 20717359, 20725343, 20734332, 20736227, 20746251
20757079, 20764012, 20766180, 20768076, 20778986, 20794034, 20798891
20800890, 20801783, 20803014, 20825533, 20828947, 20830459, 20831538
20832516, 20835241, 20839705, 20842388, 20844426, 20848335, 20856766
20859910, 20860659, 20862087, 20868862, 20869721, 20875898, 20877664
20878790, 20879709, 20879889, 20880215, 20882568, 20890311, 20897759
20898391, 20898997, 20899461, 20904530, 20907061, 20914870, 20919320
20920911, 20922010, 20925795, 20926021, 20929771, 20936731, 20936905
20938170, 20951038, 20952966, 20958816, 20977794, 20978259, 21037923
21047766, 21047803, 21052842, 21059919, 21060755, 21061354, 21063322
21068507, 21072646, 21080143, 21091431, 21091901, 21095391, 21097043
21099555, 21101873, 21106027, 21132297, 21133343, 21142837, 21147908
21153266, 21157728, 21159665, 21164318, 21171382, 21172913, 21174504
21184223, 21186167, 21188532, 21188537, 21188584, 21196809, 21197626
21220620, 21225209, 21239530, 21241052, 21241829, 21246723, 21260397
21260431, 21263635, 21266085, 21270823, 21273804, 21275255, 21281532
21281607, 21285458, 21291274, 21293600, 21294938, 21296029, 21297872
21299490, 21300341, 21308727, 21315084, 21322887, 21329301, 21354456
21373076, 21373473, 21380789, 21383171, 21385422, 21387128, 21387964
21419850, 21421886, 21422580, 21424824, 21425496, 21429602, 21442094
21450666, 21476308, 21479753, 21492036, 21502702, 21514877, 21516611
21517440, 21522582, 21526048, 21532755, 21534893, 21542577, 21555660
21560152, 21566639, 21566944, 21566993, 21575362, 21620471, 21623164
21625179, 21626377, 21629064, 21632821, 21641414, 21641760, 21644640
21649497, 21656630, 21659726, 21665897, 21668627, 21675340, 21695575
21698350, 21744290, 21756661, 21756677, 21756699, 21764119, 21773465
21780146, 21785691, 21787056, 21794615, 21795111, 21811517, 21820934
21821302, 21828126, 21837606, 21842017, 21842740, 21847223, 21856522
21863727, 21868720, 21875360, 21889720, 21893235, 21896069, 21899588
21911701, 21913183, 21915719, 21917884, 21924131, 21960504, 21967197
21977186, 21977392, 22007324, 22018363, 22022760, 22024071, 22037014
22046677, 22062026, 22062517, 22068305, 22070866, 22072818, 22075064
22077517, 22083366, 22087683, 22092979, 22118835, 22118851, 22139226
22146062, 22148226, 22160989, 22165897, 22168163, 22173980, 22175564
22176950, 22178855, 22179537, 22185234, 22205263, 22214989, 22223463
22228324, 22232606, 22233505, 22238921, 22243719, 22243983, 22250006
22256431, 22256560, 22258530, 22264489, 22268833, 22282748, 22294260
22296366, 22301880, 22305887, 22346829, 22347493, 22351572, 22353199
22353346, 22359063, 22364044, 22365117, 22366322, 22366558, 22374754
22380919, 22454326, 22458049, 22465352, 22468781, 22475617, 22492533
22495062, 22495673, 22496904, 22499356, 22501616, 22503297, 22507210
22507234, 22515353, 22517782, 22518784, 22519146, 22520320, 22528741
22529728, 22533631, 22536802, 22551446, 22568016, 22568177, 22568797
22606521, 22624709, 22645009, 22654475, 22657942, 22670385, 22670413
22674709, 22675136, 22686674, 22690648, 22695831, 22707244, 22707866
22721409, 22729345, 22730454, 22733141, 22734547, 22750215, 22757364
22760595, 22760679, 22762046, 22782647, 22806698, 22808310, 22809871
22815955, 22816287, 22820579, 22826718, 22836801, 22842151, 22855193
22862134, 22865673, 22873635, 22894949, 22897344, 22901797, 22905130
22916353, 22922076, 22923409, 22950945, 22961508, 22972770, 22977256
23002524, 23003979, 23007241, 23008056, 23019710, 23020270, 23025340
23026585, 23028781, 23029562, 23035249, 23053606, 23061453, 23061702
23065323, 23066146, 23068169, 23080557, 23084507, 23088803, 23089357
23096938, 23101501, 23104033, 23105538, 23108128, 23115139, 23124895
23125826, 23126410, 23140259, 23148260, 23149541, 23151677, 23168363
23170620, 23172924, 23177536, 23177923, 23184263, 23195445, 23197103
23209741, 23220453, 23229229, 23237313, 23240358, 23260854, 23262847
23265914, 23265965, 23266217, 23272045, 23294548, 23302839, 23314180
23315153, 23315889, 23324000, 23326313, 23328639, 23338911, 23342170
23492665, 23501901, 23514710, 23514911, 23521523, 23528412, 23533524
23533807, 23543183, 23548817, 23567857, 23571055, 23572982, 23602213
23614158, 23628685, 23642282, 23709062, 23711335, 23713236, 23717151
23725036, 23727148, 23731896, 23746128, 23854396, 24285405, 24300640

24303148, 24307571, 24308635, 24315824, 24316947, 24321547, 24326444
24341675, 24350620, 24350831, 24365589, 24385625, 24385983, 24386767
24393981, 24397438, 24401351, 24411921, 24413809, 24415926, 24416451
24421668, 24423416, 24425998, 24437510, 24448240, 24448282, 24457597
24461826, 24509056, 24523374, 24534298, 24555417, 24560906, 24563422
24570598, 24573817, 24577566, 24589081, 24600330, 24624166, 24642295
24652769, 24662775, 24674955, 24683149, 24690216, 24693382, 24701840
24713381, 24717859, 24718260, 24719736, 24737064, 24737403, 24737581
24737954, 24739928, 24752618, 24766121, 24790914, 24792678, 24796092
24801152, 24802934, 24808595, 24812585, 24825843, 24831514, 24835538
24835919, 24848928, 24907917, 24908321, 24917972, 24920582, 24929210
24966594, 25029423, 25031502, 25034396, 25042823, 25047724, 25051465
25056052, 25058080, 25060506, 25067795, 25076732, 25076756, 25079710
25091141, 25093739, 25093872, 25099339, 25107334, 25110233, 25123585
25150925, 25161298, 25165496, 25178179, 25192729, 25210690, 25240188
25248384, 25264559, 25300427, 25307368, 25313154, 25328093, 25330273
25353983, 25357142, 25377044, 25392535, 25405687, 25415713, 25417056
25417958, 25423453, 25427662, 25429959, 25437695, 25437699, 25459958
25472885, 25475853, 25476125, 25477657, 25482971, 25483815, 25484507
25486384, 25489342, 25489367, 25489607, 25490238, 25492379, 25494379
25494413, 25495682, 25539063, 25546608, 25547060, 25551676, 25555252
25575628, 25579761, 25599425, 25600342, 25600421, 25602488, 25606091
25612095, 25616268, 25633101, 25634317, 25635149, 25639019, 25643931
25649873, 25653109, 25654936, 25655390, 25669791, 25670786, 25695903
25699321, 25722055, 25733479, 25740844, 25743479, 25760195, 25764020
25766822, 25775213, 25780343, 25789277, 25790353, 25809524, 25822410
25823532, 25823754, 25856821, 25861398, 25879984, 25881255, 25885148
25897615, 25914276, 25919622, 25947799, 25957038, 25982666, 25986062
25997810, 26007010, 26023002, 26023025, 26024732, 26027162, 26029780
26039623, 26088426, 26089440, 26110259, 26110632, 26111842, 26121990
26153977, 26187943, 26198757, 26198926, 26203182, 26243698, 26245237
26248143, 26256131, 26262953, 26263721, 26318627, 26324206, 26325856
26336977, 26353617, 26366517, 26412540, 26430737, 26439748, 26444887
26446098, 26482376, 26513067, 26513709, 26544823, 26546664, 26546754
26556014, 26569225, 26570171, 26575788, 26633558, 26635845, 26637592
26637824, 26654363, 26658759, 26714910, 26758193, 26768025, 26784509
26822620, 26828994, 26832296, 26844406, 26875822, 26898563, 26968670
26999139, 27000663, 27000690, 27001733, 27012701, 27015449, 27033652
27034890, 27052607, 27060167, 27072923, 27086138, 27092508, 27097854
27101105, 27122162, 27133662, 27169796, 27199245, 27207110, 27207634
27213224, 27217412, 27223075, 27229389, 27255377, 27274536, 27276231
27303938, 27314206, 27314390, 27314697, 27337759, 27348081, 27351628
27370965, 27375542, 27397048, 27404573, 27424405, 27433385, 27441326
27445727, 27461789, 27461842, 27468303, 27475603, 27487919, 27534509
27548131, 27567477, 27611612, 27620950, 27623159, 27629756, 27634991
27642235, 27710072, 27726780, 27751755, 27829295, 27846298, 27847259
27897759, 27923320, 27929509, 27938623, 27952577, 27952584, 27986817
27995248, 27997875, 28000269, 28022101, 28023399, 28024793, 28025414
28026866, 28043157, 28079127, 28098160, 28125601, 28164480, 28174827
28199085, 28215510, 28250929, 28281094, 28302049, 28305362, 28357401
28369092, 28384353, 28390273, 28420042, 28423598, 28432129, 28440711
28501075, 28502113, 28502128, 28507324, 28542455, 28566241, 28578164
28587723, 28612674, 28636676, 28639299, 28683167, 28708023, 28714988
28730253, 28758090, 28774416, 28790654, 28797711, 28821847, 28849751
28852325, 28867992, 28891741, 28915933, 28950969, 28986231, 28993590
29006527, 29009513, 29027694, 29030780, 29142109, 29163567, 29189889
29200700, 29247712, 29250317, 29251241, 29254615, 29260956, 29343156
29372460, 29378913, 29379978, 29388020, 29408136, 29434301, 29437712
29464779, 29483626, 29483672, 29483723, 29483771, 29500257, 29500963
29511611, 29559723, 29621961, 29633753, 29637560, 29645349, 29707896
29719146, 29726695, 29774367, 29774383, 29782211, 29791152, 29817278
29828111, 29869404, 29869906, 29893132, 29944660, 29961353, 29962927
29962939, 29997937, 30116203, 30128197, 30160639, 30164714, 30179644
30200680, 30215130, 30218044, 30225443, 30252098, 30295478, 30305880
30365745, 30497057, 30499600, 30502041, 30522998, 30534664, 30624864
30659882, 30668407, 30698289, 30758943, 30803210, 30805558, 30816938

30855121, 30973003, 30987088, 31001455, 31013127, 31022858, 31031715
31113348, 31156383, 31219939, 31306274, 31335037, 31335142

Version 12.1.0.2.v20

Version 12.1.0.2.v20 includes the following:

- Patch 30700212: Database PSU 12.1.0.2.200414
- Patch 30805558: Oracle JVM Component Database PSU 12.1.0.2.200414
- Patch 29997937: DSTv34 for RDBMS (TZDATA2019G)
- Patch 29997959: DSTv34 for OJVM (TZDATA2019G)
- Patch 17969866: Oracle GoldenGate - 46719: ENH: REPLICATION SUPPORT FOR INSERTS / FULL UPDATES WITH LARGE VALUES
- Patch 20394750: Oracle GoldenGate - APPLY CDR RESOLUTION FAILING FOR LOBS, XML, LONG, AND OBJECTS
- Patch 24835919: Oracle GoldenGate - IR EXECUTING DEPENDENT TRANSACTIONS OUT OF ORDER WITH PARALLELISM GREATER THAN
- Patch 23262847: Oracle GoldenGate - MALFORMED REDO CAUSED OGG REPLICATION ABEND
- Patch 21171382: DBMS_STATS Patch
- Patch 21091901: ONLINE MOVE OF HASH OR REF PARTITION CAN LEAVE LOCAL INDEXES INCONSISTENT
- Patch 31164857: JSON bundle patch
- Patch 20033733: PART :IMC:HIT ORA 600 [KGL-HEAP-SIZE-EXCEEDED]
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- PreUpgrade Jar: preupgrade_12.1.0.2.0_18_crlf.zipn
- Support for [Purging the recycle bin \(p. 1062\)](#)
- Support for [Generating performance reports with Automatic Workload Repository \(AWR\) \(p. 1053\)](#) using the `rdsadmin.rdsadmin_diagnostic_util` package

Combined patches for version 12.1.0.2, released April 2020

Bugs fixed:

19309466, 19902195, 18250893, 25437699, 19383839, 19781326, 16756406
18456643, 26546664, 22364044, 29483723, 18913440, 18845653, 28774416
19915271, 20172151, 18417036, 19516448, 24907917, 23713236, 24796092
23140259, 21856522, 29434301, 23028781, 19243521, 19658708, 18272672
21153266, 19174430, 28250929, 18900107, 22243719, 19548064, 26556014
20493163, 20688221, 22346829, 21387964, 13542050, 25377044, 22072818
22250006, 22734547, 22243983, 21623164, 21534893, 19012119, 19932634
19869255, 22232606, 18681056, 23324000, 25427662, 22068305, 24589081
19439759, 19303936, 18856106, 22916353, 24835538, 22353346, 19790243
21106027, 20023340, 26444887, 23088803, 22529728, 26256131, 18492302
19134173, 24303148, 21101873, 20447445, 27122162, 21188584, 19390567
26513709, 25780343, 19769480, 21097043, 21225209, 27217412, 26245237
20677396, 19284031, 19450314, 23514911, 19016730, 27303938, 22205263
22517782, 20919320, 22075064, 29621961, 22551446, 29250317, 22721409
18440095, 22496904, 27611612, 16439813, 18354830, 20596234, 22022760
20936905, 22268833, 23197103, 23746128, 22515353, 27897759, 21514877
19809171, 21186167, 26111842, 18990023, 13787015, 22492533, 25405687
22233505, 20173897, 24624166, 17210525, 29707896, 21260431, 25579761
20181030, 25056052, 19370504, 21868720, 24423416, 23068169, 19124589

22690648, 21383171, 19402853, 19888853, 25107334, 24341675, 17722075
20882568, 25653109, 23026585, 18604692, 20717081, 25546608, 20768076
27370965, 19081128, 22173980, 25722055, 23514710, 29483771, 19178851
20951038, 22168163, 25161298, 20569094, 24308635, 28302049, 19791377
19050649, 20920911, 29962939, 30365745, 22475617, 19189525, 19060015
19469538, 27052607, 29633753, 20598042, 22458049, 18988834, 21159665
23302839, 25307368, 25699321, 21924131, 21837606, 17409174, 20588486
22729345, 22842151, 25051465, 19238590, 16941434, 20387265, 29378913
24397438, 20673810, 23108128, 20356733, 28215510, 22380919, 18436647
21764119, 23065323, 20825533, 19124336, 22294260, 20839705, 24790914
20284155, 23571055, 6194865, 25539063, 17365043, 25914276, 20952966
22961508, 19176223, 21300341, 23237313, 18288842, 27223075, 22353199
20011515, 22083366, 28305362, 27634991, 25670786, 21419850, 26898563
22495673, 27986817, 19577410, 26248143, 23294548, 25328093, 23101501
24737064, 19931709, 25423453, 25547060, 23533807, 27726780, 24600330
20635353, 28384353, 25600421, 18122373, 20043616, 23124895, 18856999
21450666, 24752618, 18893947, 26633558, 20076781, 20926021, 26029780
21196809, 21354456, 22533631, 23725036, 20464614, 19562381, 27375542
24808595, 19189317, 25669791, 18307021, 21917884, 19708632, 28423598
27213224, 25633101, 29006527, 20711718, 18973548, 25982666, 25472885
19718981, 20684983, 23567857, 22826718, 25655390, 21773465, 20250147
20144019, 19197175, 26263721, 19597439, 28867992, 21387128, 22007324
28797711, 18818069, 21566639, 19180770, 19879746, 21785691, 20539050
20424183, 24285405, 21425496, 26544823, 19957298, 20322560, 29962927
22228324, 23172924, 22520320, 29817278, 28164480, 30179644, 27751755
21575362, 25058080, 22365117, 22645009, 25165496, 28950969, 27133662
27433385, 18774543, 20124446, 21429602, 29189889, 26153977, 30659882
19371175, 21863727, 18940497, 19074147, 22923409, 25489342, 21380789
19154375, 25417056, 19044962, 19532017, 23080557, 19662635, 22374754
20560611, 25654936, 21492036, 18705806, 28420042, 19578247, 20705577
22024071, 22238921, 29645349, 22809871, 21184223, 19995869, 23089357
19404068, 18921743, 19065677, 19018447, 19018206, 18308268, 19777862
27314697, 29027694, 22223463, 19304354, 22519146, 23020270, 22214989
19445860, 26654363, 27199245, 22977256, 20890311, 27445727, 28281094
21142837, 20869721, 22258530, 24555417, 22179537, 21756699, 18801391
18648816, 20217801, 18819908, 19550902, 22760595, 25483815, 19543384
23628685, 25482971, 30252098, 23007241, 19593445, 21080143, 27351628
20582405, 24966594, 20031873, 29828111, 25489367, 20374572, 18618122
24737581, 21698350, 22501616, 26784509, 28043157, 19306797, 24739928
18966843, 19077215, 20704450, 19068970, 20543011, 19023822, 24713381
22836801, 20432873, 21756677, 23168363, 20328248, 18674047, 18849537
20087383, 25459958, 20315311, 29163567, 22897344, 27534509, 26768025
20686773, 25178179, 19308965, 18948177, 20764012, 27623159, 19468991
20868862, 21780146, 23315153, 20466628, 21756661, 20397490, 19706965
20302006, 24831514, 23240358, 22178855, 19032777, 20862087, 19329654
18974476, 20603378, 21275255, 20859910, 29500963, 19307662, 26203182
21847223, 20281121, 28079127, 22568797, 19075256, 19076343, 28026866
29511611, 18866977, 22808310, 25635149, 20844426, 20904530, 20441797
20175161, 20296619, 19831647, 18548246, 30497057, 21442094, 25079710
24674955, 18840932, 18740837, 20294666, 27404573, 21037923, 25602488
21517440, 22062517, 19180394, 27337759, 19174942, 27092508, 20671094
21889720, 19347458, 19450116, 18411216, 20117253, 24386767, 24737954
20641666, 19931367, 25264559, 19930276, 22092979, 25616268, 21625179
20879709, 23003979, 20165574, 28578164, 19272708, 19547370, 22624709
23084507, 23184263, 20228093, 21281532, 25093872, 19805359, 26324206
19461270, 18700762, 19434529, 18799063, 20354900, 29388020, 20378086
17008068, 21246723, 20831538, 20424899, 20361671, 18674024, 19689979
24411921, 19873610, 16619249, 20562898, 21641414, 21091431, 19440586
20001168, 22757364, 22175564, 22499356, 20725343, 21241052, 19561643
28199085, 21270823, 20736227, 19399918, 19195895, 20830459, 20017509
18475439, 25790353, 21828126, 21665897, 25555252, 20746251, 19315668
22568177, 25764020, 25612095, 25357142, 23096938, 19067244, 19943771
18043064, 19941352, 21329301, 18885870, 26243698, 26187943, 20324049
30164714, 19536415, 30305880, 23709062, 28174827, 20446883, 27314206
21299490, 25313154, 18628388, 21744290, 18254023, 27072923, 25047724
20591183, 27847259, 20459944, 19185876, 18548433, 27207110, 22465352

24385625, 24326444, 24920582, 20402832, 19627012, 22733141, 29200700
20468401, 27441326, 27620950, 16863642, 19639483, 19315691, 27567477
21479753, 19174521, 23177923, 20401975, 18306996, 18851894, 21424824
27034890, 20581111, 20318889, 20936731, 21060755, 25240188, 26828994
27629756, 22256560, 19188927, 23328639, 27229389, 20766180, 20229001
24570598, 25475853, 21172913, 17655240, 29379978, 21266085, 19028800
19035573, 19366375, 28821847, 24523374, 25599425, 25034396, 19289642
21502702, 21291274, 18007682, 23521523, 20475845, 29408136, 22148226
22528741, 25417958, 29500257, 24652769, 26088426, 19326908, 19597583
17414008, 23019710, 20897759, 26822620, 22046677, 19663176, 20938170
19891090, 24825843, 26318627, 21960504, 20524085, 24509056, 19054077
21385422, 26262953, 22657942, 20428621, 21899588, 23326313, 19723336
28891741, 19835133, 17532734, 17495022, 25300427, 19333670, 21842017
19285025, 21373473, 29483626, 23260854, 23061453, 19687159, 14643995
22146062, 20977794, 20734332, 16938780, 17551063, 27548131, 21977392
28612674, 24461826, 19676012, 20588502, 23315889, 19520602, 23053606
19841800, 20245930, 19001359, 21476308, 26546754, 19393542, 30215130
23533524, 21099555, 29961353, 17532729, 27995248, 25429959, 19141838
19644859, 21915719, 19908386, 21421886, 19358317, 27101105, 19524158
29869404, 28758090, 23548817, 25861398, 20803014, 23025340, 19335438
19058490, 23642282, 19207117, 18799993, 25919622, 26569225, 25986062
20835241, 24662775, 20958816, 19475971, 18967382, 20347562, 25740844
20348653, 29009513, 19896336, 24812585, 20048359, 21896069, 20468490
19524384, 25392353, 21147908, 21695575, 30295478, 20440930, 30973003
25789277, 19171086, 24718260, 17867700, 19791273, 26110632, 27397048
21241829, 19591608, 18662619, 22707244, 18419520, 22296366, 22654475
18914624, 19571367, 28636676, 21522582, 29893132, 19501299, 26007010
19529868, 20425790, 19708342, 27487919, 27997875, 26968670, 16870214
18202441, 24415926, 18743542, 19001390, 21157728, 20657411, 19332396
22606521, 21875360, 21821302, 25091141, 28000269, 19149990, 20382309
22855193, 16777441, 19606174, 28542455, 20848335, 25495682, 19382851
20528052, 22762046, 24563422, 27468303, 23125826, 22503297, 28993590
25192729, 23338911, 27274536, 22730454, 19354794, 20757079, 19176326
20298413, 19048007, 22018363, 24300640, 18849970, 21532755, 20860659
22905130, 26121990, 21263635, 23602213, 27710072, 23209741, 22160989
18499088, 18775971, 22894949, 21059919, 18952989, 27348081, 22518784
25856821, 24457597, 25885148, 25484507, 20794034, 20554364, 21061354
19468347, 17533661, 19883092, 20657441, 24401351, 21285458, 28023399
18051556, 25330273, 26412540, 24425998, 19699191, 24437510, 16875041
20669434, 18964978, 25415713, 23342170, 22972770, 28369092, 20828947
21373076, 25492379, 25551676, 14283239, 25766822, 21967197, 22922076
19601762, 25575628, 26110259, 20368850, 21239530, 20437153, 24848928
20880215, 20798891, 25606091, 19013183, 29782211, 21095391, 25042823
21133343, 22695831, 24365589, 25248384, 25634317, 20134113, 19587324
20273319, 28501075, 18542562, 19017309, 26758193, 21063322, 22062026
24802934, 27829295, 20134339, 22077517, 22815955, 23854396, 24690216
22507210, 16354467, 20101006, 21795111, 27938623, 23501901, 18797519
25997810, 23029562, 25879984, 26844406, 21260397, 25029423, 29726695
19354335, 19730508, 22366558, 19390620, 26658759, 25822410, 6599380
20717359, 24321547, 27097854, 21297872, 18964939, 19871910, 29437712
26366517, 21913183, 25695903, 22366322, 20171986, 20603431, 21132297
25957038, 21542577, 29791152, 22507234, 23170620, 24719736, 25600342
18868646, 28587723, 29142109, 26637824, 20627866, 18110491, 16923858
24642295, 19518079, 20914870, 19339555, 20466322, 25823754, 25110233
20169408, 24908321, 20842388, 17274537, 26575788, 20474192, 21644640
28849751, 21794615, 18899974, 20471920, 22806698, 19052488, 29944660
29260956, 26198757, 19503821, 23717151, 24350620, 23126410, 20074391
25823532, 19157754, 22495062, 21220620, 24316947, 19865345, 19065556
22816287, 25947799, 20878790, 23492665, 21322887, 22305887, 19617921
20879889, 24350831, 19578350, 28022101, 26439748, 21893235, 19363645
21072646, 20898391, 19291380, 27060167, 18382302, 27086138, 22536802
22087683, 21197626, 21656630, 20373598, 19248799, 22707866, 28432129
19155797, 19279273, 18886413, 25490238, 20922010, 19990037, 25150925
20509482, 20778986, 22282748, 27255377, 24717859, 20703000, 22862134
21526048, 28683167, 24929210, 24560906, 19079752, 25486384, 20144308
21620471, 19670108, 19068610, 20267166, 25123585, 20476175, 28639299

18549238, 19297917, 20564072, 22950945, 19385656, 23528412, 19684504
19330795, 21174504, 28357401, 20899461, 20557786, 21911701, 19143550
20118035, 19024808, 25809524, 25760195, 20009833, 19604659, 16359751
26039623, 22820579, 28024793, 19928926, 23314180, 20212067, 24737403
20480209, 18904062, 29030780, 26430737, 25476125, 20856766, 17258582
27169796, 21668627, 26325856, 23272045, 20877664, 29247712, 19487147
23149541, 24577566, 19430401, 19676905, 28025414, 20925795, 26482376
22760679, 21296029, 21629064, 24416451, 23229229, 22865673, 20708701
25353983, 19280225, 21315084, 20613079, 19375649, 19213447, 19989009
18191823, 27314390, 26336977, 25775213, 30803210, 24393981, 22568016
27033652, 25639019, 17319928, 14705949, 19703301, 20308798, 28390273
21626377, 20122715, 6418158, 23105538, 25743479, 26198926, 28714988
19258504, 21188532, 24792678, 23151677, 17890099, 21649497, 26446098
16887946, 19693090, 26024732, 18791688, 19721304, 27012701, 19490948
29483672, 19619732, 21164318, 29559723, 21516611, 23148260, 18090142
21641760, 19818513, 22468781, 23002524, 20139391, 21052842, 24693382
19978542, 25477657, 23543183, 22165897, 19373893, 22359063, 19409212
18373438, 23035249, 21820934, 20677974, 1890693, 20470877, 19452434
21422580, 21632821, 22351572, 20235511, 23220453, 18742258, 18604493
23008056, 22901797, 18610915, 20978259, 20832516, 24801152, 27276231
26089440, 20907061, 25733479, 19523462, 18733351, 20505778, 19183343
21675340, 21787056, 21273804, 22782647, 20544065, 29719146, 25093739
17835294, 25210690, 28708023, 24413809, 27846298, 18371441, 26714910
24385983, 20413820, 22176950, 28986231, 24421668, 25897615, 25643931
23195445, 21281607, 20513399, 18841764, 28098160, 20558005, 20093776
18909599, 20618595, 23572982, 23104033, 19211433, 20331945, 19512341
23066146, 22256431, 19637186, 19022470, 22686674, 18607546, 26875822
24573817, 23115139, 19649152, 19201867, 21294938, 20898997, 18510194
21293600, 30218044, 21842740, 22454326, 24683149, 19534363, 25489607
23061702, 30805558, 30855121, 30502041, 30534664, 30128197, 30160639
22070866, 29774383, 29774367, 29251241, 29254615, 19165673, 28790654
28915933, 28440711, 28502128, 28502113, 27923320, 27952584, 27952577
27642235, 27475603, 27461789, 27461842, 25649873, 27001733, 27000663
27000690, 26635845, 26637592, 26570171, 26027162, 26023002, 26023025
25437695, 25494413, 25494379, 24917972, 25067795, 24534298, 25076732
25076756, 24315824, 21659726, 24448240, 24448282, 23177536, 22675136
23265914, 23265965, 23727148, 22674709, 22670413, 22670385, 21188537
22139226, 22118835, 22118851, 21555660, 21811517, 19623450, 21566993
21566944, 19176885, 21068507, 21047803, 21047766, 20415564, 20408829
20408866, 19877336, 19855285, 19909862, 19895362, 19895326, 19153980
19231857, 19223010, 19245191, 19699946, 28730253, 16799735, 17432124
18759211, 19396455, 20875898, 22037014, 22873635, 23614158, 24701840
25881255, 27015449, 28125601, 28852325, 29997937, 29997959, 17969866
20394750, 24835919, 23262847, 21171382, 21091901, 18727933, 18758877
18778801, 18803726, 18810904, 18895170, 18952766, 18999568, 19130152
19131386, 19131607, 19146474, 19168167, 19248279, 19313563, 19317646
19327391, 19364502, 19440520, 19448499, 19461428, 19468612, 19547774
19566592, 19571055, 19571082, 19583624, 19590877, 19632912, 19647503
19680796, 19701015, 19705781, 19769625, 19784751, 19799847, 19811709
19817386, 19824871, 19859472, 19880190, 19886165, 19889230, 19933147
19952975, 19982584, 19988852, 20009569, 20011646, 20011897, 20052269
20061399, 20233181, 20565112, 20565133, 20577490, 20703629, 20707932
20717091, 20800890, 20801783, 25031502

Version 12.1.0.2.v19

Version 12.1.0.2.v19 includes the following:

- Patch 30340202: DATABASE PATCH SET UPDATE 12.1.0.2.200114
- Patch 30502041: OJVM PATCH SET UPDATE 12.1.0.2.200114
- Patch 29997937: DSTv34 for RDBMS (TZDATA2019G)
- Patch 29997959: DSTv34 for OJVM (TZDATA2019G)

- Patch 17969866: Oracle GoldenGate - 46719: ENH: REPLICATION SUPPORT FOR INSERTS / FULL UPDATES WITH LARGE VALUES
- Patch 20394750: Oracle GoldenGate - APPLY CDR RESOLUTION FAILING FOR LOBS, XML, LONG, AND OBJECTS
- Patch 24835919: Oracle GoldenGate - IR EXECUTING DEPENDENT TRANSACTIONS OUT OF ORDER WITH PARALLELISM GREATER THAN
- Patch 23262847: Oracle GoldenGate - MALFORMED REDO CAUSED OGG REPLICATION ABEND
- Patch 21171382: DBMS_STATS Patch
- Patch 21091901: ONLINE MOVE OF HASH OR REF PARTITION CAN LEAVE LOCAL INDEXES INCONSISTENT
- Patch 30708149: JSON bundle patch
- Patch 20033733: PART :IMC:HIT ORA 600 [KGL-HEAP-SIZE-EXCEEDED]
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- PreUpgrade Jar: preupgrade_12.1.0.2.0_18_crlf.zip

Oracle patch 29918340, released January 2020

Bugs fixed:

```
19309466, 19902195, 18250893, 25437699, 19383839, 19781326, 16756406
18456643, 26546664, 22364044, 29483723, 18913440, 18845653, 19915271
20172151, 18417036, 19516448, 24907917, 23713236, 24796092, 23140259
29434301, 19243521, 19658708, 18272672, 21153266, 19174430, 18900107
22243719, 19548064, 26556014, 20493163, 20688221, 22346829, 21387964
13542050, 22072818, 22250006, 22734547, 22243983, 21623164, 19012119
21534893, 19932634, 19869255, 22232606, 18681056, 23324000, 25427662
22068305, 24589081, 19439759, 19303936, 18856106, 22916353, 24835538
22353346, 19790243, 21106027, 26444887, 20023340, 23088803, 22529728
26256131, 19134173, 18492302, 24303148, 21101873, 20447445, 27122162
21188584, 19390567, 26513709, 25780343, 19769480, 21097043, 21225209
26245237, 20677396, 19284031, 19450314, 23514911, 19016730, 22205263
22517782, 20919320, 22075064, 22551446, 29250317, 22721409, 18440095
22496904, 16439813, 18354830, 20596234, 22022760, 20936905, 22268833
23197103, 23746128, 22515353, 27897759, 21514877, 19809171, 21186167
26111842, 18990023, 13787015, 22492533, 22233505, 20173897, 24624166
17210525, 29707896, 21260431, 25579761, 20181030, 25056052, 19370504
21868720, 23068169, 19124589, 22690648, 21383171, 19402853, 19888853
25107334, 24341675, 17722075, 20882568, 25653109, 23026585, 18604692
20717081, 25546608, 20768076, 27370965, 19081128, 22173980, 25722055
23514710, 29483771, 19178851, 20951038, 22168163, 25161298, 20569094
24308635, 19791377, 19050649, 20920911, 29962939, 30365745, 22475617
19189525, 19060015, 19469538, 27052607, 29633753, 20598042, 22458049
18988834, 21159665, 23302839, 25307368, 25699321, 21924131, 17409174
21837606, 22729345, 22842151, 25051465, 19238590, 16941434, 20387265
29378913, 24397438, 20673810, 23108128, 20356733, 22380919, 18436647
21764119, 23065323, 20825533, 19124336, 22294260, 20839705, 24790914
20284155, 23571055, 6194865, 25539063, 17365043, 25914276, 20952966
22961508, 19176223, 21300341, 23237313, 18288842, 27223075, 22353199
20011515, 22083366, 27634991, 25670786, 28305362, 21419850, 26898563
22495673, 27986817, 19577410, 26248143, 23294548, 23101501, 24737064
19931709, 25423453, 25547060, 23533807, 27726780, 24600330, 28384353
20635353, 25600421, 18122373, 20043616, 23124895, 18856999, 21450666
18893947, 24752618, 26633558, 20076781, 26029780, 20926021, 21196809
21354456, 22533631, 23725036, 20464614, 19562381, 27375542, 24808595
19189317, 25669791, 18307021, 21917884, 19708632, 27213224, 25633101
29006527, 20711718, 18973548, 25982666, 25472885, 19718981, 20684983
23567857, 22826718, 25655390, 21773465, 20250147, 20144019, 19197175
26263721, 19597439, 21387128, 28867992, 22007324, 18818069, 21566639
```

19180770, 19879746, 21785691, 20539050, 20424183, 24285405, 21425496
26544823, 19957298, 20322560, 22228324, 29962927, 23172924, 22520320
29817278, 27751755, 21575362, 25058080, 22365117, 22645009, 25165496
28950969, 27433385, 18774543, 20124446, 21429602, 29189889, 26153977
19371175, 21863727, 18940497, 19074147, 22923409, 25489342, 21380789
19154375, 25417056, 19044962, 19532017, 19662635, 23080557, 22374754
20560611, 25654936, 21492036, 18705806, 28420042, 19578247, 20705577
22024071, 22238921, 22809871, 29645349, 21184223, 19995869, 23089357
19404068, 18921743, 19065677, 19018447, 19018206, 18308268, 19777862
29027694, 22223463, 19304354, 22519146, 22214989, 19445860, 26654363
27199245, 22977256, 20890311, 27445727, 21142837, 20869721, 22258530
24555417, 22179537, 21756699, 18801391, 20217801, 18819908, 19550902
22760595, 25483815, 19543384, 23628685, 25482971, 23007241, 19593445
30252098, 21080143, 27351628, 20582405, 24966594, 20031873, 29828111
25489367, 18618122, 24737581, 21698350, 22501616, 26784509, 28043157
19306797, 24739928, 18966843, 19077215, 20704450, 19068970, 20543011
19023822, 24713381, 22836801, 20432873, 21756677, 20328248, 18674047
18849537, 20087383, 25459958, 20315311, 29163567, 22897344, 27534509
26768025, 25178179, 19308965, 20686773, 18948177, 20764012, 27623159
19468991, 20868862, 21780146, 23315153, 20466628, 21756661, 20397490
19706965, 20302006, 24831514, 23240358, 22178855, 19032777, 20862087
19329654, 18974476, 20603378, 21275255, 20859910, 19307662, 26203182
21847223, 20281121, 22568797, 19075256, 28079127, 19076343, 28026866
29511611, 18866977, 22808310, 25635149, 20844426, 20904530, 20441797
20296619, 19831647, 18548246, 21442094, 25079710, 24674955, 18840932
18740837, 20294666, 21037923, 25602488, 21517440, 22062517, 19180394
27337759, 19174942, 27092508, 20671094, 21889720, 19347458, 19450116
18411216, 20117253, 24386767, 20641666, 24737954, 19931367, 25264559
19930276, 22092979, 25616268, 21625179, 20879709, 23003979, 20165574
28578164, 19272708, 19547370, 22624709, 23084507, 23184263, 20228093
21281532, 25093872, 19805359, 26324206, 19461270, 19434529, 18799063
20354900, 29388020, 20378086, 17008068, 21246723, 20831538, 20424899
20361671, 18674024, 19689979, 24411921, 19873610, 16619249, 20562898
21641414, 21091431, 19440586, 20001168, 22757364, 22175564, 22499356
20725343, 21241052, 19561643, 28199085, 20736227, 19399918, 19195895
20830459, 20017509, 18475439, 25790353, 21828126, 21665897, 25555252
20746251, 19315668, 22568177, 25764020, 25612095, 25357142, 23096938
19067244, 19943771, 18043064, 19941352, 21329301, 18885870, 26243698
26187943, 20324049, 19536415, 30164714, 23709062, 28174827, 20446883
27314206, 21299490, 25313154, 18628388, 21744290, 18254023, 27072923
25047724, 20591183, 27847259, 20459944, 19185876, 18548433, 27207110
22465352, 24385625, 24326444, 20402832, 19627012, 22733141, 29200700
20468401, 27441326, 27620950, 16863642, 19639483, 19315691, 27567477
21479753, 19174521, 23177923, 20401975, 18306996, 18851894, 21424824
27034890, 20581111, 20318899, 20936731, 21060755, 25240188, 26828994
22256560, 19188927, 23328639, 27229389, 20766180, 20229001, 24570598
25475853, 21172913, 17655240, 29379978, 21266085, 19028800, 19035573
19366375, 24523374, 28821847, 25599425, 25034396, 19289642, 21502702
21291274, 18007682, 23521523, 20475845, 29408136, 22148226, 22528741
25417958, 29500257, 24652769, 26088426, 19326908, 19597583, 17414008
23019710, 20897759, 26822620, 22046677, 19663176, 20938170, 19891090
24825843, 26318627, 21960504, 24509056, 20524085, 19054077, 21385422
26262953, 22657942, 20428621, 2189588, 23326313, 19723336, 19835133
17532734, 17495022, 25300427, 19333670, 21842017, 19285025, 21373473
29483626, 23260854, 23061453, 19687159, 14643995, 22146062, 20977794
20734332, 16938780, 17551063, 27548131, 21977392, 28612674, 24461826
19676012, 20588502, 23315889, 19520602, 23053606, 19841800, 20245930
19001359, 21476308, 26546754, 19393542, 23533524, 21099555, 27995248
17532729, 25429959, 19141838, 19644859, 21915719, 19908836, 21421886
19358317, 19524158, 27101105, 29869404, 23548817, 25861398, 20803014
23025340, 19335438, 19058490, 23642282, 19207117, 18799993, 25919622
26569225, 25986062, 20835241, 24662775, 19475971, 18967382, 20347562
20348653, 29009513, 19896336, 24812585, 20048359, 21896069, 20468490
19524384, 25392535, 21147908, 21695575, 30295478, 20440930, 25789277
19171086, 24718260, 17867700, 19791273, 26110632, 27397048, 21241829
19591608, 22707244, 18662619, 18419520, 22296366, 22654475, 18914624

19571367, 28636676, 21522582, 29893132, 19501299, 26007010, 19529868
20425790, 19708342, 27997875, 26968670, 16870214, 18202441, 24415926
18743542, 19001390, 21157728, 20657411, 19332396, 22606521, 21875360
21821302, 25091141, 28000269, 19149990, 20382309, 22855193, 16777441
19606174, 28542455, 20848335, 25495682, 19382851, 20528052, 22762046
24563422, 27468303, 23125826, 22503297, 28993590, 25192729, 23338911
22730454, 27274536, 19354794, 20757079, 19176326, 20298413, 19048007
22018363, 24300640, 18849970, 21532755, 20860659, 22905130, 26121990
21263635, 27710072, 23209741, 22160989, 18499088, 18775971, 22894949
21059919, 18952989, 27348081, 22518784, 25856821, 24457597, 25484507
20794034, 25885148, 20554364, 21061354, 19468347, 17533661, 19883092
20657441, 24401351, 21285458, 28023399, 18051556, 25330273, 26412540
24425998, 19699191, 24437510, 16875041, 20669434, 18964978, 23342170
22972770, 20828947, 21373076, 25492379, 25551676, 14283239, 25766822
21967197, 22922076, 19601762, 25575628, 26110259, 20368850, 21239530
20437153, 24848928, 20880215, 20798891, 25606091, 19013183, 25042823
21133343, 22695831, 24365589, 25248384, 25634317, 20134113, 19587324
20273319, 28501075, 18542562, 19017309, 26758193, 21063322, 22062026
20134339, 22077517, 22815955, 23854396, 24690216, 22507210, 16354467
20101006, 21795111, 27938623, 23501901, 18797519, 25997810, 23029562
25879984, 26844406, 21260397, 25029423, 19354335, 29726695, 19730508
22366558, 19390620, 26658759, 25822410, 6599380, 20717359, 24321547
27097854, 21297872, 18964939, 19871910, 29437712, 26366517, 21913183
25695903, 22366322, 20171986, 20603431, 21132297, 25957038, 21542577
22507234, 23170620, 24719736, 25600342, 18868646, 28587723, 29142109
26637824, 20627866, 18110491, 16923858, 24642295, 19518079, 20914870
19339555, 20466322, 25823754, 25110233, 20169408, 24908321, 20842388
17274537, 26575788, 20474192, 21644640, 28849751, 21794615, 18899974
20471920, 22806698, 19052488, 29944660, 29260956, 26198757, 19503821
23717151, 24350620, 23126410, 20074391, 19157754, 22495062, 21220620
24316947, 19865345, 19065556, 22816287, 25947799, 20878790, 23492665
21322887, 22305887, 19617921, 20879889, 24350831, 19578350, 28022101
26439748, 21893235, 19363645, 21072646, 20898391, 19291380, 27060167
18382302, 27086138, 22536802, 22087683, 21197626, 21656630, 20373598
19248799, 22707866, 28432129, 19155797, 19279273, 18886413, 25490238
20922010, 19990037, 25150925, 20509482, 20778986, 27255377, 24717859
20703000, 22862134, 21526048, 28683167, 24929210, 24560906, 20144308
21620471, 19670108, 19068610, 20267166, 25123585, 20476175, 18549238
28639299, 19297917, 20564072, 22950945, 19385656, 23528412, 19684504
19330795, 21174504, 28357401, 20899461, 20557786, 21911701, 19143550
20118035, 19024808, 25760195, 20009833, 19604659, 16359751, 26039623
22820579, 19928926, 23314180, 20212067, 24737403, 20480209, 18904062
29030780, 26430737, 20856766, 25476125, 17258582, 27169796, 21668627
23272045, 20877664, 26325856, 19487147, 23149541, 24577566, 19430401
19676905, 20925795, 26482376, 21296029, 21629064, 23229229, 22865673
20708701, 25353983, 19280225, 21315084, 20613079, 19375649, 19213447
19989009, 18191823, 27314390, 26336977, 25775213, 24393981, 22568016
25639019, 17319928, 14705949, 19703301, 28390273, 20308798, 21626377
20122715, 6418158, 23105538, 26198926, 25743479, 19258504, 28714988
21188532, 24792678, 23151677, 17890099, 21649497, 26446098, 16887946
26024732, 18791688, 19721304, 27012701, 19490948, 29483672, 19619732
21164318, 29559723, 21516611, 23148260, 18090142, 21641760, 19818513
22468781, 23002524, 20139391, 24693382, 19978542, 25477657, 23543183
22165897, 19373893, 22359063, 19409212, 18373438, 23035249, 21820934
20677974, 18990693, 20470877, 19452434, 21422580, 21632821, 22351572
20235511, 23220453, 18742258, 18604493, 23008056, 22901797, 18610915
20978259, 20832516, 24801152, 26089440, 27276231, 20907061, 25733479
19523462, 18733351, 20505778, 19183343, 21787056, 21273804, 22782647
20544065, 25093739, 17835294, 29719146, 25210690, 28708023, 24413809
27846298, 18371441, 26714910, 24385983, 20413820, 28986231, 24421668
25897615, 25643931, 23195445, 21281607, 20513399, 18841764, 28098160
20558005, 20093776, 18909599, 20618595, 23572982, 23104033, 19211433
20331945, 19512341, 23066146, 22256431, 19637186, 19022470, 22686674
18607546, 26875822, 24573817, 23115139, 19649152, 19201867, 21294938
20898997, 18510194, 21293600, 21842740, 22454326, 24683149, 19534363
25489607, 23061702

Version 12.1.0.2.v18

Version 12.1.0.2.v18 includes the following:

- Patch 29918340: DATABASE PATCH SET UPDATE 12.1.0.2.191015
- Patch 30128197: OJVM PATCH SET UPDATE 12.1.0.2.191015
- Patch 29997937: DSTv34 for RDBMS (TZDATA2019G)
- Patch 29997959: DSTv34 for OJVM (TZDATA2019G)
- Patch 17969866: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 20394750: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 24835919: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 21171382: DBMS_STATS Patch
- Patch 21091901: ONLINE MOVE OF HASH OR REF PARTITION CAN LEAVE LOCAL INDEXES INCONSISTENT
- Patch 30370890: JSON bundle patch
- Patch 20033733: PART :IMC: HIT ORA 600 [KGL-HEAP-SIZE-EXCEEDED]
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR
- PreUpgrade Jar: preupgrade_12.1.0.2.0_18_crlf.zip
- Support for [Resizing the temporary tablespace in a read replica \(p. 1061\)](#)

Oracle patch 29918340, released October 2019

Bugs fixed:

19309466, 19902195, 18250893, 25437699, 19383839, 19781326, 16756406
18456643, 26546664, 22364044, 29483723, 18845653, 18913440, 19915271
20172151, 18417036, 19516448, 24907917, 23713236, 24796092, 23140259
19243521, 19658708, 18272672, 21153266, 19174430, 18900107, 22243719
19548064, 26556014, 20493163, 20688221, 22346829, 21387964, 13542050
22072818, 22250006, 22734547, 22243983, 21623164, 19012119, 19932634
19869255, 22232606, 18681056, 23324000, 25427662, 22068305, 24589081
19439759, 19303936, 22916353, 24835538, 22353346, 19790243, 21106027
26444887, 23088803, 22529728, 26256131, 19134173, 24303148, 21101873
20447445, 27122162, 21188584, 19390567, 26513709, 25780343, 19769480
21097043, 21225209, 26245237, 20677396, 19284031, 19450314, 19016730
23514911, 22205263, 22517782, 20919320, 22075064, 22551446, 29250317
22721409, 18440095, 22496904, 16439813, 18354830, 20596234, 22022760
20936905, 22268833, 23197103, 22515353, 23746128, 21514877, 19809171
21186167, 26111842, 18990023, 13787015, 22492533, 22233505, 20173897
24624166, 17210525, 21260431, 29707896, 25579761, 20181030, 25056052
19370504, 21868720, 23068169, 19124589, 22690648, 21383171, 19402853
19888853, 25107334, 24341675, 17722075, 20882568, 25653109, 23026585
18604692, 20717081, 25546608, 20768076, 27370965, 19081128, 22173980
25722055, 23514710, 19178851, 20951038, 22168163, 25161298, 20569094
24308635, 19791377, 19050649, 20920911, 22475617, 19189525, 19469538
27052607, 20598042, 29633753, 22458049, 18988834, 21159665, 23302839
25307368, 25699321, 21924131, 17409174, 22729345, 22842151, 25051465
19238590, 16941434, 20387265, 29378913, 24397438, 20673810, 23108128
20356733, 22380919, 18436647, 23065323, 20825533, 19124336, 22294260
24790914, 20284155, 23571055, 6194865, 25539063, 17365043, 25914276
20952966, 22961508, 19176223, 21300341, 23237313, 18288842, 27223075
22353199, 20011515, 22083366, 27634991, 25670786, 21419850, 26898563

27986817, 19577410, 22495673, 26248143, 23294548, 23101501, 24737064
19931709, 25423453, 25547060, 23533807, 27726780, 24600330, 28384353
25600421, 18122373, 20043616, 23124895, 18856999, 21450666, 18893947
26633558, 20076781, 26029780, 21196809, 21354456, 22533631, 23725036
20464614, 19562381, 27375542, 24808595, 19189317, 25669791, 18307021
21917884, 19708632, 27213224, 25633101, 29006527, 20711718, 18973548
25982666, 19718981, 20684983, 23567857, 22826718, 25655390, 21773465
20250147, 20144019, 19197175, 26263721, 19597439, 21387128, 22007324
18818069, 21566639, 19180770, 19879746, 21785691, 20424183, 20539050
24285405, 21425496, 26544823, 19957298, 20322560, 22228324, 23172924
22520320, 27751755, 21575362, 25058080, 22365117, 22645009, 25165496
28950969, 18774543, 20124446, 21429602, 29189889, 26153977, 19371175
21863727, 18940497, 19074147, 22923409, 25489342, 21380789, 19154375
25417056, 19044962, 19532017, 19662635, 22374754, 20560611, 25654936
21492036, 18705806, 28420042, 19578247, 22024071, 22238921, 22809871
21184223, 19995869, 23089357, 19404068, 18921743, 19065677, 19018447
19018206, 18308268, 19777862, 29027694, 22223463, 19304354, 22519146
22214989, 19445860, 26654363, 27199245, 22977256, 20890311, 27445727
21142837, 20869721, 24555417, 22258530, 22179537, 21756699, 18801391
20217801, 18819908, 22760595, 19550902, 25483815, 23628685, 19543384
25482971, 23007241, 19593445, 21080143, 27351628, 20582405, 24966594
20031873, 25489367, 29828111, 18618122, 24737581, 21698350, 22501616
26784509, 24739928, 18966843, 19077215, 20704450, 19068970, 20543011
19023822, 24713381, 22836801, 20432873, 21756677, 20328248, 18674047
18849537, 20087383, 25459958, 20315311, 22897344, 29163567, 27534509
26768025, 25178179, 19308965, 18948177, 20764012, 27623159, 19468991
20868862, 21780146, 23315153, 20466628, 21756661, 20397490, 19706965
20302006, 24831514, 23240358, 22178855, 19032777, 20862087, 19329654
18974476, 20603378, 20859910, 19307662, 26203182, 21847223, 20281121
22568797, 19075256, 19076343, 28026866, 29511611, 18866977, 22808310
25635149, 20844426, 20904530, 20441797, 20296619, 18548246, 21442094
25079710, 24674955, 18840932, 18740837, 20294666, 21037923, 25602488
21517440, 22062517, 19180394, 27337759, 19174942, 20671094, 21889720
19347458, 19450116, 18411216, 20117253, 24386767, 20641666, 19931367
25264559, 19930276, 22092979, 25616268, 21625179, 20879709, 23003979
20165574, 28578164, 19272708, 19547370, 22624709, 23084507, 23184263
20228093, 21281532, 25093872, 19805359, 26324206, 19461270, 19434529
18799063, 20354900, 29388020, 20378086, 17008068, 21246723, 20831538
20424899, 20361671, 18674024, 19689979, 24411921, 19873610, 16619249
20562898, 21641414, 21091431, 19440586, 20001168, 22757364, 22175564
22499356, 20725343, 21241052, 19561643, 28199085, 20736227, 19399918
19195895, 20830459, 20017509, 18475439, 25790353, 21828126, 21665897
25555252, 20746251, 22568177, 25764020, 25612095, 25357142, 23096938
19067244, 18043064, 19941352, 21329301, 18885870, 26243698, 26187943
20324049, 19536415, 23709062, 28174827, 20446883, 27314206, 21299490
25313154, 18628388, 21744290, 18254023, 27072923, 25047724, 20591183
27847259, 20459944, 19185876, 18548433, 27207110, 22465352, 24385625
24326444, 20402832, 19627012, 22733141, 29200700, 20468401, 27441326
27620950, 16863642, 19639483, 19315691, 27567477, 21479753, 19174521
23177923, 20401975, 18306996, 18851894, 27034890, 21424824, 20581111
20318889, 20936731, 21060755, 25240188, 26828994, 22256560, 19188927
23328639, 27229389, 20766180, 20229001, 24570598, 25475853, 21172913
17655240, 29379978, 21266085, 19028800, 19035573, 19366375, 24523374
25599425, 25034396, 19289642, 21502702, 21291274, 18007682, 23521523
20475845, 29408136, 22148226, 22528741, 25417958, 24652769, 26088426
19326908, 19597583, 29500257, 17414008, 23019710, 20897759, 26822620
22046677, 19663176, 20938170, 19891090, 24825843, 26318627, 21960504
24509056, 19054077, 21385422, 26262953, 22657942, 20428621, 21899588
23326313, 19723336, 19835133, 17532734, 17495022, 25300427, 19333670
21842017, 19285025, 21373473, 23260854, 23061453, 19687159, 14643995
22146062, 20977794, 20734332, 17551063, 16938780, 27548131, 21977392
28612674, 24461826, 19676012, 20588502, 23315889, 19520602, 23053606
19841800, 20245930, 19001359, 21476308, 26546754, 19393542, 23533524
21099555, 27995248, 25429959, 19141838, 19644859, 21915719, 19908836
21421886, 19358317, 19524158, 23548817, 25861398, 20803014, 23025340
19335438, 19058490, 23642282, 19207117, 18799993, 25919622, 26569225

25986062, 20835241, 24662775, 19475971, 18967382, 20347562, 20348653
19896336, 24812585, 20048359, 21896069, 20468490, 19524384, 25392535
21147908, 21695575, 20440930, 30295478, 25789277, 19171086, 24718260
17867700, 19791273, 27397048, 21241829, 19591608, 22707244, 18419520
22296366, 22654475, 18914624, 19571367, 28636676, 21522582, 19501299
29893132, 26007010, 19529868, 20425790, 19708342, 27997875, 16870214
18202441, 24415926, 18743542, 19001390, 21157728, 20657411, 19332396
22606521, 21875360, 25091141, 21821302, 28000269, 19149990, 20382309
22855193, 16777441, 19606174, 20848335, 28542455, 25495682, 19382851
20528052, 22762046, 24563422, 23125826, 22503297, 28993590, 25192729
23338911, 22730454, 19354794, 20757079, 19176326, 20298413, 19048007
22018363, 18849970, 21532755, 24300640, 20860659, 22905130, 26121990
21263635, 27710072, 23209741, 22160989, 18499088, 18775971, 22894949
21059919, 18952989, 27348081, 22518784, 25856821, 24457597, 25484507
20794034, 20554364, 21061354, 19468347, 17533661, 19883092, 20657441
24401351, 21285458, 28023399, 18051556, 25330273, 26412540, 24425998
19699191, 24437510, 16875041, 20669434, 18964978, 22972770, 23342170
20828947, 21373076, 25492379, 25551676, 14283239, 25766822, 21967197
22922076, 19601762, 25575628, 26110259, 20368850, 21239530, 20437153
24848928, 20880215, 20798891, 25606091, 19013183, 25042823, 21133343
22695831, 24365589, 25248384, 25634317, 20134113, 19587324, 20273319
28501075, 18542562, 19017309, 26758193, 21063322, 22062026, 20134339
22077517, 22815955, 23854396, 24690216, 22507210, 16354467, 20101006
21795111, 27938623, 23501901, 18797519, 25997810, 23029562, 25879984
21260397, 25029423, 26844406, 19354335, 19730508, 22366558, 19390620
26658759, 25822410, 6599380, 20717359, 24321547, 27097854, 21297872
18964939, 19871910, 29437712, 26366517, 21913183, 25695903, 22366322
20171986, 20603431, 21132297, 25957038, 21542577, 22507234, 23170620
24719736, 25600342, 18868646, 28587723, 29142109, 26637824, 20627866
18110491, 16923858, 24642295, 19518079, 19339555, 20466322, 25823754
25110233, 20169408, 24908321, 20842388, 17274537, 26575788, 20474192
21644640, 28849751, 21794615, 18899974, 20471920, 22806698, 19052488
26198757, 19503821, 23717151, 24350620, 23126410, 20074391, 19157754
22495062, 21220620, 24316947, 19865345, 19065556, 22816287, 25947799
20878790, 23492665, 21322887, 22305887, 20879889, 19617921, 24350831
19578350, 28022101, 26439748, 21893235, 19363645, 21072646, 20898391
19291380, 27060167, 18382302, 27086138, 22536802, 22087683, 21197626
21656630, 20373598, 19248799, 22707866, 28432129, 19155797, 19279273
18886413, 25490238, 20922010, 19990037, 25150925, 20509482, 20778986
27255377, 24717859, 20703000, 22862134, 21526048, 28683167, 24929210
24560906, 20144308, 21620471, 19670108, 19068610, 20267166, 25123585
20476175, 18549238, 19297917, 22950945, 19385656, 20564072, 23528412
19684504, 19330795, 21174504, 28357401, 20899461, 20557786, 21911701
19143550, 20118035, 19024808, 25760195, 20009833, 19604659, 16359751
26039623, 22820579, 19928926, 23314180, 20212067, 24737403, 20480209
18904062, 29030780, 26430737, 20856766, 17258582, 27169796, 21668627
23272045, 20877664, 19487147, 23149541, 24577566, 19430401, 19676905
20925795, 26482376, 21296029, 21629064, 23229229, 22865673, 20708701
25353983, 19280225, 21315084, 20613079, 19375649, 19213447, 19989009
18191823, 27314390, 26336977, 25775213, 24393981, 22568016, 25639019
17319928, 14705949, 19703301, 28390273, 21626377, 20122715, 6418158
23105538, 26198926, 19258504, 21188532, 24792678, 23151677, 17890099
21649497, 26446098, 16887946, 26024732, 18791688, 19721304, 27012701
19490948, 29483672, 19619732, 21164318, 29559723, 23148260, 18090142
21641760, 19818513, 23002524, 22468781, 20139391, 24693382, 19978542
25477657, 23543183, 22165897, 19373893, 22359063, 19409212, 18373438
23035249, 20677974, 18990693, 20470877, 21422580, 21632821, 22351572
20235511, 23220453, 18742258, 18604493, 23008056, 22901797, 18610915
20978259, 20832516, 24801152, 26089440, 20907061, 19523462, 25733479
20505778, 18733351, 19183343, 21787056, 21273804, 22782647, 20544065
25093739, 17835294, 28708023, 24413809, 27846298, 18371441, 26714910
24385983, 20413820, 28986231, 24421668, 25897615, 25643931, 23195445
21281607, 20513399, 18841764, 20558005, 20093776, 18909599, 20618595
23572982, 23104033, 19211433, 20331945, 19512341, 22256431, 19637186
23066146, 19022470, 22686674, 18607546, 26875822, 24573817, 23115139
19649152, 19201867, 21294938, 20898997, 18510194, 21293600, 21842740

22454326, 24683149, 19534363, 25489607, 23061702

Version 12.1.0.2.v17

Version 12.1.0.2.v17 includes the following:

- Patch 29494060: DATABASE PATCH SET UPDATE 12.1.0.2.190716
- Patch 29774383: OJVM PATCH SET UPDATE 12.1.0.2.190716
- Patch 28852325: DSTv33 for RDBMS (TZDATA2018G)
- Patch 28852334: DSTv33 for OJVM (TZDATA2018G)
- Patch 17969866: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 20394750: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 24835919: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 21171382: DBMS_STATS Patch
- Patch 21091901: ONLINE MOVE OF HASH OR REF PARTITION CAN LEAVE LOCAL INDEXES INCONSISTENT
- Patch 29958796: JSON bundle patch
- Patch 20033733: PART :IMC: HIT ORA 600 [KGL-HEAP-SIZE-EXCEEDED]
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR

Oracle patch 29494060, released July 2019

Bugs fixed:

19309466, 19902195, 18250893, 25437699, 19383839, 19781326, 16756406
18456643, 26546664, 22364044, 18845653, 19915271, 20172151, 18417036
19516448, 23713236, 24907917, 24796092, 23140259, 19243521, 19658708
18272672, 21153266, 19174430, 22243719, 19548064, 26556014, 20493163
20688221, 21387964, 13542050, 22250006, 22734547, 22243983, 21623164
19012119, 19932634, 19869255, 22232606, 18681056, 23324000, 25427662
22068305, 24589081, 19439759, 19303936, 22916353, 24835538, 22353346
19790243, 21106027, 26444887, 23088803, 22529728, 26256131, 19134173
24303148, 20447445, 21101873, 21188584, 19390567, 26513709, 25780343
19769480, 21097043, 21225209, 26245237, 20677396, 19284031, 19450314
19016730, 22517782, 20919320, 22205263, 22075064, 22551446, 22721409
18440095, 22496904, 16439813, 18354830, 20596234, 22022760, 20936905
23197103, 22515353, 21514877, 19809171, 21186167, 26111842, 18990023
13787015, 22492533, 20173897, 24624166, 17210525, 21260431, 20181030
25056052, 19370504, 21868720, 23068169, 19124589, 21383171, 19402853
22690648, 19888853, 24341675, 17722075, 25107334, 20882568, 25653109
23026585, 18604692, 20717081, 25546608, 27370965, 19081128, 20768076
22173980, 25722055, 23514710, 19178851, 20951038, 22168163, 25161298
20569094, 24308635, 19791377, 19050649, 20920911, 22475617, 19189525
19469538, 27052607, 20598042, 22458049, 18988834, 21159665, 23302839
25307368, 25699321, 17409174, 22729345, 22842151, 19238590, 25051465
16941434, 20387265, 24397438, 29378913, 20673810, 23108128, 20356733
22380919, 18436647, 23065323, 20825533, 19124336, 22294260, 24790914
20284155, 23571055, 25539063, 17365043, 25914276, 20952966, 22961508
19176223, 21300341, 23237313, 18288842, 27223075, 22353199, 20011515
22083366, 27634991, 25670786, 21419850, 26898563, 27986817, 19577410
26248143, 23294548, 23101501, 24737064, 19931709, 25423453, 25547060
23533807, 27726780, 24600330, 25600421, 18122373, 20043616, 23124895
18856999, 21450666, 18893947, 26633558, 20076781, 26029780, 21196809
21354456, 22533631, 23725036, 20464614, 19562381, 27375542, 24808595
19189317, 25669791, 18307021, 21917884, 19708632, 27213224, 25633101

29006527, 20711718, 18973548, 25982666, 19718981, 23567857, 22826718
25655390, 20684983, 21773465, 20250147, 20144019, 19197175, 26263721
19597439, 21387128, 22007324, 19180770, 18818069, 21566639, 19879746
21785691, 20424183, 24285405, 21425496, 26544823, 20322560, 22228324
23172924, 22520320, 27751755, 21575362, 25058080, 22365117, 22645009
25165496, 28950969, 18774543, 20124446, 21429602, 26153977, 29189889
19371175, 21863727, 18940497, 19074147, 22923409, 25489342, 21380789
19154375, 19044962, 25417056, 19532017, 19662635, 22374754, 20560611
25654936, 21492036, 18705806, 28420042, 19578247, 22024071, 22238921
22809871, 21184223, 19995869, 23089357, 19404068, 18921743, 19065677
19018447, 19018206, 18308268, 19777862, 29027694, 22223463, 19304354
22519146, 19445860, 26654363, 27199245, 22977256, 20890311, 27445727
21142837, 20869721, 24555417, 22179537, 21756699, 20217801, 18819908
22760595, 25483815, 23628685, 23007241, 19593445, 21080143, 27351628
20582405, 24966594, 20031873, 25489367, 18618122, 24737581, 21698350
26784509, 24739928, 18966843, 19077215, 20704450, 19068970, 20543011
19023822, 24713381, 20432873, 21756677, 22836801, 20328248, 18674047
18849537, 20087383, 25459958, 20315311, 22897344, 27534509, 25178179
19308965, 18948177, 19468991, 20868862, 21780146, 23315153, 20466628
21756661, 20397490, 19706965, 20302006, 24831514, 23240358, 22178855
19032777, 20862087, 19329654, 18974476, 20603378, 20859910, 19307662
26203182, 21847223, 20281121, 22568797, 19075256, 19076343, 18866977
28026866, 29511611, 22808310, 25635149, 20844426, 20904530, 20441797
21442094, 25079710, 24674955, 18840932, 18740837, 20294666, 25602488
21517440, 22062517, 19180394, 27337759, 19174942, 20671094, 21889720
19450116, 18411216, 20117253, 24386767, 20641666, 19931367, 25264559
19930276, 22092979, 25616268, 21625179, 20879709, 23003979, 20165574
28578164, 19272708, 19547370, 22624709, 23084507, 23184263, 20228093
21281532, 25093872, 19805359, 26324206, 19461270, 19434529, 18799063
20354900, 20378086, 29388020, 17008068, 21246723, 20831538, 20424899
20361671, 18674024, 19689979, 24411921, 19873610, 16619249, 20562898
21641414, 21091431, 19440586, 20001168, 22757364, 22175564, 20725343
21241052, 19561643, 20736227, 19399918, 19195895, 20830459, 20017509
18475439, 25790353, 21828126, 21665897, 25555252, 20746251, 22568177
25764020, 25612095, 25357142, 23096938, 19067244, 18043064, 21329301
18885870, 26243698, 26187943, 20324049, 19536415, 23709062, 28174827
20446883, 27314206, 21299490, 25313154, 18628388, 21744290, 18254023
25047724, 20591183, 27847259, 19185876, 18548433, 27207110, 22465352
24385625, 24326444, 20402832, 19627012, 29200700, 20468401, 27441326
27620950, 16863642, 19639483, 19315691, 27567477, 21479753, 19174521
23177923, 20401975, 18306996, 18851894, 27034890, 20581111, 20318889
20936731, 21060755, 25240188, 26828994, 22256560, 19188927, 23328639
27229389, 20766180, 20229001, 24570598, 25475853, 21172913, 17655240
21266085, 19028800, 29379978, 19035573, 19366375, 24523374, 25599425
25034396, 19289642, 21502702, 21291274, 18007682, 23521523, 20475845
29408136, 22148226, 22528741, 25417958, 24652769, 26088426, 19326908
19597583, 17414008, 23019710, 20897759, 26822620, 22046677, 19663176
20938170, 19891090, 24825843, 26318627, 21960504, 24509056, 19054077
26262953, 22657942, 20428621, 21899588, 23326313, 19723336, 19835133
17532734, 25300427, 19333670, 17495022, 21842017, 19285025, 21373473
23260854, 23061453, 19687159, 14643995, 20977794, 20734332, 17551063
27548131, 21977392, 24461826, 28612674, 19676012, 20588502, 23315889
19520602, 23053606, 19841800, 20245930, 19001359, 21476308, 26546754
19393542, 23533524, 21099555, 27995248, 25429959, 19141838, 19644859
21915719, 19908836, 21421886, 19358317, 19524158, 23548817, 25861398
20803014, 23025340, 19335438, 19058490, 23642282, 19207117, 18799993
25919622, 26569225, 25986062, 20835241, 24662775, 19475971, 18967382
20347562, 20348653, 19896336, 24812585, 20048359, 21896069, 20468490
19524384, 25392535, 21147908, 21695575, 20440930, 25789277, 19171086
24718260, 17867700, 19791273, 27397048, 21241829, 19591608, 22707244
18419520, 22296366, 22654475, 18914624, 19571367, 28636676, 21522582
19501299, 26007010, 19529868, 20425790, 19708342, 27997875, 16870214
18202441, 24415926, 18743542, 19001390, 21157728, 20657411, 19332396
21875360, 22606521, 25091141, 28000269, 19149990, 20382309, 22855193
16777441, 19606174, 20848335, 25495682, 19382851, 20528052, 22762046
24563422, 23125826, 22503297, 28993590, 25192729, 23338911, 22730454

19354794, 20757079, 19176326, 20298413, 19048007, 22018363, 18849970
21532755, 20860659, 22905130, 26121990, 21263635, 27710072, 22160989
23209741, 18499088, 22894949, 21059919, 18952989, 27348081, 22518784
25856821, 24457597, 25484507, 20794034, 21061354, 20554364, 19468347
17533661, 19883092, 20657441, 24401351, 21285458, 28023399, 18051556
25330273, 26412540, 19699191, 24437510, 16875041, 20669434, 18964978
22972770, 20828947, 21373076, 25492379, 25551676, 14283239, 25766822
21967197, 22922076, 25575628, 19601762, 26110259, 20368850, 21239530
20437153, 24848928, 20880215, 20798891, 25606091, 19013183, 25042823
21133343, 22695831, 24365589, 25248384, 25634317, 20134113, 19587324
20273319, 28501075, 18542562, 19017309, 26758193, 21063322, 22062026
20134339, 22077517, 22815955, 23854396, 24690216, 22507210, 16354467
20101006, 21795111, 27938623, 23501901, 18797519, 25997810, 25879984
21260397, 25029423, 19354335, 19730508, 22366558, 26658759, 25822410
6599380, 20717359, 24321547, 27097854, 21297872, 18964939, 19871910
29437712, 26366517, 21913183, 22366322, 20171986, 25695903, 20603431
21132297, 25957038, 21542577, 22507234, 23170620, 24719736, 25600342
18868646, 28587723, 29142109, 26637824, 20627866, 18110491, 16923858
24642295, 19518079, 19339555, 20466322, 25823754, 25110233, 20169408
24908321, 20842388, 17274537, 26575788, 20474192, 21644640, 28849751
21794615, 18899974, 20471920, 22806698, 19052488, 26198757, 19503821
24350620, 20074391, 19157754, 21220620, 22495062, 24316947, 19865345
19065556, 22816287, 25947799, 20878790, 23492665, 21322887, 22305887
20879889, 24350831, 19578350, 28022101, 26439748, 21893235, 19363645
21072646, 20898391, 19291380, 27060167, 18382302, 27086138, 22536802
22087683, 21197626, 21656630, 20373598, 19248799, 22707866, 28432129
19155797, 19279273, 18886413, 25490238, 20922010, 19990037, 25150925
20509482, 27255377, 24717859, 20703000, 22862134, 21526048, 28683167
24929210, 24560906, 20144308, 21620471, 19670108, 19068610, 20267166
25123585, 20476175, 18549238, 19297917, 22950945, 19385656, 23528412
19684504, 19330795, 21174504, 28357401, 20899461, 20557786, 21911701
19143550, 20118035, 19024808, 25760195, 20009833, 19604659, 16359751
26039623, 22820579, 19928926, 23314180, 20212067, 24737403, 20480209
18904062, 29030780, 26430737, 20856766, 17258582, 27169796, 21668627
20877664, 23272045, 19487147, 23149541, 24577566, 19430401, 19676905
20925795, 26482376, 21296029, 21629064, 23229229, 22865673, 20708701
25353983, 19280225, 21315084, 20613079, 19375649, 19213447, 19989009
18191823, 27314390, 26336977, 25775213, 24393981, 25639019, 17319928
14705949, 19703301, 28390273, 21626377, 20122715, 6418158, 23105538
26198926, 19258504, 21188532, 23151677, 24792678, 17890099, 21649497
26446098, 16887946, 26024732, 18791688, 19721304, 27012701, 19490948
19619732, 21164318, 23148260, 18090142, 21641760, 19818513, 23002524
20139391, 24693382, 19978542, 23543183, 22165897, 19373893, 22359063
19409212, 18373438, 23035249, 20677974, 18990693, 20470877, 21422580
21632821, 22351572, 20235511, 23220453, 18742258, 18604493, 23008056
22901797, 18610915, 20978259, 20832516, 24801152, 26089440, 20907061
19523462, 20505778, 19183343, 21787056, 21273804, 22782647, 25093739
17835294, 28708023, 24413809, 27846298, 18371441, 26714910, 24385983
20413820, 24421668, 28986231, 25897615, 25643931, 23195445, 21281607
20513399, 20558005, 20093776, 18909599, 20618595, 23572982, 19211433
20331945, 19512341, 22256431, 19637186, 19022470, 18607546, 26875822
24573817, 23115139, 19649152, 19201867, 21294938, 20898997, 18510194
21842740, 22454326, 24683149, 19534363, 25489607, 23061702

Version 12.1.0.2.v16

Version 12.1.0.2.v16 includes the following:

- Patch 29141015: Database Patch Set Update: 12.1.0.2.190416
- Patch 29251241: OJVM PATCH SET UPDATE 12.1.0.2.190416
- Patch 28852325: DSTv33 for RDBMS (TZDATA2018G)
- Patch 28852334: DSTv33 for OJVM (TZDATA2018G)

- Patch 17969866: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 20394750: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 24835919: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 21171382: DBMS_STATS Patch
- Patch 21091901: ONLINE MOVE OF HASH OR REF PARTITION CAN LEAVE LOCAL INDEXES INCONSISTENT
- Patch 29600862: JSON bundle patch
- Patch 20033733: PART :IMC:HIT ORA 600 [KGL-HEAP-SIZE-EXCEEDED]
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR

Oracle patch 22785785, released April 2019

Bugs fixed:

```
19309466, 19902195, 18250893, 25437699, 19383839, 19781326, 16756406
18456643, 26546664, 22364044, 18845653, 19915271, 20172151, 18417036
19516448, 23713236, 24796092, 23140259, 19243521, 19658708, 18272672
21153266, 19174430, 22243719, 26556014, 20493163, 20688221, 21387964
13542050, 22734547, 22243983, 21623164, 19012119, 19932634, 19869255
22232606, 18681056, 23324000, 25427662, 22068305, 24589081, 19439759
19303936, 22916353, 24835538, 22353346, 21106027, 19790243, 26444887
23088803, 22529728, 26256131, 19134173, 20447445, 21188584, 19390567
26513709, 25780343, 19769480, 21097043, 21225209, 26245237, 20677396
19284031, 19450314, 19016730, 20919320, 22517782, 22075064, 22551446
22721409, 18440095, 22496904, 16439813, 18354830, 20596234, 22022760
20936905, 23197103, 21514877, 21186167, 26111842, 18990023, 13787015
22492533, 20173897, 24624166, 17210525, 21260431, 20181030, 25056052
19370504, 21868720, 23068169, 19124589, 21383171, 19402853, 19888853
24341675, 17722075, 20882568, 25653109, 23026585, 18604692, 20717081
25546608, 27370965, 19081128, 22173980, 23514710, 25722055, 19178851
20951038, 22168163, 25161298, 20569094, 24308635, 19791377, 19050649
20920911, 22475617, 19189525, 19469538, 27052607, 20598042, 22458049
18988834, 23302839, 25307368, 17409174, 22729345, 22842151, 19238590
16941434, 20387265, 24397438, 20673810, 23108128, 20356733, 22380919
18436647, 23065323, 20825533, 19124336, 22294260, 24790914, 20284155
23571055, 25539063, 17365043, 25914276, 20952966, 22961508, 19176223
21300341, 23237313, 18288842, 27223075, 22353199, 20011515, 22083366
25670786, 27634991, 21419850, 26898563, 19577410, 27986817, 26248143
23294548, 24737064, 23101501, 19931709, 25423453, 25547060, 23533807
27726780, 24600330, 25600421, 18122373, 20043616, 23124895, 18856999
21450666, 18893947, 26633558, 20076781, 26029780, 21196809, 21354456
23725036, 20464614, 22533631, 19562381, 27375542, 24808595, 19189317
25669791, 18307021, 21917884, 19708632, 27213224, 25633101, 20711718
29006527, 18973548, 25982666, 19718981, 23567857, 22826718, 25655390
21773465, 20250147, 20144019, 19197175, 26263721, 19597439, 21387128
22007324, 19180770, 19879746, 21785691, 20424183, 24285405, 26544823
20322560, 22228324, 23172924, 22520320, 21575362, 27751755, 25058080
22365117, 22645009, 25165496, 28950969, 18774543, 20124446, 21429602
26153977, 19371175, 21863727, 18940497, 19074147, 22923409, 25489342
21380789, 19154375, 19044962, 19532017, 19662635, 22374754, 20560611
25654936, 21492036, 18705806, 28420042, 19578247, 22024071, 22238921
22809871, 21184223, 19995869, 23089357, 19404068, 18921743, 19065677
19018447, 19018206, 18308268, 19777862, 22223463, 19304354, 29027694
22519146, 27199245, 19445860, 26654363, 22977256, 20890311, 27445727
21142837, 20869721, 24555417, 22179537, 21756699, 20217801, 18819908
22760595, 25483815, 23628685, 23007241, 19593445, 21080143, 27351628
20582405, 20031873, 25489367, 18618122, 24737581, 26784509, 24739928
18966843, 19077215, 20704450, 19068970, 20543011, 19023822, 24713381
20432873, 21756677, 20328248, 18674047, 18849537, 20087383, 25459958
```

20315311, 22897344, 27534509, 25178179, 19308965, 18948177, 19468991
20868862, 21780146, 23315153, 20466628, 21756661, 20397490, 19706965
20302006, 24831514, 23240358, 22178855, 19032777, 20862087, 19329654
18974476, 20603378, 20859910, 19307662, 21847223, 20281121, 22568797
19075256, 19076343, 18866977, 22808310, 25635149, 20844426, 20904530
20441797, 21442094, 25079710, 24674955, 18840932, 18740837, 20294666
25602488, 21517440, 22062517, 27337759, 19174942, 19180394, 20671094
21889720, 18411216, 20117253, 24386767, 20641666, 25264559, 22092979
25616268, 21625179, 20879709, 23003979, 20165574, 28578164, 19272708
19547370, 22624709, 23084507, 23184263, 20228093, 21281532, 25093872
19805359, 19461270, 26324206, 19434529, 18799063, 20378086, 17008068
21246723, 20831538, 20424899, 20361671, 18674024, 19689979, 24411921
19873610, 16619249, 20562898, 21641414, 21091431, 19440586, 22757364
20001168, 22175564, 20725343, 21241052, 19561643, 20736227, 19399918
19195895, 20830459, 20017509, 18475439, 25790353, 21828126, 21665897
25555252, 20746251, 22568177, 25764020, 25612095, 25357142, 23096938
19067244, 18043064, 21329301, 18885870, 26243698, 26187943, 20324049
19536415, 23709062, 28174827, 20446883, 27314206, 21299490, 25313154
18628388, 21744290, 18254023, 25047724, 20591183, 27847259, 19185876
27207110, 22465352, 24385625, 24326444, 20402832, 19627012, 20468401
27441326, 27620950, 16863642, 19639483, 19315691, 27567477, 21479753
19174521, 20401975, 18306996, 18851894, 27034890, 20581111, 20318889
20936731, 21060755, 25240188, 26828994, 22256560, 19188927, 27229389
20766180, 20229001, 24570598, 25475853, 21172913, 17655240, 21266085
19028800, 19035573, 19366375, 24523374, 25599425, 25034396, 19289642
21502702, 21291274, 18007682, 23521523, 20475845, 22148226, 22528741
29408136, 25417958, 24652769, 26088426, 19326908, 19597583, 17414008
23019710, 20897759, 26822620, 22046677, 19663176, 20938170, 19891090
24825843, 26318627, 21960504, 24509056, 19054077, 26262953, 22657942
20428621, 21899588, 23326313, 19723336, 19835133, 17532734, 19333670
25300427, 21842017, 19285025, 21373473, 23260854, 23061453, 19687159
14643995, 20977794, 20734332, 17551063, 27548131, 21977392, 24461826
19676012, 20588502, 23315889, 19520602, 23053606, 19841800, 20245930
19001359, 21476308, 26546754, 19393542, 23533524, 21099555, 27995248
25429959, 19141838, 19644859, 21915719, 19908836, 21421886, 19358317
19524158, 23548817, 25861398, 20803014, 23025340, 19335438, 19058490
23642282, 19207117, 18799993, 25919622, 26569225, 20835241, 25986062
24662775, 19475971, 18967382, 20347562, 20348653, 19896336, 24812585
20048359, 21896069, 20468490, 19524384, 25392535, 21147908, 20440930
25789277, 19171086, 24718260, 17867700, 19791273, 27397048, 21241829
19591608, 22707244, 18419520, 22296366, 22654475, 18914624, 19571367
28636676, 21522582, 19501299, 19529868, 20425790, 26007010, 19708342
27997875, 16870214, 18202441, 24415926, 18743542, 19001390, 21157728
19332396, 21875360, 25091141, 2800269, 19149990, 20382309, 22855193
16777441, 19606174, 20848335, 25495682, 19382851, 20528052, 22762046
24563422, 23125826, 22503297, 28993590, 25192729, 23338911, 22730454
19176326, 20298413, 19048007, 18849970, 21532755, 20860659, 22905130
26121990, 21263635, 22160989, 18499088, 22894949, 21059919, 18952989
22518784, 27348081, 25856821, 24457597, 25484507, 20794034, 21061354
19468347, 17533661, 19883092, 20657441, 24401351, 21285458, 18051556
25330273, 28023399, 26412540, 19699191, 24437510, 20669434, 16875041
18964978, 22972770, 20828947, 21373076, 25492379, 25551676, 14283239
25766822, 21967197, 22922076, 25575628, 26110259, 20368850, 21239530
20437153, 24848928, 20880215, 20798891, 25606091, 19013183, 21133343
22695831, 24365589, 25634317, 19587324, 20134113, 20273319, 18542562
26758193, 19017309, 21063322, 22062026, 20134339, 22077517, 22815955
23854396, 24690216, 22507210, 16354467, 20101006, 21795111, 27938623
23501901, 18797519, 25879984, 21260397, 25029423, 19354335, 19730508
22366558, 26658759, 25822410, 6599380, 20717359, 24321547, 27097854
21297872, 18964939, 26366517, 21913183, 22366322, 20171986, 20603431
21132297, 25957038, 21542577, 22507234, 23170620, 24719736, 25600342
18868646, 26637824, 20627866, 28587723, 29142109, 18110491, 16923858
24642295, 19518079, 20466322, 19339555, 25823754, 25110233, 24908321
20842388, 17274537, 26575788, 20474192, 21644640, 28849751, 21794615
18899974, 20471920, 22806698, 19052488, 19503821, 24350620, 20074391
19157754, 21220620, 24316947, 19865345, 19065556, 22816287, 25947799

20878790, 23492665, 21322887, 22305887, 20879889, 24350831, 19578350
28022101, 26439748, 21893235, 19363645, 21072646, 20898391, 19291380
27060167, 27086138, 18382302, 22536802, 22087683, 21197626, 21656630
20373598, 19248799, 22707866, 19155797, 19279273, 28432129, 18886413
25490238, 20922010, 19990037, 25150925, 20509482, 27255377, 24717859
20703000, 22862134, 21526048, 28683167, 24929210, 24560906, 20144308
21620471, 19670108, 19068610, 20267166, 25123585, 20476175, 18549238
22950945, 19385656, 23528412, 19684504, 21174504, 20899461, 20557786
21911701, 19143550, 20118035, 19024808, 25760195, 20009833, 19604659
16359751, 26039623, 19928926, 23314180, 20212067, 24737403, 20480209
18904062, 26430737, 29030780, 20856766, 27169796, 21668627, 17258582
20877664, 19487147, 23149541, 24575766, 19430401, 19676905, 20925795
21296029, 21629064, 23229229, 22865673, 20708701, 25353983, 19280225
21315084, 20613079, 19375649, 19213447, 19989009, 18191823, 27314390
26336977, 25775213, 24393981, 25639019, 17319928, 14705949, 19703301
21626377, 20122715, 6418158, 23105538, 26198926, 19258504, 21188532
23151677, 17890099, 21649497, 26446098, 16887946, 26024732, 18791688
19721304, 27012701, 19490948, 19619732, 21164318, 23148260, 18090142
21641760, 19818513, 23002524, 20139391, 24693382, 19978542, 23543183
22165897, 22359063, 19373893, 19409212, 18373438, 23035249, 18990693
20470877, 21422580, 21632821, 22351572, 20235511, 23220453, 18742258
18604493, 23008056, 22901797, 18610915, 20978259, 20832516, 24801152
26089440, 20907061, 19523462, 20505778, 19183343, 21787056, 21273804
22782647, 25093739, 17835294, 28708023, 24413809, 18371441, 26714910
24385983, 20413820, 24421668, 25897615, 25643931, 23195445, 21281607
20513399, 20558005, 20093776, 18909599, 20618595, 23572982, 19211433
20331945, 19512341, 22256431, 19637186, 19022470, 18607546, 26875822
24573817, 23115139, 19649152, 19201867, 21294938, 20898997, 18510194
21842740, 22454326, 24683149, 19534363, 25489607

Version 12.1.0.2.v15

Version 12.1.0.2.v15 includes the following:

- Patch 28729169: Oracle Database Patch Set Update 12.1.0.2.190115
- Patch 28790654: Oracle JVM Patch Set Update 12.1.0.2.190115
- Patch 28125601: DSTv32 for RDBMS (TZDATA2018E)
- Patch 28127287: DSTv32 for OJVM (TZDATA2018E)
- Patch 17969866: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 20394750: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 21171382: DBMS_STATS Patch
- Patch 29125200: JSON bundle patch
- Patch 20033733: KGL heap size patch

Oracle patch 28729169, released January 2019

Bugs fixed:

19309466, 19902195, 18250893, 25437699, 19383839, 16756406, 18456643
26546664, 22364044, 18845653, 19915271, 20172151, 18417036, 23713236
24796092, 23140259, 19243521, 19658708, 18272672, 21153266, 19174430
22243719, 20493163, 20688221, 21387964, 13542050, 22734547, 21623164
19012119, 19932634, 19869255, 22232606, 18681056, 23324000, 25427662
22068305, 24589081, 19439759, 19303936, 22916353, 24835538, 22353346
21106027, 26444887, 23088803, 22529728, 26256131, 19134173, 20447445
21188584, 19390567, 26513709, 25780343, 19769480, 21097043, 21225209
26245237, 20677396, 19284031, 19450314, 19016730, 20919320, 22075064

22551446, 22721409, 18440095, 22496904, 16439813, 18354830, 20596234
22022760, 20936905, 23197103, 21514877, 26111842, 18990023, 13787015
22492533, 20173897, 24624166, 17210525, 21260431, 20181030, 25056052
19370504, 21868720, 23068169, 19124589, 19402853, 21383171, 19888853
24341675, 17722075, 20882568, 25653109, 23026585, 18604692, 20717081
25546608, 27370965, 19081128, 22173980, 23514710, 19178851, 20951038
22168163, 25161298, 20569094, 24308635, 19791377, 19050649, 20920911
19189525, 22475617, 19469538, 27052607, 20598042, 22458049, 18988834
23302839, 25307368, 17409174, 22729345, 22842151, 19238590, 16941434
20387265, 24397438, 20673810, 23108128, 20356733, 22380919, 18436647
23065323, 20825533, 19124336, 22294260, 24790914, 20284155, 23571055
25539063, 17365043, 25914276, 20952966, 22961508, 19176223, 21300341
23237313, 18288842, 22353199, 27223075, 22083366, 25670786, 21419850
26898563, 19577410, 26248143, 23294548, 24737064, 19931709, 25423453
25547060, 23533807, 27726780, 24600330, 25600421, 18122373, 20043616
23124895, 18856999, 21450666, 18893947, 26633558, 20076781, 26029780
21196809, 21354456, 23725036, 20464614, 19562381, 27375542, 24808595
19189317, 25669791, 18307021, 21917884, 19708632, 27213224, 25633101
20711718, 18973548, 25982666, 19718981, 23567857, 22826718, 25655390
21773465, 20250147, 20144019, 19197175, 26263721, 19597439, 21387128
22007324, 19180770, 19879746, 21785691, 20424183, 24285405, 26544823
20322560, 22228324, 23172924, 22520320, 21575362, 25058080, 22365117
22645009, 25165496, 28950969, 18774543, 20124446, 21429602, 26153977
19371175, 21863727, 18940497, 19074147, 22923409, 25489342, 21380789
19154375, 19044962, 19532017, 19662635, 22374754, 20560611, 25654936
21492036, 18705806, 19578247, 22024071, 22238921, 22809871, 21184223
23089357, 19404068, 18921743, 19065677, 19018447, 19018206, 18308268
19777862, 22223463, 19304354, 22519146, 27199245, 22977256, 20890311
27445727, 21142837, 20869721, 24555417, 22179537, 21756699, 20217801
18819908, 22760595, 25483815, 23628685, 23007241, 19593445, 21080143
27351628, 20031873, 25489367, 18618122, 24737581, 26784509, 24739928
18966843, 19077215, 20704450, 19068970, 20543011, 19023822, 24713381
20432873, 21756677, 20328248, 18674047, 18849537, 20087383, 25459958
20315311, 22897344, 27534509, 25178179, 19308965, 18948177, 19468991
20868862, 21780146, 23315153, 20466628, 21756661, 20397490, 19706965
20302006, 24831514, 23240358, 22178855, 19032777, 20862087, 19329654
18974476, 20603378, 20859910, 19307662, 21847223, 20281121, 19075256
22568797, 19076343, 18866977, 22808310, 25635149, 20844426, 20904530
20441797, 21442094, 25079710, 24674955, 18840932, 18740837, 20294666
25602488, 21517440, 22062517, 27337759, 19174942, 20671094, 21889720
18411216, 20117253, 24386767, 20641666, 25264559, 22092979, 25616268
21625179, 20879709, 23003979, 20165574, 19272708, 19547370, 22624709
23084507, 20228093, 21281532, 25093872, 19805359, 19461270, 19434529
18799063, 20378086, 17008068, 21246723, 20831538, 20424899, 20361671
18674024, 19689979, 24411921, 19873610, 16619249, 20562898, 21641414
21091431, 19440586, 22757364, 22175564, 20725343, 21241052, 19561643
20736227, 19399918, 19195895, 20830459, 20017509, 18475439, 25790353
21828126, 21665897, 25555252, 20746251, 25764020, 25612095, 22568177
25357142, 23096938, 19067244, 18043064, 21329301, 18885870, 26243698
26187943, 20324049, 19536415, 23709062, 28174827, 20446883, 27314206
21299490, 25313154, 21744290, 18254023, 20591183, 25047724, 27847259
19185876, 27207110, 22465352, 24326444, 20402832, 19627012, 20468401
27441326, 27620950, 16863642, 19639483, 19315691, 27567477, 21479753
19174521, 20401975, 18306996, 18851894, 27034890, 20581111, 20318889
20936731, 21060755, 25240188, 26828994, 22256560, 19188927, 27229389
24570598, 20229001, 25475853, 21172913, 17655240, 21266085, 19028800
19035573, 19366375, 24523374, 25034396, 25599425, 19289642, 21291274
18007682, 23521523, 20475845, 22148226, 22528741, 25417958, 24652769
26088426, 19326908, 19597583, 17414008, 23019710, 20897759, 26822620
22046677, 20938170, 19891090, 24825843, 26318627, 21960504, 24509056
19054077, 26262953, 22657942, 20428621, 21899588, 23326313, 19723336
19835133, 17532734, 19333670, 21842017, 19285025, 21373473, 23260854
23061453, 19687159, 14643995, 20977794, 20734332, 17551063, 27548131
21977392, 24461826, 19676012, 20588502, 23315889, 19520602, 23053606
19841800, 20245930, 19001359, 21476308, 26546754, 19393542, 23533524
21099555, 27995248, 25429959, 19141838, 19644859, 21915719, 19908836

21421886, 19358317, 19524158, 23548817, 25861398, 20803014, 23025340
19335438, 19058490, 23642282, 19207117, 18799993, 25919622, 26569225
20835241, 24662775, 19475971, 18967382, 20347562, 20348653, 19896336
24812585, 20048359, 21896069, 19524384, 20468490, 25392535, 21147908
20440930, 25789277, 19171086, 24718260, 17867700, 19791273, 21241829
27397048, 19591608, 22707244, 18419520, 22296366, 22654475, 18914624
19571367, 28636676, 21522582, 19501299, 20425790, 19529868, 19708342
27997875, 16870214, 18202441, 24415926, 18743542, 19001390, 21875360
25091141, 28000269, 19149990, 20382309, 22855193, 16777441, 19606174
20848335, 25495682, 19382851, 20528052, 22762046, 24563422, 23125826
22503297, 28993590, 25192729, 23338911, 22730454, 19176326, 19048007
18849970, 21532755, 20860659, 22905130, 26121990, 21263635, 22160989
18499088, 22894949, 21059919, 18952989, 22518784, 25856821, 25484507
20794034, 19468347, 17533661, 19883092, 20657441, 24401351, 21285458
18051556, 25330273, 26412540, 19699191, 24437510, 20669434, 18964978
22972770, 20828947, 21373076, 25492379, 25551676, 14283239, 25766822
21967197, 22922076, 25575628, 26110259, 20368850, 21239530, 20437153
24848928, 20880215, 20798891, 25606091, 19013183, 21133343, 22695831
24365589, 25634317, 19587324, 20273319, 18542562, 26758193, 21063322
22062026, 20134339, 22077517, 22815955, 24690216, 22507210, 16354467
20101006, 21795111, 27938623, 23501901, 18797519, 25879984, 21260397
25029423, 19354335, 19730508, 22366558, 26658759, 25822410, 6599380
20717359, 24321547, 21297872, 27097854, 18964939, 26366517, 21913183
22366322, 20171986, 20603431, 21132297, 25957038, 21542577, 22507234
23170620, 24719736, 25600342, 18868646, 26637824, 20627866, 18110491
16923858, 24642295, 19518079, 20466322, 25823754, 25110233, 24908321
20842388, 17274537, 26575788, 20474192, 21644640, 21794615, 18899974
20471920, 22806698, 19052488, 19503821, 24350620, 20074391, 19157754
21220620, 24316947, 19865345, 19065556, 22816287, 25947799, 20878790
23492665, 21322887, 22305887, 20879889, 24350831, 19578350, 28022101
19363645, 21072646, 20898391, 19291380, 27060167, 27086138, 22536802
22087683, 21656630, 20373598, 19248799, 22707866, 19155797, 19279273
18886413, 25490238, 20922010, 19990037, 25150925, 20509482, 24717859
20703000, 22862134, 21526048, 24929210, 24560906, 28683167, 20144308
21620471, 19670108, 19068610, 20267166, 25123585, 20476175, 18549238
22950945, 19385656, 23528412, 19684504, 21174504, 20899461, 20557786
21911701, 19143550, 20118035, 19024808, 25760195, 20009833, 19604659
16359751, 26039623, 19928926, 23314180, 20212067, 24737403, 20480209
26430737, 20856766, 27169796, 21668627, 20877664, 19487147, 23149541
24577566, 19430401, 19676905, 20925795, 21296029, 21629064, 23229229
22865673, 20708701, 25353983, 19280225, 21315084, 20613079, 19375649
19213447, 19989009, 18191823, 27314390, 25775213, 26336977, 24393981
25639019, 17319928, 19703301, 21626377, 20122715, 6418158, 23105538
26198926, 19258504, 21188532, 23151677, 17890099, 21649497, 26446098
16887946, 26024732, 18791688, 19721304, 27012701, 19490948, 19619732
21164318, 23148260, 18090142, 21641760, 19818513, 20139391, 24693382
19978542, 23543183, 22165897, 22359063, 19409212, 23035249, 18990693
20470877, 21422580, 21632821, 22351572, 20235511, 23220453, 18604493
18742258, 23008056, 22901797, 18610915, 20978259, 20832516, 24801152
26089440, 20907061, 20505778, 19183343, 21787056, 21273804, 25093739
17835294, 24413809, 28708023, 18371441, 26714910, 24385983, 20413820
24421668, 25897615, 25643931, 23195445, 21281607, 20513399, 20558005
20093776, 18909599, 20618595, 23572982, 19211433, 20331945, 19512341
22256431, 19637186, 19022470, 18607546, 24573817, 23115139, 19649152
19201867, 21294938, 20898997, 18510194, 21842740, 22454326, 24683149
19534363, 25489607

Version 12.1.0.2.v14

Version 12.1.0.2.v14 includes the following:

- Patch 28259833: Oracle Database Patch Set Update 12.1.0.2.181016
- Patch 28440711: Oracle JVM Patch Set Update 12.1.0.2.181016

- Patch 28125601: DSTv32 for RDBMS (TZDATA2018E)
- Patch 28127287: DSTv32 for OJVM (TZDATA2018E)
- Patch 17969866: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 20394750: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 21171382: DBMS_STATS Patch
- Patch 28697469: JSON Database Patch
- Patch 20033733: KGL heap size patch

Oracle patch 28259833, released October 2018

Bugs fixed:

19309466, 19902195, 18250893, 25437699, 19383839, 16756406, 18456643 26546664, 22364044, 18845653, 19915271, 20172151, 18417036, 23713236 24796092, 23140259, 19243521, 19658708, 18272672, 21153266, 19174430 22243719, 20688221, 20493163, 21387964, 13542050, 22734547, 21623164 19012119, 19932634, 19869255, 22232606, 18681056, 23324000, 25427662 22068305, 24589081, 19439759, 19303936, 22916353, 24835538, 22353346 21106027, 26444887, 23088803, 22529728, 26256131, 19134173, 20447445 21188584, 19390567, 26513709, 25780343, 19769480, 21097043, 21225209 26245237, 20677396, 19284031, 19450314, 19016730, 20919320, 22075064 22551446, 22721409, 18440095, 22496904, 16439813, 18354830, 20596234 22022760, 20936905, 23197103, 21514877, 26111842, 18990023, 22492533 20173897, 24624166, 17210525, 21260431, 20181030, 25056052, 19370504 21868720, 23068169, 19124589, 19402853, 19888853, 24341675, 17722075 20882568, 25653109, 23026585, 18604692, 20717081, 25546608, 27370965 19081128, 22173980, 23514710, 19178851, 20951038, 22168163, 25161298 20569094, 24308635, 19791377, 19050649, 20920911, 19189525, 19469538 27052607, 20598042, 22458049, 18988834, 23302839, 25307368, 17409174 22729345, 22842151, 19238590, 16941434, 20387265, 24397438, 20673810 23108128, 20356733, 22380919, 18436647, 23065323, 20825533, 19124336 22294260, 24790914, 20284155, 25539063, 17365043, 25914276, 20952966 22961508, 19176223, 21300341, 23237313, 18288842, 22353199, 22083366 25670786, 21419850, 26898563, 19577410, 23294548, 24737064, 19931709 25423453, 25547060, 23533807, 27726780, 24600330, 25600421, 18122373 20043616, 23124895, 18856999, 21450666, 18893947, 20076781, 26633558 26029780, 21196809, 21354456, 23725036, 20464614, 19562381, 24808595 27375542, 19189317, 25669791, 18307021, 21917884, 19708632, 27213224 25633101, 20711718, 18973548, 25982666, 19718981, 22826718, 25655390 23567857, 21773465, 20250147, 19197175, 26263721, 19597439, 21387128 22007324, 19180770, 19879746, 21785691, 20424183, 24285405, 26544823 20322560, 22228324, 23172924, 22520320, 21575362, 25058080, 22365117 22645009, 25165496, 18774543, 20124446, 21429602, 26153977, 19371175 21863727, 18940497, 19074147, 22923409, 25489342, 21380789, 19154375 19044962, 19532017, 19662635, 22374754, 20560611, 25654936, 21492036 18705806, 19578247, 22024071, 22238921, 22809871, 21184223, 23089357 19404068, 18921743, 19065677, 19018447, 19018206, 18308268, 19777862 22223463, 19304354, 22519146, 2719245, 20890311, 22977256, 21142837 20869721, 24555417, 22179537, 21756699, 20217801, 18819908, 22760595 25483815, 23007241, 19593445, 21080143, 27351628, 20031873, 18618122 24737581, 26784509, 24739928, 18966843, 19077215, 20704450, 19068970 20543011, 19023822, 24713381, 20432873, 21756677, 20328248, 18674047 18849537, 25459958, 20315311, 22897344, 27534509, 25178179, 19308965 18948177, 19468991, 20868862, 21780146, 20466628, 21756661, 20397490 23315153, 19706965, 20302006, 24831514, 23240358, 22178855, 19032777 20862087, 19329654, 18974476, 20603378, 20859910, 19307662, 21847223 20281121, 19075256, 19076343, 18866977, 22808310, 25635149, 20844426 20904530, 20441797, 21442094, 25079710, 24674955, 18840932, 18740837 20294666, 25602488, 21517440, 22062517, 27337759, 19174942, 20671094 21889720, 18411216, 20117253, 24386767, 20641666, 25264559, 22092979

21625179, 20879709, 23003979, 20165574, 19272708, 19547370, 22624709
23084507, 20228093, 21281532, 19805359, 19461270, 19434529, 18799063
20378086, 17008068, 21246723, 20831538, 20424899, 20361671, 18674024
19689979, 24411921, 19873610, 16619249, 20562898, 21641414, 21091431
19440586, 22757364, 22175564, 21241052, 20725343, 19561643, 20736227
19399918, 19195895, 20830459, 20017509, 25790353, 21828126, 21665897
25555252, 20746251, 25764020, 25612095, 25357142, 23096938, 19067244
18043064, 21329301, 18885870, 26243698, 26187943, 20324049, 19536415
23709062, 28174827, 20446883, 27314206, 21299490, 25313154, 21744290
18254023, 20591183, 27847259, 19185876, 27207110, 22465352, 24326444
20402832, 19627012, 27441326, 27620950, 16863642, 19639483, 19315691
21479753, 19174521, 20401975, 18306996, 18851894, 27034890, 20581111
20318889, 20936731, 21060755, 26828994, 22256560, 19188927, 27229389
24570598, 25475853, 21172913, 17655240, 21266085, 19028800, 19035573
19366375, 24523374, 25034396, 19289642, 21291274, 18007682, 23521523
20475845, 22148226, 22528741, 25417958, 24652769, 26088426, 19326908
19597583, 17414008, 23019710, 20897759, 26822620, 22046677, 20938170
24825843, 19891090, 21960504, 26318627, 24509056, 19054077, 26262953
22657942, 20428621, 21899588, 19723336, 19835133, 17532734, 19333670
21842017, 19285025, 21373473, 23260854, 19687159, 23061453, 14643995
20977794, 20734332, 17551063, 27548131, 21977392, 24461826, 19676012
20588502, 23315889, 19520602, 23053606, 19841800, 20245930, 19001359
21476308, 26546754, 19393542, 23533524, 21099555, 25429959, 19141838
19644859, 21915719, 19908836, 21421886, 19358317, 19524158, 23548817
25861398, 20803014, 23025340, 19335438, 19058490, 19207117, 23642282
18799993, 25919622, 26569225, 20835241, 24662775, 19475971, 18967382
20347562, 20348653, 19896336, 24812585, 20048359, 21896069, 19524384
25392535, 21147908, 20440930, 25789277, 19171086, 24718260, 17867700
19791273, 21241829, 19591608, 22707244, 18419520, 22296366, 18914624
19571367, 22654475, 21522582, 19501299, 20425790, 19708342, 27997875
16870214, 18202441, 24415926, 18743542, 19001390, 21875360, 25091141
28000269, 19149990, 20382309, 22855193, 16777441, 19606174, 20848335
25495682, 19382851, 20528052, 22762046, 24563422, 23125826, 22503297
25192729, 23338911, 22730454, 19176326, 19048007, 18849970, 21532755
20860659, 22905130, 21263635, 22160989, 18499088, 21059919, 18952989
22894949, 22518784, 25856821, 25484507, 20794034, 19468347, 17533661
19883092, 20657441, 24401351, 21285458, 18051556, 25330273, 19699191
24437510, 20669434, 18964978, 22972770, 20828947, 21373076, 25551676
25492379, 14283239, 25766822, 22922076, 25575628, 20368850, 21239530
20437153, 24848928, 20880215, 20798891, 25606091, 19013183, 21133343
22695831, 24365589, 25634317, 19587324, 20273319, 18542562, 26758193
21063322, 22062026, 20134339, 22077517, 22815955, 24690216, 22507210
20101006, 16354467, 21795111, 27938623, 23501901, 18797519, 25879984
21260397, 25029423, 19354335, 19730508, 22366558, 26658759, 6599380
20717359, 24321547, 21297872, 18964939, 26366517, 21913183, 22366322
20171986, 20603431, 21132297, 25957038, 21542577, 22507234, 23170620
24719736, 25600342, 18868646, 20627866, 26637824, 18110491, 16923858
24642295, 19518079, 20466322, 25823754, 25110233, 24908321, 20842388
17274537, 26575788, 20474192, 21644640, 21794615, 18899974, 20471920
22806698, 19052488, 19503821, 24350620, 20074391, 19157754, 21220620
24316947, 19865345, 19065556, 22816287, 25947799, 20878790, 23492665
21322887, 22305887, 20879889, 24350831, 19578350, 19363645, 21072646
20898391, 19291380, 27060167, 27086138, 22536802, 22087683, 21656630
20373598, 19248799, 22707866, 19155797, 19279273, 18886413, 25490238
20922010, 19990037, 25150925, 20509482, 24717859, 20703000, 22862134
21526048, 24929210, 24560906, 20144308, 21620471, 19670108, 19068610
20267166, 25123585, 20476175, 18549238, 22950945, 19385656, 23528412
19684504, 21174504, 20899461, 20557786, 21911701, 19143550, 20118035
19024808, 25760195, 20009833, 19604659, 16359751, 26039623, 19928926
23314180, 20212067, 24737403, 20480209, 26430737, 20856766, 27169796
21668627, 20877664, 19487147, 23149541, 24577566, 19430401, 19676905
20925795, 21296029, 21629064, 23229229, 22865673, 20708701, 25353983
19280225, 21315084, 19213447, 19989009, 18191823, 27314390, 25775213
24393981, 25639019, 17319928, 19703301, 21626377, 20122715, 6418158
23105538, 26198926, 19258504, 21188532, 23151677, 17890099, 21649497
26446098, 16887946, 26024732, 18791688, 19721304, 19490948, 27012701

19619732, 21164318, 18090142, 21641760, 19818513, 20139391, 24693382
19978542, 23543183, 22165897, 22359063, 19409212, 23035249, 18990693
20470877, 21422580, 21632821, 22351572, 20235511, 23220453, 18604493
23008056, 22901797, 18610915, 20832516, 24801152, 26089440, 20907061
20505778, 19183343, 21787056, 21273804, 25093739, 17835294, 24413809
18371441, 26714910, 24385983, 20413820, 24421668, 25897615, 25643931
23195445, 21281607, 20513399, 20558005, 20093776, 18909599, 20618595
23572982, 19211433, 20331945, 19512341, 22256431, 19637186, 19022470
18607546, 24573817, 19649152, 23115139, 19201867, 21294938, 20898997
18510194, 21842740, 22454326, 24683149, 19534363, 25489607

Version 12.1.0.2.v13

Version 12.1.0.2.v13 includes the following:

- Patch 27547329: Oracle Database Patch Set Update 12.1.0.2.180717
- Patch 27923320: Oracle JVM Patch Set Update 12.1.0.2.180717
- Patch 28125601: DSTv32 for RDBMS (TZDATA2018E)
- Patch 28127287: DSTv32 for OJVM (TZDATA2018E)
- Patch 17969866: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 20394750: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 21171382: DBMS_STATS Patch
- Patch 28307069: JSON Database Patch
- Patch 20033733: KGL heap size patch

Oracle patch 27547329, released July 2018

Bugs fixed:

19309466, 19902195, 18250893, 25437699, 19383839, 16756406, 18456643
26546664, 18845653, 19915271, 20172151, 18417036, 23713236, 24796092
19243521, 19658708, 21153266, 19174430, 22243719, 20688221, 21387964
13542050, 22734547, 21623164, 19012119, 19932634, 19869255, 22232606
18681056, 23324000, 25427662, 22068305, 24589081, 19439759, 19303936
22916353, 24835538, 22353346, 21106027, 26444887, 23088803, 22529728
26256131, 19134173, 20447445, 21188584, 19390567, 26513709, 19769480
21097043, 21225209, 20677396, 19284031, 26245237, 19450314, 19016730
20919320, 22075064, 22551446, 22721409, 18440095, 22496904, 16439813
18354830, 20596234, 22022760, 20936905, 23197103, 21514877, 26111842
18990023, 22492533, 20173897, 24624166, 17210525, 21260431, 20181030
25056052, 19370504, 21868720, 23068169, 19124589, 19402853, 19888853
24341675, 17722075, 20882568, 23026585, 25653109, 20717081, 25546608
19081128, 27370965, 22173980, 19178851, 20951038, 22168163, 25161298
20569094, 24308635, 19791377, 19050649, 20920911, 19189525, 19469538
20598042, 22458049, 18988834, 17409174, 22729345, 22842151, 19238590
16941434, 20387265, 24397438, 20673810, 23108128, 20356733, 22380919
18436647, 23065323, 20825533, 19124336, 22294260, 24790914, 20284155
25539063, 17365043, 20952966, 22961508, 19176223, 21300341, 23237313
18288842, 22353199, 22083366, 21419850, 26898563, 19577410, 23294548
19931709, 25423453, 25547060, 23533807, 24600330, 25600421, 18122373
20043616, 23124895, 18856999, 21450666, 18893947, 20076781, 26029780
21196809, 21354456, 20464614, 23725036, 19562381, 24808595, 19189317
18307021, 25669791, 21917884, 19708632, 27213224, 25633101, 20711718
18973548, 25982666, 22826718, 25655390, 21773465, 20250147, 19197175
19597439, 26263721, 21387128, 19180770, 19879746, 21785691, 20424183
24285405, 26544823, 20322560, 22228324, 22520320, 23172924, 21575362

22365117, 22645009, 25165496, 18774543, 20124446, 21429602, 19371175
21863727, 18940497, 19074147, 22923409, 21380789, 19154375, 19044962
19532017, 19662635, 22374754, 20560611, 25654936, 21492036, 18705806
19578247, 22024071, 22238921, 22809871, 21184223, 23089357, 19404068
18921743, 19065677, 19018447, 19018206, 18308268, 19777862, 22223463
19304354, 22519146, 27199245, 20890311, 21142837, 20869721, 24555417
22179537, 21756699, 20217801, 18819908, 22760595, 25483815, 23007241
19593445, 21080143, 20031873, 18618122, 26784509, 24739928, 18966843
19077215, 20704450, 19068970, 20543011, 19023822, 24713381, 20432873
21756677, 20328248, 18674047, 18849537, 25459958, 20315311, 22897344
27534509, 25178179, 19308965, 18948177, 19468991, 20868862, 21780146
20466628, 21756661, 20397490, 19706965, 24831514, 23240358, 22178855
20302006, 19032777, 20862087, 19329654, 18974476, 20603378, 20859910
19307662, 21847223, 20281121, 19075256, 19076343, 18866977, 20844426
20904530, 20441797, 21442094, 25079710, 24674955, 18840932, 18740837
20294666, 25602488, 21517440, 22062517, 27337759, 19174942, 20671094
21889720, 18411216, 20117253, 24386767, 20641666, 25264559, 22092979
21625179, 20879709, 23003979, 20165574, 19272708, 19547370, 22624709
23084507, 20228093, 21281532, 19805359, 19461270, 19434529, 18799063
20378086, 17008068, 21246723, 20831538, 20424899, 20361671, 18674024
19689979, 24411921, 19873610, 16619249, 20562898, 21091431, 21641414
19440586, 22757364, 22175564, 21241052, 19561643, 19399918, 19195895
20830459, 20017509, 25790353, 21828126, 21665897, 20746251, 25764020
25612095, 25357142, 23096938, 19067244, 18043064, 21329301, 18885870
26187943, 20324049, 19536415, 20446883, 21299490, 27314206, 25313154
21744290, 18254023, 20591183, 27847259, 19185876, 22465352, 27207110
20402832, 19627012, 27441326, 27620950, 16863642, 19639483, 19315691
21479753, 19174521, 20401975, 18306996, 18851894, 27034890, 20581111
20318889, 20936731, 21060755, 22256560, 19188927, 24570598, 25475853
21172913, 17655240, 21266085, 19028800, 19035573, 19366375, 24523374
25034396, 19289642, 21291274, 18007682, 23521523, 20475845, 22148226
22528741, 25417958, 24652769, 26088426, 19326908, 19597583, 17414008
23019710, 20897759, 22046677, 20938170, 24825843, 21960504, 24509056
19054077, 22657942, 26262953, 20428621, 21899588, 19723336, 19835133
17532734, 19333670, 21842017, 19285025, 21373473, 23260854, 19687159
14643995, 20977794, 20734332, 17551063, 27548131, 21977392, 24461826
19676012, 20588502, 23315889, 19520602, 23053606, 19841800, 20245930
19001359, 21476308, 26546754, 19393542, 23533524, 21099555, 25429959
19141838, 19644859, 21915719, 19908836, 21421886, 19358317, 19524158
23548817, 25861398, 20803014, 23025340, 19335438, 19058490, 19207117
18799993, 26569225, 25919622, 20835241, 24662775, 19475971, 18967382
20347562, 20348653, 19896336, 24812585, 20048359, 21896069, 19524384
25392535, 20440930, 25789277, 19171086, 24718260, 17867700, 19791273
21241829, 19591608, 22707244, 18419520, 22296366, 18914624, 19571367
19501299, 20425790, 19708342, 27997875, 16870214, 18202441, 24415926
18743542, 19001390, 21875360, 25091141, 19149990, 20382309, 22855193
16777441, 19606174, 20848335, 25495682, 19382851, 20528052, 22762046
24563422, 23125826, 22503297, 25192729, 23338911, 22730454, 19176326
19048007, 18849970, 21532755, 20860659, 22905130, 21263635, 22160989
18499088, 21059919, 18952989, 22518784, 25856821, 25484507, 20794034
19468347, 17533661, 19883092, 20657441, 24401351, 21285458, 18051556
25330273, 19699191, 24437510, 20669434, 18964978, 20828947, 21373076
25551676, 14283239, 25766822, 22922076, 25575628, 20368850, 21239530
20437153, 20880215, 20798891, 25606091, 19013183, 21133343, 22695831
24365589, 19587324, 18542562, 26758193, 22062026, 20134339, 22077517
22815955, 24690216, 22507210, 20101006, 21795111, 27938623, 23501901
18797519, 21260397, 25029423, 19354335, 19730508, 22366558, 26658759
6599380, 20717359, 24321547, 21297872, 18964939, 26366517, 21913183
22366322, 20171986, 20603431, 21132297, 25957038, 21542577, 22507234
23170620, 24719736, 25600342, 18868646, 20627866, 18110491, 16923858
24642295, 19518079, 20466322, 25823754, 25110233, 24908321, 20842388
17274537, 26575788, 20474192, 21644640, 21794615, 18899974, 20471920
22806698, 19052488, 19503821, 24350620, 20074391, 19157754, 21220620
24316947, 19865345, 19065556, 22816287, 25947799, 20878790, 23492665
21322887, 20879889, 24350831, 19578350, 19363645, 21072646, 20898391
19291380, 27060167, 27086138, 22536802, 22087683, 20373598, 19248799

22707866, 19155797, 19279273, 18886413, 25490238, 20922010, 19990037
25150925, 20509482, 24717859, 20703000, 22862134, 21526048, 24929210
24560906, 20144308, 21620471, 19670108, 19068610, 20267166, 25123585
20476175, 18549238, 22950945, 19385656, 23528412, 19684504, 21174504
20899461, 20557786, 21911701, 19143550, 19024808, 20118035, 20009833
25760195, 19604659, 16359751, 26039623, 19928926, 23314180, 20212067
24737403, 20480209, 26430737, 27169796, 21668627, 20877664, 19487147
23149541, 24577566, 19430401, 19676905, 20925795, 21296029, 21629064
23229229, 22865673, 20708701, 19280225, 25353983, 21315084, 19213447
19989009, 18191823, 24393981, 25639019, 17319928, 19703301, 21626377
20122715, 6418158, 23105538, 26198926, 19258504, 21188532, 17890099
21644947, 26446098, 16887946, 26024732, 18791688, 19721304, 19490948
19619732, 21164318, 18090142, 21641760, 19818513, 20139391, 24693382
19978542, 23543183, 22165897, 22359063, 19409212, 23035249, 18990693
20470877, 21422580, 21632821, 22351572, 20235511, 23220453, 18604493
23008056, 18610915, 20832516, 24801152, 26089440, 20907061, 20505778
19183343, 21787056, 21273804, 25093739, 17835294, 24413809, 18371441
24385983, 20413820, 26714910, 24421668, 25897615, 25643931, 21281607
20513399, 23195445, 20558005, 20093776, 18909599, 20618595, 23572982
19211433, 20331945, 19512341, 22256431, 19637186, 19022470, 18607546
24573817, 19649152, 19201867, 21294938, 20898997, 18510194, 22454326
19534363, 24683149, 25489607

Version 12.1.0.2.v12

Version 12.1.0.2.v12 includes the following:

- Patch 27338041: DATABASE PATCH SET UPDATE 12.1.0.2.180417
- Patch 27475603: OJVM PATCH SET UPDATE 12.1.0.2.180417
- Patch 27015449: RDBMS - PROACTIVE DSTV31 UPDATE - TZDATA2017C
- Patch 27015468: PROACTIVE DSTV31 UPDATE - TZDATA2017C - NEED OJVM FIX
- Patch 17969866: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 20394750: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 21171382: AUTO DOP COMPUTES A HIGH DOP UNNECESSARILY
- Patch 27666699: JSON Database Patch
- Patch 20033733: PART :IMC:HIT ORA 600 [KGL-HEAP-SIZE-EXCEEDED]

Oracle patch 27338041, released April 2018

Bugs fixed:

19309466, 24570598, 25475853, 21172913, 19902195, 18250893, 17655240
25437699, 19383839, 21266085, 19028800, 19035573, 16756406, 19366375
18456643, 26546664, 24523374, 25034396, 19289642, 18845653, 19915271
21291274, 18007682, 20172151, 18417036, 23713236, 24796092, 23521523
20475845, 22148226, 22528741, 19243521, 19658708, 21153266, 24652769
26088426, 19326908, 19597583, 17414008, 20897759, 23019710, 19174430
22046677, 22243719, 20938170, 24825843, 21960504, 24509056, 19054077
22657942, 20688221, 20428621, 21899588, 21387964, 13542050, 19723336
19835133, 17532734, 19333670, 21842017, 19285025, 21373473, 22734547
23260854, 19687159, 14643995, 21623164, 20977794, 20734332, 19012119
19869255, 19932634, 17551063, 18681056, 22232606, 27548131, 21977392
23324000, 24461826, 19676012, 20588502, 25427662, 22068305, 23315889
19520602, 23053606, 19841800, 19439759, 20245930, 19303936, 19001359
21476308, 26546754, 22916353, 19393542, 23533524, 21099555, 24835538
22353346, 25429959, 19141838, 19644859, 21106027, 21915719, 26444887
23088803, 19908836, 21421886, 22529728, 26256131, 19358317, 19134173

19524158, 20447445, 23548817, 25861398, 20803014, 23025340, 21188584
19335438, 19390567, 19058490, 19207117, 26513709, 18799993, 26569225
20835241, 24662775, 19769480, 19475971, 21097043, 21225209, 20677396
19284031, 19450314, 19016730, 18967382, 20919320, 22075064, 20347562
20348653, 22551446, 19896336, 22721409, 24812585, 20048359, 21896069
18440095, 22496904, 19524384, 25392535, 16439813, 18354830, 20596234
20440930, 22022760, 20936905, 19171086, 23197103, 24718260, 17867700
19791273, 21514877, 26111842, 18990023, 21241829, 19591608, 22707244
18419520, 22492533, 22296366, 20173897, 24624166, 17210525, 18914624
19571367, 21260431, 19501299, 20181030, 25056052, 20425790, 19708342
19370504, 21868720, 23068169, 19124589, 19402853, 19888853, 16870214
24341675, 17722075, 18202441, 24415926, 18743542, 19001390, 20882568
23026585, 20717081, 25546608, 19081128, 22173980, 21875360, 25091141
19178851, 19149990, 20382309, 20951038, 22855193, 22168163, 16777441
25161298, 19606174, 20569094, 24308635, 20848335, 19791377, 19050649
19382851, 20920911, 20528052, 22762046, 19189525, 24563422, 23125826
22503297, 19469538, 25192729, 23338911, 20598042, 22458049, 18988834
22730454, 19176326, 19048007, 17409174, 22729345, 18849970, 21532755
20860659, 22842151, 22905130, 19238590, 16941434, 20387265, 21263635
24397438, 20673810, 23108128, 22160989, 20356733, 22380919, 18499088
18436647, 23065323, 21059919, 20825533, 18952989, 22518784, 19124336
25856821, 22294260, 25484507, 20794034, 19468347, 20284155, 17533661
19883092, 20657441, 24401351, 25539063, 17365043, 21285458, 20952966
22961508, 18051556, 25330273, 19176223, 21300341, 23237313, 18288842
19699191, 22353199, 24437510, 22083366, 21419850, 20669434, 18964978
26898563, 19577410, 23294548, 20828947, 21373076, 25551676, 14283239
25766822, 19931709, 22922076, 25423453, 25547060, 25575628, 23533807
20368850, 21239530, 20437153, 20880215, 25600421, 20798891, 25606091
18122373, 20043616, 23124895, 19013183, 18856999, 21450666, 21133343
22695831, 18893947, 24365589, 20076781, 21196809, 21354456, 19587324
20464614, 19562381, 18542562, 26758193, 24808595, 22062026, 19189317
18307021, 21917884, 19708632, 27213224, 25633101, 20711718, 20134339
22077517, 22815955, 24690216, 18973548, 25982666, 22507210, 22826718
25655390, 21773465, 20250147, 20101006, 21795111, 19197175, 23501901
18797519, 19597439, 21387128, 19180770, 19879746, 19354335, 21785691
19730508, 20424183, 22366558, 26658759, 24285405, 6599380, 20717359
26544823, 21297872, 20322560, 18964939, 22520320, 21575362, 26366517
21913183, 22366322, 20171986, 22365117, 22645009, 25165496, 20603431
21132297, 25957038, 21542577, 22507234, 18774543, 23170620, 24719736
25600342, 20627866, 20124446, 18110491, 21429602, 16923858, 24642295
19518079, 19371175, 20466322, 21863727, 18940497, 19074147, 22923409
25823754, 25110233, 24908321, 20842388, 17274537, 21380789, 26575788
19154375, 20474192, 19044962, 19532017, 21644640, 19662635, 22374754
20560611, 25654936, 21794615, 18899974, 21492036, 18705806, 20471920
22806698, 19052488, 22024071, 22238921, 19503821, 24350620, 22809871
20074391, 21184223, 23089357, 19157754, 21220620, 19404068, 24316947
18921743, 19865345, 19065677, 19065556, 22816287, 19018447, 19018206
19777862, 25947799, 22223463, 19304354, 20878790, 22519146, 23492665
21322887, 20879889, 24350831, 20890311, 19578350, 21142837, 20869721
24555417, 22179537, 21756699, 20217801, 18819908, 19363645, 25483815
21072646, 20898391, 19291380, 27060167, 27086138, 23007241, 19593445
21080143, 22536802, 22087683, 20373598, 19248799, 20031873, 22707866
19155797, 19279273, 18886413, 18618122, 25490238, 20922010, 19990037
25150925, 20509482, 24739928, 20703000, 18966843, 19077215, 22862134
21526048, 24929210, 24560906, 20704450, 20144308, 19068970, 20543011
21620471, 19023822, 19670108, 19068610, 20267166, 24713381, 20432873
21756677, 20476175, 25123585, 18549238, 20328248, 18674047, 22950945
19385656, 18849537, 23528412, 19684504, 25459958, 20315311, 22897344
20899461, 25178179, 20557786, 21911701, 19308965, 19143550, 19024808
18948177, 19468991, 20009833, 20868862, 21780146, 20466628, 21756661
20397490, 19706965, 24831514, 23240358, 22178855, 19604659, 16359751
19032777, 20862087, 19329654, 19928926, 18974476, 23314180, 20212067
20603378, 24737403, 20480209, 20859910, 26430737, 19307662, 21847223
21668627, 20281121, 27169796, 19075256, 20877664, 19487147, 19076343
23149541, 18866977, 24577566, 19430401, 19676905, 20844426, 20904530
20925795, 20441797, 21296029, 21629064, 21442094, 23229229, 25079710

22865673, 20708701, 19280225, 21315084, 24674955, 19213447, 18840932
18740837, 20294666, 19989009, 25602488, 18191823, 21517440, 22062517
19174942, 27337759, 17319928, 20671094, 21889720, 19703301, 21626377
20122715, 23105538, 18411216, 6418158, 26198926, 20117253, 19258504
21188532, 24386767, 17890099, 21649497, 26446098, 16887946, 26024732
25264559, 18791688, 19721304, 22092979, 19490948, 19619732, 21164318
21625179, 20879709, 23003979, 20165574, 18090142, 19272708, 21641760
19818513, 19547370, 22624709, 20139391, 23084507, 24693382, 20228093
21281532, 19978542, 23543183, 22165897, 22359063, 19409212, 19805359
19461270, 23035249, 19434529, 1879063, 18990693, 20470877, 20378086
17008068, 21246723, 21422580, 21632821, 20831538, 22351572, 20424899
20361671, 18674024, 19689979, 20235511, 23220453, 24411921, 19873610
16619249, 18604493, 20562898, 21091431, 19440586, 22757364, 18610915
22175564, 21241052, 19561643, 19399918, 19195895, 20832516, 20830459
20017509, 24801152, 21828126, 20907061, 21665897, 20746251, 20505778
19183343, 25764020, 25612095, 25357142, 23096938, 21787056, 21273804
19067244, 18043064, 21329301, 18885870, 20324049, 26187943, 19536415
25093739, 17835294, 20446883, 21299490, 25313154, 24413809, 21744290
18254023, 20591183, 18371441, 24385983, 20413820, 24421668, 25897615
19185876, 25643931, 21281607, 20513399, 22465352, 20558005, 20402832
19627012, 20093776, 18909599, 20618595, 27441326, 27620950, 23572982
16863642, 19639483, 19315691, 19211433, 20331945, 19512341, 22256431
21479753, 19637186, 19174521, 19022470, 18607546, 20401975, 18306996
24573817, 18851894, 19649152, 27034890, 20581111, 19201867, 20318889
20936731, 21060755, 21294938, 20898997, 18510194, 22256560, 22454326
19534363, 25489607, 19188927

Version 12.1.0.2.v11

Version 12.1.0.2.v11 includes the following:

- Patch 26925311: DATABASE PATCH SET UPDATE 12.1.0.2.180116
- Patch 27001733: OJVM PATCH SET UPDATE 12.1.0.2.180116
- Patch 27015449: RDBMS - PROACTIVE DSTV31 UPDATE - TZDATA2017C
- Patch 27015468: PROACTIVE DSTV31 UPDATE - TZDATA2017C - NEED OJVM FIX
- Patch 17969866: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 20394750: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 21171382: AUTO DOP COMPUTES A HIGH DOP UNNECESSARILY
- Patch 27315904: JSON Database Patch
- Patch 20033733: ORA 600 [KGL-HEAP-SIZE-EXCEEDED]

Oracle patch 26925311, released January 2018

Bugs fixed:

21099555, 22175564, 19141838, 22083366, 20842388, 19865345, 20117253
20830459, 19791273, 20671094, 21542577, 23105538, 19243521, 20951038
22165897, 19238590, 21281532, 17008068, 19908836, 24401351, 24577566
21184223, 25427662, 20717359, 19134173, 20569094, 20031873, 20387265
20322560, 21575362, 19149990, 21263635, 18886413, 17551063, 24719736
22160989, 22519146, 21623164, 22507210, 19703301, 23338911, 19366375
18007682, 19001390, 18202441, 24285405, 25655390, 20267166, 19358317
19706965, 19068970, 24739928, 18549238, 22148226, 18797519, 26544823
20825533, 23521523, 21196809, 18940497, 19670108, 19649152, 18866977
18948177, 19404068, 22496904, 22826718, 18964978, 19176326, 19035573
20413820, 20717081, 19176223, 21106027, 20904530, 20134339, 19074147
20868862, 18411216, 23035249, 25475853, 21072646, 21322887, 22507234

20425790, 20862087, 18966843, 25861398, 24929210, 24624166, 21329301
20562898, 19333670, 19468991, 20124446, 19883092, 23543183, 20878790
22855193, 18510194, 19658708, 19591608, 19402853, 23149541, 24796092
20618595, 22238921, 21795111, 21787056, 22380919, 19469538, 21266085
17835294, 19721304, 19068610, 19791377, 22178855, 16777441, 22173980
20746251, 20048359, 21896069, 19185876, 20898391, 20281121, 20907061
22950945, 21281607, 6599380, 19577410, 22092979, 19001359, 20603378
23089357, 23572982, 19490948, 21387964, 22294260, 20832516, 17532734
22351572, 18849970, 19309466, 19081128, 20627866, 20844426, 24908321
21188532, 18791688, 21442094, 20890311, 20596234, 20368850, 26366517
18973548, 19303936, 21296029, 22536802, 20882568, 21479753, 19461270
20235511, 20936905, 22077517, 21220620, 18964939, 19430401, 22806698
22296366, 21153266, 19409212, 20703000, 22657942, 20657441, 19879746
20557786, 26758193, 23237313, 26198926, 19684504, 26088426, 21294938
19024808, 24693382, 20528052, 20977794, 18799993, 20466322, 24642295
18740837, 19662635, 18440095, 21794615, 20382309, 20228093, 19065556
20212067, 25547060, 21868720, 22905130, 20938170, 19524384, 25459958
24350831, 17722075, 20446883, 20144308, 25056052, 18952989, 24523374
16870214, 21773465, 19928926, 19835133, 21629064, 21354456, 20466628
23007241, 24386767, 25490238, 19931709, 19730508, 18819908, 20250147
23124895, 25643931, 23220453, 1918927, 20074391, 18307021, 23533807
20356733, 14643995, 26430737, 18090142, 19065677, 19547370, 26024732
21225209, 21960504, 18371441, 20397490, 26575788, 23315889, 20172151
18967382, 22729345, 19174430, 22068305, 25654936, 18419520, 21241829
19536415, 26546664, 19171086, 21889720, 21132297, 20470877, 22465352
22168163, 19335438, 24397438, 20076781, 20447445, 18856999, 20471920
19869255, 21620471, 18990693, 23096938, 17890099, 19124336, 24812585
18990023, 20101006, 21300341, 20848335, 21744290, 21241052, 20897759
21668627, 19304354, 19052488, 20543011, 20794034, 23025340, 25606091
23260854, 18681056, 19562381, 24570598, 20952966, 19896336, 20828947
25539063, 18618122, 20328248, 24365589, 20440930, 18456643, 19699191
23065323, 22865673, 19201867, 22816287, 21514877, 22022760, 18743542
20798891, 20347562, 25161298, 23294548, 19777862, 24560906, 22551446
19687159, 21373076, 19174942, 20424899, 24461826, 21641760, 21899588
22862134, 18899974, 21476308, 20598042, 21297872, 24308635, 19058490
19032777, 20171986, 22815955, 25150925, 19399918, 24718260, 19434529
22492533, 19018447, 21273804, 18051556, 22757364, 18851894, 23125826
20424183, 21842017, 19022470, 19284031, 18043064, 26898563, 20173897
23713236, 22062026, 20475845, 17274537, 19440586, 16887946, 22374754
18974476, 22961508, 24825843, 17319928, 20401975, 20708701, 22062517
24674955, 17655240, 22809871, 19805359, 16439813, 19155797, 20859910
19393542, 17210525, 22024071, 19189525, 21847223, 21649497, 19075256
25079710, 25823754, 19370504, 20315311, 22762046, 22075064, 20936731
20437153, 25165496, 18845653, 19280225, 19248799, 20560611, 18988834
21756699, 22256431, 18921743, 20245930, 21532755, 18799063, 22454326
20373598, 20476175, 19571367, 20925795, 19018206, 25264559, 24385983
20509482, 20711718, 24509056, 20588502, 20181030, 21911701, 18849537
23501901, 25034396, 19183343, 22842151, 21917884, 21142837, 20603431
19189317, 23003979, 19644859, 19390567, 19279273, 26546754, 20669434
16863642, 22528741, 22707244, 25546608, 19619732, 20348653, 18607546
19315691, 19676905, 20165574, 17867700, 23528412, 20558005, 20734332
19532017, 20922010, 19818513, 19450314, 22353346, 16941434, 20361671
25423453, 20009833, 22366558, 20294666, 23197103, 18191823, 20860659
22707866, 19195895, 19371175, 19307662, 19154375, 20043616, 20324049
21977392, 18914624, 22529728, 22256560, 25330273, 19708342, 20139391
19593445, 21291274, 19382851, 19520602, 19174521, 21875360, 19676012
19326908, 20217801, 20093776, 18840932, 21097043, 21246723, 20803014
21665897, 19143550, 23026585, 20428621, 19627012, 24415926, 22087683
23548817, 14283239, 21422580, 19213447, 19518079, 26446098, 18610915
23492665, 18674024, 24831514, 21863727, 24413809, 18306996, 19915271
21626377, 19524158, 20122715, 20513399, 18110491, 22366322, 20284155
25091141, 21080143, 20017509, 22359063, 19363645, 19597439, 21239530
23108128, 19888853, 19383839, 20880215, 21756677, 22458049, 19534363
19354335, 19044962, 19639483, 25982666, 19475971, 22353199, 21060755
22243719, 22916353, 20378086, 21260431, 21756661, 24808595, 22923409
19028800, 20877664, 22518784, 21059919, 20879889, 21380789, 19723336

19077215, 21421886, 19604659, 21285458, 23533524, 26569225, 23170620
22365117, 18288842, 19048007, 19308965, 19689979, 17409174, 19503821
23068169, 24662775, 21526048, 25429959, 19197175, 19180770, 24555417
24573817, 19902195, 26444887, 25313154, 24835538, 23324000, 20318889
21492036, 19013183, 20591183, 19012119, 20464614, 22645009, 21625179
19067244, 25178179, 23053606, 21632821, 19841800, 19512341, 19211433
22695831, 20331945, 19587324, 24316947, 19578350, 19637186, 19054077
18674047, 19708632, 20898997, 21091431, 19285025, 19289642, 25947799
21133343, 20835241, 20869721, 21172913, 25602488, 19258504, 17365043
21419850, 21644640, 19468347, 21373473, 25093739, 22721409, 16359751
24421668, 21164318, 25484507, 25489607, 22520320, 19769480, 19439759
19272708, 23088803, 19978542, 19329654, 20402832, 19873610, 23229229
21517440, 13542050, 25897615, 19291380, 21915719, 25600342, 25192729
20879709, 20677396, 19076343, 19561643, 19990037, 18909599, 19487147
22897344, 20831538, 25600421, 19016730, 18250893, 23240358, 22179537
16619249, 18354830, 24411921, 25764020, 18254023, 16756406, 21188584
19989009, 25766822, 17414008, 20688221, 20441797, 20704450, 21780146
25612095, 25957038, 24652769, 25483815, 19157754, 19207117, 24437510
18885870, 21785691, 20673810, 24341675, 21450666, 18893947, 18705806
22223463, 18417036, 16923858, 23084507, 23314180, 20919320, 22503297
20474192, 22046677, 21299490, 19501299, 19385656, 20432873, 18542562
20920911, 20899461, 21315084, 21429602, 21387128, 18122373, 20581111
22624709, 26111842, 19606174, 24690216, 18436647, 19023822, 25110233
19124589, 19178851, 19597583, 20480209, 18499088, 19050649

Version 12.1.0.2.v10

Version 12.1.0.2.v10 includes the following:

- Oracle October 2017 PSU, a combination of database PSU (patch 26713565) + OJVM component PSU (patch 26635845)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 17969866)
- DBMS_STATS AUTO DOP COMPUTES A HIGH DOP UNNECESSARILY (patch 21171382)
- JSON bundle patch (patch 26750145)
- KGL heap size patch (patch 20033733)
- Timezone file DSTv30 (patch 25881255, OJVM patch 25881271)

Oracle patch 26713565, released October 2017

Bugs fixed:

21099555, 22175564, 19141838, 22083366, 20842388, 19865345, 20117253
20830459, 19791273, 20671094, 21542577, 19243521, 20951038, 22165897
19238590, 21281532, 17008068, 19908836, 24577566, 21184223, 25427662
19134173, 20569094, 20031873, 20387265, 20322560, 21575362, 19149990
21263635, 17551063, 18886413, 24719736, 22160989, 22519146, 21623164
22507210, 23338911, 19703301, 19366375, 18007682, 19001390, 18202441
24285405, 25655390, 20267166, 19358317, 19706965, 19068970, 24739928
18549238, 22148226, 18797519, 26544823, 20825533, 23521523, 21196809
18940497, 19670108, 19649152, 18866977, 18948177, 22496904, 19404068
18964978, 19176326, 19035573, 20413820, 20717081, 19176223, 21106027
20904530, 20134339, 19074147, 20868862, 23035249, 18411216, 21072646
25475853, 21322887, 22507234, 20425790, 20862087, 18966843, 25861398
21329301, 20562898, 19333670, 19468991, 20124446, 19883092, 22855193
20878790, 18510194, 19658708, 19591608, 19402853, 23149541, 20618595
22238921, 21795111, 21787056, 22380919, 19469538, 21266085, 17835294
19721304, 19068610, 19791377, 22178855, 16777441, 22173980, 20746251
20048359, 21896069, 19185876, 20898391, 20281121, 20907061, 22950945

6599380, 19577410, 22092979, 19001359, 20603378, 23089357, 21387964
19490948, 22294260, 20832516, 17532734, 22351572, 19309466, 19081128
20627866, 20844426, 24908321, 21188532, 18791688, 21442094, 20890311
20596234, 20368850, 18973548, 19303936, 21296029, 20882568, 21479753
19461270, 20235511, 22077517, 20936905, 21220620, 18964939, 19430401
22806698, 22296366, 21153266, 19409212, 22657942, 20703000, 20657441
19879746, 20557786, 26198926, 26088426, 19684504, 21294938, 19024808
24693382, 20528052, 20977794, 18799993, 20466322, 24642295, 18740837
19662635, 18440095, 21794615, 20228093, 19065556, 20212067, 25547060
21868720, 20938170, 22905130, 19524384, 25459958, 24350831, 17722075
20144308, 20446883, 25056052, 18952989, 24523374, 16870214, 19928926
19835133, 21629064, 21354456, 20466628, 24386767, 25490238, 19931709
19730508, 18819908, 20250147, 23124895, 25643931, 23220453, 19188927
20074391, 18307021, 23533807, 20356733, 26430737, 14643995, 18090142
19065677, 19547370, 21225209, 21960504, 18371441, 20397490, 26575788
23315889, 20172151, 18967382, 19174430, 22068305, 25654936, 21241829
19536415, 19171086, 26546664, 21132297, 21889720, 22465352, 22168163
19335438, 24397438, 20076781, 20447445, 18856999, 20471920, 19869255
21620471, 18990693, 23096938, 19124336, 17890099, 24812585, 18990023
21300341, 20101006, 20848335, 21744290, 21241052, 20897759, 21668627
19304354, 19052488, 20543011, 20794034, 23025340, 25606091, 23260854
18681056, 19562381, 20952966, 19896336, 20828947, 25539063, 18618122
20328248, 20440930, 18456643, 19699191, 22865673, 19201867, 22816287
22022760, 21514877, 18743542, 20798891, 20347562, 25161298, 23294548
24560906, 22551446, 19777862, 19687159, 21373076, 19174942, 20424899
21899588, 22862134, 18899974, 21476308, 20598042, 24308635, 21297872
19058490, 19032777, 20171986, 22815955, 19399918, 19434529, 19018447
18051556, 21273804, 22757364, 18851894, 23125826, 20424183, 21842017
19022470, 19284031, 18043064, 23713236, 20173897, 22062026, 20475845
17274537, 19440586, 22961508, 24825843, 18974476, 22374754, 16887946
17319928, 20401975, 20708701, 24674955, 22062517, 22809871, 17655240
19805359, 16439813, 19155797, 20859910, 19393542, 17210525, 22024071
19189525, 21847223, 21649497, 19075256, 25823754, 25079710, 20315311
22762046, 22075064, 20936731, 20437153, 18845653, 19280225, 19248799
20560611, 18988834, 21756699, 22256431, 21532755, 18921743, 20245930
22454326, 18799063, 20373598, 20476175, 19571367, 20925795, 19018206
25264559, 20711718, 20509482, 20181030, 20588502, 21911701, 18849537
23501901, 25034396, 19183343, 22842151, 21917884, 21142837, 20603431
19189317, 23003979, 19644859, 19390567, 19279273, 26546754, 20669434
16863642, 22528741, 22707244, 25546608, 19619732, 20348653, 18607546
19315691, 19676905, 20165574, 17867700, 20558005, 20734332, 19532017
20922010, 19818513, 19450314, 22353346, 16941434, 20361671, 25423453
20009833, 22366558, 20294666, 23197103, 18191823, 20860659, 19195895
19371175, 19307662, 19154375, 20043616, 21977392, 18914624, 22529728
19708342, 20139391, 25330273, 19593445, 21291274, 19382851, 19520602
19174521, 21875360, 19676012, 19326908, 20217801, 20093776, 18840932
21097043, 21246723, 20803014, 21665897, 19143550, 23026585, 20428621
19627012, 22087683, 23548817, 14283239, 21422580, 19213447, 26446098
19518079, 23492665, 18610915, 18674024, 21863727, 24413809, 18306996
19915271, 21626377, 19524158, 20122715, 20513399, 18110491, 20284155
25091141, 21080143, 20017509, 22359063, 19363645, 19597439, 21239530
23108128, 19383839, 20880215, 21756677, 19888853, 22458049, 19534363
19354335, 19044962, 19639483, 25982666, 19475971, 22353199, 21060755
22243719, 22916353, 20378086, 24808595, 21756661, 21260431, 22923409
19028800, 20877664, 21059919, 20879889, 21380789, 19723336, 19077215
21421886, 19604659, 21285458, 23533524, 23170620, 22365117, 18288842
19048007, 19308965, 19689979, 17409174, 23068169, 19503821, 24662775
25429959, 21526048, 19197175, 19180770, 24555417, 24573817, 19902195
26444887, 24835538, 23324000, 20318889, 21492036, 19013183, 20591183
19012119, 20464614, 21625179, 19067244, 23053606, 21632821, 19841800
19512341, 22695831, 20331945, 19587324, 24316947, 19578350, 19637186
19054077, 18674047, 19708632, 20898997, 19285025, 21091431, 19289642
25947799, 21133343, 20835241, 20869721, 21172913, 25602488, 19258504
17365043, 21419850, 21644640, 19468347, 21373473, 25093739, 16359751
24421668, 21164318, 25489607, 25484507, 22520320, 19769480, 19439759
19272708, 19978542, 19329654, 20402832, 19873610, 23229229, 13542050

21517440, 25897615, 19291380, 21915719, 25600342, 20879709, 20677396
19076343, 19561643, 19990037, 22897344, 18909599, 19487147, 25600421
20831538, 19016730, 18250893, 23240358, 22179537, 16619249, 18354830
24411921, 18254023, 16756406, 21188584, 19989009, 25766822, 17414008
20688221, 20441797, 20704450, 21780146, 25612095, 25957038, 24652769
25483815, 19157754, 19207117, 24437510, 18885870, 21785691, 20673810
24341675, 21450666, 18893947, 18705806, 22223463, 18417036, 16923858
23084507, 23314180, 20919320, 22503297, 20474192, 22046677, 21299490
19501299, 19385656, 20432873, 18542562, 20920911, 20899461, 21429602
21387128, 21315084, 18122373, 20581111, 26111842, 22624709, 19606174
24690216, 18436647, 19023822, 25110233, 19124589, 19178851, 19597583
18499088, 19050649

Version 12.1.0.2.v9

Version 12.1.0.2.v9 includes the following:

- Oracle July 2017 PSU, a combination of database PSU (patch 26609783) + OJVM component PSU (patch 26027162)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 17969866)
- DBMS_STATS AUTO DOP COMPUTES A HIGH DOP UNNECESSARILY (patch 21171382)
- JSON bundle patch (patch 26083365)
- KGL heap size patch (patch 20033733 for 12.1.0.2)
- Timezone file DSTv30 (patch 25881255, OJVM patch 25881271)
- Support for [Validating DB instance files \(p. 1072\)](#) with the RMAN logical validation utility
- Support for [Setting the default edition for a DB instance \(p. 1057\)](#)

Oracle patch 26609783, released July 2017

Bugs fixed:

21099555, 22175564, 19141838, 22083366, 20842388, 19865345, 20117253
19791273, 20671094, 21542577, 20951038, 19243521, 22165897, 19238590
21281532, 17008068, 19908836, 24577566, 21184223, 25427662, 19134173
20569094, 20031873, 20387265, 20322560, 21575362, 19149990, 21263635
17551063, 18886413, 22160989, 22507210, 19703301, 19366375, 18007682
19001390, 18202441, 24285405, 25655390, 20267166, 19358317, 19706965
19068970, 24739928, 18549238, 22148226, 18797519, 26544823, 20825533
21196809, 18940497, 19670108, 19649152, 18866977, 18948177, 22496904
19404068, 18964978, 19176326, 19035573, 20413820, 20717081, 19176223
21106027, 20904530, 20134339, 19074147, 20868862, 18411216, 21072646
25475853, 21322887, 22507234, 20425790, 20862087, 18966843, 21329301
20562898, 19333670, 19468991, 20124446, 19883092, 20878790, 18510194
19658708, 19591608, 19402853, 20618595, 21787056, 22380919, 21266085
19469538, 17835294, 19721304, 19068610, 19791377, 22178855, 16777441
22173980, 20746251, 20048359, 21896069, 19185876, 20898391, 20281121
20907061, 6599380, 19577410, 22092979, 19001359, 20603378, 23089357
21387964, 19490948, 22294260, 20832516, 17532734, 22351572, 19309466
19081128, 20627866, 20844426, 24908321, 21188532, 18791688, 21442094
20890311, 20596234, 20368850, 18973548, 19303936, 21296029, 20882568
21479753, 19461270, 20235511, 22077517, 20936905, 21220620, 18964939
19430401, 22296366, 21153266, 19409212, 22657942, 20703000, 20657441
19879746, 20557786, 19684504, 21294938, 19024808, 24693382, 20528052
20977794, 18799993, 20466322, 18740837, 19662635, 18440095, 20228093
19065556, 20212067, 25547060, 21868720, 22905130, 19524384, 25459958
24350831, 17722075, 20446883, 25056052, 18952989, 24523374, 16870214
19928926, 19835133, 21629064, 21354456, 20466628, 24386767, 25490238

19931709, 19730508, 18819908, 20250147, 23124895, 25643931, 23220453
19188927, 20074391, 18307021, 23533807, 20356733, 14643995, 18090142
19065677, 19547370, 21225209, 21960504, 26575788, 20397490, 20172151
18967382, 19174430, 21241829, 19536415, 26546664, 19171086, 21132297
21889720, 22465352, 22168163, 19335438, 24397438, 20076781, 20447445
18856999, 20471920, 19869255, 21620471, 18990693, 23096938, 19124336
17890099, 24812585, 18990023, 21300341, 20101006, 20848335, 21744290
20897759, 21668627, 19304354, 19052488, 20543011, 20794034, 23025340
25606091, 23260854, 18681056, 19562381, 20952966, 19896336, 20828947
25539063, 18618122, 20328248, 20440930, 18456643, 19699191, 22865673
19201867, 22022760, 21514877, 18743542, 20798891, 20347562, 25161298
23294548, 24560906, 22551446, 19777862, 19687159, 21373076, 19174942
20424899, 21899588, 18899974, 21476308, 20598042, 24308635, 21297872
19058490, 19032777, 20171986, 22815955, 19399918, 19434529, 19018447
18051556, 21273804, 22757364, 18851894, 19022470, 19284031, 18043064
20173897, 22062026, 20475845, 17274537, 19440586, 24825843, 18974476
22374754, 16887946, 17319928, 20401975, 20708701, 24674955, 22062517
22809871, 17655240, 19805359, 16439813, 19155797, 20859910, 19393542
17210525, 22024071, 19189525, 21847223, 21649497, 19075256, 25823754
25079710, 20315311, 22762046, 22075064, 20936731, 20437153, 18845653
19280225, 19248799, 20560611, 18988834, 21756699, 18921743, 20245930
18799063, 20373598, 20476175, 19571367, 20925795, 19018206, 25264559
20711718, 20509482, 20181030, 2058502, 21911701, 18849537, 23501901
19183343, 21917884, 21142837, 20603431, 19189317, 19644859, 19390567
26546754, 19279273, 20669434, 16863642, 22528741, 25546608, 19619732
20348653, 18607546, 19315691, 19676905, 20165574, 17867700, 20558005
20734332, 19532017, 20922010, 19818513, 19450314, 22353346, 16941434
20361671, 25423453, 20009833, 22366558, 20294666, 23197103, 18191823
19195895, 19371175, 19307662, 19154375, 20043616, 21977392, 18914624
22529728, 20139391, 25330273, 19593445, 21291274, 19382851, 19520602
19174521, 21875360, 19676012, 19326908, 20217801, 20093776, 18840932
21097043, 21246723, 20803014, 21665897, 19143550, 23026585, 20428621
19627012, 14283239, 21422580, 19213447, 19518079, 18610915, 18674024
24413809, 18306996, 19915271, 21626377, 19524158, 20122715, 20513399
20284155, 25091141, 21080143, 20017509, 22359063, 19363645, 19597439
21239530, 19383839, 20880215, 21756677, 19888853, 22458049, 19534363
19354335, 19044962, 19639483, 25982666, 19475971, 22353199, 21060755
22243719, 22916353, 20378086, 24808595, 21756661, 21260431, 22923409
19028800, 20877664, 21059919, 20879889, 21380789, 19723336, 19077215
21421886, 19604659, 21285458, 23533524, 23170620, 22365117, 18288842
19048007, 19308965, 19689979, 17409174, 19503821, 21526048, 19197175
19180770, 24573817, 19902195, 24835538, 23324000, 20318889, 19013183
20591183, 19012119, 20464614, 19067244, 21632821, 19841800, 19512341
22695831, 20331945, 19587324, 24316947, 19578350, 19637186, 19054077
18674047, 19708632, 20898997, 21091431, 19289642, 21133343, 20835241
20869721, 21172913, 19258504, 17365043, 21419850, 21644640, 19468347
21373473, 25093739, 16359751, 21164318, 25484507, 22520320, 19769480
19439759, 19272708, 19978542, 19329654, 20402832, 19873610, 23229229
13542050, 21517440, 19291380, 21915719, 25600342, 20879709, 20677396
19076343, 19561643, 19990037, 18909599, 19487147, 25600421, 20831538
19016730, 18250893, 16619249, 18354830, 24411921, 16756406, 18254023
21188584, 19989009, 25766822, 17414008, 20688221, 20441797, 20704450
21780146, 25612095, 25957038, 25483815, 19157754, 19207117, 24437510
18885870, 21785691, 20673810, 21450666, 18893947, 18705806, 22223463
18417036, 16923858, 23314180, 20919320, 20474192, 22046677, 21299490
19501299, 19385656, 20432873, 20920911, 20899461, 21387128, 21315084
18122373, 20581111, 22624709, 19606174, 24690216, 18436647, 19023822
25110233, 19124589, 19178851, 19597583, 18499088, 19050649

Version 12.1.0.2.v8

Version 12.1.0.2.v8 includes the following:

- Oracle patch 25433980, a combination of database PSU (patch 25171037) + OJVM component PSU (patch 25437695)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 17969866 for 12.1.0.2)
- Oracle Forms patch 18307021 for 12.1.0.2
- DBMS_STATS Patch (patch 21171382 for 12.1.0.2)
- JSON bundle patch (patch 25531469 for 12.1.0.2)
- KGL heap size patch (patch 20033733 for 12.1.0.2)
- Fixed a bug that affected PSU apply after upgrade to 12.1.0.2.v5, v6, and v7
- Timezone file DSTv28 (patch 24701840)
- Support for the DBMS_CHANGE_NOTIFICATION package
- Support for XSTREAM packages and views (may require additional licensing)

Oracle patch 25171037, released April 2017

Bugs fixed:

```
21099555, 22175564, 19141838, 22083366, 20842388, 20117253, 19865345
19791273, 21542577, 20951038, 19243521, 22165897, 17008068, 19908836
21281532, 19238590, 24577566, 21184223, 19134173, 20569094, 20031873
20322560, 20387265, 21575362, 19149990, 21263635, 17551063, 18886413
22160989, 22507210, 19366375, 19703301, 19001390, 24285405, 18202441
20267166, 19358317, 19706965, 19068970, 18549238, 24739928, 18797519
22148226, 20825533, 21196809, 19649152, 19670108, 18940497, 18948177
22496904, 18964978, 19176326, 19035573, 20413820, 19176223, 21106027
20904530, 20134339, 19074147, 20868862, 18411216, 25475853, 21322887
21072646, 22507234, 20425790, 20862087, 18966843, 21329301, 20562898
19333670, 20124446, 19468991, 19883092, 20878790, 18510194, 19658708
19591608, 19402853, 20618595, 21787056, 22380919, 19469538, 21266085
17835294, 19721304, 19068610, 19791377, 22178855, 16777441, 22173980
20048359, 20746251, 21896069, 19185876, 20898391, 20907061, 20281121
6599380, 19577410, 22092979, 19001359, 20603378, 23089357, 21387964
19490948, 22294260, 17532734, 20832516, 22351572, 19309466, 20627866
19081128, 20844426, 21188532, 18791688, 20890311, 21442094, 20596234
20368850, 18973548, 19303936, 21296029, 20882568, 19461270, 21479753
22077517, 20936905, 20235511, 21220620, 18964939, 19430401, 22296366
21153266, 19409212, 20703000, 22657942, 19879746, 20657441, 21294938
19684504, 19024808, 20528052, 24693382, 20977794, 18799993, 20466322
18740837, 19662635, 18440095, 20228093, 19065556, 20212067, 21868720
22905130, 19524384, 24350831, 17722075, 20446883, 25056052, 18952989
24523374, 16870214, 19928926, 19835133, 21629064, 21354456, 20466628
24386767, 25490238, 19931709, 19730508, 18819908, 20250147, 23124895
23220453, 19188927, 20074391, 18307021, 20356733, 14643995, 19065677
19547370, 21960504, 21225209, 20397490, 18967382, 19174430, 21241829
19536415, 19171086, 21889720, 22465352, 22168163, 19335438, 24397438
20447445, 18856999, 19869255, 20471920, 21620471, 23096938, 18990693
19124336, 17890099, 24812585, 18990023, 21300341, 20101006, 20848335
21744290, 20897759, 21668627, 19304354, 20543011, 19052488, 20794034
23025340, 23260854, 18681056, 20952966, 19896336, 25539063, 18618122
20328248, 20440930, 18456643, 19699191, 19201867, 22865673, 22022760
20798891, 18743542, 25161298, 20347562, 22551446, 19777862, 19687159
21373076, 19174942, 20424899, 21899588, 18899974, 21476308, 20598042
21297872, 24308635, 20171986, 19058490, 19032777, 22815955, 19399918
19434529, 21273804, 19018447, 22757364, 18851894, 19022470, 19284031
18043064, 20173897, 22062026, 20475845, 17274537, 19440586, 18974476
24825843, 22374754, 16887946, 17319928, 20401975, 20708701, 22062517
22809871, 17655240, 16439813, 19805359, 19155797, 20859910, 19393542
22024071, 17210525, 19189525, 21847223, 21649497, 25079710, 19075256
20315311, 22762046, 22075064, 20936731, 18845653, 19280225, 19248799
```

20560611, 18988834, 21756699, 18921743, 20245930, 18799063, 20373598
19571367, 20476175, 20925795, 19018206, 25264559, 20711718, 20509482
20181030, 2058802, 21911701, 18849537, 23501901, 19183343, 21917884
21142837, 19189317, 19644859, 19390567, 19279273, 20669434, 16863642
22528741, 25546608, 19619732, 18607546, 20348653, 19315691, 19676905
20165574, 17867700, 20558005, 20734332, 19532017, 20922010, 19818513
19450314, 22353346, 16941434, 20361671, 20009833, 22366558, 20294666
18191823, 23197103, 19195895, 19371175, 19307662, 19154375, 20043616
21977392, 18914624, 22529728, 25330273, 20139391, 19593445, 21291274
19382851, 19520602, 19174521, 21875360, 19676012, 19326908, 20217801
20093776, 18840932, 21097043, 21246723, 20803014, 21665897, 19143550
20428621, 19627012, 14283239, 21422580, 19213447, 19518079, 18610915
18674024, 24413809, 18306996, 19915271, 19524158, 20122715, 20284155
20017509, 22359063, 19363645, 19597439, 21239530, 19383839, 20880215
21756677, 19888853, 22458049, 19534363, 19354335, 19044962, 19639483
19475971, 22353199, 22243719, 21060755, 22916353, 20378086, 24808595
21756661, 21260431, 22923409, 19028800, 20877664, 21059919, 20879889
21380789, 19723336, 19077215, 19604659, 21421886, 21285458, 23533524
23170620, 22365117, 18288842, 19048007, 19308965, 19689979, 19503821
21526048, 19197175, 19180770, 19902195, 23324000, 20318889, 19013183
20591183, 19012119, 20464614, 19067244, 21632821, 19841800, 19512341
22695831, 20331945, 19587324, 24316947, 19578350, 19637186, 19054077
18674047, 19708632, 20898997, 21091431, 19289642, 21133343, 20869721
21172913, 19258504, 17365043, 21419850, 19468347, 21373473, 25093739
16359751, 21164318, 22520320, 19769480, 19439759, 19272708, 19978542
19329654, 20402832, 19873610, 23229229, 13542050, 21517440, 19291380
21915719, 20879709, 20677396, 19076343, 19561643, 19990037, 19487147
18909599, 20831538, 19016730, 18250893, 16619249, 18354830, 24411921
16756406, 18254023, 21188584, 19989009, 17414008, 20688221, 20704450
20441797, 25483815, 19157754, 24437510, 18885870, 21785691, 20673810
21450666, 18893947, 18705806, 22223463, 16923858, 18417036, 23314180
20919320, 20474192, 22046677, 21299490, 19501299, 19385656, 20920911
20899461, 21387128, 21315084, 18122373, 20581111, 19606174, 24690216
18436647, 19023822, 19124589, 19178851, 19597583, 18499088, 19050649

Version 12.1.0.2.v7

Version 12.1.0.2.v7 includes the following:

- Oracle patch 24917069, a combination of database PSU (patch 24732082) + OJVM component PSU (patch 24917972)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 17969866 for 12.1.0.2)
- Oracle Forms patch 18307021 for 12.1.0.2
- DBMS_STATS Patch (patch 21171382 for 12.1.0.2)
- JSON bundle patch (patch 25089615 for 12.1.0.2)
- KGL heap size patch (patch 20033733 for 12.1.0.2)

Oracle patch 24917069, released January 2017

Bugs fixed:

24917972, 25067795, 24534298, 25076732, 25076756, 24315824, 21659726
24448240, 24448282, 23177536, 22675136, 23265914, 23265965, 23727148
22674709, 22670413, 22670385, 21188537, 22139226, 22118835, 22118851
21555660, 21811517, 19623450, 21566993, 21566944, 19176885, 21068507
21047803, 21047766, 20415564, 20408829, 20408866, 19877336, 19855285
19909862, 19895362, 19895326, 19153980, 19231857, 19223010, 19245191,

19699946, 21099555, 22175564, 19141838, 22083366, 20842388, 20117253, 19865345 19791273, 21542577, 20951038, 19243521, 22165897, 19908836, 21281532 19238590, 24577566, 21184223, 19134173, 20031873, 20387265, 21575362 19149990, 21263635, 17551063, 18886413, 22160989, 22507210, 19366375 19703301, 19001390, 24285405, 18202441, 20267166, 19358317, 19706965 24739928, 19068970, 18549238, 18797519, 22148226, 20825533, 21196809 19649152, 19670108, 18940497, 18948177, 22496904, 18964978, 19035573 19176326, 20413820, 19176223, 21106027, 20904530, 20134339, 19074147 20868862, 18411216, 21072646, 21322887, 22507234, 20425790, 18966843 21329301, 20562898, 19333670, 20124446, 19468991, 19883092, 18510194 19658708, 19591608, 19402853, 20618595, 21787056, 22380919, 19469538 21266085, 17835294, 19721304, 19791377, 19068610, 22178855, 16777441 22173980, 20048359, 20746251, 21896069, 20898391, 19185876, 20907061 20281121, 6599380, 19577410, 22092979, 19001359, 20603378, 23089357 19490948, 21387964, 22294260, 20832516, 17532734, 19309466, 20627866 19081128, 20844426, 21188532, 18791688, 20890311, 21442094, 20596234 18973548, 21296029, 19303936, 20882568, 19461270, 21479753, 22077517 20936905, 20235511, 21220620, 18964939, 19430401, 22296366, 21153266 19409212, 22657942, 19879746, 20657441, 21294938, 19684504, 24693382 20528052, 19024808, 20977794, 18799993, 20466322, 18740837, 19662635 20228093, 20212067, 19065556, 19524384, 17722075, 20446883, 25056052 24523374, 18952989, 16870214, 19928926, 19835133, 21629064, 21354456 20466628, 24386767, 19931709, 19730508, 18819908, 23124895, 23220453 19188927, 20074391, 18307021, 20356733, 14643995, 19547370, 19065677 21960504, 21225209, 20397490, 18967382, 19174430, 21241829, 19536415 19171086, 22465352, 22168163, 19335438, 24397438, 20447445, 18856999 19869255, 20471920, 21620471, 18990693, 17890099, 24812585, 18990023 21300341, 20101006, 20848335, 21744290, 20897759, 21668627, 19304354 19052488, 20794034, 23025340, 23260854, 18681056, 20952966, 19896336 20328248, 18618122, 20440930, 18456643, 19699191, 19201867, 22865673 22022760, 20798891, 18743542, 25161298, 20347562, 19777862, 22551446 19687159, 21373076, 19174942, 20424899, 21899588, 18899974, 21476308 20598042, 24308635, 19032777, 19058490, 22815955, 19399918, 19434529 21273804, 19018447, 22757364, 18851894, 19022470, 19284031, 18043064 20173897, 22062026, 20475845, 17274537, 19440586, 24825843, 18974476 22374754, 16887946, 17319928, 20401975, 20708701, 22809871, 17655240 16439813, 19805359, 19155797, 20859910, 19393542, 17210525, 22024071 21847223, 19189525, 21649497, 19075256, 20315311, 22762046, 22075064 20936731, 19280225, 18845653, 20560611, 19248799, 21756699, 18988834 20245930, 18921743, 18799063, 20373598, 19571367, 20476175, 20925795 25264559, 19018206, 20711718, 20509482, 20181030, 20588502, 18849537 23501901, 19183343, 21917884, 19189317, 19644859, 19390567, 19279273 20669434, 22528741, 16863642, 19619732, 18607546, 20348653, 19315691 19676905, 20165574, 17867700, 20558005, 20734332, 19532017, 20922010 19818513, 19450314, 22353346, 20361671, 20009833, 22366558, 20294666 23197103, 18191823, 19195895, 19307662, 19371175, 20043616, 19154375 18914624, 22529728, 20139391, 21291274, 19382851, 19520602, 19174521 21875360, 19676012, 19326908, 20217801, 20093776, 18840932, 21097043 21246723, 20803014, 21665897, 19143550, 20428621, 19627012, 14283239 19518079, 18610915, 18674024, 24413809, 18306996, 19524158, 19915271 20122715, 20284155, 20017509, 22359063, 19363645, 19597439, 21239530 19888853, 21756677, 20880215, 22458049, 19534363, 19354335, 19044962 19639483, 19475971, 22353199, 21060755, 22243719, 22916353, 20378086 24808595, 21260431, 21756661, 22923409, 20877664, 19028800, 21059919 20879889, 21380789, 19723336, 19077215, 19604659, 21421886, 21285458 23533524, 23170620, 22365117, 18288842, 19308965, 19048007, 19689979 21526048, 19197175, 19180770, 19902195, 23324000, 20318889, 19013183 20591183, 19012119, 20464614, 19067244, 21632821, 19512341, 19841800 22695831, 20331945, 19587324, 24316947, 19578350, 19637186, 18674047 19054077, 20898997, 19708632, 21091431, 19289642, 21133343, 20869721 21172913, 19258504, 17365043, 19468347, 21373473, 16359751, 19769480 19439759, 19272708, 19978542, 20402832, 19329654, 19873610, 23229229 21517440, 13542050, 19291380, 21915719, 20879709, 20677396, 19076343 19561643, 19990037, 19487147, 18909599, 20831538, 18250893, 19016730 16619249, 18354830, 18254023, 21188584, 19989009, 17414008, 20688221
--

20704450, 20441797, 19157754, 24437510, 18885870, 21785691, 18893947
21450666, 18705806, 22223463, 16923858, 18417036, 23314180, 20919320
20474192, 22046677, 19385656, 19501299, 20920911, 20899461, 21315084
21387128, 18122373, 20581111, 19606174, 24690216, 18436647, 19023822
19178851, 19124589, 19597583, 18499088, 19050649

Version 12.1.0.2.v6

Version 12.1.0.2.v6 includes the following:

- Oracle patch 24433133, a combination of database PSU (patch 24006101) + OJVM component PSU (patch 24315824)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 17969866 for 12.1.0.2)
- Oracle Forms patch 18307021 for 12.1.0.2
- DBMS_STATS Patch (patch 21171382 for 12.1.0.2)
- JSON bundle patch (patch 24568656 for 12.1.0.2)
- Fixed a bug that caused 12c upgrade scripts to drop customer directories
- Made DIAG log directory available to customers

Baseline: Oracle database patch set update 12.1.0.2.161018 (patch 24006101, released October 2016)

Bugs fixed:

21099555, 22175564, 19141838, 22083366, 20842388, 20117253, 19865345
19791273, 19243521, 20951038, 19908836, 21281532, 19238590, 24577566
21184223, 19134173, 20387265, 19149990, 21263635, 18886413, 17551063
22160989, 22507210, 19703301, 19366375, 19001390, 18202441, 20267166
19358317, 19706965, 18549238, 19068970, 18797519, 22148226, 20825533
19649152, 19670108, 18940497, 18948177, 18964978, 19035573, 19176326
20413820, 19176223, 20904530, 20134339, 19074147, 20868862, 18411216
21322887, 22507234, 20425790, 18966843, 21329301, 19333670, 19468991
20124446, 19883092, 19658708, 19591608, 19402853, 20618595, 21787056
22380919, 21266085, 17835294, 19721304, 19791377, 19068610, 22178855
22173980, 20746251, 20048359, 20898391, 19185876, 20281121, 20907061
6599380, 19577410, 22092979, 20603378, 19001359, 19490948, 21387964
20832516, 17532734, 19309466, 19081128, 20627866, 20844426, 21188532
18791688, 21442094, 20890311, 20596234, 18973548, 21296029, 19303936
19461270, 21479753, 20936905, 20235511, 21220620, 18964939, 19430401
22296366, 21153266, 19409212, 22657942, 20657441, 19879746, 19684504
20528052, 19024808, 20977794, 18799993, 20466322, 18740837, 19662635
20228093, 19065556, 20212067, 19524384, 17722075, 20446883, 18952989
16870214, 19928926, 19835133, 21629064, 20466628, 24386767, 19931709
19730508, 18819908, 23124895, 19188927, 20074391, 20356733, 14643995
19547370, 19065677, 21960504, 21225209, 20397490, 18967382, 19174430
21241829, 19536415, 19171086, 22465352, 22168163, 19335438, 20447445
18856999, 20471920, 19869255, 21620471, 18990693, 17890099, 18990023
20101006, 21300341, 20848335, 21744290, 20897759, 21668627, 19304354
19052488, 20794034, 23260854, 18681056, 20952966, 19896336, 18618122
20328248, 20440930, 18456643, 19699191, 19201867, 22865673, 18743542
20798891, 20347562, 22551446, 19777862, 19687159, 21373076, 19174942
20424899, 21899588, 18899974, 20598042, 19032777, 19058490, 22815955
19399918, 19434529, 21273804, 19018447, 22757364, 18851894, 19284031
19022470, 18043064, 20173897, 22062026, 20475845, 17274537, 19440586
16887946, 22374754, 17319928, 20708701, 17655240, 16439813, 19805359
19155797, 20859910, 19393542, 22024071, 17210525, 21847223, 19189525
21649497, 19075256, 22762046, 22075064, 19280225, 18845653, 20560611

19248799, 21756699, 18988834, 20245930, 18921743, 18799063, 20373598
20476175, 19571367, 20925795, 19018206, 20509482, 20711718, 20588502
18849537, 19183343, 21917884, 19189317, 19644859, 19390567, 19279273
20669434, 16863642, 22528741, 19619732, 18607546, 20348653, 19315691
19676905, 20165574, 17867700, 20558005, 20734332, 19532017, 20922010
19450314, 22353346, 20361671, 20009833, 22366558, 20294666, 18191823
19307662, 19371175, 19195895, 20043616, 19154375, 18914624, 20139391
21291274, 19174521, 19520602, 19382851, 21875360, 19676012, 19326908
20217801, 20093776, 21097043, 21246723, 21665897, 19143550, 20428621
19627012, 14283239, 19518079, 18610915, 18674024, 18306996, 19524158
19915271, 20122715, 20284155, 20017509, 19363645, 19597439, 21239530
19888853, 20880215, 21756677, 19534363, 19354335, 19044962, 19639483
22353199, 22243719, 22916353, 20378086, 21756661, 21260431, 22923409
20877664, 19028800, 20879889, 19723336, 19077215, 21421886, 19604659
19308965, 19048007, 18288842, 19689979, 21526048, 19180770, 19197175
19902195, 20318889, 19013183, 19012119, 20464614, 19067244, 21632821
19512341, 19841800, 20331945, 19587324, 24316947, 19578350, 19637186
18674047, 19054077, 20898997, 19708632, 21091431, 19289642, 20869721
19258504, 17365043, 19468347, 21373473, 16359751, 19439759, 19769480
19272708, 19978542, 20402832, 19329654, 19873610, 23229229, 21517440
13542050, 19291380, 21915719, 19076343, 19561643, 19990037, 19487147
18909599, 20831538, 18250893, 19016730, 16619249, 18354830, 21188584
19989009, 17414008, 20688221, 20704450, 20441797, 19157754, 18885870
21785691, 21450666, 18893947, 18705806, 22223463, 16923858, 18417036
20919320, 20474192, 22046677, 19385656, 19501299, 20920911, 20899461
21387128, 21315084, 18122373, 20581111, 19606174, 18436647, 19023822
19178851, 19124589, 19597583, 18499088, 19050649

Version 12.1.0.2.v5

Version 12.1.0.2.v5 includes the following:

- Oracle patch 23615289, a combination of database PSU (patch 23054246) + OJVM component PSU (patch 23177536)
- Timezone file DSTv26 (patch 22873635 for 12.1.0.2)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 17969866 for 12.1.0.2)
- Oracle Forms patch 18307021 for 12.1.0.2
- Ability to create custom password verify functions (see [Creating custom functions to verify passwords \(p. 1042\)](#))
- Fixed a bug that prevented implicit recompilation of views owned by SYS

Baseline: Oracle database patch set update 12.1.0.2.160719 (patch 23054246, released July 2016)

Bugs fixed:

19189525, 21847223, 21099555, 21649497, 19075256, 19141838, 22762046
22075064, 20117253, 19865345, 19791273, 18845653, 19280225, 19248799
19243521, 20951038, 18988834, 21756699, 21281532, 19238590, 21184223
18921743, 20245930, 18799063, 19134173, 20373598, 19571367, 20476175
20925795, 19018206, 20509482, 20711718, 20387265, 20588502, 19149990
21263635, 18849537, 18886413, 17551063, 22507210, 19183343, 19366375
19703301, 21917884, 19001390, 18202441, 19189317, 20267166, 19644859
19390567, 19358317, 19279273, 19706965, 18549238, 16863642, 19068970
22528741, 18797519, 20825533, 19619732, 18607546, 20348653, 19649152
19670108, 18940497, 18948177, 19315691, 19676905, 18964978, 19176326
20165574, 19035573, 20413820, 17867700, 20558005, 19176223, 19532017

20904530, 20134339, 19450314, 19074147, 22353346, 20868862, 18411216
22507234, 20361671, 20425790, 18966843, 20009833, 22366558, 21329301
20294666, 18191823, 19333670, 19195895, 19371175, 19307662, 19154375
20043616, 20124446, 18914624, 19468991, 19883092, 21291274, 19382851
19520602, 19174521, 21875360, 19676012, 19326908, 19658708, 19591608
19402853, 20093776, 20618595, 21787056, 22380919, 21246723, 17835294
19721304, 19068610, 19791377, 21665897, 22178855, 22173980, 20048359
20746251, 19143550, 20898391, 19185876, 19627012, 20281121, 19577410
22092979, 19001359, 14283239, 19518079, 18610915, 19490948, 17532734
18674024, 18306996, 19309466, 19081128, 19524158, 19915271, 20122715
21188532, 18791688, 20284155, 20890311, 21442094, 20596234, 18973548
21296029, 19303936, 19597439, 20936905, 20235511, 21220620, 20880215
18964939, 21756677, 19888853, 19534363, 19430401, 19354335, 19044962
19639483, 22296366, 22353199, 21153266, 19409212, 19879746, 20657441
19684504, 20528052, 19024808, 20977794, 20378086, 18799993, 21756661
21260431, 18740837, 22923409, 19028800, 20877664, 20228093, 20879889
19065556, 19723336, 19077215, 19604659, 21421886, 19524384, 17722075
19308965, 18288842, 19048007, 19689979, 20446883, 18952989, 16870214
19928926, 19835133, 21629064, 21526048, 19197175, 19180770, 20466628
19902195, 19931709, 20318889, 19013183, 19730508, 19012119, 19067244
20074391, 20356733, 14643995, 19512341, 19841800, 20331945, 19587324
19065677, 19547370, 19578350, 21225209, 19637186, 20397490, 18967382
19174430, 21241829, 19054077, 18674047, 20898997, 19708632, 19536415
21091431, 19289642, 20869721, 22168163, 19335438, 19258504, 20447445
17365043, 18856999, 19468347, 19869255, 20471920, 21373473, 21620471
16359751, 18990693, 17890099, 19769480, 19439759, 19272708, 18990023
19978542, 19329654, 20101006, 21300341, 20402832, 19873610, 20848335
23229229, 21744290, 21668627, 21517440, 13542050, 19304354, 19052488
20794034, 19291380, 21915719, 23260854, 18681056, 20952966, 19896336
19076343, 19561643, 18618122, 19990037, 20440930, 18456643, 19699191
19201867, 19487147, 18909599, 20831538, 19016730, 18250893, 20798891
18743542, 20347562, 16619249, 18354830, 22551446, 19777862, 19687159
21373076, 19174942, 20424899, 21188584, 19989009, 17414008, 20688221
21899588, 20441797, 19157754, 19058490, 19032777, 22815955, 19399918
18885870, 19434529, 21273804, 19018447, 21450666, 18893947, 18851894
16923858, 18417036, 20919320, 19022470, 19284031, 20474192, 20173897
22046677, 22062026, 19501299, 19385656, 20920911, 17274537, 20899461
21315084, 19440586, 16887946, 22374754, 17319928, 19606174, 20708701
18436647, 17655240, 19023822, 19124589, 19178851, 16439813, 19805359
19597583, 18499088, 19155797, 19050649, 19393542

Version 12.1.0.2.v4

Version 12.1.0.2.v4 includes the following:

- Oracle PSU 12.1.0.2.160419 (22291127)
- Timezone file DSTv25 (patch 22037014)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 17969866)
- Adds the ability for the master user to grant the EM_EXPRESS_BASIC and EM_EXPRESS_ALL roles
- Adds the ability for the master user to grant privileges on SYS objects with the grant option using the RDSADMIN.RDSADMIN_UTIL.GRANT_SYS_OBJECT procedure
- Adds master user privileges to support most common schemas created by the Oracle Fusion Middleware Repository Creation Utility (RCU)

Baseline: Oracle database patch set update 12.1.0.2.160419 (patch 22291127, released April 2016)

Bugs fixed:

21847223, 19189525, 19075256, 19141838, 22762046, 20117253, 19865345
19791273, 19280225, 18845653, 19248799, 20951038, 19243521, 21756699
18988834, 21281532, 19238590, 18921743, 20245930, 18799063, 19134173
20373598, 19571367, 20476175, 20925795, 19018206, 20711718, 20387265
20509482, 20588502, 19149990, 18849537, 17551063, 18886413, 19183343
19703301, 21917884, 19001390, 18202441, 19189317, 19644859, 19358317
19390567, 19279273, 19706965, 22528741, 19068970, 20825533, 19619732
18607546, 20348653, 19649152, 19670108, 18940497, 18948177, 19315691
19676905, 18964978, 19035573, 20165574, 19176326, 20413820, 20558005
19176223, 19532017, 20904530, 20134339, 19450314, 22353346, 19074147
18411216, 20361671, 20425790, 18966843, 21329301, 20294666, 19333670
19195895, 19307662, 19371175, 20043616, 19154375, 20124446, 18914624
19468991, 19883092, 19382851, 19520602, 19174521, 21875360, 19676012
19326908, 19658708, 19591608, 20093776, 20618595, 21787056, 17835294
19721304, 19791377, 19068610, 22173980, 20746251, 20048359, 19143550
19185876, 19627012, 20281121, 19577410, 22092979, 19001359, 19518079
18610915, 19490948, 18674024, 18306996, 19309466, 19081128, 19915271
20122715, 21188532, 18791688, 20284155, 20890311, 21442094, 20596234
18973548, 19303936, 19597439, 20936905, 20235511, 19888853, 21756677
18964939, 19354335, 19430401, 19044962, 19639483, 21153266, 22353199
19409212, 20657441, 19879746, 19684504, 19024808, 21260431, 21756661
18799993, 20877664, 19028800, 20879889, 19065556, 19723336, 19077215
19604659, 21421886, 19524384, 18288842, 19048007, 19689979, 20446883
18952989, 16870214, 19928926, 19835133, 21526048, 20466628, 19197175
19180770, 19902195, 20318889, 19730508, 19012119, 19067244, 20074391
20356733, 14643995, 19512341, 19841800, 20331945, 19587324, 19547370
19065677, 21225209, 19637186, 20397490, 18967382, 19174430, 19054077
18674047, 19536415, 19708632, 21091431, 19289642, 22168163, 20869721
19335438, 19258504, 20447445, 17365043, 18856999, 19468347, 20471920
19869255, 21620471, 16359751, 18990693, 17890099, 19769480, 19439759
19272708, 18990023, 19978542, 20402832, 20101006, 21300341, 19329654
19873610, 21744290, 13542050, 21517440, 21668627, 19304354, 19052488
20794034, 19291380, 21915719, 18681056, 20952966, 19896336, 19076343
19561643, 19990037, 18618122, 20440930, 18456643, 19699191, 19487147
18909599, 20831538, 18250893, 19016730, 18743542, 20347562, 16619249
18354830, 19777862, 19687159, 19174942, 20424899, 19989009, 20688221
21899588, 20441797, 19157754, 19032777, 19058490, 19399918, 18885870
19434529, 21273804, 19018447, 18893947, 16923858, 18417036, 20919320
19022470, 19284031, 20474192, 22046677, 20173897, 22062026, 19385656
19501299, 17274537, 20899461, 21315084, 19440586, 22374754, 16887946
19606174, 18436647, 17655240, 19023822, 19178851, 19124589, 16439813
19805359, 19597583, 18499088, 19155797, 19050649, 19393542

Version 12.1.0.2.v3

Version 12.1.0.2.v3 includes the following:

- Oracle PSU 12.1.0.2.160119 (21948354).
- Timezone file DSTv25 (patch 22037014 for 12.1.0.2). 12.1.0.1 includes DSTv24, patch 20875898 (unchanged from 12.1.0.1.v3), because a backport of DSTv25 was unavailable at build time.
- Fixed an issue that prevented customers from creating more than 10 Directory objects in the database.
- Fixed an issue that prevented customers from re-granting read privileges on the ADUMP and BDUMP Directory objects.

Baseline: Oracle database patch set update 12.1.0.2.160119 (patch 21948354, released January 2016)

Bugs fixed:

19189525, 19075256, 19141838, 19865345, 19791273, 19280225, 18845653
20951038, 19243521, 19248799, 21756699, 18988834, 19238590, 21281532
20245930, 18921743, 18799063, 19134173, 19571367, 20476175, 20925795
19018206, 20509482, 20387265, 20588502, 19149990, 18849537, 18886413
17551063, 19183343, 19703301, 19001390, 18202441, 19189317, 19644859
19358317, 19390567, 19279273, 19706965, 19068970, 19619732, 20348653
18607546, 18940497, 19670108, 19649152, 18948177, 19315691, 19676905
18964978, 19035573, 20165574, 19176326, 20413820, 20558005, 19176223
19532017, 20134339, 19074147, 18411216, 20361671, 20425790, 18966843
20294666, 19307662, 19371175, 19195895, 19154375, 19468991, 19174521
19520602, 19382851, 21875360, 19326908, 19658708, 20093776, 20618595
21787056, 17835294, 19791377, 19068610, 20048359, 20746251, 19143550
19185876, 19627012, 20281121, 19577410, 22092979, 19001359, 19518079
18610915, 19490948, 18674024, 18306996, 19309466, 19081128, 19915271
20122715, 21188532, 20284155, 18791688, 20890311, 21442094, 18973548
19303936, 19597439, 20235511, 18964939, 19430401, 19044962, 19409212
19879746, 20657441, 19684504, 19024808, 18799993, 20877664, 19028800
19065556, 19723336, 19077215, 19604659, 21421886, 19524384, 19048007
18288842, 19689979, 20446883, 18952989, 16870214, 19928926, 21526048
19180770, 19197175, 19902195, 20318889, 19730508, 19012119, 19067244
20074391, 19512341, 19841800, 14643995, 20331945, 19587324, 19547370
19065677, 19637186, 21225209, 20397490, 18967382, 19174430, 18674047
19054077, 19536415, 19708632, 19289642, 20869721, 19335438, 17365043
18856999, 19869255, 20471920, 19468347, 21620471, 16359751, 18990693
17890099, 19439759, 19769480, 19272708, 19978542, 20101006, 21300341
20402832, 19329654, 19873610, 21668627, 21517440, 19304354, 19052488
20794034, 19291380, 18681056, 19896336, 19076343, 19561643, 18618122
20440930, 18456643, 19699191, 18909599, 19487147, 18250893, 19016730
18743542, 20347562, 16619249, 18354830, 19687159, 19174942, 20424899
19989009, 20688221, 20441797, 19157754, 19032777, 19058490, 19399918
18885870, 19434529, 19018447, 18417036, 20919320, 19022470, 19284031
20474192, 20173897, 22062026, 19385656, 19501299, 17274537, 20899461
19440586, 16887946, 19606174, 18436647, 17655240, 19023822, 19178851
19124589, 19805359, 19597583, 19155797, 19393542, 19050649

Version 12.1.0.2.v2

Version 12.1.0.2.v2 includes the following:

- Oracle PSU 12.1.0.2.5 (21359755)
- Includes the Daylight Saving Time Patch, patch 20875898: DST-24, that came out after the April 2015 PSU.

Baseline: Oracle database patch set update 12.1.0.2.5 (patch 21359755, released October 2015)

Bugs fixed:

19189525, 19075256, 19865345, 19791273, 19280225, 18845653, 19248799
19243521, 18988834, 19238590, 21281532, 18921743, 20245930, 19134173
19571367, 20476175, 20925795, 19018206, 20387265, 19149990, 18849537
19183343, 19703301, 19001390, 18202441, 19189317, 19644859, 19390567
19358317, 19279273, 19706965, 19068970, 19619732, 18607546, 20348653
18940497, 19670108, 19649152, 18948177, 19315691, 19676905, 18964978
20165574, 19035573, 19176326, 20413820, 20558005, 19176223, 19532017
20134339, 19074147, 18411216, 20361671, 20425790, 18966843, 20294666
19371175, 19307662, 19195895, 19154375, 19468991, 19174521, 19520602
19382851, 19658708, 20093776, 17835294, 19068610, 19791377, 20746251

20048359, 19143550, 19185876, 19627012, 20281121, 19577410, 19001359
19518079, 18610915, 18674024, 18306996, 19309466, 19081128, 19915271
20122715, 20284155, 18791688, 21442094, 19303936, 19597439, 20235511
18964939, 19430401, 19044962, 19409212, 20657441, 19684504, 19024808
19028800, 19065556, 19723336, 19077215, 21421886, 19524384, 19048007
18288842, 18952989, 16870214, 19928926, 19180770, 19197175, 19730508
19012119, 19067244, 20074391, 19841800, 19512341, 14643995, 20331945
19587324, 19065677, 19547370, 19637186, 21225209, 20397490, 18967382
19174430, 18674047, 19054077, 19708632, 19536415, 19289642, 19335438
17365043, 18856999, 20471920, 19468347, 21620471, 16359751, 18990693
19439759, 19769480, 19272708, 19978542, 19329654, 20402832, 19873610
19304354, 19052488, 19291380, 18681056, 19896336, 19076343, 19561643
18618122, 20440930, 18456643, 19699191, 18909599, 19487147, 18250893
19016730, 18743542, 20347562, 16619249, 18354830, 19687159, 19174942
20424899, 19989009, 20688221, 20441797, 19157754, 19058490, 19032777
19399918, 18885870, 19434529, 19018447, 18417036, 20919320, 19284031
19022470, 20474192, 22062026, 19385656, 19501299, 17274537, 20899461
19440586, 19606174, 18436647, 19023822, 19178851, 19124589, 19805359
19597583, 19155797, 19393542, 19050649

Version 12.1.0.2.v1

Version 12.1.0.2.v1 includes the following:

- Oracle PSU 12.1.0.2.3 (patch 20299023)
- The In-Memory option allows storing a subset of data in an in-memory column format optimized for performance.
- Installs additional Oracle Text knowledge bases from Oracle Database Examples media (English and French)
- Provides access to DBMS_REPAIR through RDSADMIN.RDSADMIN_DBMS_REPAIR
- Grants ALTER DATABASE LINK, ALTER PUBLIC DATABASE LINK, EXEMPT ACCESS POLICY, EXEMPT IDENTITY POLICY, and EXEMPT REDACTION POLICY to master user

Note

Version 12.1.0.2.v1 supports Enterprise Edition only.

Baseline: Oracle database patch set update 12.1.0.2.3 (patch 20299023, released April 2015)

Bugs fixed:

19189525, 19065556, 19075256, 19723336, 19077215, 19865345, 18845653
19280225, 19524384, 19248799, 18988834, 19048007, 18288842, 19238590
18921743, 18952989, 16870214, 19928926, 19134173, 19180770, 19018206
19197175, 19149990, 18849537, 19730508, 19183343, 19012119, 19001390
18202441, 19067244, 19189317, 19644859, 19358317, 19390567, 20074391
19279273, 19706965, 19068970, 19841800, 19512341, 14643995, 19619732
20348653, 18607546, 18940497, 19670108, 19649152, 19065677, 19547370
18948177, 19315691, 19637186, 19676905, 18964978, 19035573, 19176326
18967382, 19174430, 19176223, 19532017, 18674047, 19074147, 19054077
19536415, 19708632, 19289642, 20425790, 19335438, 18856999, 19371175
19468347, 19195895, 19154375, 16359751, 18990693, 19439759, 19769480
19272708, 19978542, 19329654, 19873610, 19174521, 19520602, 19382851
19658708, 19304354, 19052488, 19291380, 18681056, 19896336, 17835294
19076343, 19791377, 19068610, 19561643, 18618122, 20440930, 18456643
18909599, 19487147, 19143550, 19185876, 19016730, 18250893, 20347562
19627012, 16619249, 18354830, 19577410, 19687159, 19001359, 19174942
19518079, 18610915, 18674024, 18306996, 19309466, 19081128, 19915271

19157754, 19058490, 20284155, 18791688, 18885870, 19303936, 19434529
19018447, 18417036, 19597439, 20235511, 19022470, 18964939, 19430401
19044962, 19385656, 19501299, 17274537, 19409212, 19440586, 19606174
18436647, 19023822, 19684504, 19178851, 19124589, 19805359, 19024808
19597583, 19155797, 19393542, 19050649, 19028800

Database engine: 11.2.0.4

The following versions are available for database engine 11.2.0.4

- [Version 11.2.0.4.v26 \(p. 1411\)](#)
- [Version 11.2.0.4.v25 \(p. 1414\)](#)
- [Version 11.2.0.4.v24 \(p. 1417\)](#)
- [Version 11.2.0.4.v23 \(p. 1420\)](#)
- [Version 11.2.0.4.v22 \(p. 1422\)](#)
- [Version 11.2.0.4.v21 \(p. 1424\)](#)
- [Version 11.2.0.4.v20 \(p. 1426\)](#)
- [Version 11.2.0.4.v19 \(p. 1428\)](#)
- [Version 11.2.0.4.v18 \(p. 1430\)](#)
- [Version 11.2.0.4.v17 \(p. 1432\)](#)
- [Version 11.2.0.4.v16 \(p. 1433\)](#)
- [Version 11.2.0.4.v15 \(p. 1435\)](#)
- [Version 11.2.0.4.v14 \(p. 1437\)](#)
- [Version 11.2.0.4.v13 \(p. 1438\)](#)
- [Version 11.2.0.4.v12 \(p. 1440\)](#)
- [Version 11.2.0.4.v11 \(p. 1441\)](#)
- [Version 11.2.0.4.v10 \(p. 1443\)](#)
- [Version 11.2.0.4.v9 \(p. 1444\)](#)
- [Version 11.2.0.4.v8 \(p. 1445\)](#)
- [Version 11.2.0.4.v7 \(p. 1447\)](#)
- [Version 11.2.0.4.v6 \(p. 1448\)](#)
- [Version 11.2.0.4.v5 \(p. 1448\)](#)
- [Version 11.2.0.4.v4 \(p. 1449\)](#)
- [Version 11.2.0.4.v3 \(p. 1451\)](#)
- [Version 11.2.0.4.v2 \(deprecated\) \(p. 1451\)](#)
- [Version 11.2.0.4.v1 \(p. 1452\)](#)

Important

RDS for Oracle Database 11g is deprecated. This information is only useful if you want to upgrade an Oracle Database 11g snapshot.

Version 11.2.0.4.v26

Important

This patch is currently only available for Oracle Database Enterprise Edition.

Version 11.2.0.4.v26 includes the following:

- [Patch 31537677: Oracle Database Patch Set Update 11.2.0.4.201020](#)
- [Patch 31668908: Oracle JavaVM Component 11.2.0.4.201020 Database PSU](#)

- Patch 22188219: "L1 VALIDATION" WAIT EVENT USED TO BACK OFF WHEN HW ENQUEUE CANNOT BE ACQUIRED
- Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 32076719: MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.201020 FOR BUGS 2990912 13254780
- Patch 24010393: MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.6 FOR BUGS 12897813 21281961
- Patch 17031322: OCIXMLDBREWRITEXML RETURNED BIND VARIABLES ARE NOT WHITESPACE PRESERVING
- Patch 19277336: INTEGRATED REPLICAT INVALIDATES DEPENDENT PACKAGES RESULTING IN AN ORA-4068
- Patch 19306797: HEARTBEAT REDO IS NOT GENERATED NON-RAC HOSTS WHEN SUPPLEMENTAL LOGGING ENABLED
- Patch 19440386: FAILED TO RAISE ORA-1 FOR PK UPDATE WHEN CONSTRAINT=IMMEDIATE
- Patch 19563715: LOGMINER DOES NOT MAKE PROGRESS WHEN 4GB OR MORE MEMORY IS USED IN GOLDENGATE
- Patch 20647412: EDITION NAME LOGGED WITH KNLD SHOULD BE THE CURRENT EDITION, NOT SESSION EDITION
- Patch 26744595: LGSB:APPLY ABORTS W/ ORA-26786 (ROW-EXISTS) COLLISION WITH HCC(PR)-NO HCC(SB)
- Patch 12668795: ORA-00600: [KDUCFA:ENDBIT] SEEN DURING XSTREAMS ONEWAY REPLICATION
- PreUpgrade Jar: preupgrade_19_cbuild_8_1f.zip
- Support for [Setting and unsetting system diagnostic events \(p. 1046\)](#) using procedures in the rdsadmin.rdsadmin_util package
- Support for the procedure rdsadmin_util.truncate_apply\$_cdr_info described in [Integrated REPLICAT slow due to query on sys."_DBA_APPLY_CDR_INFO" \(p. 1236\)](#)

Combined patches for version 11.2.0.4.v26, released November 2020

Bugs fixed:

```
2990912, 6599380, 8322815, 9756271, 10136473, 11733603, 11786053
11883252, 12364061, 12611721, 12668795, 12747740, 12816846, 12897813
12905058, 12982566, 13254780, 13364795, 13498382, 13558557, 13609098
13645875, 13680635, 13829543, 13837378, 13853126, 13854364, 13866822
13871092, 13936038, 13944971, 13951456, 13955826, 13960236, 14010183
14015842, 14034426, 14054676, 14084247, 14106803, 14133975, 14176370
14245531, 14285317, 14312810, 14338435, 14354737, 14368995, 14458214
14521218, 14521849, 14565184, 14602788, 14657740, 14692762, 14705949
14735792, 14764829, 14774730, 14786201, 14829250, 14852021, 15861775
15913355, 15955387, 15979965, 15990359, 16042673, 16043574, 16065166
16069901, 16091637, 16180763, 16194160, 16198143, 16220077, 16228604
16233738, 16268425, 16285691, 16306373, 16314254, 16315398, 16344544
16354467, 16360112, 16384983, 16392068, 16399083, 16410570, 16422541
16450169, 16472716, 16494615, 16524926, 16538760, 16542886, 16571443
16579084, 16595641, 16596890, 16613964, 16618694, 16668584, 16674686
16685417, 16692232, 16721594, 16731148, 16756406, 16777840, 16785708
16799735, 16819962, 16832076, 16833527, 16833845, 16837842, 16850630
16854386, 16855292, 16863422, 16870214, 16875230, 16875449, 16898135
16901385, 16903536, 16912439, 16929165, 16934803, 16941434, 16943711
16956380, 16989630, 16992075, 17006183, 17006570, 17008068, 17011832
17016369, 17019086, 17019345, 17019356, 17025461, 17027426, 17030189
17031322, 17036973, 17037130, 17040527, 17040764, 17042658, 17047404
17050888, 17056813, 17071721, 17080436, 17082359, 17082983, 17088068
```

17156148, 17165204, 17174582, 17184721, 17186905, 17201047, 17201159
17205719, 17208934, 17215560, 17227073, 17227277, 17231779, 17232014
17235750, 17237521, 17238511, 17239687, 17242746, 17246576, 17254374
17258090, 17258582, 17265217, 17267114, 17274537, 17279227, 17282229
17284817, 17285560, 17288409, 17291347, 17296856, 17297939, 17299889
17302277, 17308789, 17311728, 17313525, 17323222, 17325413, 17332800
17341326, 17343514, 17344412, 17346091, 17346671, 17348614, 17359610
17360606, 17365043, 17375354, 17381384, 17385178, 17389192, 17390160
17390431, 17392698, 17393683, 17393915, 17394950, 17397545, 17432124
17437634, 17441661, 17443671, 17446237, 17449815, 17465741, 17468141
17477958, 17478145, 17478514, 17484731, 17484762, 17495022, 17501491
17518652, 17528315, 17532245, 17532729, 17545847, 17546761, 17546973
17551063, 17551674, 17551699, 17551709, 17570240, 17570606, 17571039
17571306, 17586955, 17587063, 17588480, 17596908, 17600719, 17602269
17610798, 17612828, 17614134, 17614227, 17621643, 17622427, 17630484
17634921, 17643573, 17644091, 17648596, 17649265, 17655240, 17655634
17672719, 17694209, 17695685, 17705023, 17716305, 17717883, 17721717
17722535, 17726838, 17752121, 17752995, 17754782, 17761775, 17762296
17767676, 17783445, 17783588, 17785870, 17786278, 17786518, 17787259
17798953, 17801017, 17804361, 17806696, 17811429, 17811438, 17811447
17811456, 17811789, 17816865, 17820741, 17824637, 17835048, 17835627
17842825, 17847764, 17848897, 17851160, 17852463, 17853456, 17853498
17865671, 17877323, 17883081, 17889549, 17889583, 17890099, 17891943
17891946, 17892268, 17903598, 17912217, 17922254, 17936109, 17945983
17951233, 17957017, 17982555, 17982832, 18000422, 18009564, 18018515
18029658, 18031668, 18043064, 18051556, 18061914, 18084625, 18086801
18090142, 18091059, 18092127, 18093615, 18094246, 18096714, 18098207
18125929, 18135678, 18139690, 18155762, 18159793, 18166013, 18166577
18180390, 18189036, 18191164, 18193833, 18199537, 18202441, 18203835
18203837, 18203838, 18228645, 18230522, 18232865, 18235390, 18244962
18247991, 18259031, 18260550, 18262334, 18264060, 18272672, 18273830
18277454, 18280813, 18282562, 18293054, 18306996, 18308268, 18315328
18316692, 18317531, 18325460, 18328509, 18331812, 18331850, 18334586
18339044, 18356166, 18362222, 18373438, 18382302, 18384391, 18384537
18388363, 18411336, 18413820, 18430495, 18436307, 18436647, 18440047
18440095, 18441944, 18456514, 18458318, 18460587, 18471685, 18482502
18492302, 18508861, 18510194, 18515268, 18522509, 18554763, 18554871
18604493, 18604692, 18607546, 18610915, 18614015, 18619917, 18628388
18641419, 18641451, 18641461, 18662619, 18673304, 18673325, 18673342
18674024, 18674047, 18674465, 18676416, 18681862, 18682983, 18685892
18704244, 18705484, 18723434, 18740837, 18744139, 18747196, 18759211
18762750, 18765602, 18774543, 18783224, 18798250, 18819257, 18828868
18832544, 18841764, 18849970, 18856106, 18856999, 18868646, 18886413
18899974, 18933818, 18948177, 18964939, 18973548, 18973907, 18996843
19006757, 19006849, 19007266, 19013183, 19032777, 19032867, 19049453
19058059, 19060015, 19121551, 19153980, 19174430, 19175543, 19176885
19187988, 19197175, 19207117, 19207156, 19211433, 19211724, 19223010
19231857, 19258504, 19271443, 19272701, 19277336, 19285025, 19289642
19297917, 19306797, 19309466, 19315668, 19330795, 19358317, 19359219
19373893, 19374518, 19393542, 19396455, 19403858, 19429927, 19433930
19440386, 19442102, 19445860, 19455741, 19458377, 19461270, 19463893
19463897, 19466309, 19469538, 19475971, 19487147, 19490948, 19516448
19540573, 19544839, 19554106, 19554117, 19563300, 19563715, 19578350
19584068, 19587324, 19601228, 19601762, 19615136, 19644859, 19680952
19689979, 19692824, 19693090, 19697993, 19699191, 19699946, 19718981
19721304, 19727057, 19730508, 19768226, 19769489, 19777862, 19781326
19788303, 19788842, 19791273, 19794897, 19827973, 19831647, 19835133
19852360, 19854503, 19871910, 19888853, 19891090, 19895326, 19896336
19909862, 19915271, 19930276, 19943771, 19972564, 19972566, 19972568
19972569, 19972570, 20004021, 20004087, 20017509, 20023340, 20031873
20067212, 20074391, 20134113, 20142975, 20144308, 20169408, 20175161
20250147, 20273319, 20294666, 20296213, 20299013, 20299015, 20324049
20331945, 20334344, 20382309, 20387265, 20390564, 20394750, 20408829
20425790, 20441797, 20448824, 20475845, 20476175, 20506699, 20506706
20506715, 20509482, 20513399, 20524085, 20558005, 20563314, 20569094
20596234, 20598042, 20627866, 20631274, 20631846, 20647412, 20657411

20657441, 20671094, 20672075, 20686773, 20717359, 20725343, 20777150
20803583, 20828947, 20856766, 20860659, 20861693, 20869721, 20875898
20879889, 20882568, 20907061, 20914870, 20925795, 20926021, 20936905
21047407, 21047766, 21051833, 21051840, 21051852, 21051858, 21051862
21059919, 21063322, 21067387, 21097043, 21132297, 21142837, 21168487
21172913, 21174504, 21179898, 21197626, 21263635, 21275255, 21281607
21281961, 21286665, 21330264, 21343775, 21343838, 21343897, 21351877
21352646, 21354456, 21380789, 21387964, 21394225, 21419850, 21422580
21424824, 21425496, 21429602, 21453153, 21502702, 21515534, 21516611
21517440, 21526048, 21532755, 21534893, 21538485, 21538558, 21538567
21566639, 21566944, 21612959, 21629064, 21641760, 21656630, 21668627
21698350, 21756661, 21756677, 21756699, 21764119, 21787056, 21794615
21795111, 21811517, 21820934, 21834568, 21842740, 21847223, 21868720
21893235, 21897746, 21911701, 21911849, 21972320, 21983325, 22037014
22083366, 22092979, 22118835, 22148226, 22168163, 22175564, 22185234
22188219, 22195441, 22195448, 22195457, 22195465, 22195477, 22195485
22195492, 22228324, 22243719, 22250006, 22253904, 22296366, 22321741
22321756, 22351572, 22353199, 22380919, 22465352, 22468255, 22499356
22502493, 22507210, 22507234, 22551446, 22568797, 22594718, 22606521
22657942, 22666802, 22670385, 22675136, 22683212, 22683225, 22686674
22730454, 22750215, 22760679, 22782647, 22809871, 22820579, 22826067
22836801, 22873635, 22893153, 22901797, 22905130, 22977256, 23003979
23007241, 23008056, 23026585, 23065323, 23082876, 23105538, 23115139
23140259, 23177648, 23184013, 23194294, 23209741, 23265914, 23266217
23294548, 23302839, 23315889, 23328639, 23330119, 23330124, 23536835
23571055, 23614158, 23628685, 23713236, 23725036, 23727132, 24307571
24316947, 24348685, 24385983, 24411921, 24433711, 24448240, 24473736
24476265, 24476274, 24528741, 24534298, 24555417, 24560906, 24563422
24570598, 24589081, 24624166, 24652769, 24662775, 24701840, 24717859
24719736, 24766121, 24790914, 24817447, 24835538, 24842886, 24908321
24975421, 25042823, 25067795, 25076732, 25077278, 25093656, 25115178
25165496, 25205368, 25248384, 25328093, 25364628, 25369547, 25423453
25427662, 25489607, 25494379, 25505371, 25505382, 25505394, 25505407
25555252, 25600421, 25634317, 25635149, 25649873, 25654936, 25655390
25764020, 25775213, 25809524, 25823754, 25879656, 25879984, 25881255
25885148, 25897615, 25914276, 25947799, 25957038, 26007010, 26023002
26030218, 26039623, 26078387, 26198926, 26203182, 26243698, 26245237
26318200, 26336977, 26354017, 26439748, 26474853, 26482376, 26513067
26544823, 26569225, 26575788, 26631046, 26637592, 26654363, 26667015
26667023, 26667032, 26679352, 26716835, 26744595, 26746894, 26910644
26999139, 27000663, 27015449, 27053456, 27072923, 27086138, 27097854
27255377, 27351628, 27374796, 27404573, 27441326, 27461842, 27534509
27567477, 27642235, 27710072, 27825893, 27870645, 27952577, 28000269
28022101, 28076295, 28079127, 28100487, 28125601, 28199085, 28254374
28305362, 28357401, 28364007, 28384353, 28394726, 28501075, 28502128
28566241, 28612674, 28730253, 28734355, 28790634, 28806384, 28819280
28849751, 28852325, 28855981, 28876684, 28915933, 29027694, 29033139
29200700, 29254615, 29343156, 29434301, 29448234, 29483672, 29483723
29483771, 29511611, 29621961, 29633753, 29774367, 29782211, 29944660
29962927, 29962939, 29965888, 29992392, 29997937, 30018017, 30160639
30179644, 30200680, 30215130, 30237239, 30252098, 30275351, 30275359
30305880, 30365745, 30387666, 30393318, 30421204, 30517516, 30534664
30559616, 30561292, 30562891, 30562907, 30562909, 30562923, 30562936
30578221, 30624864, 30758943, 30772207, 30803210, 30855121, 31001455
31010960, 31022191, 31022281, 31031715, 31125948, 31172207, 31194264
31306274, 31335037, 31335142, 31338362, 31476032, 31492144, 31492164
31492176, 31506720, 31537677, 31668061, 31668867, 31834759, 31883489
31884535, 31885162, 31885173, 31885179, 31885190, 31885201, 31885213
31885223, 31885230

Version 11.2.0.4.v25

Version 11.2.0.4.v25 includes the following:

- Patch 31103343: Database Patch Set Update 11.2.0.4.200714
- Patch 31219953: Oracle JVM Component Database PSU 11.2.0.4.200714
- Patch 31335037: DSTV35 for RDBMS (TZDATA2020A)
- Patch 31335142: DSTV35 for OJVM (TZDATA2020A)
- Patch 31596256: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 22188219: "L1 VALIDATION" WAIT EVENT USED TO BACK OFF WHEN HW ENQUEUE CANNOT BE ACQUIRED
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR

Combined patches for version 11.2.0.4.v25, released July 2020

Bugs fixed:

```
2990912, 6599380, 8322815, 9756271, 10136473, 11733603, 11786053
11883252, 12364061, 12611721, 12668795, 12747740, 12816846, 12897813
12905058, 12982566, 13254780, 13364795, 13498382, 13558557, 13609098
13645875, 13680635, 13829543, 13837378, 13853126, 13866822, 13871092
13936038, 13944971, 13951456, 13955826, 13960236, 14010183, 14015842
14034426, 14054676, 14084247, 14106803, 14133975, 14176370, 14245531
14285317, 14312810, 14338435, 14354737, 14368995, 14458214, 14521218
14521849, 14565184, 14602788, 14657740, 14692762, 14705949, 14735792
14764829, 14774730, 14829250, 14852021, 15861775, 15913355, 15955387
15979965, 15990359, 16042673, 16043574, 16065166, 16069901, 16091637
16180763, 16194160, 16198143, 16220077, 16228604, 16233738, 16268425
16285691, 16306373, 16314254, 16315398, 16344544, 16354467, 16360112
16384983, 16392068, 16399083, 16410570, 16422541, 16450169, 16472716
16494615, 16524926, 16538760, 16542886, 16571443, 16579084, 16595641
16596890, 16613964, 16618694, 16668584, 16674686, 16685417, 16692232
16721594, 16731148, 16756406, 16777840, 16785708, 16799735, 16819962
16832076, 16833527, 16833845, 16837842, 16850630, 16854386, 16855292
16863422, 16870214, 16875230, 16875449, 16898135, 16901385, 16903536
16912439, 16929165, 16934803, 16941434, 16943711, 16956380, 16989630
16992075, 17006183, 17006570, 17008068, 17011832, 17016369, 17019086
17019345, 17019356, 17025461, 17027426, 17030189, 17031322, 17036973
17037130, 17040527, 17040764, 17042658, 17047404, 17050888, 17056813
17071721, 17080436, 17082359, 17082983, 17088068, 17156148, 17165204
17174582, 17184721, 17186905, 17201047, 17201159, 17205719, 17208934
17215560, 17227073, 17227277, 17231779, 17232014, 17235750, 17237521
17238511, 17239687, 17242746, 17246576, 17254374, 17258090, 17258582
17265217, 17267114, 17274537, 17279227, 17282229, 17284817, 17285560
17288409, 17291347, 17296856, 17297939, 17299889, 17302277, 17308789
17311728, 17313525, 17323222, 17325413, 17332800, 17341326, 17343514
17344412, 17346091, 17346671, 17348614, 17359610, 17360606, 17365043
17375354, 17381384, 17385178, 17389192, 17390160, 17390431, 17392698
17393683, 17393915, 17394950, 17397545, 17432124, 17437634, 17441661
17443671, 17446237, 17449815, 17465741, 17468141, 17477958, 17478145
17478514, 17484731, 17484762, 17495022, 17501491, 17518652, 17528315
17532245, 17532729, 17545847, 17546761, 17546973, 17551063, 17551674
17551699, 17551709, 17570240, 17570606, 17571039, 17571306, 17586955
17587063, 17588480, 17596908, 17600719, 17602269, 17610798, 17612828
17614134, 17614227, 17621643, 17622427, 17630484, 17634921, 17643573
17644091, 17648596, 17649265, 17655240, 17655634, 17672719, 17694209
17695685, 17705023, 17716305, 17717883, 17721717, 17722535, 17726838
17752121, 17752995, 17754782, 17761775, 17762296, 17767676, 17783445
17783588, 17785870, 17786278, 17786518, 17787259, 17798953, 17801017
17804361, 17806696, 17811429, 17811438, 17811447, 17811456, 17811789
17816865, 17820741, 17824637, 17835048, 17835627, 17842825, 17847764
17848897, 17851160, 17852463, 17853456, 17853498, 17865671, 17877323
17883081, 17889549, 17889583, 17890099, 17891943, 17891946, 17892268
```

17903598, 17912217, 17922254, 17936109, 17945983, 17951233, 17957017
17982555, 17982832, 18000422, 18009564, 18018515, 18029658, 18031668
18043064, 18051556, 18061914, 18084625, 18086801, 18090142, 18091059
18092127, 18093615, 18094246, 18096714, 18098207, 18125929, 18135678
18139690, 18155762, 18159793, 18166013, 18166577, 18180390, 18189036
18191164, 18193833, 18199537, 18202441, 18203835, 18203837, 18203838
18228645, 18230522, 18232865, 18235390, 18244962, 18247991, 18259031
18260550, 18262334, 18264060, 18272672, 18273830, 18277454, 18280813
18282562, 18293054, 18306996, 18308268, 18315328, 18316692, 18317531
18325460, 18328509, 18331812, 18331850, 18334586, 18339044, 18356166
18362222, 18373438, 18382302, 18384391, 18384537, 18388363, 18411336
18413820, 18430495, 18436307, 18436647, 18440047, 18440095, 18441944
18456514, 18458318, 18460587, 18471685, 18482502, 18492302, 18508861
18510194, 18515268, 18522509, 18554763, 18554871, 18604493, 18604692
18607546, 18610915, 18614015, 18619917, 18628388, 18641419, 18641451
18641461, 18662619, 18673090, 18673304, 18673325, 18673342, 18674024
18674047, 18674465, 18676416, 18681862, 18682983, 18685892, 18704244
18705484, 18723434, 18740837, 18744139, 18747196, 18759211, 18762750
18765602, 18774543, 18783224, 18798250, 18819257, 18828868, 18832544
18841764, 18849970, 18856106, 18856999, 18868646, 18886413, 18899974
18933818, 18948177, 18964939, 18973548, 18973907, 18996843, 19006757
19006849, 19007266, 19013183, 19032777, 19032867, 19049453, 19058059
19060015, 19121551, 19153980, 19174430, 19175543, 19176885, 19187988
19197175, 19207117, 19207156, 19211433, 19211724, 19223010, 19231857
19258504, 19271443, 19272701, 19277336, 19285025, 19289642, 19297917
19306797, 19309466, 19315668, 19330795, 19358317, 19359219, 19373893
19374518, 19393542, 19396455, 19403858, 19429927, 19433930, 19440386
19442102, 19445860, 19455741, 19458377, 19461270, 19463893, 19463897
19466309, 19469538, 19475971, 19487147, 19490948, 19516448, 19540573
19544839, 19554106, 19554117, 19563300, 19563715, 19578350, 19584068
19587324, 19601228, 19601762, 19615136, 19644859, 19680952, 19689979
19692824, 19693090, 19697993, 19699191, 19699946, 19718981, 19721304
19727057, 19730508, 19768226, 19769489, 19777862, 19781326, 19788303
19788842, 19791273, 19794897, 19827973, 19831647, 19835133, 19852360
19854503, 19871910, 19888853, 19891090, 19895326, 19896336, 19909862
19915271, 19930276, 19943771, 19972564, 19972566, 19972568, 19972569
19972570, 20004021, 20004087, 20017509, 20023340, 20031873, 20067212
20074391, 20134113, 20142975, 20144308, 20169408, 20175161, 20250147
20273319, 20294666, 20296213, 20299015, 20324049, 20331945, 20334344
20382309, 20387265, 20390564, 20394750, 20408829, 20425790, 20441797
20448824, 20475845, 20476175, 20506699, 20506706, 20506715, 20509482
20513399, 20524085, 20558005, 20563314, 20569094, 20596234, 20598042
20627866, 20631274, 20631846, 20647412, 20657411, 20657441, 20671094
20672075, 20686773, 20717359, 20725343, 20777150, 20803583, 20828947
20856766, 20860659, 20861693, 20869721, 20875898, 20879889, 20882568
20907061, 20914870, 20925795, 20926021, 20936905, 21047407, 21047766
21051833, 21051840, 21051852, 21051858, 21051862, 21059919, 21063322
21067387, 21097043, 21132297, 21142837, 21168487, 21172913, 21174504
21179898, 21197626, 21227138, 21263635, 21275255, 21281607, 21281961
21286665, 21330264, 21343775, 21343838, 21343897, 21351877, 21352646
21354456, 21380789, 21387964, 21394225, 21419850, 21422580, 21424824
21425496, 21429602, 21453153, 21502702, 21515534, 21516611, 21517440
21526048, 21532755, 21534893, 21538485, 21538558, 21538567, 21566639
21566944, 21612959, 21629064, 21641760, 21656630, 21668627, 21698350
21756661, 21756677, 2175699, 21764119, 21787056, 21794615, 21795111
21811517, 21820934, 21834568, 21842740, 21847223, 21868720, 21893235
21897746, 21911701, 21911849, 21972320, 21983325, 22037014, 22083366
22092979, 22118835, 22148226, 22168163, 22175564, 22185234, 22188219
22195441, 22195448, 22195457, 22195465, 22195477, 22195485, 22195492
22228324, 22243719, 22250006, 22253904, 22296366, 22321741, 22321756
22351572, 22353199, 22380919, 22465352, 22499356, 22502493, 22507210
22507234, 22551446, 22568797, 22594718, 22606521, 22657942, 22666802
22670385, 22675136, 22683212, 22683225, 22686674, 22730454, 22760679
22782647, 22809871, 22820579, 22836801, 22873635, 22893153, 22901797
22905130, 22977256, 23003979, 23007241, 23008056, 23026585, 23065323
23105538, 23115139, 23140259, 23177648, 23194294, 23209741, 23262847

23265914, 23294548, 23302839, 23315889, 23328639, 23330119, 23330124
23536835, 23571055, 23614158, 23628685, 23713236, 23725036, 23727132
24307571, 24316947, 24348685, 24385983, 24411921, 24433711, 24448240
24476265, 24476274, 24528741, 24534298, 24555417, 24560906, 24563422
24570598, 24589081, 24624166, 24652769, 24662775, 24701840, 24717859
24719736, 24766121, 24790914, 24835538, 24842886, 24908321, 24975421
25042823, 25067795, 25076732, 25077278, 25093656, 25165496, 25248384
25328093, 25364628, 25369547, 25423453, 25427662, 25489607, 25494379
25505371, 25505382, 25505394, 25505407, 25555252, 25600421, 25634317
25635149, 25649873, 25654936, 25655390, 25764020, 25775213, 25809524
25823754, 25879656, 25879984, 25881255, 25885148, 25897615, 25914276
25947799, 25957038, 26007010, 26023002, 26030218, 26039623, 26078387
26198926, 26203182, 26243698, 26245237, 26336977, 26354017, 26439748
26474853, 26482376, 26513067, 26544823, 26569225, 26575788, 26631046
26637592, 26654363, 26667015, 26667023, 26667032, 26679352, 26744595
26746894, 26910644, 26999139, 27000663, 27015449, 27053456, 27072923
27086138, 27097854, 27255377, 27351628, 27374796, 27441326, 27461842
27534509, 27567477, 27642235, 27710072, 27825893, 27870645, 27952577
28000269, 28022101, 28076295, 28079127, 28100487, 28125601, 28199085
28254374, 28305362, 28357401, 28364007, 28384353, 28501075, 28502128
28612674, 28730253, 28734355, 28790634, 28806384, 28819280, 28849751
28852325, 28855981, 28876684, 28915933, 29027694, 29033139, 29200700
29254615, 29343156, 29434301, 29448234, 29483672, 29483723, 29483771
29511611, 29621961, 29633753, 29774367, 29944660, 29962927, 29962939
299972392, 29997937, 30160639, 30179644, 30215130, 30237239, 30252098
30275351, 30275359, 30305880, 30365745, 30393318, 30534664, 30559616
30562891, 30562907, 30562909, 30562923, 30562936, 30624864, 30758943
30772207, 30803210, 30855121, 31001455, 31010960, 31022191, 31022281
31031715, 31103343, 31125948, 31172207, 31306274, 31335037, 31335142
31338362, 31492144, 31492164, 31492176

Version 11.2.0.4.v24

Version 11.2.0.4.v24 includes the following:

- Patch 30670774: Database PSU 11.2.0.4.200414
- Patch 30805543: Oracle JVM Component Database PSU 11.2.0.4.200414
- Patch 29997937: DSTV34 for RDBMS (TZDATA2019G)
- Patch 29997959: DSTV34 OJVM (TZDATA2019B)
- Patch 31192454: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 22188219: "L1 VALIDATION" WAIT EVENT USED TO BACK OFF WHEN HW ENQUEUE CANNOT BE ACQUIRED
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR support for Purging the recycle bin (p. 1062).
- Support for Generating performance reports with Automatic Workload Repository (AWR) (p. 1053) using the rdsadmin.rdsadmin_diagnostic_util package

Combined patches for version 11.2.0.4, released April 2020

Bugs fixed:

18619917, 19309466, 28876684, 28855981, 18189036, 19781326, 13609098
16285691, 16756406, 18430495, 17323222, 29483723, 19915271, 19516448
14458214, 23713236, 23140259, 29434301, 22502493, 18272672, 16410570
16494615, 19174430, 21352646, 16901385, 16596890, 22243719, 18996843
21387964, 20334344, 17174582, 22250006, 17798953, 14015842, 18031668

15955387, 21534893, 16832076, 16065166, 16579084, 25427662, 21179898
11786053, 15990359, 24589081, 17982832, 18685892, 20142975, 24835538
16315398, 20861693, 17037130, 17284817, 17891946, 17279227, 17588480
17291347, 16731148, 21097043, 24528741, 22321741, 17165204, 26245237
17891943, 17359610, 17265217, 17465741, 29621961, 22551446, 18191164
16721594, 18614015, 27825893, 18440095, 19769489, 20596234, 18482502
16043574, 17360606, 20936905, 22321756, 19211724, 17392698, 19463893
29033139, 17477958, 17040764, 18362222, 19463897, 24624166, 17853456
14521849, 17816865, 19692824, 21868720, 17951233, 25505407, 17040527
31022191, 24975421, 19888853, 18009564, 20882568, 20803583, 23026585
18604692, 17622427, 16903536, 29483771, 17865671, 17883081, 16228604
17325413, 17082359, 12747740, 22168163, 16091637, 20569094, 17468141
30365745, 29962939, 19469538, 29633753, 20598042, 16042673, 23302839
17437634, 28734355, 19049453, 20387265, 16941434, 16833527, 21343775
17297939, 16069901, 14285317, 22380919, 18436647, 23065323, 21983325
17853498, 24790914, 23571055, 16542886, 21286665, 17365043, 17752995
25914276, 17296856, 18783224, 22353199, 22083366, 28305362, 21419850
16180763, 23294548, 26679352, 13960236, 25328093, 25423453, 18339044
17282229, 25600421, 18856999, 18259031, 28806384, 21354456, 23725036
18471685, 30237239, 17258090, 16344544, 17903598, 17011832, 18135678
18704244, 17786518, 19718981, 25655390, 17242746, 20250147, 19197175
17390431, 17835627, 17672719, 17393915, 21566639, 18765602, 21425496
26544823, 22228324, 29962927, 18682983, 30179644, 25165496, 12816846
18774543, 18747196, 17824637, 19429927, 21429602, 16524926, 17343514
19271443, 17019345, 18681862, 17186905, 23330119, 17811438, 26474853
17215560, 16875449, 21380789, 17184721, 18508861, 19466309, 23330124
17811429, 17019356, 25654936, 17754782, 17752121, 22809871, 17201159
18308268, 19777862, 16198143, 29027694, 18828868, 17586955, 28076295
26654363, 22977256, 16692232, 27374796, 21142837, 20869721, 17649265
25879656, 17847764, 21756699, 19697993, 28364007, 17787259, 23628685
30252098, 23007241, 27351628, 18094246, 20031873, 17375354, 21698350
26513067, 21538567, 22683212, 16450169, 17478145, 17311728, 17648596
17308789, 22836801, 21756677, 18674047, 14084247, 19788303, 22683225
27534509, 16833845, 18948177, 17205719, 21756661, 20004021, 17922254
13837378, 18084625, 17912217, 11883252, 24842886, 12982566, 26203182
14176370, 14764829, 21847223, 16875230, 28079127, 22568797, 17237521
29511611, 25635149, 16934803, 17848897, 20441797, 20175161, 16613964
18334586, 17288409, 17341326, 17449815, 15913355, 16399083, 18740837
20294666, 14565184, 21517440, 17614134, 19854503, 14245531, 16194160
18325460, 15979965, 30562923, 20671094, 27870645, 25093656, 18247991
16912439, 30562936, 24433711, 19930276, 22092979, 20506715, 23003979
20506706, 13871092, 19272701, 17397545, 16785708, 19461270, 21051862
13829543, 16220077, 17008068, 18061914, 20448824, 30275359, 18674024
19689979, 24411921, 30275351, 17596908, 17036973, 22175564, 17612828
20725343, 28199085, 23194294, 17630484, 21051858, 20017509, 21051852
17767676, 17232014, 22893153, 12611721, 25555252, 18356166, 17071721
19315668, 25764020, 16863422, 21051840, 17267114, 17820741, 18043064
21538558, 26243698, 20324049, 30305880, 16392068, 18744139, 24348685
26746894, 18628388, 27072923, 14010183, 16595641, 17080436, 17332800
20777150, 21453153, 20299015, 18413820, 18264060, 16819962, 22465352
21351877, 21051833, 18673342, 30562907, 30562909, 29200700, 27441326
16571443, 18328509, 27567477, 18674465, 16422541, 18306996, 19359219
21424824, 17443671, 17478514, 21067387, 16268425, 17381384, 18723434
17235750, 23328639, 22195448, 24570598, 21172913, 17655240, 18384391
16992075, 22195441, 17025461, 30562891, 16472716, 19289642, 21502702
22195457, 20475845, 22148226, 26030218, 18331850, 17945983, 13498382
24652769, 18673304, 17610798, 19891090, 25369547, 18456514, 8322815
22657942, 17313525, 17050888, 18317531, 19835133, 17495022, 11733603
18798250, 19285025, 18260550, 17390160, 18316692, 19458377, 14368995
17551063, 21343838, 12905058, 14735792, 28612674, 16855292, 23315889
13364795, 18235390, 18293054, 18673325, 19393542, 30215130, 14657740
17532729, 17393683, 17389192, 17783588, 17852463, 19358317, 17441661
14034426, 28254374, 20631274, 19207117, 26569225, 17518652, 24662775
19475971, 18282562, 19896336, 17348614, 19827973, 17346671, 31022281
19791273, 24476274, 22296366, 13853126, 18273830, 17570606, 13558557
26007010, 16685417, 18180390, 14692762, 18159793, 17027426, 24476265

23177648, 17851160, 16870214, 18202441, 17227073, 20657411, 19006849
22606521, 20506699, 28000269, 23536835, 17761775, 20382309, 16306373
17801017, 19680952, 16850630, 17694209, 26667015, 17877323, 18230522
24563422, 17446237, 17889549, 17551674, 16233738, 22730454, 17571039
26667023, 19972570, 18849970, 21532755, 20860659, 22905130, 21168487
17016369, 21263635, 17231779, 21343897, 17717883, 27710072, 18522509
23209741, 17484731, 21972320, 19972569, 19972568, 17716305, 21059919
19972566, 19972564, 26667032, 17394950, 20657441, 17551699, 17006570
18051556, 12364061, 18029658, 17546973, 18262334, 19699191, 17227277
18018515, 16943711, 17982555, 20828947, 18098207, 18436307, 19584068
16898135, 13936038, 19601762, 31010960, 14054676, 25505394, 18228645
19013183, 25042823, 17721717, 17239687, 25248384, 25634317, 20134113
20273319, 28501075, 21063322, 17344412, 22507210, 16354467, 21795111
25505371, 16777840, 25879984, 17811456, 19730508, 17385178, 18166013
17484762, 10136473, 6599380, 20717359, 20296213, 27097854, 13955826
18193833, 17545847, 16837842, 18964939, 19871910, 25505382, 17811447
18554763, 21132297, 25957038, 20004087, 17889583, 19544839, 26631046
22507234, 24719736, 18868646, 17042658, 20627866, 14106803, 13951456
18139690, 18277454, 13680635, 25823754, 18554871, 18515268, 20169408
24908321, 17274537, 17602269, 26575788, 19032867, 17762296, 14829250
16929165, 14602788, 28849751, 21794615, 18899974, 29944660, 18441944
17811789, 20074391, 14852021, 17705023, 13645875, 24316947, 16668584
17786278, 25947799, 20879889, 19578350, 28022101, 22594718, 16384983
26439748, 17957017, 19121551, 17570240, 19788842, 18382302, 27086138
21330264, 21197626, 14338435, 13944971, 21656630, 18886413, 17156148
17936109, 20509482, 27255377, 24717859, 18762750, 21526048, 24560906
18096714, 17238511, 26078387, 27053456, 20144308, 25364628, 18244962
19433930, 20476175, 19297917, 21174504, 18280813, 28819280, 17614227
28357401, 21911701, 17006183, 25809524, 18092127, 19727057, 17695685
26039623, 22820579, 20856766, 15861775, 17258582, 21668627, 19487147
20925795, 28100487, 26482376, 19554106, 22760679, 21629064, 18199537
18091059, 17299889, 21538485, 17546761, 26336977, 25775213, 18155762
30803210, 16956380, 19207156, 14705949, 23105538, 26198926, 19258504
16314254, 17246576, 17655634, 17890099, 16989630, 20067212, 19721304
25077278, 19490948, 18203835, 18203838, 18973907, 18203837, 29483672
19615136, 17587063, 18000422, 18641451, 18090142, 21641760, 17019086
30559616, 19373893, 18373438, 21820934, 18641461, 17346091, 21422580
22351572, 18604493, 23008056, 22901797, 18610915, 17892268, 17501491
20907061, 14354737, 17835048, 21787056, 22195485, 22782647, 17082983
18641419, 16618694, 14133975, 22195492, 18331812, 18093615, 24385983
25897615, 20513399, 21281607, 13866822, 18841764, 17600719, 17842825
20558005, 17088068, 9756271, 22195465, 18440047, 19211433, 21515534
20331945, 22686674, 18384537, 18607546, 17254374, 18315328, 23115139
28790634, 21394225, 16360112, 22195477, 17726838, 18510194, 17571306
24766121, 17302277, 21842740, 17551709, 26910644, 17634921, 25489607
16538760, 18933818, 19176885, 17201047, 25649873, 25067795, 28502128
27952577, 14774730, 27461842, 19153980, 21911849, 23727132, 18166577
27000663, 24448240, 17056813, 21811517, 19909862, 25494379, 22675136
24534298, 19895326, 22253904, 17804361, 19231857, 27642235, 26023002
17528315, 19058059, 30534664, 29992392, 19554117, 19007266, 28915933
30855121, 30160639, 17285560, 29254615, 22670385, 18458318, 19187988
23265914, 19699946, 19006757, 19374518, 29774367, 19223010, 29448234
25076732, 22118835, 26637592, 19852360, 20408829, 21047766, 21566944
28730253, 16799735, 17432124, 18759211, 19396455, 20875898, 22037014
22873635, 23614158, 24701840, 25881255, 27015449, 28125601, 28852325
29997937, 29997959, 26354017, 21893235, 18125929, 19601228, 12668795
20524085, 19403858, 18086801, 16854386, 21612959, 20563314, 18705484
16674686, 20425790, 20390564, 19032777, 21275255, 17783445, 19943771
18973548, 18411336, 18673090, 19768226, 21047407, 17030189, 18856106
19693090, 21227138, 18492302, 19831647, 12897813, 19563300, 20914870
19587324, 13254780, 18676416, 21834568, 19794897, 26744595, 17208934
17031322, 19060015, 19277336, 19455741, 21764119, 17785870, 20631846
22185234, 20023340, 20647412, 19440386, 21281961, 25885148, 17722535
20926021, 20686773, 17621643, 18662619, 19563715, 19442102, 21516611
14312810, 20672075, 21897746, 2990912, 23262847, 19644859, 19175543
17644091, 20394750, 19306797, 18819257, 22188219

Version 11.2.0.4.v23

Version 11.2.0.4.v23 includes the following:

- Patch 30298532: Database Patch Set Update: 11.2.0.4.200114
- Patch 30503372: OJVM PATCH SET UPDATE 11.2.0.4.200114
- Patch 29997937: DSTv34 for RDBMS (TZDATA2019G)
- Patch 29997959: DSTV34 OJVM (TZDATA2019B)
- Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 19440386: FAILED TO RAISE ORA-1 FOR PK UPDATE WHEN CONSTRAINT=IMMEDIATE
- Patch 19277336: INTEGRATED REPLICAT INVALIDATES DEPENDENT PACKAGES RESULTING IN AN ORA-4068
- Patch 24286409: MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.6 FOR BUGS 20647412 21534893
- Patch 24010393: MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.6 FOR BUGS 12897813 21281961
- Patch 19306797: HEARTBEAT REDO IS NOT GENERATED NON-RAC HOSTS WHEN SUPPLIMENTAL LOGGING ENABLED
- Patch 19563715: LOGMINER DOES NOT MAKE PROGRESS WHEN 4GB OR MORE MEMORY IS USED IN GOLDENATE
- Patch 20425790: LOGMINER PATCHES SHOULD TRANSPARENTLY FUNCTION IN A NON-PARTITIONING ENABLED DB
- Patch 17031322: 46719: OCIXMLDBREWRITEXML RETURNED BIND VARIABLES ARE NOT WHITESPACE PRESERVING
- Patch 30303921: MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.190416 FOR BUGS 29879564 14312810
- Patch 30293609: MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.190416 FOR BUGS 29600521 23262847
- Patch 26744595: LGSB:APPLY ABORTS W/ ORA-26786 (ROW-EXISTS) COLLISION WITH HCC(PR)-NO HCC(SB)
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR

Oracle patch 30298532, released January 2020

Bugs fixed:

18619917, 19309466, 28876684, 28855981, 18189036, 19781326, 13609098
16285691, 16756406, 18430495, 17323222, 29483723, 19915271, 19516448
14458214, 23713236, 23140259, 29434301, 22502493, 18272672, 16410570
16494615, 19174430, 21352646, 16901385, 16596890, 22243719, 18996843
21387964, 20334344, 17174582, 22250006, 17798953, 14015842, 18031668
15955387, 16832076, 16065166, 16579084, 25427662, 21179898, 11786053
15990359, 17982832, 18685892, 20142975, 24835538, 16315398, 20861693
17037130, 17284817, 17891946, 17279227, 17588480, 17291347, 16731148
21097043, 24528741, 22321741, 17165204, 26245237, 17891943, 17359610
17265217, 17465741, 22551446, 18191164, 16721594, 18614015, 27825893
18440095, 19769489, 20596234, 18482502, 16043574, 17360606, 22321756
19211724, 17392698, 19463893, 29033139, 17477958, 17040764, 18362222
19463897, 24624166, 17853456, 14521849, 17816865, 19692824, 21868720
17951233, 25505407, 17040527, 24975421, 19888853, 18009564, 20882568

20803583, 23026585, 18604692, 17622427, 16903536, 17865671, 29483771
17883081, 16228604, 17325413, 17082359, 12747740, 22168163, 16091637
20569094, 17468141, 30365745, 29962939, 19469538, 29633753, 20598042
16042673, 17437634, 23302839, 28734355, 19049453, 20387265, 16833527
21343775, 17297939, 16069901, 14285317, 22380919, 18436647, 23065323
21983325, 17853498, 24790914, 23571055, 16542886, 21286665, 17365043
17752995, 25914276, 17296856, 18783224, 22353199, 22083366, 28305362
16180763, 21419850, 23294548, 26679352, 13960236, 25423453, 18339044
17282229, 25600421, 18856999, 18259031, 28806384, 21354456, 23725036
18471685, 30237239, 17258090, 16344544, 17903598, 17011832, 18135678
18704244, 17786518, 19718981, 25655390, 17242746, 20250147, 19197175
17390431, 17835627, 17672719, 17393915, 21566639, 18765602, 21425496
26544823, 22228324, 29962927, 18682983, 25165496, 12816846, 18774543
18747196, 17824637, 19429927, 21429602, 16524926, 17343514, 19271443
17019345, 18681862, 17186905, 23330119, 17811438, 26474853, 17215560
16875449, 21380789, 17184721, 18508861, 19466309, 23330124, 17811429
17019356, 25654936, 17754782, 17752121, 22809871, 17201159, 18308268
19777862, 16198143, 29027694, 18828868, 17586955, 28076295, 22977256
16692232, 27374796, 21142837, 20869721, 17649265, 25879656, 21756699
19697993, 28364007, 17787259, 23628685, 23007241, 30252098, 27351628
18094246, 20031873, 17375354, 21698350, 21538567, 22683212, 16450169
17478145, 17311728, 17648596, 17308789, 22836801, 21756677, 18674047
14084247, 19788303, 22683225, 27534509, 16833845, 18948177, 17205719
21756661, 20004021, 17922254, 13837378, 18084625, 17912217, 11883252
24842886, 12982566, 26203182, 14176370, 14764829, 21847223, 16875230
28079127, 22568797, 17237521, 29511611, 25635149, 16934803, 17848897
20441797, 16613964, 18334586, 17288409, 17341326, 17449815, 15913355
16399083, 18740837, 20294666, 14565184, 21517440, 17614134, 19854503
14245531, 16194160, 18325460, 15979965, 30562923, 20671094, 27870645
25093656, 18247991, 16912439, 24433711, 19930276, 22092979, 20506715
23003979, 20506706, 13871092, 19272701, 17397545, 16785708, 19461270
21051862, 13829543, 16220077, 17008068, 18061914, 20448824, 30275359
18674024, 19689979, 24411921, 30275351, 17596908, 17036973, 17612828
20725343, 28199085, 23194294, 17630484, 21051858, 20017509, 21051852
17767676, 17232014, 22893153, 12611721, 25555252, 18356166, 17071721
25764020, 16863422, 21051840, 17267114, 18043064, 21538558, 26243698
20324049, 16392068, 18744139, 24348685, 26746894, 27072923, 14010183
16595641, 17080436, 17332800, 20777150, 21453153, 20299015, 18413820
18264060, 16819962, 22465352, 21351877, 21051833, 18673342, 30562907
30562909, 29200700, 27441326, 16571443, 18328509, 27567477, 18674465
16422541, 18306996, 17443671, 19359219, 21424824, 17478514, 21067387
16268425, 17381384, 18723434, 17235750, 23328639, 22195448, 24570598
21172913, 17655240, 18384391, 16992075, 22195441, 17025461, 30562891
16472716, 19289642, 21502702, 22195457, 20475845, 22148226, 26030218
18331850, 17945983, 13498382, 24652769, 18673304, 17610798, 19891090
25369547, 18456514, 8322815, 22657942, 17313525, 17050888, 18317531
17495022, 11733603, 18798250, 19285025, 18260550, 17390160, 18316692
19458377, 14368995, 17551063, 21343838, 12905058, 14735792, 28612674
16855292, 23315889, 13364795, 18235390, 18293054, 18673325, 19393542
14657740, 17393683, 17389192, 17783588, 17852463, 19358317, 17441661
14034426, 28254374, 20631274, 19207117, 26569225, 17518652, 24662775
19475971, 18282562, 17348614, 19827973, 17346671, 24476274, 22296366
13853126, 18273830, 17570606, 13558557, 26007010, 16685417, 18180390
14692762, 17027426, 18159793, 24476265, 23177648, 17851160, 16870214
18202441, 17227073, 20657411, 19006849, 20506699, 22606521, 28000269
23536835, 17761775, 20382309, 16306373, 19680952, 16850630, 17694209
26667015, 17877323, 18230522, 24563422, 17446237, 17889549, 17551674
16233738, 22730454, 17571039, 26667023, 19972570, 18849970, 21532755
20860659, 22905130, 21168487, 17016369, 17231779, 21263635, 21343897
27710072, 18522509, 23209741, 17484731, 21972320, 19972569, 19972568
17716305, 21059919, 19972566, 19972564, 26667032, 17394950, 20657441
17551699, 17006570, 18051556, 12364061, 18029658, 17546973, 18262334
19699191, 17227277, 18018515, 16943711, 17982555, 20828947, 18098207
18436307, 19584068, 16898135, 13936038, 19601762, 14054676, 25505394
18228645, 19013183, 25042823, 17721717, 17239687, 25248384, 25634317
20134113, 20273319, 28501075, 21063322, 17344412, 22507210, 16354467

21795111, 25505371, 16777840, 25879984, 17811456, 19730508, 17385178
18166013, 17484762, 10136473, 6599380, 20717359, 20296213, 27097854
13955826, 18193833, 17545847, 16837842, 18964939, 19871910, 25505382
17811447, 18554763, 21132297, 25957038, 20004087, 17889583, 19544839
26631046, 22507234, 24719736, 18868646, 17042658, 20627866, 14106803
13951456, 18139690, 18277454, 13680635, 25823754, 18554871, 20169408
18515268, 24908321, 17274537, 17602269, 26575788, 19032867, 17762296
14829250, 16929165, 14602788, 28849751, 21794615, 18899974, 18441944
29944660, 17811789, 20074391, 14852021, 17705023, 13645875, 24316947
16668584, 17786278, 25947799, 20879889, 19578350, 28022101, 22594718
16384983, 26439748, 17957017, 19121551, 17570240, 19788842, 18382302
27086138, 21330264, 21197626, 14338435, 13944971, 21656630, 18886413
17156148, 17936109, 20509482, 27255377, 24717859, 18762750, 21526048
24560906, 18096714, 17238511, 26078387, 27053456, 20144308, 18244962
19433930, 20476175, 19297917, 21174504, 18280813, 28819280, 17614227
28357401, 17006183, 18092127, 19727057, 17695685, 26039623, 22820579
20856766, 15861775, 17258582, 21668627, 19487147, 20925795, 28100487
26482376, 19554106, 21629064, 18199537, 18091059, 17299889, 21538485
17546761, 25775213, 26336977, 18155762, 16956380, 19207156, 14705949
23105538, 26198926, 19258504, 16314254, 17246576, 17655634, 16989630
20067212, 19721304, 19490948, 25077278, 18203835, 18203838, 18973907
18203837, 29483672, 19615136, 17587063, 18000422, 18641451, 18090142
21641760, 17019086, 30559616, 19373893, 18373438, 18641461, 17346091
21422580, 22351572, 18604493, 23008056, 22901797, 18610915, 17892268
17501491, 20907061, 14354737, 17835048, 21787056, 22195485, 22782647
17082983, 18641419, 16618694, 14133975, 22195492, 18331812, 18093615
24385983, 25897615, 20513399, 21281607, 13866822, 18841764, 17600719
17842825, 20558005, 17088068, 9756271, 22195465, 18440047, 19211433
21515534, 20331945, 22686674, 18384537, 18607546, 17254374, 18315328
23115139, 28790634, 21394225, 16360112, 22195477, 17726838, 18510194
17571306, 17302277, 24766121, 21842740, 17551709, 26910644, 17634921
25489607, 16538760

Version 11.2.0.4.v22

Version 11.2.0.4.v22 includes the following:

- Patch 29913194: DATABASE PATCH SET UPDATE 11.2.0.4.191015
- Patch 30132974: OJVM PATCH SET UPDATE 11.2.0.4.191015
- Patch 29997937: DSTv34 for RDBMS (TZDATA2019G)
- Patch 29997959: DSTV34 OJVM (TZDATA2019B)
- Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 19440386: FAILED TO RAISE ORA-1 FOR PK UPDATE WHEN CONSTRAINT=IMMEDIATE
- Patch 19277336: INTEGRATED REPLICAT INVALIDATES DEPENDENT PACKAGES RESULTING IN AN ORA-4068
- Patch 24286409: MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.6 FOR BUGS 20647412 21534893
- Patch 24010393: MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.6 FOR BUGS 12897813 21281961
- Patch 19306797: HEARTBEAT REDO IS NOT GENERATED NON-RAC HOSTS WHEN SUPPLEMENTAL LOGGING ENABLED
- Patch 19563715: LOGMINER DOES NOT MAKE PROGRESS WHEN 4GB OR MORE MEMORY IS USED IN GOLDENGATE
- Patch 20425790: LOGMINER PATCHES SHOULD TRANSPARENTLY FUNCTION IN A NON-PARITIONING ENABLED DB
- Patch 17031322: 46719: OCIXMLDBREWRITEXML RETURNED BIND VARIABLES ARE NOT WHITESPACE PRESERVING

- Patch 30303921: MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.190416 FOR BUGS 29879564 14312810
- Patch 30293609: MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.190416 FOR BUGS 29600521 23262847
- Patch 26744595: LGSB:APPLY ABORTS W/ ORA-26786 (ROW-EXISTS) COLLISION WITH HCC(PR)-NO HCC(SB)
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR

Oracle patch 29913194, released October 2019

Bugs fixed:

```
17184721, 21174504, 20169408, 21538558, 16091637, 18092127, 17381384
15979965, 20671094, 16731148, 16314254, 18441944, 13837378, 17835048
17291347, 23105538, 28254374, 13558557, 21656630, 21842740, 17008068
18382302, 17201159, 17853498, 25427662, 21197626, 17246576, 20717359
18356166, 18681862, 18440047, 20569094, 20031873, 16875449, 20387265
19788842, 17296856, 21330264, 14010183, 17648596, 17025461, 18886413
17551063, 24719736, 17258582, 17267114, 21063322, 22507210, 17912217
17889583, 18202441, 17040764, 17478145, 16524926, 25655390, 19358317
22730454, 22148226, 18747196, 26544823, 18641419, 17036973, 18948177
17811789, 16542886, 14285317, 18009564, 17359610, 16618694, 8322815
16832076, 18247991, 16692232, 22507234, 28022101, 17570240, 13871092
24624166, 26631046, 24348685, 19429927, 17848897, 17441661, 14034426
17465741, 20273319, 19207156, 16596890, 17437634, 20506706, 18510194
21343897, 28849751, 18339044, 21453153, 17951233, 21795111, 22321741
18430495, 21787056, 22380919, 20506715, 19692824, 19469538, 17811429
17903598, 19721304, 11786053, 29511611, 18230522, 19554106, 19458377
21281607, 17612828, 6599380, 18029658, 22092979, 19516448, 17040527
22321756, 17811438, 18641461, 18682983, 14657740, 25635149, 21502702
13364795, 19490948, 21387964, 17346671, 17588480, 22351572, 18235390
26474853, 18849970, 17982832, 17889549, 19309466, 16472716, 23008056
20627866, 24908321, 20134113, 25775213, 20596234, 18331850, 18641451
17019356, 20882568, 17344412, 19461270, 21179898, 17546761, 24842886
17231779, 14521849, 18203835, 18203838, 18964939, 18203837, 17313525
22195457, 18139690, 16837842, 14106803, 22296366, 17842825, 22657942
21352646, 16360112, 22594718, 20657441, 22195441, 17389192, 26198926
14565184, 19781326, 17019345, 17205719, 18740837, 18440095, 14764829
14354737, 22195448, 17019086, 13944971, 16571443, 21868720, 17186905
17080436, 18673342, 28501075, 22905130, 17027426, 19972569, 27374796
19972568, 16833845, 19972566, 20144308, 17282229, 19972564, 16870214
16410570, 21629064, 19615136, 21354456, 26039623, 19871910, 17390431
18762750, 23007241, 25248384, 16613964, 18098207, 17957017, 17484762
18471685, 19730508, 18264060, 21538485, 17323222, 17754782, 17600719
18317531, 17852463, 17596908, 17655634, 18166013, 16228604, 20074391
27053456, 24790914, 19972570, 26482376, 20856766, 18090142, 19891090
18996843, 16042673, 19854503, 17835627, 22901797, 20334344, 17393683
20861693, 18000422, 17551709, 26575788, 23315889, 20506699, 19006849
18277454, 18456514, 17258090, 19174430, 20657411, 17174582, 25654936
17242746, 27097854, 16399083, 17824637, 21132297, 17762296, 22465352
22168163, 28612674, 18604692, 17397545, 16450169, 12364061, 20067212
19373893, 18856999, 19211724, 19463893, 19463897, 27351628, 21343775
17853456, 18373438, 18673304, 20004021, 28000269, 26030218, 21668627
16194160, 17477958, 23140259, 16538760, 12982566, 24570598, 20828947
27255377, 18259031, 20296213, 21425496, 28855981, 18293054, 17610798
19699191, 23065323, 17311728, 18135678, 18774543, 23294548, 16785708
10136473, 22551446, 19777862, 24560906, 17786518, 18315328, 25879984
18334586, 12747740, 22250006, 18096714, 19032867, 21641760, 17390160
18899974, 17232014, 20598042, 16354467, 26245237, 26679352, 17484731
18673325, 16422541, 18155762, 19827973, 14015842, 22683225, 17726838
18554871, 23177648, 18051556, 20803583, 18282562, 17922254, 15990359
```

21972320, 16855292, 16668584, 21343838, 20299015, 29483672, 17446237
18043064, 18093615, 17694209, 23713236, 17288409, 18308268, 20475845
17274537, 13955826, 16934803, 18841764, 17634921, 17501491, 16315398
23725036, 22683212, 17006183, 13829543, 18191164, 26746894, 22809871
17655240, 28819280, 19393542, 18384391, 29633753, 21538567, 17695685
16198143, 21847223, 28199085, 25823754, 17892268, 20142975, 19584068
17165204, 25165496, 27072923, 18604493, 18508861, 21756699, 18554763
16901385, 21532755, 18189036, 17443671, 17385178, 14829250, 17936109
20476175, 20925795, 20509482, 17478514, 27441326, 16850630, 13951456
16595641, 14054676, 15861775, 21142837, 16912439, 17299889, 17297939
23003979, 16833527, 18619917, 17798953, 17630484, 19697993, 17816865
25914276, 18607546, 17571306, 21286665, 16898135, 17341326, 16819962
26910644, 17851160, 17586955, 20558005, 19049453, 21051840, 17587063
16956380, 18328509, 25042823, 14735792, 25423453, 14133975, 29033139
19718981, 18061914, 18522509, 16233738, 17518652, 21051833, 18765602
20294666, 23194294, 20860659, 18272672, 20324049, 18199537, 17332800
13609098, 22502493, 18384537, 14338435, 17945983, 27710072, 16392068
21067387, 17752995, 21097043, 21051862, 16863422, 17237521, 25505382
29483723, 18244962, 19544839, 28357401, 19433930, 24433711, 24717859
17156148, 18973907, 23026585, 17449815, 17877323, 18180390, 17088068
17037130, 20004087, 21422580, 19466309, 11733603, 25505371, 18610915
21051858, 18084625, 29027694, 18674024, 26243698, 21051852, 18091059
18306996, 16306373, 25369547, 19930276, 17787259, 19915271, 18193833
20631274, 20513399, 16344544, 26439748, 25879656, 14692762, 18614015
22782647, 17346091, 18413820, 19297917, 18228645, 17721717, 13960236
18685892, 18436307, 11883252, 19888853, 21756677, 17891943, 19475971
22353199, 16384983, 19121551, 27825893, 25634317, 12816846, 17982555
17761775, 17227073, 13936038, 22243719, 17265217, 25505394, 17071721
16721594, 18262334, 21756661, 17891946, 15913355, 17672719, 17602269
17239687, 17042658, 25555252, 17238511, 21059919, 17811456, 17284817
17752121, 20879889, 28806384, 21380789, 19601762, 17394950, 17011832
28305362, 16579084, 22195465, 16875230, 14602788, 28790634, 18325460
27567477, 30275351, 26569225, 24476265, 24476274, 12611721, 18674465
16903536, 17006570, 19689979, 28076295, 16043574, 18783224, 22836801
14705949, 24662775, 16494615, 21526048, 17392698, 19197175, 16069901
17811447, 29200700, 27870645, 28876684, 17308789, 24835538, 22195477
17865671, 17343514, 19013183, 17325413, 18316692, 16180763, 30275359
17348614, 14368995, 21983325, 17393915, 16285691, 19788303, 19211433
20331945, 17883081, 17705023, 24316947, 17614227, 23571055, 19578350
22195485, 14084247, 24975421, 26078387, 23115139, 13645875, 23328639
16777840, 19727057, 21698350, 14852021, 18744139, 18674047, 17716305
19285025, 18482502, 17622427, 19289642, 27534509, 25947799, 22195492
14458214, 20869721, 21172913, 17767676, 18723434, 25505407, 17786278
19258504, 17082983, 17365043, 21351877, 13498382, 18331812, 16065166
25489607, 16685417, 21566639, 18031668, 22893153, 17551674, 16943711
19272701, 21517440, 25897615, 17649265, 13866822, 18094246, 24528741
17783588, 14245531, 17082359, 18280813, 26007010, 20448824, 23330119
16268425, 19487147, 25600421, 18018515, 17302277, 17215560, 24411921
19271443, 25764020, 14176370, 17016369, 20777150, 16756406, 23330124
22977256, 20441797, 19769489, 28100487, 17545847, 25093656, 18260550
13853126, 17227277, 17551699, 23536835, 30237239, 25957038, 24652769
20725343, 19207117, 9756271, 17495022, 18868646, 17614134, 26667023
17546973, 19680952, 18704244, 26667015, 17050888, 18273830, 18828868
17360606, 16992075, 24563422, 17375354, 12905058, 18362222, 21429602
17254374, 27086138, 28364007, 26667032, 17571039, 17468141, 18436647
17570606, 17235750, 21168487, 17279227, 16220077, 16929165

Version 11.2.0.4.v21

Version 11.2.0.4.v21 includes the following:

- Patch 29497421: DATABASE PATCH SET UPDATE 11.2.0.4.190716

- Patch 29610422: OJVM PATCH SET UPDATE 11.2.0.4.190716
- Patch 28852325: DSTv33 for RDBMS (TZDATA2018G)
- Patch 28852334: DSTv33 for OJVM (TZDATA2018G)
- Patch 30018733: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR

Oracle patch 29497421, released July 2019

Bugs fixed:

```
17184721, 21174504, 20169408, 21538558, 16091637, 18092127, 17381384
15979965, 20671094, 16731148, 16314254, 18441944, 13837378, 17835048
23105538, 17291347, 28254374, 13558557, 21842740, 21656630, 17008068
18382302, 17201159, 25427662, 17853498, 21197626, 20717359, 17246576
18356166, 18681862, 18440047, 20569094, 20031873, 16875449, 20387265
19788842, 17296856, 21330264, 14010183, 17648596, 17025461, 18886413
17551063, 17258582, 24719736, 17267114, 21063322, 22507210, 17912217
17889583, 18202441, 17040764, 17478145, 16524926, 25655390, 19358317
22730454, 22148226, 18747196, 26544823, 18641419, 17036973, 18948177
17811789, 16542886, 14285317, 18009564, 17359610, 16618694, 8322815
16832076, 18247991, 16692232, 22507234, 28022101, 17570240, 13871092
24624166, 26631046, 24348685, 19429927, 17848897, 17441661, 14034426
17465741, 20273319, 19207156, 16596890, 17437634, 18510194, 21343897
20506706, 28849751, 18339044, 21453153, 17951233, 21795111, 22321741
18430495, 21787056, 22380919, 19692824, 19469538, 20506715, 17811429
17903598, 19721304, 11786053, 29511611, 18230522, 19554106, 19458377
21281607, 17612828, 6599380, 18029658, 22092979, 17040527, 22321756
17811438, 18641461, 18682983, 21502702, 14657740, 25635149, 13364795
19490948, 21387964, 17346671, 17588480, 22351572, 18235390, 18849970
26474853, 17889549, 19309466, 16472716, 23008056, 20627866, 24908321
20134113, 25775213, 20596234, 18331850, 18641451, 20882568, 17019356
17344412, 19461270, 21179898, 17546761, 24842886, 17231779, 14521849
18203835, 18203838, 18964939, 18203837, 17313525, 22195457, 18139690
16837842, 14106803, 22296366, 17842825, 22657942, 21352646, 16360112
22594718, 20657441, 22195441, 17389192, 26198926, 14565184, 17019345
19781326, 17205719, 18740837, 18440095, 14764829, 14354737, 22195448
17019086, 13944971, 16571443, 21868720, 17186905, 17080436, 18673342
28501075, 22905130, 17027426, 19972569, 27374796, 19972568, 16833845
19972566, 20144308, 17282229, 19972564, 16870214, 16410570, 21629064
19615136, 19871910, 21354456, 26039623, 17390431, 18762750, 23007241
25248384, 16613964, 18098207, 17957017, 17484762, 18471685, 19730508
18264060, 21538485, 17323222, 17754782, 17600719, 18317531, 17852463
17596908, 17655634, 18166013, 16228604, 20074391, 27053456, 24790914
19972570, 26482376, 20856766, 18090142, 19891090, 18996843, 16042673
19854503, 17835627, 22901797, 20334344, 17393683, 20861693, 18000422
17551709, 26575788, 23315889, 20506699, 19006849, 18277454, 18456514
17258090, 19174430, 20657411, 17174582, 25654936, 17242746, 27097854
16399083, 17824637, 21132297, 22465352, 17762296, 22168163, 28612674
18604692, 17397545, 16450169, 12364061, 20067212, 18856999, 19211724
19463893, 19463897, 27351628, 21343775, 17853456, 18373438, 18673304
20004021, 26030218, 28000269, 21668627, 16194160, 17477958, 23140259
16538760, 12982566, 24570598, 20828947, 27255377, 18259031, 20296213
28855981, 21425496, 18293054, 17610798, 19699191, 23065323, 17311728
18135678, 18774543, 23294548, 16785708, 10136473, 19777862, 22551446
24560906, 17786518, 18315328, 25879984, 18334586, 22250006, 12747740
18096714, 19032867, 21641760, 17390160, 18899974, 17232014, 20598042
26679352, 26245237, 16354467, 17484731, 18673325, 16422541, 18155762
19827973, 14015842, 22683225, 17726838, 18554871, 23177648, 18051556
20803583, 18282562, 17922254, 15990359, 21972320, 16855292, 16668584
21343838, 20299015, 17446237, 18043064, 18093615, 23713236, 17694209
```

17288409, 20475845, 18308268, 17274537, 13955826, 16934803, 17634921
17501491, 16315398, 23725036, 22683212, 17006183, 13829543, 18191164
26746894, 22809871, 17655240, 28819280, 18384391, 19393542, 21538567
17695685, 16198143, 21847223, 25823754, 17892268, 20142975, 19584068
17165204, 25165496, 18604493, 18508861, 21756699, 18554763, 16901385
21532755, 18189036, 17443671, 17385178, 14829250, 17936109, 20925795
20509482, 17478514, 27441326, 16850630, 13951456, 16595641, 14054676
15861775, 21142837, 16912439, 17299889, 17297939, 23003979, 16833527
18619917, 17798953, 19697993, 17816865, 25914276, 18607546, 17571306
21286665, 16898135, 17341326, 16819962, 26910644, 17851160, 17586955
20558005, 19049453, 21051840, 17587063, 16956380, 25042823, 14735792
18328509, 25423453, 14133975, 29033139, 19718981, 18061914, 16233738
18522509, 17518652, 21051833, 20294666, 18765602, 23194294, 20860659
18272672, 20324049, 18199537, 17332800, 13609098, 22502493, 18384537
14338435, 27710072, 17945983, 16392068, 21067387, 17752995, 21097043
21051862, 16863422, 17237521, 25505382, 18244962, 28357401, 19544839
19433930, 24433711, 24717859, 17156148, 18973907, 23026585, 17449815
17877323, 18180390, 17088068, 17037130, 20004087, 21422580, 19466309
11733603, 25505371, 18610915, 21051858, 18084625, 29027694, 18674024
26243698, 21051852, 18091059, 18306996, 25369547, 16306373, 19930276
17787259, 18193833, 19915271, 20513399, 20631274, 16344544, 26439748
25879656, 14692762, 18614015, 22782647, 17346091, 18413820, 19297917
18228645, 17721717, 13960236, 18685892, 18436307, 11883252, 19888853
21756677, 17891943, 19475971, 22353199, 16384983, 19121551, 27825893
25634317, 12816846, 17982555, 17761775, 13936038, 17227073, 22243719
17265217, 25505394, 17071721, 16721594, 18262334, 21756661, 17891946
15913355, 17672719, 17602269, 17239687, 17042658, 25555252, 17238511
21059919, 17811456, 17284817, 17752121, 20879889, 28806384, 19601762
21380789, 17394950, 28305362, 17011832, 16579084, 22195465, 16875230
14602788, 28790634, 18325460, 27567477, 26569225, 24476265, 24476274
12611721, 18674465, 16903536, 17006570, 19689979, 28076295, 16043574
18783224, 22836801, 24662775, 16494615, 21526048, 17392698, 19197175
16069901, 29200700, 17811447, 27870645, 28876684, 17308789, 24835538
22195477, 17865671, 17343514, 19013183, 17325413, 18316692, 16180763
17348614, 14368995, 21983325, 17393915, 16285691, 19211433, 20331945
17883081, 17705023, 24316947, 23571055, 17614227, 19578350, 22195485
14084247, 24975421, 26078387, 23115139, 13645875, 23328639, 16777840
21698350, 19727057, 14852021, 18744139, 18674047, 17716305, 19285025
18482502, 17622427, 19289642, 27534509, 25947799, 22195492, 14458214
20869721, 21172913, 17767676, 18723434, 25505407, 17786278, 19258504
17082983, 17365043, 21351877, 13498382, 18331812, 16065166, 25489607
16685417, 21566639, 18031668, 22893153, 17551674, 16943711, 19272701
21517440, 25897615, 17649265, 13866822, 18094246, 24528741, 17783588
14245531, 17082359, 18280813, 26007010, 20448824, 23330119, 16268425
19487147, 25600421, 18018515, 17302277, 17215560, 24411921, 19271443
25764020, 14176370, 17016369, 20777150, 16756406, 23330124, 22977256
20441797, 19769489, 28100487, 17545847, 25093656, 18260550, 13853126
17227277, 17551699, 23536835, 25957038, 24652769, 20725343, 19207117
9756271, 17495022, 18868646, 17614134, 26667023, 17546973, 19680952
18704244, 26667015, 17050888, 18273830, 18828868, 17360606, 16992075
24563422, 17375354, 12905058, 18362222, 21429602, 17254374, 27086138
26667032, 28364007, 17571039, 17468141, 18436647, 17570606, 17235750
21168487, 17279227, 16220077, 16929165

Version 11.2.0.4.v20

Version 11.2.0.4.v20 includes the following:

- Patch 29141056: DATABASE PATCH SET UPDATE 11.2.0.4.190416
- Patch 29251270: OJVM PATCH SET UPDATE 11.2.0.4.190416
- Patch 28852325: DSTv33 for RDBMS (TZDATA2018G)
- Patch 28852334: DSTv33 for OJVM (TZDATA2018G)

- Patch 29638593: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 28730253: SUPPORT NEW ERA REIWA FOR JAPANESE IMPERIAL CALENDAR

Oracle patch 22768427, released April 2019

Bugs fixed:

17184721, 21174504, 21538558, 16091637, 18092127, 17381384, 15979965 20671094, 16731148, 16314254, 13837378, 18441944, 17291347, 23105538 17835048, 28254374, 13558557, 21656630, 21842740, 17008068, 18382302 17201159, 25427662, 17853498, 21197626, 20717359, 17246576, 18356166 18681862, 18440047, 20569094, 20031873, 16875449, 20387265, 19788842 17296856, 21330264, 14010183, 17648596, 18886413, 17025461, 17551063 24719736, 17267114, 21063322, 22507210, 17912217, 17889583, 18202441 17040764, 17478145, 16524926, 25655390, 19358317, 22730454, 22148226 18747196, 26544823, 18641419, 17036973, 18948177, 17811789, 16542886 14285317, 18009564, 17359610, 16618694, 8322815, 16832076, 18247991 16692232, 22507234, 17570240, 13871092, 24624166, 26631046, 19429927 24348685, 17848897, 17441661, 14034426, 17465741, 20273319, 19207156 16596890, 18510194, 17437634, 21343897, 20506706, 28849751, 18339044 21453153, 17951233, 21795111, 22321741, 18430495, 21787056, 22380919 19469538, 19692824, 20506715, 17811429, 17903598, 19721304, 11786053 18230522, 19554106, 19458377, 21281607, 17612828, 6599380, 22092979 17040527, 22321756, 17811438, 18641461, 18682983, 14657740, 25635149 13364795, 19490948, 21387964, 17346671, 17588480, 22351572, 18235390 18849970, 26474853, 17889549, 19309466, 20627866, 23008056, 16472716 24908321, 20134113, 25775213, 20596234, 18331850, 18641451, 20882568 17019356, 17344412, 19461270, 21179898, 17546761, 24842886, 17231779 14521849, 18203835, 18203838, 18964939, 18203837, 17313525, 22195457 18139690, 16837842, 14106803, 22296366, 17842825, 22657942, 21352646 16360112, 22594718, 20657441, 22195441, 17389192, 26198926, 14565184 19781326, 17019345, 17205719, 18740837, 18440095, 14764829, 14354737 22195448, 17019086, 13944971, 16571443, 21868720, 17186905, 17080436 18673342, 22905130, 17027426, 27374796, 19972569, 19972568, 20144308 19972566, 17282229, 19972564, 16870214, 16410570, 21629064, 19615136 21354456, 26039623, 17390431, 18762750, 23007241, 16613964, 17957017 18098207, 17484762, 18471685, 19730508, 18264060, 21538485, 17323222 17754782, 17600719, 18317531, 17852463, 17596908, 17655634, 18166013 16228604, 20074391, 27053456, 24790914, 19972570, 20856766, 18090142 19891090, 18996843, 16042673, 19854503, 17835627, 22901797, 20334344 17393683, 20861693, 18000422, 17551709, 26575788, 23315889, 20506699 19006849, 18277454, 18456514, 19174430, 17258090, 17174582, 25654936 17242746, 27097854, 16399083, 17824637, 21132297, 22465352, 17762296 22168163, 18604692, 17397545, 16450169, 12364061, 20067212, 18856999 19211724, 19463893, 27351628, 19463897, 21343775, 17853456, 18373438 18673304, 20004021, 28000269, 26030218, 21668627, 16194160, 17477958 23140259, 16538760, 12982566, 24570598, 20828947, 27255377, 18259031 20296213, 21425496, 28855981, 18293054, 17610798, 19699191, 23065323 17311728, 18135678, 18774543, 23294548, 16785708, 10136473, 24560906 19777862, 22551446, 17786518, 25879984, 18315328, 18334586, 12747740 18096714, 19032867, 21641760, 17390160, 18899974, 17232014, 20598042 26679352, 26245237, 16354467, 17484731, 18673325, 16422541, 18155762 19827973, 14015842, 22683225, 17726838, 18554871, 23177648, 18051556 20803583, 18282562, 21972320, 15990359, 17922254, 16855292, 16668584 21343838, 20299015, 17446237, 18043064, 18093615, 23713236, 17694209 17288409, 20475845, 18308268, 17274537, 13955826, 16934803, 17634921 17501491, 16315398, 23725036, 22683212, 17006183, 13829543, 18191164 26746894, 22809871, 17655240, 28819280, 18384391, 19393542, 21538567 16198143, 21847223, 25823754, 17892268, 20142975, 19584068, 17165204 25165496, 18604493, 21756699, 18508861, 18554763, 16901385, 21532755 18189036, 17443671, 17385178, 14829250, 17936109, 20925795, 20509482 17478514, 27441326, 16850630, 13951456, 16595641, 14054676, 15861775
--

21142837, 16912439, 17299889, 17297939, 23003979, 16833527, 18619917
17798953, 19697993, 17816865, 25914276, 18607546, 17571306, 21286665
16898135, 16819962, 17341326, 26910644, 17851160, 17586955, 20558005
19049453, 21051840, 17587063, 16956380, 18328509, 25423453, 14133975
29033139, 19718981, 18061914, 18522509, 17518652, 21051833, 20294666
18765602, 20860659, 18272672, 20324049, 18199537, 17332800, 13609098
22502493, 18384537, 14338435, 17945983, 16392068, 21067387, 17752995
21051862, 16863422, 17237521, 25505382, 18244962, 19544839, 24433711
24717859, 17156148, 18973907, 23026585, 17449815, 17877323, 18180390
17088068, 17037130, 20004087, 21422580, 19466309, 11733603, 25505371
18610915, 21051858, 18084625, 29027694, 18674024, 26243698, 21051852
18091059, 18306996, 25369547, 16306373, 17787259, 18193833, 19915271
20513399, 20631274, 26439748, 16344544, 25879656, 14692762, 18614015
22782647, 17346091, 18228645, 17721717, 13960236, 18685892, 18436307
11883252, 19888853, 21756677, 17891943, 19475971, 22353199, 16384983
19121551, 25634317, 27825893, 12816846, 17982555, 17761775, 17227073
22243719, 17265217, 25505394, 17071721, 16721594, 18262334, 21756661
17891946, 15913355, 17672719, 17602269, 17239687, 17042658, 25555252
17238511, 21059919, 17811456, 17284817, 17752121, 20879889, 28806384
21380789, 17394950, 17011832, 16579084, 22195465, 14602788, 28790634
18325460, 27567477, 26569225, 24476265, 24476274, 12611721, 18674465
16903536, 17006570, 19689979, 28076295, 16043574, 18783224, 24662775
16494615, 21526048, 17392698, 19197175, 16069901, 17811447, 27870645
28876684, 17308789, 24835538, 22195477, 17865671, 17343514, 19013183
17325413, 18316692, 16180763, 17348614, 14368995, 21983325, 17393915
16285691, 19211433, 20331945, 17883081, 17705023, 24316947, 17614227
19578350, 22195485, 14084247, 24975421, 26078387, 23115139, 13645875
16777840, 19727057, 14852021, 18744139, 18674047, 17716305, 19285025
18482502, 17622427, 19289642, 27534509, 25947799, 22195492, 14458214
20869721, 21172913, 17767676, 18723434, 25505407, 17786278, 19258504
17082983, 17365043, 21351877, 13498382, 18331812, 16065166, 25489607
16685417, 18031668, 22893153, 17551674, 16943711, 19272701, 21517440
25897615, 17649265, 13866822, 18094246, 24528741, 17783588, 14245531
17082359, 26007010, 18280813, 20448824, 23330119, 16268425, 19487147
25600421, 18018515, 17302277, 17215560, 24411921, 19271443, 25764020
14176370, 17016369, 20777150, 16756406, 23330124, 22977256, 20441797
19769489, 28100487, 17545847, 25093656, 18260550, 13853126, 17227277
17551699, 23536835, 25957038, 24652769, 20725343, 19207117, 9756271
18868646, 17614134, 26667023, 17546973, 19680952, 18704244, 26667015
17050888, 18273830, 18828868, 17360606, 16992075, 24563422, 17375354
12905058, 18362222, 21429602, 17254374, 26667032, 28364007, 27086138
17571039, 17468141, 18436647, 17570606, 17235750, 21168487, 17279227
16220077, 16929165

Version 11.2.0.4.v19

Version 11.2.0.4.v19 includes the following:

- Patch 28729262: Oracle Database Patch Set Update 11.2.0.4.190115
- Patch 28790660: Oracle JVM Patch Set Update 11.2.0.4.190115
- Patch 28125601: DSTv32 for RDBMS (TZDATA2018E)
- Patch 27015468: DSTv32 for OJVM (TZDATA2018E)
- Patch 27216420: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches

[Oracle patch 28729262, released January 2019](#)

Bugs fixed:

17288409, 21051852, 24316947, 17811429, 17205719, 18607546, 25654936
17484762, 17816865, 20506699, 24835538, 25957038, 19692824, 23330119
17922254, 17754782, 13364795, 16934803, 17311728, 18604692, 20387265
26679352, 17284817, 17441661, 20671094, 24560906, 16992075, 25635149
17446237, 14015842, 19972569, 21756677, 17375354, 17449815, 20925795
21538558, 17019086, 19463897, 26575788, 13866822, 17235750, 17982555
17478514, 18317531, 14338435, 18235390, 19461270, 20803583, 13944971
19475971, 20142975, 17811789, 16929165, 18704244, 24662775, 20506706
17546973, 21422580, 17359610, 20334344, 14054676, 25489607, 17088068
17570606, 18264060, 17346091, 17343514, 21538567, 19680952, 18471685
19211724, 21132297, 23105538, 13951456, 25775213, 16315398, 21847223
18744139, 16850630, 23177648, 19049453, 18673304, 18090142, 17883081
19915271, 18641419, 18262334, 25600421, 17006183, 16065166, 18277454
18685892, 16833527, 10136473, 18051556, 17865671, 25879984, 18554871
17852463, 18774543, 17853498, 18334586, 19487147, 20879889, 17551709
17588480, 19827973, 17344412, 17842825, 18828868, 20509482, 17025461
13609098, 19429927, 26039623, 11883252, 17239687, 16410570, 23007241
17602269, 19197175, 22195457, 18316692, 17313525, 12611721, 21174504
19544839, 18964939, 20294666, 17600719, 26667015, 18191164, 17571306
19393542, 18482502, 20777150, 27086138, 19466309, 22243719, 17165204
17040527, 18098207, 16785708, 24790914, 19891090, 17465741, 16180763
17174582, 12982566, 16777840, 27097854, 19463893, 22195465, 16875449
22148226, 12816846, 17237521, 6599380, 19358317, 17811438, 25505394
17811447, 21983325, 17945983, 18762750, 16912439, 17184721, 18061914
20598042, 26631046, 21380789, 17282229, 18948177, 18331850, 21142837
18202441, 17082359, 18723434, 19554106, 21532755, 21972320, 25505371
20273319, 14034426, 18339044, 19458377, 17752995, 20448824, 17891943
17767676, 17258090, 16668584, 18384391, 21063322, 17040764, 17381384
15913355, 18356166, 14084247, 20596234, 21641760, 20506715, 13853126
21756661, 18203837, 18610915, 14245531, 16043574, 21756699, 22195441
17848897, 17877323, 26667032, 28790634, 19272701, 21453153, 20569094
17468141, 17786518, 20861693, 17912217, 17037130, 16956380, 18155762
17478145, 17394950, 18189036, 18641461, 17551674, 18619917, 17027426
17019356, 21352646, 16268425, 24476274, 22195492, 19584068, 26544823
18436307, 22507210, 17265217, 13498382, 17634921, 19469538, 21526048
19258504, 23003979, 16354467, 18043064, 19174430, 20004087, 17443671
22195485, 18000422, 22321756, 20004021, 17571039, 27053456, 25897615
16832076, 21067387, 22905130, 16344544, 21429602, 18009564, 14354737
18135678, 21286665, 18614015, 14521849, 20441797, 28876684, 18362222
25655390, 16472716, 17835048, 17050888, 17936109, 14010183, 17325413
18747196, 19207156, 17231779, 21842740, 17761775, 16721594, 17082983
20067212, 21179898, 17279227, 17302277, 18084625, 20717359, 24624166
15990359, 24842886, 26746894, 18203835, 23026585, 17297939, 17811456
16731148, 22380919, 21168487, 14133975, 17215560, 13829543, 18740837
17694209, 17385178, 18091059, 8322815, 18259031, 17586955, 19689979
25165496, 28254374, 17201159, 17655634, 18331812, 19730508, 17551699
17648596, 18868646, 16220077, 16069901, 17393915, 17348614, 17957017
17274537, 18096714, 17308789, 18436647, 14285317, 19289642, 14764829
17622427, 18328509, 23115139, 16943711, 22195477, 22502493, 14368995
17346671, 18996843, 17783588, 18604493, 21343838, 16618694, 17672719
18856999, 18783224, 17851160, 17546761, 22168163, 17798953, 18273830
22092979, 16596890, 19972566, 13871092, 20828947, 26667023, 17726838
16384983, 22296366, 17360606, 13645875, 22321741, 16542886, 18199537
25879656, 25634317, 21787056, 23140259, 17889549, 21172913, 14565184
26245237, 20475845, 27825893, 17071721, 21281607, 17610798, 18308268
20299015, 21343897, 22893153, 22594718, 28076295, 20657441, 17397545
18230522, 16360112, 19769489, 12905058, 18641451, 12747740, 18430495
25423453, 17016369, 17042658, 14602788, 17551063, 26243698, 19972568
21517440, 23725036, 19788842, 18508861, 14657740, 17332800, 13837378
17186905, 19972564, 17019345, 19699191, 18315328, 27441326, 17437634
24570598, 22353199, 18093615, 19006849, 28806384, 17392698, 19013183
17296856, 18674024, 26569225, 17232014, 16855292, 21051840, 14692762
17762296, 17705023, 23294548, 22351572, 22507234, 19121551, 20324049
21330264, 26198926, 19854503, 23315889, 26030218, 26910644, 21868720
19309466, 27567477, 25764020, 18681862, 17365043, 17390160, 18554763
20031873, 20558005, 24717859, 21795111, 18456514, 13955826, 16306373

18139690, 17501491, 17752121, 17299889, 21668627, 23713236, 24652769
17889583, 18673325, 22551446, 17242746, 18293054, 18674465, 19721304
19211433, 19888853, 25914276, 24563422, 17951233, 18094246, 17649265
19615136, 17011832, 17477958, 16870214, 18522509, 20631274, 16091637
17323222, 16595641, 16524926, 18228645, 17484731, 18282562, 17596908
18272672, 18031668, 17156148, 16494615, 22683225, 20869721, 17545847
25093656, 28819280, 18682983, 17655240, 24528741, 17614134, 13558557
25427662, 17341326, 22465352, 29033139, 17891946, 17716305, 22657942
27374796, 16392068, 18440095, 19271443, 21351877, 20513399, 18092127
17614227, 18440047, 18849970, 14106803, 16903536, 20725343, 18973907
18673342, 17389192, 25505382, 22809871, 19032867, 17612828, 17006570
16194160, 25369547, 25505407, 16685417, 17721717, 21354456, 17390431
17570240, 16863422, 13960236, 28100487, 18325460, 17008068, 19727057
28855981, 16422541, 17267114, 19972570, 18244962, 21538485, 18203838
18765602, 16198143, 17246576, 14829250, 28364007, 17835627, 20860659
21629064, 18247991, 14458214, 21051862, 17786278, 16692232, 17227277
24348685, 16042673, 24476265, 24975421, 22901797, 16314254, 19285025
16228604, 16756406, 14176370, 16837842, 20144308, 17393683, 23536835
25823754, 18899974, 17787259, 24719736, 20331945, 26078387, 20074391
19490948, 15861775, 16399083, 25947799, 25555252, 18018515, 22683212
18260550, 21051858, 17080436, 16613964, 17036973, 16579084, 24433711
18384537, 27870645, 18280813, 20296213, 16901385, 15979965, 17518652
23330124, 20856766, 18441944, 16450169, 9756271, 27534509, 22730454
19718981, 17291347, 17892268, 11733603, 16285691, 17587063, 21343775
18180390, 16538760, 26474853, 18193833, 21387964, 21051833, 17238511
19777862, 17824637, 23065323, 21656630, 19697993, 17903598, 16571443
18306996, 18166013, 19578350, 14852021, 18674047, 17853456, 12364061
24411921, 19207117, 22195448

Version 11.2.0.4.v18

Version 11.2.0.4.v18 includes the following:

- Patch 28204707: Oracle Database Patch Set Update 11.2.0.4.181016
- Patch 28440700: Oracle JVM Patch Set Update 11.2.0.4.181016
- Patch 28125601: DSTv32 for RDBMS (TZDATA2018E)
- Patch 27015468: DSTv32 for OJVM (TZDATA2018E)
- Patch 27216420: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patches 27659043 and 19692824 are now included in the Database Patch Set Update

Oracle patch 28204707, released October 2018

Bugs fixed:

17288409, 21051852, 24316947, 17811429, 17205719, 18607546, 25654936
17484762, 17816865, 20506699, 24835538, 25957038, 19692824, 23330119
17922254, 17754782, 13364795, 16934803, 17311728, 18604692, 26679352
20387265, 17284817, 17441661, 20671094, 24560906, 25635149, 16992075
17446237, 14015842, 19972569, 21756677, 17375354, 21538558, 20925795
17449815, 17019086, 19463897, 26575788, 13866822, 17235750, 17982555
17478514, 18317531, 14338435, 18235390, 19461270, 20803583, 13944971
19475971, 20142975, 17811789, 16929165, 18704244, 24662775, 20506706
17359610, 17546973, 21422580, 20334344, 14054676, 25489607, 17570606
17088068, 17346091, 18264060, 17343514, 21538567, 19680952, 18471685
19211724, 21132297, 25775213, 13951456, 16315398, 21847223, 18744139
16850630, 23177648, 19049453, 18090142, 18673304, 17883081, 19915271
18641419, 18262334, 25600421, 17006183, 16065166, 18277454, 18685892
16833527, 10136473, 18051556, 17865671, 25879984, 18554871, 17852463

18774543, 17853498, 18334586, 19487147, 20879889, 17551709, 17588480
19827973, 17344412, 17842825, 18828868, 20509482, 17025461, 26039623
19429927, 13609098, 11883252, 16410570, 17239687, 23007241, 17602269
19197175, 22195457, 18316692, 17313525, 12611721, 21174504, 19544839
20294666, 18964939, 17600719, 26667015, 18191164, 17571306, 19393542
20777150, 18482502, 27086138, 19466309, 22243719, 17165204, 17040527
18098207, 24790914, 16785708, 19891090, 17465741, 16180763, 17174582
12982566, 16777840, 19463893, 22195465, 16875449, 22148226, 12816846
17237521, 6599380, 19358317, 17811438, 25505394, 17811447, 21983325
17945983, 18762750, 16912439, 17184721, 18061914, 20598042, 26631046
21380789, 17282229, 18948177, 18331850, 21142837, 18202441, 17082359
18723434, 21972320, 21532755, 19554106, 25505371, 20273319, 14034426
18339044, 19458377, 17752995, 20448824, 17891943, 17767676, 17258090
16668584, 18384391, 21063322, 17040764, 17381384, 15913355, 18356166
14084247, 20596234, 21641760, 20506715, 13853126, 21756661, 18610915
18203837, 14245531, 16043574, 21756699, 22195441, 17848897, 17877323
26667032, 21453153, 19272701, 20569094, 17468141, 17786518, 20861693
17912217, 17037130, 16956380, 18155762, 17478145, 17394950, 18641461
18189036, 17551674, 18619917, 17019356, 17027426, 21352646, 16268425
24476274, 22195492, 19584068, 26544823, 18436307, 22507210, 17265217
13498382, 17634921, 19469538, 21526048, 19258504, 23003979, 16354467
18043064, 19174430, 20004087, 17443671, 22195485, 18000422, 22321756
20004021, 17571039, 25897615, 27053456, 16832076, 21067387, 22905130
16344544, 21429602, 18009564, 14354737, 21286665, 18135678, 14521849
18614015, 20441797, 18362222, 25655390, 16472716, 17835048, 17050888
17936109, 14010183, 17325413, 18747196, 19207156, 17231779, 21842740
17761775, 16721594, 17082983, 20067212, 21179898, 17279227, 17302277
18084625, 20717359, 24624166, 15990359, 24842886, 26746894, 18203835
23026585, 17297939, 17811456, 16731148, 22380919, 21168487, 14133975
13829543, 17215560, 18740837, 17694209, 17385178, 18091059, 8322815
18259031, 28254374, 19689979, 25165496, 17586955, 17201159, 17655634
18331812, 17551699, 19730508, 17648596, 18868646, 16220077, 16069901
17393915, 17348614, 17957017, 17274537, 18096714, 17308789, 18436647
14285317, 19289642, 14764829, 17622427, 18328509, 23115139, 16943711
22195477, 22502493, 14368995, 17346671, 18996843, 17783588, 18604493
21343838, 16618694, 17672719, 18856999, 18783224, 17851160, 17546761
22168163, 17798953, 18273830, 22092979, 16596890, 19972566, 13871092
20828947, 26667023, 17726838, 16384983, 22296366, 17360606, 13645875
22321741, 25634317, 16542886, 18199537, 25879656, 21787056, 23140259
17889549, 21172913, 26245237, 14565184, 27825893, 20475845, 17071721
21281607, 17610798, 18308268, 20290105, 21343897, 22893153, 22594718
20657441, 17397545, 18230522, 16360112, 19769489, 12905058, 18641451
12747740, 18430495, 25423453, 17016369, 17042658, 14602788, 17551063
26243698, 19972568, 21517440, 23725036, 19788842, 18508861, 14657740
17332800, 13837378, 17186905, 19972564, 17019345, 19699191, 18315328
27441326, 17437634, 24570598, 22353199, 18093615, 19006849, 17392698
19013183, 17296856, 18674024, 26569225, 17232014, 16855292, 21051840
14692762, 17762296, 17705023, 23294548, 22351572, 22507234, 19121551
20324049, 21330264, 26198926, 19854503, 23315889, 26910644, 26030218
21868720, 19309466, 25764020, 18681862, 17365043, 17390160, 20031873
20558005, 18554763, 24717859, 21795111, 18456514, 13955826, 16306373
18139690, 17501491, 17752121, 17299889, 21668627, 23713236, 24652769
17889583, 18673325, 22551446, 18674465, 17242746, 19721304, 18293054
19211433, 19888853, 25914276, 24563422, 17951233, 18094246, 17649265
19615136, 17011832, 17477958, 16870214, 18522509, 20631274, 16091637
17323222, 16595641, 16524926, 17484731, 18228645, 18282562, 17596908
18272672, 18031668, 17156148, 16494615, 22683225, 20869721, 17545847
25093656, 18682983, 17655240, 24528741, 17614134, 25427662, 13558557
17341326, 22465352, 17891946, 17716305, 22657942, 27374796, 16392068
18440095, 19271443, 21351877, 20513399, 18092127, 17614227, 18440047
18849970, 16903536, 14106803, 20725343, 18973907, 18673342, 17389192
19032867, 25505382, 22809871, 17612828, 17006570, 16194160, 25369547
25505407, 16685417, 17721717, 21354456, 17390431, 17570240, 13960236
16863422, 28100487, 18325460, 17008068, 19727057, 16422541, 17267114
19972570, 18244962, 21538485, 18203838, 18765602, 16198143, 17246576
14829250, 28364007, 17835627, 20860659, 21629064, 18247991, 14458214

21051862, 17786278, 16692232, 17227277, 24348685, 24476265, 16042673
24975421, 22901797, 16314254, 19285025, 16228604, 16756406, 14176370
16837842, 20144308, 17393683, 23536835, 25823754, 18899974, 17787259
24719736, 20331945, 26078387, 19490948, 20074391, 15861775, 16399083
25555252, 25947799, 18018515, 22683212, 18260550, 21051858, 17080436
16613964, 17036973, 16579084, 24433711, 18384537, 27870645, 18280813
20296213, 16901385, 15979965, 17518652, 23330124, 20856766, 18441944
16450169, 9756271, 27534509, 22730454, 19718981, 17291347, 17892268
11733603, 16285691, 17587063, 21343775, 18180390, 26474853, 16538760
18193833, 21387964, 21051833, 17238511, 19777862, 17824637, 23065323
21656630, 17903598, 16571443, 18166013, 18306996, 19578350, 14852021
17853456, 18674047, 12364061, 24411921, 19207117, 22195448

Version 11.2.0.4.v17

Version 11.2.0.4.v17 includes the following:

- Patch 27734982: Oracle Database Patch Set Update 11.2.0.4.180717
- Patch 27923163: Oracle JVM Patch Set Update 11.2.0.4.180717
- Patch 28125601: DSTv32 for RDBMS (TZDATA2018E)
- Patch 27015468: DSTv32 for OJVM (TZDATA2018E)
- Patch 27216420: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 27659043: MES Bundle 405
- Patch 19692824: DBCONTROL is not coming up on OEL 7

Oracle patch 27734982, released July 2018

Bugs fixed:

17288409, 21051852, 24316947, 17811429, 17205719, 18607546, 25654936
17816865, 20506699, 24835538, 25957038, 23330119, 17922254, 17754782
13364795, 16934803, 17311728, 20387265, 17284817, 17441661, 20671094
24560906, 16992075, 17446237, 14015842, 19972569, 21756677, 17375354
21538558, 20925795, 17449815, 19463897, 26575788, 13866822, 17235750
17982555, 17478514, 18317531, 14338435, 18235390, 19461270, 20803583
13944971, 19475971, 20142975, 17811789, 16929165, 18704244, 24662775
20506706, 17546973, 21422580, 20334344, 14054676, 25489607, 17088068
17346091, 18264060, 17343514, 21538567, 19680952, 18471685, 19211724
21132297, 13951456, 16315398, 21847223, 18744139, 16850630, 23177648
19049453, 18090142, 18673304, 17883081, 19915271, 18641419, 18262334
25600421, 17006183, 16065166, 18277454, 16833527, 10136473, 18051556
17865671, 18554871, 17852463, 18774543, 17853498, 18334586, 19487147
20879889, 17551709, 17588480, 19827973, 17344412, 17842825, 18828868
20509482, 17025461, 13609098, 11883252, 17239687, 23007241, 17602269
19197175, 22195457, 18316692, 17313525, 12611721, 21174504, 19544839
20294666, 18964939, 17600719, 26667015, 18191164, 17571306, 19393542
20777150, 18482502, 27086138, 19466309, 22243719, 17165204, 17040527
18098207, 16785708, 17465741, 16180763, 17174582, 12982566, 16777840
19463893, 22195465, 16875449, 22148226, 12816846, 17237521, 6599380
19358317, 17811438, 25505394, 17811447, 21983325, 17945983, 18762750
16912439, 17184721, 18061914, 20598042, 21380789, 17282229, 18948177
18331850, 21142837, 18202441, 17082359, 18723434, 21972320, 21532755
19554106, 25505371, 14034426, 18339044, 19458377, 17752995, 20448824
17891943, 17767676, 17258090, 16668584, 18384391, 17040764, 17381384
15913355, 18356166, 14084247, 20596234, 21641760, 20506715, 13853126
21756661, 18203837, 14245531, 16043574, 21756699, 22195441, 17848897
17877323, 21453153, 19272701, 20569094, 17468141, 17786518, 20861693

17912217, 17037130, 16956380, 18155762, 17478145, 17394950, 18641461
18189036, 18619917, 17027426, 21352646, 16268425, 24476274, 22195492
19584068, 26544823, 18436307, 22507210, 17265217, 13498382, 17634921
19469538, 21526048, 19258504, 23003979, 18043064, 19174430, 20004087
17443671, 22195485, 18000422, 20004021, 22321756, 17571039, 27053456
25897615, 21067387, 16832076, 22905130, 16344544, 21429602, 18009564
14354737, 21286665, 18135678, 14521849, 18614015, 20441797, 18362222
25655390, 16472716, 17835048, 17050888, 17936109, 14010183, 17325413
18747196, 17761775, 16721594, 17082983, 20067212, 21179898, 17302277
18084625, 20717359, 24624166, 15990359, 24842886, 26746894, 18203835
23026585, 17297939, 17811456, 16731148, 22380919, 21168487, 14133975
13829543, 17215560, 17694209, 17385178, 18091059, 8322815, 18259031
19689979, 25165496, 17586955, 17201159, 17655634, 18331812, 19730508
17648596, 18868646, 16220077, 16069901, 17393915, 17348614, 17957017
17274537, 18096714, 17308789, 18436647, 14285317, 19289642, 14764829
17622427, 18328509, 16943711, 22195477, 22502493, 14368995, 17346671
18996843, 17783588, 18604493, 21343838, 16618694, 17672719, 18856999
18783224, 17851160, 17546761, 22168163, 17798953, 18273830, 22092979
16596890, 19972566, 20828947, 13871092, 26667023, 17726838, 16384983
22296366, 17360606, 13645875, 22321741, 16542886, 18199537, 25879656
21787056, 17889549, 21172913, 14565184, 27825893, 20475845, 17071721
21281607, 18308268, 17610798, 20299015, 21343897, 22893153, 20657441
17397545, 18230522, 16360112, 19769489, 12905058, 18641451, 12747740
18430495, 25423453, 17016369, 17042658, 14602788, 17551063, 19972568
21517440, 23725036, 19788842, 18508861, 14657740, 17332800, 13837378
17186905, 19972564, 19699191, 18315328, 27441326, 17437634, 24570598
22353199, 18093615, 19006849, 17392698, 19013183, 17296856, 18674024
26569225, 17232014, 16855292, 21051840, 14692762, 17762296, 17705023
23294548, 22351572, 22507234, 19121551, 20324049, 21330264, 26198926
19854503, 23315889, 26910644, 26030218, 21868720, 19309466, 25764020
18681862, 17365043, 17390160, 20031873, 20558005, 18554763, 24717859
21795111, 18456514, 16306373, 13955826, 18139690, 17501491, 17752121
17299889, 21668627, 23713236, 24652769, 17889583, 18673325, 22551446
17242746, 19721304, 18293054, 19211433, 19888853, 24563422, 17951233
18094246, 17649265, 19615136, 17011832, 17477958, 16870214, 18522509
20631274, 16091637, 17323222, 16595641, 16524926, 18228645, 18282562
17596908, 18031668, 17156148, 16494615, 22683225, 20869721, 17545847
25093656, 17655240, 24528741, 17614134, 25427662, 13558557, 17341326
22465352, 17891946, 17716305, 22657942, 27374796, 16392068, 18440095
19271443, 21351877, 20513399, 18092127, 17614227, 18440047, 18849970
16903536, 14106803, 18973907, 18673342, 17389192, 19032867, 25505382
22809871, 17612828, 17006570, 16194160, 25369547, 25505407, 16685417
17721717, 21354456, 17390431, 17570240, 16863422, 28100487, 18325460
17008068, 19727057, 16422541, 17267114, 19972570, 18244962, 21538485
18203838, 18765602, 16198143, 17246576, 14829250, 17835627, 20860659
21629064, 18247991, 14458214, 21051862, 17786278, 16692232, 24348685
17227277, 24476265, 16042673, 16314254, 19285025, 16228604, 16756406
16837842, 20144308, 17393683, 23536835, 25823754, 18899974, 17787259
24719736, 20331945, 19490948, 20074391, 15861775, 16399083, 25947799
18018515, 22683212, 18260550, 21051858, 17080436, 16613964, 17036973
16579084, 24433711, 18384537, 27870645, 18280813, 20296213, 16901385
15979965, 23330124, 18441944, 16450169, 27534509, 9756271, 17892268
11733603, 16285691, 17587063, 21343775, 18180390, 26474853, 16538760
18193833, 21387964, 21051833, 17238511, 19777862, 17824637, 23065323
17903598, 16571443, 18306996, 19578350, 14852021, 17853456, 18674047
12364061, 24411921, 19207117, 22195448

Version 11.2.0.4.v16

Version 11.2.0.4.v16 includes the following:

- Patch 27338049: DATABASE PATCH SET UPDATE 11.2.0.4.180417
- Patch 27475598: OJVM PATCH SET UPDATE 11.2.0.4.180417

- Patch 27015449: RDBMS - PROACTIVE DSTV31 UPDATE - TZDATA2017C
- Patch 27015468: PROACTIVE DSTV31 UPDATE - TZDATA2017C - NEED OJVM FIX
- Patch 27216420: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 27659043: MES 405 BUNDLE ON TOP OF RDBMS 11.2.0.4.180116 PSU
- Patch 19692824: DBCONTROL is not coming up on OEL 7
- Support for the DBMS_ADVANCED_REWRITE package
- Fixed a bug where DBA_LOCKS and associated views available in new DB instances of 11.2.0.4.v15 were not created in upgrades to 11.2.0.4.v15. Views are now created in new and upgraded DB instances of 11.2.0.4.v16 and later.

Oracle patch 27338049, released April 2018

Bugs fixed:

```
21174504, 17184721, 21538558, 16091637, 18092127, 17381384, 15979965
20671094, 16731148, 16314254, 13837378, 18441944, 17835048, 13558557
17008068, 17201159, 25427662, 17853498, 20717359, 17246576, 18356166
18681862, 18440047, 20569094, 20031873, 16875449, 20387265, 19788842
17296856, 21330264, 14010183, 17648596, 17551063, 17025461, 24719736
17267114, 22507210, 17912217, 17889583, 18202441, 17040764, 17478145
16524926, 25655390, 19358317, 22148226, 18747196, 26544823, 18641419
17036973, 18948177, 17811789, 16542886, 14285317, 18009564, 16618694
8322815, 16832076, 18247991, 16692232, 22507234, 17570240, 13871092
24624166, 17848897, 17441661, 14034426, 17465741, 16596890, 17437634
21343897, 20506706, 21453153, 18339044, 22321741, 21795111, 17951233
18430495, 21787056, 22380919, 19469538, 20506715, 17811429, 19721304
17903598, 18230522, 19554106, 19458377, 21281607, 17612828, 6599380
22092979, 22321756, 17040527, 17811438, 18641461, 14657740, 13364795
21387964, 19490948, 22351572, 17346671, 17588480, 18235390, 26474853
18849970, 17889549, 19309466, 16472716, 20596234, 18331850, 18641451
17344412, 21179898, 19461270, 17546761, 24842886, 14521849, 18203835
18203838, 18964939, 18203837, 17313525, 22195457, 18139690, 16837842
22296366, 14106803, 17842825, 21352646, 22657942, 16360112, 20657441
22195441, 17389192, 26198926, 14565184, 17205719, 18440095, 14764829
22195448, 14354737, 13944971, 16571443, 21868720, 17186905, 17080436
18673342, 22905130, 17027426, 27374796, 19972569, 19972568, 20144308
19972566, 17282229, 19972564, 16870214, 21629064, 19615136, 21354456
17390431, 18762750, 23007241, 16613964, 17957017, 18098207, 18471685
19730508, 21538485, 18264060, 17323222, 17754782, 17600719, 18317531
17852463, 17596908, 17655634, 16228604, 27053456, 20074391, 19972570
18090142, 18996843, 19854503, 16042673, 17835627, 20334344, 17393683
20861693, 18000422, 17551709, 26575788, 23315889, 20506699, 19006849
18277454, 18456514, 19174430, 17258090, 17174582, 25654936, 17242746
16399083, 17824637, 21132297, 22465352, 17762296, 22168163, 17397545
16450169, 12364061, 20067212, 18856999, 19211724, 19463893, 19463897
21343775, 17853456, 18673304, 20004021, 26030218, 21668627, 16194160
17477958, 16538760, 12982566, 24570598, 20828947, 18259031, 20296213
18293054, 17610798, 19699191, 23065323, 17311728, 18135678, 18774543
23294548, 16785708, 10136473, 24560906, 22551446, 19777862, 17786518
18315328, 18334586, 12747740, 18096714, 19032867, 21641760, 18899974
17390160, 17232014, 20598042, 18673325, 16422541, 18155762, 14015842
19827973, 22683225, 17726838, 18554871, 23177648, 18051556, 20803583
21972320, 15990359, 17922254, 18282562, 16855292, 16668584, 21343838
20299015, 17446237, 18093615, 18043064, 23713236, 17694209, 17288409
20475845, 17274537, 13955826, 16934803, 17634921, 17501491, 16315398
22683212, 17006183, 13829543, 18191164, 17655240, 26746894, 22809871
18384391, 19393542, 21538567, 16198143, 21847223, 25823754, 17892268
20142975, 19584068, 17165204, 25165496, 18604493, 21756699, 18508861
16901385, 18554763, 21532755, 18189036, 17443671, 17385178, 14829250
```

17936109, 20925795, 20509482, 17478514, 27441326, 16850630, 13951456
16595641, 14054676, 15861775, 21142837, 16912439, 17299889, 17297939
23003979, 18619917, 16833527, 17798953, 17816865, 18607546, 17571306
21286665, 17341326, 26910644, 17851160, 20558005, 17586955, 19049453
21051840, 17587063, 16956380, 18328509, 25423453, 14133975, 18061914
18522509, 21051833, 18765602, 20860659, 20324049, 18199537, 17332800
13609098, 22502493, 18384537, 14338435, 17945983, 16392068, 21067387
17752995, 21051862, 16863422, 25505382, 17237521, 18244962, 19544839
24433711, 24717859, 17156148, 18973907, 23026585, 17877323, 17449815
18180390, 17088068, 17037130, 20004087, 21422580, 19466309, 11733603
25505371, 21051858, 18084625, 18674024, 21051852, 18091059, 25369547
16306373, 18306996, 18193833, 19915271, 17787259, 20513399, 20631274
25879656, 16344544, 14692762, 18614015, 17346091, 18228645, 17721717
18436307, 21756677, 19888853, 11883252, 17891943, 19475971, 22353199
16384983, 19121551, 12816846, 17982555, 17761775, 22243719, 17265217
25505394, 17071721, 16721594, 21756661, 18262334, 17891946, 15913355
17672719, 17602269, 17239687, 17042658, 17238511, 17811456, 17284817
17752121, 20879889, 21380789, 17394950, 17011832, 16579084, 22195465
14602788, 18325460, 24476265, 26569225, 24476274, 12611721, 16903536
17006570, 19689979, 16043574, 18783224, 24662775, 16494615, 21526048
17392698, 19197175, 16069901, 17811447, 17308789, 22195477, 24835538
17865671, 17343514, 19013183, 17325413, 18316692, 16180763, 17348614
14368995, 21983325, 17393915, 16285691, 19211433, 20331945, 17883081
17705023, 24316947, 17614227, 19578350, 22195485, 14084247, 13645875
16777840, 19727057, 14852021, 18744139, 18674047, 17716305, 19285025
18482502, 17622427, 19289642, 22195492, 25947799, 14458214, 20869721
21172913, 17767676, 18723434, 25505407, 17786278, 19258504, 17082983
21351877, 17365043, 13498382, 18331812, 16065166, 25489607, 16685417
18031668, 22893153, 16943711, 19272701, 21517440, 25897615, 17649265
13866822, 18094246, 24528741, 17783588, 14245531, 17082359, 18280813
20448824, 23330119, 16268425, 19487147, 25600421, 18018515, 17302277
17215560, 24411921, 19271443, 25764020, 17016369, 20777150, 23330124
16756406, 20441797, 19769489, 17545847, 25093656, 18260550, 13853126
17227277, 23536835, 25957038, 24652769, 19207117, 9756271, 18868646
17614134, 26667023, 17546973, 18704244, 19680952, 26667015, 17050888
18828868, 18273830, 17360606, 24563422, 16992075, 17375354, 12905058
18362222, 21429602, 27086138, 17571039, 17468141, 18436647, 17235750
21168487, 16220077, 16929165

Version 11.2.0.4.v15

Version 11.2.0.4.v15 includes the following:

- Patch 26925576: DATABASE PATCH SET UPDATE 11.2.0.4.180116
- Patch 26925532: OJVM PATCH SET UPDATE 11.2.0.4.180116
- Patch 27015449: RDBMS - PROACTIVE DSTV31 UPDATE - TZDATA2017C
- Patch 27015468: PROACTIVE DSTV31 UPDATE - TZDATA2017C - NEED OJVM FIX
- Patch 27216420: Oracle GoldenGate - Oracle RDBMS Server Recommended Patches
- Patch 27244661: MES 405 BUNDLE ON TOP OF RDBMS 11.2.0.4.180116 PSU
- Patch 19692824: DBCONTROL is not coming up on OEL 7
- Support for DBA_LOCKS and associated views

Oracle patch 26925576, released January 2018

Bugs fixed:

17288409, 21051852, 24316947, 17811429, 17205719, 18607546, 25654936
17816865, 20506699, 24835538, 25957038, 23330119, 17922254, 17754782
13364795, 16934803, 17311728, 20387265, 17284817, 17441661, 20671094
24560906, 16992075, 17446237, 14015842, 19972569, 21756677, 17375354
21538558, 20925795, 17449815, 26575788, 19463897, 13866822, 17235750
17982555, 17478514, 18317531, 14338435, 18235390, 20803583, 19461270
19475971, 13944971, 20142975, 17811789, 16929165, 18704244, 24662775
20506706, 21422580, 17546973, 20334344, 14054676, 25489607, 17088068
17346091, 18264060, 17343514, 21538567, 19680952, 18471685, 19211724
21132297, 13951456, 16315398, 21847223, 18744139, 16850630, 23177648
19049453, 18090142, 18673304, 17883081, 19915271, 18641419, 18262334
25600421, 17006183, 16065166, 18277454, 16833527, 10136473, 18051556
17865671, 18554871, 17852463, 17853498, 18334586, 20879889, 17551709
17588480, 19827973, 17344412, 17842825, 18828868, 20509482, 17025461
13609098, 11883252, 17239687, 23007241, 17602269, 19197175, 18316692
22195457, 17313525, 12611721, 21174504, 19544839, 18964939, 17600719
26667015, 18191164, 17571306, 19393542, 20777150, 18482502, 19466309
22243719, 17165204, 17040527, 18098207, 16785708, 17465741, 16180763
17174582, 12982566, 16777840, 19463893, 22195465, 16875449, 22148226
12816846, 17237521, 6599380, 19358317, 17811438, 25505394, 17811447
21983325, 17945983, 18762750, 16912439, 17184721, 20598042, 18061914
21380789, 17282229, 18948177, 18331850, 21142837, 18202441, 17082359
18723434, 21972320, 21532755, 19554106, 25505371, 14034426, 18339044
19458377, 17752995, 20448824, 17891943, 17767676, 17258090, 16668584
18384391, 17040764, 17381384, 15913355, 18356166, 14084247, 20596234
21641760, 20506715, 13853126, 21756661, 18203837, 14245531, 16043574
21756699, 22195441, 17848897, 17877323, 21453153, 19272701, 20569094
17468141, 17786518, 20861693, 17912217, 17037130, 16956380, 18155762
17478145, 17394950, 18641461, 18189036, 18619917, 17027426, 21352646
16268425, 24476274, 22195492, 19584068, 26544823, 18436307, 22507210
17265217, 13498382, 17634921, 19469538, 21526048, 19258504, 23003979
19174430, 18043064, 20004087, 17443671, 22195485, 18000422, 20004021
22321756, 17571039, 25897615, 27053456, 21067387, 16832076, 22905130
16344544, 21429602, 18009564, 14354737, 21286665, 18135678, 14521849
18614015, 20441797, 18362222, 25655390, 16472716, 17835048, 17050888
17936109, 14010183, 17325413, 18747196, 17761775, 16721594, 17082983
20067212, 21179898, 17302277, 18084625, 20717359, 24624166, 15990359
26746894, 24842886, 18203835, 23026585, 17297939, 17811456, 16731148
22380919, 21168487, 14133975, 13829543, 17215560, 17694209, 17385178
18091059, 8322815, 18259031, 25165496, 19689979, 17586955, 17201159
17655634, 18331812, 19730508, 18868646, 17648596, 16220077, 16069901
17393915, 17348614, 17957017, 17274537, 18096714, 17308789, 18436647
14285317, 19289642, 14764829, 17622427, 18328509, 16943711, 22195477
22502493, 14368995, 17346671, 18996843, 17783588, 21343838, 16618694
17672719, 18856999, 18783224, 17851160, 17546761, 22168163, 17798953
18273830, 22092979, 16596890, 19972566, 20828947, 13871092, 26667023
17726838, 16384983, 22296366, 17360606, 13645875, 22321741, 16542886
25879656, 18199537, 21787056, 17889549, 21172913, 14565184, 20475845
17071721, 21281607, 17610798, 20299015, 21343897, 22893153, 20657441
17397545, 18230522, 16360112, 19769489, 12905058, 18641451, 12747740
18430495, 25423453, 17016369, 17042658, 14602788, 17551063, 19972568
21517440, 19788842, 18508861, 14657740, 17332800, 13837378, 17186905
19972564, 19699191, 18315328, 17437634, 24570598, 22353199, 18093615
19006849, 19013183, 17296856, 18674024, 26569225, 17232014, 16855292
21051840, 14692762, 17762296, 17705023, 23294548, 22507234, 191211551
20324049, 21330264, 26198926, 19854503, 23315889, 26910644, 26030218
21868720, 19309466, 25764020, 18681862, 17365043, 20031873, 20558005
18554763, 17390160, 24717859, 21795111, 18456514, 16306373, 13955826
18139690, 17501491, 17752121, 21668627, 17299889, 23713236, 24652769
17889583, 18673325, 22551446, 19721304, 18293054, 17242746, 19211433
19888853, 17951233, 18094246, 17649265, 19615136, 17011832, 17477958
16870214, 18522509, 20631274, 16091637, 17323222, 16595641, 16524926
18228645, 18282562, 17596908, 18031668, 17156148, 16494615, 22683225
20869721, 17545847, 25093656, 17655240, 24528741, 17614134, 25427662
13558557, 22465352, 17341326, 17891946, 17716305, 22657942, 16392068
18440095, 19271443, 21351877, 20513399, 18092127, 17614227, 18440047

18849970, 16903536, 14106803, 18973907, 18673342, 22809871, 17389192
19032867, 25505382, 17612828, 17006570, 16194160, 25369547, 16685417
25505407, 17721717, 21354456, 17390431, 17570240, 16863422, 18325460
17008068, 19727057, 16422541, 19972570, 17267114, 18244962, 21538485
18203838, 18765602, 16198143, 17246576, 14829250, 17835627, 20860659
21629064, 18247991, 14458214, 21051862, 17786278, 16692232, 17227277
24476265, 16042673, 16314254, 19285025, 16228604, 16756406, 16837842
20144308, 17393683, 23536835, 25823754, 18899974, 17787259, 24719736
20331945, 19490948, 20074391, 15861775, 16399083, 25947799, 18018515
22683212, 21051858, 18260550, 17080436, 16613964, 17036973, 16579084
24433711, 18384537, 18280813, 20296213, 16901385, 15979965, 23330124
18441944, 16450169, 9756271, 17892268, 11733603, 16285691, 17587063
21343775, 18180390, 26474853, 16538760, 18193833, 21387964, 21051833
17238511, 19777862, 23065323, 17824637, 16571443, 17903598, 18306996
19578350, 14852021, 17853456, 18674047, 12364061, 19207117, 24411921,
22195448

Version 11.2.0.4.v14

Version 11.2.0.4.v14 includes the following:

- Oracle October 2017 PSU, a combination of database PSU (patch 26392168) + OJVM component PSU (patch 26635834)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 26950781)
- RSA Micro-Edition Suite Bundle (patch 26963526)
- Timezone file DSTv30 (patch 25881255, OJVM patch 25881271)

Oracle patch 26392168, released October 2017

Bugs fixed:

17288409, 21051852, 24316947, 17811429, 17205719, 18607546, 25654936
20506699, 17816865, 25957038, 23330119, 17922254, 17754782, 13364795
16934803, 17311728, 20387265, 17284817, 17441661, 24560906, 16992075
17446237, 14015842, 19972569, 21756677, 17375354, 21538558, 20925795
17449815, 26575788, 19463897, 13866822, 17235750, 17982555, 17478514
18317531, 14338435, 18235390, 20803583, 19461270, 13944971, 20142975
17811789, 16929165, 18704244, 24662775, 20506706, 17546973, 20334344
25489607, 14054676, 17088068, 17346091, 18264060, 17343514, 21538567
19680952, 18471685, 19211724, 21132297, 13951456, 21847223, 16315398
18744139, 16850630, 23177648, 19049453, 18673304, 17883081, 19915271
18641419, 18262334, 25600421, 17006183, 16065166, 18277454, 16833527
10136473, 18051556, 17865671, 17852463, 18554871, 17853498, 18334586
20879889, 17551709, 17588480, 19827973, 17344412, 17842825, 18828868
20509482, 17025461, 11883252, 13609098, 17239687, 17602269, 19197175
18316692, 22195457, 17313525, 12611721, 19544839, 18964939, 26667015
17600719, 18191164, 19393542, 17571306, 20777150, 18482502, 19466309
22243719, 17040527, 17165204, 18098207, 16785708, 17465741, 16180763
17174582, 12982566, 16777840, 19463893, 22195465, 16875449, 22148226
12816846, 17237521, 6599380, 19358317, 17811438, 25505394, 17811447
17945983, 21983325, 18762750, 16912439, 17184721, 18061914, 17282229
18331850, 18202441, 17082359, 18723434, 21532755, 21972320, 19554106
25505371, 14034426, 18339044, 19458377, 17752995, 20448824, 17891943
17258090, 17767676, 16668584, 18384391, 17040764, 17381384, 15913355
18356166, 14084247, 20596234, 20506715, 21756661, 13853126, 18203837
14245531, 16043574, 21756699, 22195441, 17848897, 17877323, 19272701
21453153, 20569094, 17468141, 20861693, 17786518, 17912217, 17037130
16956380, 18155762, 17478145, 17394950, 18641461, 18189036, 18619917

17027426, 21352646, 16268425, 24476274, 22195492, 19584068, 26544823
18436307, 22507210, 17265217, 17634921, 13498382, 19469538, 21526048
19258504, 18043064, 20004087, 17443671, 22195485, 18000422, 20004021
22321756, 17571039, 21067387, 16832076, 22905130, 16344544, 21429602
18009564, 14354737, 21286665, 18135678, 14521849, 18614015, 20441797
18362222, 25655390, 16472716, 17835048, 17050888, 17936109, 14010183
17325413, 18747196, 17761775, 16721594, 17082983, 20067212, 21179898
17302277, 18084625, 24624166, 15990359, 26746894, 24842886, 23026585
18203835, 17297939, 17811456, 16731148, 22380919, 21168487, 14133975
13829543, 17215560, 17694209, 17385178, 18091059, 8322815, 18259031
19689979, 17586955, 17201159, 17655634, 18331812, 19730508, 18868646
17648596, 16220077, 16069901, 17348614, 17393915, 17957017, 17274537
18096714, 17308789, 18436647, 14285317, 19289642, 14764829, 17622427
18328509, 16943711, 22195477, 14368995, 22502493, 17346671, 18996843
17783588, 21343838, 16618694, 17672719, 18856999, 18783224, 17851160
17546761, 22168163, 17798953, 18273830, 22092979, 16596890, 19972566
20828947, 13871092, 26667023, 17726838, 16384983, 22296366, 17360606
22321741, 13645875, 25879656, 18199537, 16542886, 21787056, 17889549
14565184, 20475845, 21281607, 17071721, 17610798, 20299015, 21343897
22893153, 20657441, 17397545, 18230522, 16360112, 19769489, 12905058
18641451, 12747740, 18430495, 25423453, 17016369, 17042658, 14602788
17551063, 19972568, 21517440, 19788842, 18508861, 14657740, 17332800
13837378, 17186905, 19972564, 19699191, 18315328, 17437634, 22353199
18093615, 19006849, 19013183, 17296856, 18674024, 17232014, 16855292
17762296, 14692762, 21051840, 17705023, 23294548, 22507234, 19121551
21330264, 26198926, 19854503, 23315889, 26030218, 21868720, 19309466
18681862, 17365043, 20558005, 18554763, 17390160, 18456514, 16306373
13955826, 18139690, 17501491, 17752121, 21668627, 17299889, 23713236
24652769, 17889583, 18673325, 22551446, 19721304, 18293054, 17242746
19211433, 19888853, 17951233, 18094246, 17649265, 19615136, 17011832
16870214, 17477958, 18522509, 20631274, 16091637, 17323222, 16595641
16524926, 18228645, 18282562, 17596908, 18031668, 17156148, 16494615
22683225, 20869721, 17545847, 25093656, 17655240, 24528741, 17614134
25427662, 13558557, 17341326, 17891946, 17716305, 22657942, 18440095
16392068, 19271443, 21351877, 18092127, 17614227, 18440047, 18849970
16903536, 14106803, 18973907, 18673342, 17389192, 25505382, 19032867
17612828, 16194160, 17006570, 25369547, 25505407, 16685417, 17721717
17390431, 17570240, 16863422, 18325460, 17008068, 19727057, 16422541
19972570, 17267114, 18244962, 21538485, 18203838, 18765602, 16198143
17246576, 14829250, 17835627, 18247991, 14458214, 21051862, 17786278
16692232, 17227277, 24476265, 16042673, 16314254, 19285025, 16228604
16837842, 20144308, 17393683, 23536835, 25823754, 18899974, 17787259
24719736, 20331945, 19490948, 20074391, 15861775, 16399083, 25947799
18018515, 22683212, 21051858, 18260550, 17080436, 16613964, 17036973
16579084, 24433711, 18384537, 18280813, 20296213, 16901385, 15979965
23330124, 18441944, 16450169, 9756271, 17892268, 11733603, 16285691
17587063, 21343775, 26474853, 18180390, 16538760, 18193833, 21387964
21051833, 17238511, 19777862, 23065323, 17824637, 17903598, 16571443
18306996, 19578350, 14852021, 17853456, 18674047, 12364061, 24411921
19207117, 22195448

Version 11.2.0.4.v13

Version 11.2.0.4.v13 includes the following:

- Oracle July 2017 PSU, a combination of database PSU (patch 26609445) + OJVM component PSU (patch 26027154)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 26554712)
- RSA Micro-Edition Suite Bundle (patch 26770426)
- Timezone file DSTv30 (patch 25881255, OJVM patch 25881271)
- Support for [Validating DB instance files \(p. 1072\)](#) with the `RMAN` logical validation utility

- Support for [Setting the default edition for a DB instance \(p. 1057\)](#)

Oracle patch 26609445, released July 2017

Bugs fixed:

17288409, 21051852, 24316947, 17811429, 17205719, 18607546, 20506699
17816865, 25957038, 23330119, 17922254, 17754782, 13364795, 16934803
17311728, 20387265, 17284817, 17441661, 24560906, 16992075, 17446237
14015842, 19972569, 21756677, 17375354, 21538558, 20925795, 17449815
26575788, 19463897, 13866822, 17235750, 17982555, 17478514, 18317531
14338435, 18235390, 20803583, 19461270, 13944971, 20142975, 17811789
16929165, 18704244, 20506706, 17546973, 20334344, 14054676, 17088068
17346091, 18264060, 17343514, 21538567, 19680952, 18471685, 19211724
13951456, 21847223, 16315398, 18744139, 16850630, 23177648, 19049453
18673304, 17883081, 19915271, 18641419, 18262334, 25600421, 17006183
16065166, 18277454, 16833527, 10136473, 18051556, 17865671, 17852463
18554871, 17853498, 18334586, 20879889, 17551709, 17588480, 19827973
17344412, 17842825, 18828868, 20509482, 17025461, 11883252, 13609098
17239687, 17602269, 19197175, 18316692, 22195457, 17313525, 12611721
19544839, 18964939, 17600719, 18191164, 19393542, 17571306, 20777150
18482502, 19466309, 22243719, 17040527, 17165204, 18098207, 16785708
17465741, 16180763, 17174582, 12982566, 16777840, 19463893, 22195465
16875449, 22148226, 12816846, 17237521, 6599380, 19358317, 17811438
25505394, 17811447, 17945983, 21983325, 18762750, 16912439, 17184721
18061914, 17282229, 18331850, 18202441, 17082359, 18723434, 21972320
19554106, 25505371, 14034426, 18339044, 19458377, 17752995, 20448824
17891943, 17258090, 17767676, 16668584, 18384391, 17040764, 17381384
15913355, 18356166, 14084247, 20596234, 20506715, 21756661, 13853126
18203837, 14245531, 16043574, 21756699, 22195441, 17848897, 17877323
21453153, 17468141, 20861693, 17786518, 17912217, 17037130, 16956380
18155762, 17478145, 17394950, 18641461, 18189036, 18619917, 17027426
21352646, 16268425, 24476274, 22195492, 19584068, 26544823, 18436307
22507210, 17265217, 17634921, 13498382, 19469538, 21526048, 19258504
18043064, 20004087, 17443671, 22195485, 18000422, 20004021, 22321756
17571039, 21067387, 16832076, 22905130, 16344544, 18009564, 14354737
21286665, 18135678, 14521849, 18614015, 20441797, 18362222, 25655390
16472716, 17835048, 17050888, 17936109, 14010183, 17325413, 18747196
17761775, 16721594, 17082983, 20067212, 21179898, 17302277, 18084625
15990359, 24842886, 18203835, 17297939, 17811456, 16731148, 22380919
21168487, 14133975, 13829543, 17215560, 17694209, 17385178, 18091059
8322815, 18259031, 19689979, 17586955, 17201159, 17655634, 18331812
19730508, 18868646, 17648596, 16220077, 16069901, 17348614, 17393915
17957017, 17274537, 18096714, 17308789, 18436647, 14285317, 19289642
14764829, 17622427, 18328509, 16943711, 22195477, 14368995, 22502493
17346671, 18996843, 17783588, 21343838, 16618694, 17672719, 18856999
18783224, 17851160, 17546761, 22168163, 17798953, 18273830, 22092979
16596890, 19972566, 13871092, 17726838, 16384983, 22296366, 17360606
22321741, 13645875, 25879656, 18199537, 16542886, 21787056, 17889549
14565184, 17071721, 17610798, 20290105, 21343897, 22893153, 20657441
17397545, 18230522, 16360112, 19769489, 12905058, 18641451, 12747740
18430495, 25423453, 17016369, 17042658, 14602788, 17551063, 19972568
21517440, 19788842, 18508861, 14657740, 17332800, 13837378, 17186905
19972564, 19699191, 18315328, 17437634, 22353199, 18093615, 19006849
19013183, 17296856, 18674024, 17232014, 16855292, 17762296, 14692762
21051840, 17705023, 22507234, 19121551, 21330264, 19854503, 26030218
21868720, 19309466, 18681862, 17365043, 20558005, 18554763, 17390160
18456514, 16306373, 13955826, 18139690, 17501491, 17752121, 21668627
17299889, 17889583, 18673325, 19721304, 18293054, 17242746, 19888853
17951233, 18094246, 17649265, 19615136, 17011832, 16870214, 17477958
18522509, 20631274, 16091637, 17323222, 16595641, 16524926, 18228645
18282562, 17596908, 18031668, 17156148, 16494615, 22683225, 17545847
25093656, 17655240, 24528741, 17614134, 25427662, 13558557, 17341326

17891946, 17716305, 22657942, 18440095, 16392068, 19271443, 21351877
18092127, 17614227, 18440047, 16903536, 14106803, 18973907, 18673342
17389192, 25505382, 19032867, 17612828, 16194160, 17006570, 25369547
25505407, 16685417, 17721717, 17390431, 17570240, 16863422, 18325460
19727057, 16422541, 19972570, 17267114, 18244962, 21538485, 18203838
18765602, 16198143, 17246576, 14829250, 17835627, 18247991, 14458214
21051862, 16692232, 17786278, 17227277, 24476265, 16042673, 16314254
16228604, 16837842, 17393683, 23536835, 25823754, 18899974, 17787259
20331945, 20074391, 15861775, 16399083, 18018515, 22683212, 21051858
18260550, 17080436, 16613964, 17036973, 16579084, 24433711, 18384537
18280813, 20296213, 16901385, 15979965, 23330124, 18441944, 16450169
9756271, 17892268, 11733603, 16285691, 17587063, 21343775, 18180390
16538760, 18193833, 21387964, 21051833, 17238511, 19777862, 17824637
16571443, 18306996, 19578350, 14852021, 17853456, 18674047, 12364061
24411921, 19207117, 22195448

Version 11.2.0.4.v12

Version 11.2.0.4.v12 includes the following:

- Oracle patch 25440428, a combination of database PSU (patch 24732075) + OJVM component PSU (patch 25434033)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 25734992)
- MES Bundle (patch 24975421 for 11.2.0.4)
- Timezone file DSTv28 (patch 24701840)
- Support for the DBMS_CHANGE_NOTIFICATION package
- Support for XSTREAM packages and views (may require additional licensing)

Oracle patch 24732075, released April 2017

Bugs fixed:

17288409, 21051852, 24316947, 17811429, 17205719, 18607546, 20506699
17816865, 17922254, 23330119, 17754782, 16934803, 13364795, 17311728
17284817, 17441661, 24560906, 16992075, 17446237, 14015842, 19972569
21756677, 17375354, 20925795, 21538558, 17449815, 19463897, 13866822
17235750, 17982555, 17478514, 18317531, 14338435, 18235390, 20803583
13944971, 20142975, 17811789, 16929165, 18704244, 20506706, 17546973
20334344, 14054676, 17088068, 17346091, 18264060, 17343514, 21538567
19680952, 18471685, 19211724, 13951456, 21847223, 16315398, 18744139
16850630, 23177648, 19049453, 18673304, 17883081, 19915271, 18641419
18262334, 17006183, 16065166, 18277454, 16833527, 10136473, 18051556
17865671, 17852463, 18554871, 17853498, 18334586, 17551709, 17588480
19827973, 17344412, 17842825, 18828868, 17025461, 11883252, 13609098
17239687, 17602269, 19197175, 18316692, 22195457, 17313525, 12611721
19544839, 18964939, 17600719, 18191164, 19393542, 17571306, 20777150
18482502, 19466309, 22243719, 17040527, 17165204, 18098207, 16785708
17465741, 17174582, 16180763, 12982566, 16777840, 19463893, 22195465
16875449, 12816846, 22148226, 17237521, 6599380, 19358317, 25505394
17811438, 17811447, 17945983, 21983325, 18762750, 16912439, 17184721
18061914, 17282229, 18331850, 18202441, 17082359, 18723434, 21972320
19554106, 25505371, 14034426, 18339044, 19458377, 17752995, 20448824
17891943, 17258090, 17767676, 16668584, 18384391, 17040764, 17381384
15913355, 18356166, 14084247, 20596234, 20506715, 21756661, 13853126
18203837, 14245531, 16043574, 21756699, 22195441, 17848897, 17877323
21453153, 17468141, 20861693, 17786518, 17912217, 17037130, 16956380
18155762, 17478145, 17394950, 18641461, 18189036, 18619917, 17027426

21352646, 16268425, 24476274, 22195492, 19584068, 18436307, 22507210
17265217, 17634921, 13498382, 21526048, 19258504, 20004087, 17443671
22195485, 18000422, 22321756, 20004021, 17571039, 21067387, 22905130
16344544, 18009564, 14354737, 21286665, 18135678, 18614015, 20441797
18362222, 17835048, 16472716, 17936109, 17050888, 14010183, 17325413
18747196, 17761775, 16721594, 17082983, 20067212, 21179898, 17302277
18084625, 15990359, 24842886, 18203835, 17297939, 17811456, 22380919
16731148, 21168487, 14133975, 13829543, 17215560, 17694209, 17385178
18091059, 8322815, 17586955, 17201159, 17655634, 18331812, 19730508
18868646, 17648596, 16220077, 16069901, 17348614, 17393915, 17274537
17957017, 18096714, 17308789, 18436647, 14285317, 19289642, 14764829
17622427, 18328509, 16943711, 22195477, 14368995, 22502493, 17346671
18996843, 17783588, 21343838, 16618694, 17672719, 18856999, 18783224
17851160, 17546761, 17798953, 18273830, 22092979, 16596890, 19972566
16384983, 17726838, 22296366, 17360606, 22321741, 13645875, 18199537
16542886, 21787056, 17889549, 14565184, 17071721, 17610798, 20299015
21343897, 22893153, 20657441, 17397545, 18230522, 16360112, 19769489
12905058, 18641451, 12747740, 18430495, 17016369, 17042658, 14602788
17551063, 19972568, 21517440, 18508861, 19788842, 14657740, 17332800
13837378, 19972564, 17186905, 18315328, 19699191, 17437634, 22353199
18093615, 19006849, 19013183, 17296856, 18674024, 17232014, 16855292
17762296, 14692762, 21051840, 17705023, 22507234, 19121551, 21330264
19854503, 21868720, 19309466, 18681862, 20558005, 18554763, 17390160
18456514, 16306373, 13955826, 18139690, 17501491, 17752121, 21668627
17299889, 17889583, 18673325, 19721304, 18293054, 17242746, 17951233
18094246, 17649265, 19615136, 17011832, 16870214, 17477958, 18522509
20631274, 16091637, 17323222, 16595641, 16524926, 18228645, 18282562
17596908, 18031668, 17156148, 16494615, 22683225, 17545847, 25093656
17655240, 24528741, 17614134, 13558557, 17341326, 17891946, 17716305
22657942, 18440095, 16392068, 19271443, 21351877, 18092127, 17614227
18440047, 16903536, 14106803, 18973907, 18673342, 25505382, 19032867
17389192, 17612828, 16194160, 17006570, 25369547, 25505407, 17721717
17390431, 17570240, 16863422, 18325460, 19727057, 16422541, 19972570
17267114, 18244962, 21538485, 18765602, 18203838, 16198143, 17246576
14829250, 17835627, 18247991, 14458214, 21051862, 16692232, 17786278
17227277, 24476265, 16042673, 16314254, 16228604, 16837842, 17393683
23536835, 17787259, 20331945, 20074391, 15861775, 16399083, 18018515
22683212, 18260550, 21051858, 17080436, 16613964, 17036973, 16579084
24433711, 18384537, 18280813, 20296213, 16901385, 15979965, 23330124
18441944, 16450169, 9756271, 17892268, 11733603, 16285691, 17587063
21343775, 18180390, 16538760, 18193833, 21387964, 21051833, 17238511
17824637, 16571443, 18306996, 14852021, 17853456, 18674047, 12364061
24411921, 22195448

Version 11.2.0.4.v11

Version 11.2.0.4.v11 includes the following:

- Oracle patch 24918033, a combination of database PSU (patch 24006111) + OJVM component PSU (patch 24917954)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 24491261)
- MES Bundle (patch 24975421 for 11.2.0.4)

Oracle patch 24918033, released January 2017

Bugs fixed:

18933818, 19176885, 17201047, 25067795, 14774730, 19153980, 21911849
23727132, 18166577, 24448240, 17056813, 21811517, 19909862, 22675136

24534298, 19895326, 22253904, 17804361, 19231857, 17528315, 19058059
19554117, 19007266, 17285560, 22670385, 18458318, 19187988, 23265914
19006757, 19374518, 19223010, 25076732, 22118835, 19852360, 20408829
21047766, 21566944,
17288409, 21051852, 24316947, 17811429, 18607546, 17205719, 20506699
17816865, 17922254, 23330119, 17754782, 16934803, 13364795, 17311728
17441661, 17284817, 16992075, 17446237, 14015842, 19972569, 21756677
17375354, 20925795, 21538558, 17449815, 19463897, 13866822, 17235750
17982555, 17478514, 18317531, 14338435, 18235390, 20803583, 13944971
20142975, 17811789, 16929165, 18704244, 20506706, 17546973, 20334344
14054676, 17088068, 17346091, 18264060, 17343514, 21538567, 19680952
18471685, 19211724, 13951456, 21847223, 16315398, 18744139, 16850630
23177648, 19049453, 18673304, 17883081, 19915271, 18641419, 18262334
17006183, 16065166, 18277454, 16833527, 10136473, 18051556, 17865671
17852463, 18554871, 17853498, 18334586, 17551709, 17588480, 19827973
17344412, 17842825, 18828868, 17025461, 11883252, 13609098, 17239687
17602269, 19197175, 22195457, 18316692, 17313525, 12611721, 19544839
18964939, 17600719, 18191164, 19393542, 17571306, 20777150, 18482502
19466309, 22243719, 17040527, 17165204, 18098207, 16785708, 17465741
17174582, 16180763, 16777840, 12982566, 19463893, 22195465, 22148226
16875449, 12816846, 17237521, 65999380, 19358317, 17811438, 17811447
17945983, 21983325, 18762750, 16912439, 17184721, 18061914, 17282229
18331850, 18202441, 17082359, 18723434, 21972320, 19554106, 14034426
18339044, 19458377, 17752995, 20448824, 17891943, 17258090, 17767676
16668584, 18384391, 17040764, 17381384, 15913355, 18356166, 14084247
20596234, 20506715, 21756661, 13853126, 18203837, 14245531, 16043574
21756699, 22195441, 17848897, 17877323, 21453153, 17468141, 20861693
17786518, 17912217, 17037130, 16956380, 18155762, 17478145, 17394950
18641461, 18189036, 18619917, 17027426, 21352646, 16268425, 24476274
22195492, 19584068, 18436307, 22507210, 17265217, 17634921, 13498382
21526048, 19258504, 20004087, 17443671, 22195485, 18000422, 22321756
20004021, 17571039, 21067387, 16344544, 18009564, 14354737, 21286665
18135678, 18614015, 20441797, 18362222, 17835048, 16472716, 17936109
17050888, 17325413, 14010183, 18747196, 17761775, 16721594, 17082983
20067212, 21179898, 17302277, 18084625, 15990359, 18203835, 17297939
17811456, 22380919, 16731148, 21168487, 14133975, 13829543, 17215560
17694209, 17385178, 18091059, 8322815, 17586955, 17201159, 17655634
18331812, 19730508, 18868646, 17648596, 16220077, 16069901, 17348614
17393915, 17274537, 17957017, 18096714, 17308789, 18436647, 14285317
19289642, 14764829, 18328509, 17622427, 16943711, 22195477, 14368995
22502493, 17346671, 18996843, 17783588, 21343838, 16618694, 17672719
18856999, 18783224, 17851160, 17546761, 17798953, 18273830, 22092979
16596890, 19972566, 16384983, 17726838, 22296366, 17360606, 22321741
13645875, 18199537, 16542886, 21787056, 17889549, 14565184, 17071721
17610798, 20299015, 21343897, 22893153, 20657441, 17397545, 18230522
16360112, 19769489, 12905058, 18641451, 12747740, 18430495, 17016369
17042658, 14602788, 17551063, 19972568, 21517440, 18508861, 19788842
14657740, 17332800, 13837378, 19972564, 17186905, 18315328, 19699191
17437634, 22353199, 18093615, 19006849, 19013183, 17296856, 18674024
17232014, 16855292, 17762296, 14692762, 21051840, 17705023, 22507234
19121551, 21330264, 19854503, 21868720, 19309466, 18681862, 20558005
18554763, 17390160, 18456514, 16306373, 13955826, 18139690, 17501491
17752121, 21668627, 17299889, 17889583, 18673325, 19721304, 18293054
17242746, 17951233, 18094246, 17649265, 19615136, 17011832, 16870214
17477958, 18522509, 20631274, 16091637, 17323222, 16595641, 16524926
18228645, 18282562, 17596908, 18031668, 17156148, 16494615, 22683225
17545847, 17655240, 24528741, 17614134, 13558557, 17341326, 17891946
17716305, 22657942, 16392068, 19271443, 21351877, 18092127, 17614227
18440047, 16903536, 14106803, 18973907, 18673342, 19032867, 17389192
17612828, 16194160, 17006570, 17721717, 17390431, 17570240, 16863422
18325460, 19727057, 16422541, 19972570, 17267114, 18244962, 21538485
18765602, 18203838, 16198143, 17246576, 14829250, 17835627, 18247991
14458214, 21051862, 16692232, 17786278, 17227277, 24476265, 16042673
16314254, 16228604, 16837842, 17393683, 23536835, 17787259, 20331945
20074391, 15861775, 16399083, 18018515, 22683212, 18260550, 21051858
17080436, 16613964, 17036973, 16579084, 24433711, 18384537, 18280813

20296213, 16901385, 15979965, 23330124, 18441944, 16450169, 9756271
17892268, 11733603, 16285691, 17587063, 21343775, 18180390, 16538760
18193833, 21387964, 21051833, 17238511, 17824637, 16571443, 18306996
14852021, 17853456, 18674047, 12364061, 22195448

Version 11.2.0.4.v10

Version 11.2.0.4.v10 includes the following:

- Oracle patch 24436313, a combination of database PSU (patch 24006111) + OJVM component PSU (patch 24315821)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 24491261)
- MES Bundle (patch 24975421 for 11.2.0.4)

Baseline: Oracle database patch set update 11.2.0.4.161018 (patch 24006111, released October 2016)

Bugs fixed:

17288409, 21051852, 24316947, 17811429, 18607546, 17205719, 20506699
17816865, 17922254, 23330119, 17754782, 16934803, 13364795, 17311728
17441661, 17284817, 16992075, 17446237, 14015842, 19972569, 21756677
17375354, 20925795, 21538558, 17449815, 19463897, 13866822, 17235750
17982555, 17478514, 18317531, 14338435, 18235390, 20803583, 13944971
20142975, 17811789, 16929165, 18704244, 20506706, 17546973, 20334344
14054676, 17088068, 17346091, 18264060, 17343514, 21538567, 19680952
18471685, 19211724, 13951456, 21847223, 16315398, 18744139, 16850630
23177648, 19049453, 18673304, 17883081, 19915271, 18641419, 18262334
17006183, 16065166, 18277454, 16833527, 10136473, 18051556, 17865671
17852463, 18554871, 17853498, 18334586, 17551709, 17588480, 19827973
17344412, 17842825, 18828868, 17025461, 11883252, 13609098, 17239687
17602269, 19197175, 22195457, 18316692, 17313525, 12611721, 19544839
18964939, 17600719, 18191164, 19393542, 17571306, 20777150, 18482502
19466309, 22243719, 17040527, 17165204, 18098207, 16785708, 17465741
17174582, 16180763, 16777840, 12982566, 19463893, 22195465, 22148226
16875449, 12816846, 17237521, 6599380, 19358317, 17811438, 17811447
17945983, 21983325, 18762750, 16912439, 17184721, 18061914, 17282229
18331850, 18202441, 17082359, 18723434, 21972320, 19554106, 14034426
18339044, 19458377, 17752995, 20448824, 17891943, 17258090, 17767676
16668584, 18384391, 17040764, 17381384, 15913355, 18356166, 14084247
20596234, 20506715, 21756661, 13853126, 18203837, 14245531, 16043574
21756699, 22195441, 17848897, 17877323, 21453153, 17468141, 20861693
17786518, 17912217, 17037130, 16956380, 18155762, 17478145, 17394950
18641461, 18189036, 18619917, 17027426, 21352646, 16268425, 24476274
22195492, 19584068, 18436307, 22507210, 17265217, 17634921, 13498382
21526048, 19258504, 20004087, 17443671, 22195485, 18000422, 22321756
20004021, 17571039, 21067387, 16344544, 18009564, 14354737, 21286665
18135678, 18614105, 20441797, 18362222, 17835048, 16472716, 17936109
17050888, 17325413, 14010183, 18747196, 17761775, 16721594, 17082983
20067212, 21179898, 17302277, 18084625, 15990359, 18203835, 17297939
17811456, 22380919, 16731148, 21168487, 14133975, 13829543, 17215560
17694209, 17385178, 18091059, 8322815, 17586955, 17201159, 17655634
18331812, 19730508, 18868646, 17648596, 16220077, 16069901, 17348614
17393915, 17274537, 17957017, 18096714, 17308789, 18436647, 14285317
19289642, 14764829, 18328509, 17622427, 16943711, 22195477, 14368995
22502493, 17346671, 18996843, 17783588, 21343838, 16618694, 17672719
18856999, 18783224, 17851160, 17546761, 17798953, 18273830, 22092979
16596890, 19972566, 16384983, 17726838, 22296366, 17360606, 22321741

13645875, 18199537, 16542886, 21787056, 17889549, 14565184, 17071721
17610798, 20299015, 21343897, 22893153, 20657441, 17397545, 18230522
16360112, 19769489, 12905058, 18641451, 12747740, 18430495, 17016369
17042658, 14602788, 17551063, 19972568, 21517440, 18508861, 19788842
14657740, 17332800, 13837378, 19972564, 17186905, 18315328, 19699191
17437634, 22353199, 18093615, 19006849, 19013183, 17296856, 18674024
17232014, 16855292, 17762296, 14692762, 21051840, 17705023, 22507234
19121551, 21330264, 19854503, 21868720, 19309466, 18681862, 20558005
18554763, 17390160, 18456514, 16306373, 13955826, 18139690, 17501491
17752121, 21668627, 17299889, 17889583, 18673325, 19721304, 18293054
17242746, 17951233, 18094246, 17649265, 19615136, 17011832, 16870214
17477958, 18522509, 20631274, 16091637, 17323222, 16595641, 16524926
18228645, 18282562, 17596908, 18031668, 17156148, 16494615, 22683225
17545847, 17655240, 24528741, 17614134, 13558557, 17341326, 17891946
17716305, 22657942, 16392068, 19271443, 21351877, 18092127, 17614227
18440047, 16903536, 14106803, 18973907, 18673342, 19032867, 17389192
17612828, 16194160, 17006570, 17721717, 17390431, 17570240, 16863422
18325460, 19727057, 16422541, 19972570, 17267114, 18244962, 21538485
18765602, 18203838, 16198143, 17246576, 14829250, 17835627, 18247991
14458214, 21051862, 16692232, 17786278, 17227277, 24476265, 16042673
16314254, 16228604, 16837842, 17393683, 23536835, 17787259, 20331945
20074391, 15861775, 16399083, 18018515, 22683212, 18260550, 21051858
17080436, 16613964, 17036973, 16579084, 24433711, 18384537, 18280813
20296213, 16901385, 15979965, 23330124, 18441944, 16450169, 9756271
17892268, 11733603, 16285691, 17587063, 21343775, 18180390, 16538760
18193833, 21387964, 21051833, 17238511, 17824637, 16571443, 18306996
14852021, 17853456, 18674047, 12364061, 22195448

Version 11.2.0.4.v9

Version 11.2.0.4.v9 includes the following:

- Oracle patch 23615392, a combination of database PSU (patch 23054359) + OJVM component PSU (patch 23177551)
- Timezone file DSTv26 (patch 22873635 for 11.2.0.4)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 24320398 for 11.2.0.4.160719)
- MES Bundle (patch 22695784 for 11.2.0.4)
- Added the ability to create custom password verify functions. For more information, see [Creating custom functions to verify passwords \(p. 1042\)](#).
- Fixed a bug that prevented implicit recompilation of views owned by SYS

Baseline: Oracle database patch set update 11.2.0.4.160719 (patch 23054359, released July 2016)

Bugs fixed:

17288409, 21051852, 17811429, 18607546, 17205719, 20506699, 17816865
23330119, 17922254, 17754782, 16934803, 13364795, 17311728, 17441661
17284817, 16992075, 17446237, 14015842, 19972569, 21756677, 17375354
21538558, 20925795, 17449815, 19463897, 13866822, 17982555, 17235750
17478514, 18317531, 14338435, 18235390, 20803583, 13944971, 20142975
17811789, 16929165, 18704244, 20506706, 17546973, 20334344, 14054676
17088068, 17346091, 18264060, 17343514, 21538567, 19680952, 18471685
19211724, 13951456, 21847223, 16315398, 18744139, 16850630, 23177648
19049453, 18673304, 17883081, 19915271, 18641419, 18262334, 17006183
16065166, 18277454, 16833527, 10136473, 18051556, 17865671, 17852463
18554871, 17853498, 18334586, 17551709, 17588480, 19827973, 17344412

17842825, 18828868, 17025461, 11883252, 13609098, 17239687, 17602269
19197175, 22195457, 18316692, 17313525, 12611721, 19544839, 18964939
17600719, 18191164, 19393542, 17571306, 18482502, 20777150, 19466309
17040527, 17165204, 18098207, 16785708, 17465741, 17174582, 16180763
16777840, 12982566, 19463893, 22195465, 16875449, 12816846, 17237521
19358317, 17811438, 17811447, 17945983, 21983325, 18762750, 16912439
17184721, 18061914, 17282229, 18331850, 18202441, 17082359, 18723434
21972320, 19554106, 14034426, 18339044, 19458377, 17752995, 20448824
17891943, 17258090, 17767676, 16668584, 18384391, 17040764, 17381384
15913355, 18356166, 14084247, 20596234, 20506715, 21756661, 13853126
18203837, 14245531, 16043574, 21756699, 22195441, 17848897, 17877323
21453153, 17468141, 20861693, 17786518, 17912217, 17037130, 16956380
18155762, 17478145, 17394950, 18641461, 18189036, 18619917, 17027426
21352646, 16268425, 22195492, 19584068, 18436307, 22507210, 17265217
17634921, 13498382, 21526048, 19258504, 20004087, 17443671, 22195485
18000422, 22321756, 20004021, 17571039, 21067387, 16344544, 18009564
14354737, 21286665, 18135678, 18614015, 20441797, 18362222, 17835048
16472716, 17936109, 17050888, 17325413, 14010183, 18747196, 17761775
16721594, 17082983, 20067212, 21179898, 17302277, 18084625, 15990359
18203835, 17297939, 22380919, 17811456, 16731148, 21168487, 13829543
17215560, 14133975, 17694209, 17385178, 18091059, 8322815, 17586955
17201159, 17655634, 18331812, 19730508, 18868646, 17648596, 16220077
16069901, 17348614, 17393915, 17274537, 17957017, 18096714, 17308789
18436647, 14285317, 19289642, 14764829, 18328509, 17622427, 16943711
22195477, 14368995, 22502493, 17346671, 18996843, 17783588, 21343838
16618694, 17672719, 18856999, 18783224, 17851160, 17546761, 17798953
18273830, 22092979, 16596890, 19972566, 16384983, 17726838, 22296366
17360606, 22321741, 13645875, 18199537, 16542886, 21787056, 17889549
14565184, 17071721, 17610798, 20299015, 21343897, 22893153, 20657441
17397545, 18230522, 16360112, 19769489, 12905058, 18641451, 12747740
18430495, 17016369, 17042658, 14602788, 17551063, 19972568, 21517440
18508861, 19788842, 14657740, 17332800, 13837378, 19972564, 17186905
18315328, 19699191, 17437634, 22353199, 18093615, 19006849, 19013183
17296856, 18674024, 17232014, 16855292, 17762296, 14692762, 21051840
17705023, 22507234, 19121551, 21330264, 19854503, 21868720, 19309466
18681862, 18554763, 20558005, 17390160, 18456514, 16306373, 13955826
18139690, 17501491, 17752121, 21668627, 17299889, 17889583, 18673325
19721304, 18293054, 17242746, 17951233, 18094246, 17649265, 19615136
17011832, 16870214, 17477958, 18522509, 20631274, 16091637, 17323222
16595641, 16524926, 18228645, 18282562, 17596908, 18031668, 17156148
16494615, 22683225, 17545847, 17655240, 17614134, 13558557, 17341326
17891946, 17716305, 16392068, 19271443, 21351877, 18092127, 17614227
18440047, 16903536, 14106803, 18973907, 18673342, 19032867, 17389192
17612828, 16194160, 17006570, 17721717, 17390431, 17570240, 16863422
18325460, 19727057, 16422541, 19972570, 17267114, 18244962, 21538485
18765602, 18203838, 16198143, 17246576, 14829250, 17835627, 18247991
14458214, 21051862, 16692232, 17786278, 17227277, 16042673, 16314254
16228604, 16837842, 17393683, 23536835, 17787259, 20331945, 20074391
15861775, 16399083, 18018515, 22683212, 18260550, 21051858, 17080436
16613964, 17036973, 16579084, 18384537, 18280813, 20296213, 16901385
15979965, 23330124, 18441944, 16450169, 9756271, 17892268, 11733603
16285691, 17587063, 21343775, 16538760, 18180390, 18193833, 21387964
21051833, 17238511, 17824637, 16571443, 18306996, 14852021, 17853456
18674047, 12364061, 22195448

Version 11.2.0.4.v8

Version 11.2.0.4.v8 includes the following:

- Oracle PSU 11.2.0.4.160419 (22502456)
- Timezone file DSTv25 (patch 22037014)
- Oracle recommended RDBMS patches for Oracle GoldenGate (patch 22576728)

- MES Bundle (patch 22695784 for 11.2.0.4)
- Adds the ability for the master user to grant privileges on SYS objects with the grant option using the RDSADMIN.RDSADMIN_UTIL.GRANT_SYS_OBJECT procedure
- Adds master user privileges to support most common schemas created by the Oracle Fusion Middleware Repository Creation Utility (RCU)

Baseline: Oracle database patch set update 11.2.0.4.160419 (patch 22502456, released April 2016)

Bugs fixed:

17288409, 21051852, 17811429, 18607546, 17205719, 20506699, 17816865 17922254, 17754782, 16934803, 13364795, 17311728, 17441661, 17284817 16992075, 17446237, 14015842, 19972569, 21756677, 21538558, 20925795 17449815, 17375354, 19463897, 13866822, 17982555, 17235750, 17478514 18317531, 14338435, 18235390, 20803583, 13944971, 20142975, 17811789 16929165, 18704244, 20506706, 17546973, 20334344, 14054676, 17088068 17346091, 18264060, 17343514, 21538567, 19680952, 18471685, 19211724 13951456, 21847223, 16315398, 18744139, 16850630, 19049453, 18673304 17883081, 19915271, 18641419, 18262334, 17006183, 16065166, 18277454 16833527, 10136473, 18051556, 17865671, 17852463, 18554871, 17853498 18334586, 17551709, 17588480, 19827973, 17344412, 17842825, 18828868 17025461, 11883252, 13609098, 17239687, 17602269, 19197175, 22195457 18316692, 17313525, 12611721, 19544839, 18964939, 17600719, 18191164 19393542, 17571306, 18482502, 20777150, 19466309, 17040527, 17165204 18098207, 16785708, 17465741, 17174582, 16180763, 16777840, 12982566 19463893, 22195465, 16875449, 12816846, 17237521, 19358317, 17811438 17811447, 21983325, 17945983, 18762750, 16912439, 17184721, 18061914 17282229, 18331850, 18202441, 17082359, 18723434, 21972320, 19554106 14034426, 18339044, 19458377, 17752995, 20448824, 17891943, 17258090 17767676, 16668584, 18384391, 17040764, 17381384, 15913355, 18356166 14084247, 20596234, 20506715, 21756661, 13853126, 18203837, 14245531 21756699, 16043574, 22195441, 17848897, 17877323, 21453153, 17468141 20861693, 17786518, 17912217, 17037130, 18155762, 16956380, 17478145 17394950, 18641461, 18189036, 18619917, 17027426, 21352646, 16268425 22195492, 19584068, 18436307, 17265217, 17634921, 13498382, 21526048 19258504, 20004087, 17443671, 22195485, 18000422, 20004021, 22321756 17571039, 21067387, 16344544, 18009564, 14354737, 21286665, 18135678 18614015, 20441797, 18362222, 17835048, 16472716, 17936109, 17050888 17325413, 14010183, 18747196, 17761775, 16721594, 17082983, 20067212 21179898, 17302277, 18084625, 15990359, 18203835, 17297939, 17811456 16731148, 21168487, 13829543, 17215560, 14133975, 17694209, 17385178 18091059, 8322815, 17586955, 17201159, 17655634, 18331812, 19730508 18868646, 17648596, 16220077, 16069901, 17348614, 17393915, 17274537 17957017, 18096714, 17308789, 18436647, 14285317, 19289642, 14764829 18328509, 17622427, 22195477, 16943711, 22502493, 14368995, 17346671 18996843, 17783588, 21343838, 16618694, 17672719, 18856999, 18783224 17851160, 17546761, 17798953, 18273830, 22092979, 16596890, 19972566 16384983, 17726838, 17360606, 22321741, 13645875, 18199537, 16542886 21787056, 17889549, 14565184, 17071721, 17610798, 20299015, 21343897 22893153, 20657441, 17397545, 18230522, 16360112, 19769489, 12905058 18641451, 12747740, 18430495, 17016369, 17042658, 14602788, 17551063 19972568, 21517440, 18508861, 19788842, 14657740, 17332800, 13837378 19972564, 17186905, 18315328, 19699191, 17437634, 22353199, 18093615 19006849, 19013183, 17296856, 18674024, 17232014, 16855292, 17762296 14692762, 21051840, 17705023, 19121551, 21330264, 19854503, 21868720 19309466, 18681862, 18554763, 20558005, 17390160, 18456514, 16306373 13955826, 18139690, 17501491, 17752121, 21668627, 17299889, 17889583 18673325, 19721304, 18293054, 17242746, 17951233, 17649265, 18094246 19615136, 17011832, 16870214, 17477958, 18522509, 20631274, 16091637 17323222, 16595641, 16524926, 18228645, 18282562, 17596908, 17156148

18031668, 16494615, 22683225, 17545847, 17655240, 17614134, 13558557
17341326, 17891946, 17716305, 16392068, 19271443, 21351877, 18092127
18440047, 17614227, 14106803, 16903536, 18973907, 18673342, 19032867
17389192, 17612828, 16194160, 17006570, 17721717, 17390431, 17570240
16863422, 18325460, 19727057, 16422541, 19972570, 17267114, 18244962
21538485, 18765602, 18203838, 16198143, 17246576, 14829250, 17835627
18247991, 14458214, 21051862, 16692232, 17786278, 17227277, 16042673
16314254, 16228604, 16837842, 17393683, 17787259, 20331945, 20074391
15861775, 16399083, 18018515, 22683212, 18260550, 21051858, 17036973
16613964, 17080436, 16579084, 18384537, 18280813, 20296213, 16901385
15979965, 18441944, 16450169, 9756271, 17892268, 11733603, 16285691
17587063, 21343775, 16538760, 18180390, 18193833, 21387964, 21051833
17238511, 17824637, 16571443, 18306996, 14852021, 18674047, 17853456
12364061, 22195448

Version 11.2.0.4.v7

Version 11.2.0.4.v7 includes the following:

- Oracle PSU 11.2.0.4.160119 (21948347)
- Timezone file DSTv25 - patch 22037014 for 11.2.0.4 and 12.1.0.2 (12.1.0.1 includes DSTv24, patch 20875898 (unchanged from 12.1.0.1.v3), as a backport of DSTv25 was unavailable at build time)
- Fixed an issue that prevented customers from creating more than 10 Directory objects in the database
- Fixed an issue that prevented customers from re-granting read privileges on the ADUMP and BDUMP Directory objects

Baseline: Oracle database patch set update 11.2.0.4.160119 (patch 21948347, released January 2016)

Bugs fixed:

17288409, 21051852, 18607546, 17205719, 17811429, 17816865, 20506699
17922254, 17754782, 16934803, 13364795, 17311728, 17441661, 17284817
16992075, 17446237, 14015842, 19972569, 17449815, 21538558, 20925795
17375354, 19463897, 17982555, 17235750, 13866822, 17478514, 18317531
18235390, 14338435, 20803583, 13944971, 20142975, 17811789, 16929165
18704244, 20506706, 17546973, 20334344, 14054676, 17088068, 18264060
17346091, 17343514, 21538567, 19680952, 18471685, 19211724, 13951456
21847223, 16315398, 18744139, 16850630, 19049453, 18673304, 17883081
19915271, 18641419, 18262334, 17006183, 16065166, 18277454, 16833527
10136473, 18051556, 17865671, 17852463, 18554871, 17853498, 18334586
17588480, 17551709, 19827973, 17842825, 17344412, 18828868, 17025461
11883252, 13609098, 17239687, 17602269, 19197175, 22195457, 18316692
17313525, 12611721, 19544839, 18964939, 17600719, 18191164, 19393542
17571306, 18482502, 20777150, 19466309, 17040527, 17165204, 18098207
16785708, 17174582, 16180763, 17465741, 16777840, 12982566, 19463893
22195465, 12816846, 16875449, 17237521, 19358317, 17811438, 17811447
17945983, 18762750, 17184721, 16912439, 18061914, 17282229, 18331850
18202441, 17082359, 18723434, 21972320, 19554106, 14034426, 18339044
19458377, 17752995, 20448824, 17891943, 17258090, 17767676, 16668584
18384391, 17040764, 17381384, 15913355, 18356166, 14084247, 20506715
13853126, 18203837, 14245531, 21756699, 16043574, 22195441, 17848897
17877323, 21453153, 17468141, 20861693, 17786518, 17912217, 17037130
18155762, 16956380, 17478145, 17394950, 18189036, 18641461, 18619917
17027426, 21352646, 16268425, 22195492, 19584068, 18436307, 17265217
17634921, 13498382, 21526048, 20004087, 22195485, 17443671, 18000422
22321756, 20004021, 17571039, 21067387, 16344544, 18009564, 14354737
18135678, 18614015, 20441797, 18362222, 17835048, 16472716, 17936109

17050888, 17325413, 14010183, 18747196, 17761775, 16721594, 17082983
20067212, 21179898, 17302277, 18084625, 15990359, 18203835, 17297939
17811456, 16731148, 21168487, 17215560, 13829543, 14133975, 17694209
18091059, 17385178, 8322815, 17586955, 17201159, 17655634, 18331812
19730508, 18868646, 17648596, 16220077, 16069901, 17348614, 17393915
17274537, 17957017, 18096714, 17308789, 18436647, 14285317, 19289642
14764829, 18328509, 17622427, 22195477, 16943711, 14368995, 17346671
18996843, 17783588, 21343838, 16618694, 17672719, 18856999, 18783224
17851160, 17546761, 17798953, 18273830, 22092979, 19972566, 16384983
17726838, 17360606, 22321741, 13645875, 18199537, 16542886, 21787056
17889549, 14565184, 17071721, 17610798, 20299015, 21343897, 20657441
17397545, 18230522, 16360112, 19769489, 12905058, 18641451, 12747740
18430495, 17042658, 17016369, 14602788, 17551063, 19972568, 21517440
18508861, 19788842, 14657740, 17332800, 13837378, 19972564, 17186905
18315328, 19699191, 17437634, 19006849, 19013183, 17296856, 18674024
17232014, 16855292, 21051840, 14692762, 17762296, 17705023, 19121551
21330264, 19854503, 19309466, 18681862, 18554763, 20558005, 17390160
18456514, 16306373, 13955826, 18139690, 17501491, 21668627, 17299889
17752121, 17889583, 18673325, 18293054, 17242746, 17951233, 17649265
18094246, 19615136, 17011832, 16870214, 17477958, 18522509, 20631274
16091637, 17323222, 16595641, 16524926, 18228645, 18282562, 17596908
17156148, 18031668, 16494615, 17545847, 17655240, 17614134, 13558557
17341326, 17891946, 17716305, 16392068, 19271443, 21351877, 18092127
18440047, 17614227, 14106803, 16903536, 18973907, 18673342, 19032867
17389192, 17612828, 16194160, 17006570, 17721717, 17570240, 17390431
16863422, 18325460, 19727057, 16422541, 19972570, 17267114, 18244962
21538485, 18765602, 18203838, 16198143, 17246576, 14829250, 17835627
18247991, 14458214, 21051862, 16692232, 17786278, 17227277, 16042673
16314254, 16228604, 16837842, 17393683, 17787259, 20331945, 20074391
15861775, 16399083, 18018515, 21051858, 18260550, 17036973, 16613964
17080436, 16579084, 18384537, 18280813, 20296213, 16901385, 15979965
18441944, 16450169, 9756271, 17892268, 11733603, 16285691, 17587063
21343775, 16538760, 18180390, 18193833, 21051833, 17238511, 17824637
16571443, 18306996, 14852021, 18674047, 17853456, 12364061, 22195448

Version 11.2.0.4.v6

Version 11.2.0.4.v6 includes the following:

- Enable SSL encryption for Standard Edition and Standard Edition One

Version 11.2.0.4.v5

Version 11.2.0.4.v5 includes the following:

- Oracle PSU 11.2.0.4.8 (21352635)
- Includes the Daylight Saving Time Patch, patch 20875898: DST-24, that came out after the April 2015 PSU.

Baseline: Oracle database patch set update 11.2.0.4.8 (patch 21352635, released October 2015)

Bugs fixed:

17288409, 21051852, 18607546, 17205719, 17811429, 17816865, 20506699
17922254, 17754782, 16934803, 13364795, 17311728, 17441661, 17284817

16992075, 17446237, 14015842, 19972569, 21538558, 20925795, 17449815
17375354, 19463897, 17982555, 17235750, 13866822, 18317531, 17478514
18235390, 14338435, 20803583, 13944971, 20142975, 17811789, 16929165
18704244, 20506706, 17546973, 20334344, 14054676, 17088068, 18264060
17346091, 17343514, 21538567, 19680952, 18471685, 19211724, 13951456
16315398, 18744139, 16850630, 19049453, 18673304, 17883081, 19915271
18641419, 18262334, 17006183, 16065166, 18277454, 16833527, 10136473
18051556, 17865671, 17852463, 18554871, 17853498, 18334586, 17588480
17551709, 19827973, 17842825, 17344412, 18828868, 17025461, 11883252
13609098, 17239687, 17602269, 19197175, 18316692, 17313525, 12611721
19544839, 18964939, 17600719, 18191164, 19393542, 17571306, 18482502
20777150, 19466309, 17040527, 17165204, 18098207, 16785708, 17174582
16180763, 17465741, 16777840, 12982566, 19463893, 12816846, 16875449
17237521, 19358317, 17811438, 17811447, 17945983, 18762750, 17184721
16912439, 18061914, 17282229, 18331850, 18202441, 17082359, 18723434
19554106, 14034426, 18339044, 19458377, 17752995, 20448824, 17891943
17258090, 17767676, 16668584, 18384391, 17040764, 17381384, 15913355
18356166, 14084247, 20506715, 13853126, 18203837, 14245531, 16043574
17848897, 17877323, 17468141, 17786518, 17912217, 17037130, 18155762
16956380, 17478145, 17394950, 18189036, 18641461, 18619917, 17027426
21352646, 16268425, 19584068, 18436307, 17265217, 17634921, 13498382
20004087, 17443671, 18000422, 20004021, 17571039, 21067387, 16344544
18009564, 14354737, 18135678, 18614015, 20441797, 18362222, 17835048
16472716, 17936109, 17050888, 17325413, 14010183, 18747196, 17761775
16721594, 17082983, 20067212, 21179898, 17302277, 18084625, 15990359
18203835, 17297939, 17811456, 16731148, 17215560, 13829543, 14133975
17694209, 18091059, 17385178, 8322815, 17586955, 17201159, 17655634
18331812, 19730508, 18868646, 17648596, 16220077, 16069901, 17348614
17393915, 17274537, 17957017, 18096714, 17308789, 18436647, 14285317
19289642, 14764829, 18328509, 17622427, 16943711, 14368995, 17346671
18996843, 17783588, 16618694, 17672719, 18856999, 18783224, 17851160
17546761, 17798953, 18273830, 19972566, 16384983, 17726838, 17360606
13645875, 18199537, 16542886, 17889549, 14565184, 17071721, 20299015
17610798, 20657441, 17397545, 18230522, 16360112, 19769489, 12905058
18641451, 12747740, 18430495, 17042658, 17016369, 14602788, 19972568
18508861, 19788842, 14657740, 17332800, 13837378, 19972564, 17186905
18315328, 19699191, 17437634, 19006849, 19013183, 17296856, 18674024
17232014, 16855292, 21051840, 14692762, 17762296, 17705023, 19121551
19854503, 19309466, 18681862, 18554763, 20558005, 17390160, 18456514
16306373, 13955826, 18139690, 17501491, 17299889, 17752121, 17889583
18673325, 18293054, 17242746, 17951233, 17649265, 18094246, 19615136
17011832, 16870214, 17477958, 18522509, 20631274, 16091637, 17323222
16595641, 16524926, 18228645, 18282562, 17596908, 17156148, 18031668
16494615, 17545847, 17614134, 13558557, 17341326, 17891946, 17716305
16392068, 19271443, 18092127, 18440047, 17614227, 14106803, 16903536
18973907, 18673342, 17389192, 16194160, 17006570, 17612828, 17721717
17570240, 17390431, 16863422, 18325460, 19727057, 16422541, 19972570
17267114, 18244962, 21538485, 18765602, 18203838, 16198143, 17246576
14829250, 17835627, 18247991, 14458214, 21051862, 16692232, 17786278
17227277, 16042673, 16314254, 16228604, 16837842, 17393683, 17787259
20331945, 20074391, 15861775, 16399083, 18018515, 18260550, 21051858
17036973, 16613964, 17080436, 16579084, 18384537, 18280813, 20296213
16901385, 15979965, 18441944, 16450169, 9756271, 17892268, 11733603
16285691, 17587063, 16538760, 18180390, 18193833, 21051833, 17238511
17824637, 16571443, 18306996, 14852021, 18674047, 17853456, 12364061

Version 11.2.0.4.v4

Version 11.2.0.4.v4 includes the following:

- Oracle PSU 11.2.0.4.6 (20299013)
- Installs additional Oracle Text knowledge bases from Oracle Database. Examples media (English and French)

- Provides access to DBMS_REPAIR through RDSADMIN.RDSADMIN_DBMS_REPAIR
- Grants ALTER DATABASE LINK, ALTER PUBLIC DATABASE LINK, EXEMPT ACCESS POLICY, EXEMPT IDENTITY POLICY, and EXEMPT REDACTION POLICY to master user

Baseline: Oracle database patch set update 11.2.0.4.6 (patch 20299013, released April 2015)

Bugs fixed:

```
17288409, 17798953, 18273830, 18607546, 17811429, 17205719, 20506699
17816865, 19972566, 17922254, 17754782, 16384983, 17726838, 13364795
16934803, 17311728, 17284817, 17441661, 17360606, 13645875, 18199537
16992075, 16542886, 17446237, 14015842, 17889549, 14565184, 19972569
17071721, 20299015, 17610798, 17375354, 17449815, 17397545, 19463897
18230522, 13866822, 17235750, 17982555, 16360112, 18317531, 17478514
19769489, 12905058, 14338435, 18235390, 13944971, 18641451, 20142975
17811789, 16929165, 18704244, 12747740, 18430495, 20506706, 17546973
14054676, 17088068, 17346091, 18264060, 17016369, 17042658, 17343514
14602788, 19972568, 19680952, 18471685, 19788842, 18508861, 14657740
17332800, 19211724, 13837378, 13951456, 16315398, 17186905, 18744139
19972564, 16850630, 18315328, 17437634, 19049453, 18673304, 17883081
19006849, 19915271, 19013183, 18641419, 17296856, 18674024, 18262334
17006183, 18277454, 16833527, 17232014, 16855292, 10136473, 17762296
14692762, 17705023, 18051556, 17865671, 17852463, 18554871, 17853498
19121551, 18334586, 19854503, 17551709, 19309466, 17588480, 19827973
17344412, 17842825, 18828868, 18681862, 18554763, 17390160, 18456514
16306373, 17025461, 13955826, 18139690, 11883252, 13609098, 17501491
17239687, 17752121, 17299889, 17602269, 19197175, 17889583, 18316692
17313525, 18673325, 12611721, 19544839, 18293054, 17242746, 18964939
17600719, 18191164, 19393542, 17571306, 18482502, 19466309, 17951233
17649265, 18094246, 19615136, 17040527, 17011832, 17165204, 18098207
16785708, 16870214, 17465741, 16180763, 17174582, 17477958, 12982566
16777840, 18522509, 20631274, 16091637, 17323222, 19463893, 16595641
16875449, 12816846, 16524926, 17237521, 18228645, 18282562, 17596908
19358317, 17811438, 17811447, 17945983, 18762750, 17156148, 18031668
16912439, 17184721, 16494615, 18061914, 17282229, 17545847, 18331850
18202441, 17082359, 18723434, 19554106, 17614134, 13558557, 17341326
14034426, 17891946, 18339044, 17716305, 19458377, 17752995, 16392068
192711443, 17891943, 18092127, 17258090, 17767676, 16668584, 18384391
17614227, 17040764, 16903536, 17381384, 14106803, 15913355, 18973907
18356166, 18673342, 17389192, 14084247, 16194160, 17612828, 17006570
20506715, 17721717, 13853126, 17390431, 18203837, 17570240, 14245531
16043574, 16863422, 17848897, 17877323, 18325460, 19727057, 17468141
17786518, 17912217, 16422541, 19972570, 17267114, 17037130, 18244962
18765602, 18203838, 18155762, 16956380, 16198143, 17246576, 17478145
17394950, 14829250, 18189036, 18641461, 18619917, 17835627, 17027426
16268425, 18247991, 19584068, 14458214, 18436307, 17265217, 17634921
13498382, 16692232, 17786278, 17227277, 16042673, 16314254, 17443671
18000422, 16228604, 16837842, 17571039, 17393683, 16344544, 17787259
18009564, 20074391, 14354737, 15861775, 18135678, 18614015, 16399083
18362222, 18018515, 16472716, 17835048, 17050888, 17936109, 14010183
17325413, 18747196, 17080436, 16613964, 17036973, 17761775, 16579084
16721594, 17082983, 18384537, 18280813, 20296213, 17302277, 16901385
18084625, 15979965, 15990359, 18203835, 17297939, 17811456, 16731148
13829543, 14133975, 17215560, 17694209, 18091059, 17385178, 8322815
17586955, 18441944, 17201159, 16450169, 9756271, 17655634, 19730508
17892268, 18868646, 17648596, 16220077, 16069901, 11733603, 16285691
17587063, 18180390, 16538760, 18193833, 17348614, 17393915, 17957017
17274537, 18096714, 17308789, 17238511, 18436647, 17824637, 14285317
19289642, 14764829, 17622427, 18328509, 16571443, 16943711, 14368995
18306996, 17346671, 14852021, 18996843, 17783588, 16618694, 17853456
```

18674047, 17672719, 18856999, 12364061, 18783224, 17851160, 17546761

Version 11.2.0.4.v3

Version 11.2.0.4.v3 includes the following:

- Oracle PSU 11.2.0.4.4 (19121551)
- Latest DST file (DSTv23 – patch 19396455, released Oct 2014). This patch is incorporated by default in new instances only.

Baseline: Oracle database patch set update 11.2.0.4.4 (patch 19121551, released October 2014)

Bugs fixed:

19396455, 18759211, 17432124, 16799735,
17288409, 17205719, 17811429, 17754782, 17726838, 13364795, 17311728
17284817, 17441661, 13645875, 18199537, 16992075, 16542886, 17446237
14565184, 17071721, 17610798, 17375354, 17449815, 17397545, 19463897
18230522, 17235750, 16360112, 13866822, 17982555, 17478514, 12905058
14338435, 13944971, 16929165, 12747740, 17546973, 14054676, 17088068
18264060, 17343514, 17016369, 17042658, 14602788, 14657740, 17332800
19211724, 13951456, 16315398, 17186905, 18744139, 16850630, 17437634
19049453, 18673304, 17883081, 18641419, 17296856, 18262334, 17006183
18277454, 17232014, 16855292, 10136473, 17705023, 17865671, 18554871
19121551, 17588480, 17551709, 17344412, 17842825, 18681862, 17390160
13955826, 13609098, 18139690, 17501491, 17239687, 17752121, 17299889
17602269, 18673325, 17313525, 17242746, 19544839, 17600719, 18191164
17571306, 19466309, 17951233, 18094246, 17165204, 17011832, 17040527
16785708, 16180763, 17477958, 17174582, 17465741, 18522509, 17323222
19463893, 16875449, 16524926, 17237521, 17596908, 17811438, 17811447
18031668, 16912439, 16494615, 18061914, 17545847, 17082359, 19554106
17614134, 17341326, 17891946, 19458377, 17716305, 17752995, 16392068
19271443, 17767676, 17614227, 17040764, 17381384, 18973907, 18673342
14084247, 17389192, 17006570, 17612828, 17721717, 13853126, 18203837
17390431, 17570240, 14245531, 16043574, 16863422, 19727057, 17468141
17786518, 17037130, 17267114, 18203838, 16198143, 16956380, 17478145
14829250, 17394950, 17027426, 16268425, 18247991, 19584068, 14458214
18436307, 17265217, 13498382, 16692232, 17786278, 17227277, 16042673
16314254, 17443671, 16228604, 16837842, 17393683, 17787259, 18009564
15861775, 16399083, 18018515, 16472716, 17050888, 14010183, 17325413
16613964, 17080436, 17036973, 17761775, 16721594, 18280813, 15979965
18203835, 17297939, 16731148, 17811456, 14133975, 17385178, 17586955
16450169, 17655634, 9756271, 17892268, 17648596, 16220077, 16069901
11733603, 16285691, 17587063, 18180390, 17393915, 18096714, 17238511
17824637, 14285317, 19289642, 14764829, 18328509, 17622427, 16943711
17346671, 18996843, 14852021, 17783588, 16618694, 17672719, 17546761

Version 11.2.0.4.v2 (deprecated)

Version 11.2.0.4.v2 includes the following:

- Oracle PSU 11.2.0.4.3 (18522509)
- User access to DBMS_TRANSACTION package to clean-up failed distributed transactions

- Latest DST file (DSTv22 – patch 18759211, released June 2014). This patch is incorporated by default only in new Oracle DB instances.
- Grants DBMS_REPUTIL to DBA role (upgrade to 11.2.0.4 revokes it from public)
- Privileges granted on DBMS_TRANSACTION, v\$pending_xatrans\$, and v\$xatrans\$
- Resolves a problem with DDL commands when user objects have "SYSTEM" in their names
- Installs schema objects to support XA Transactions, allowing transactions to be managed by an external transaction manager
- Permits truncation of temporary SYS and SYSTEM objects, allowing tools like LogMiner to function correctly

Baseline: Oracle database patch set update 11.2.0.4.3 (patch 18522509, released July 2014)

Bugs fixed:

```
17432124, 18759211, 18522509, 18031668, 17478514,  
17752995, 17288409, 16392068, 17205719, 17811429, 17767676, 17614227  
17040764, 17381384, 17754782, 17726838, 13364795, 17311728, 17389192  
17006570, 17612828, 17284817, 17441661, 13853126, 17721717, 13645875  
18203837, 17390431, 16542886, 16992075, 16043574, 17446237, 16863422  
14565184, 17071721, 17610798, 17468141, 17786518, 17375354, 17397545  
18203838, 16956380, 17478145, 16360112, 17235750, 17394950, 13866822  
17478514, 17027426, 12905058, 14338435, 16268425, 13944971, 18247991  
14458214, 16929165, 17265217, 13498382, 17786278, 17227277, 17546973  
14054676, 17088068, 16314254, 17016369, 14602788, 17443671, 16228604  
16837842, 17332800, 17393683, 13951456, 16315398, 18744139, 17186905  
16850630, 17437634, 19049453, 17883081, 15861775, 17296856, 18277454  
16399083, 16855292, 18018515, 10136473, 16472716, 17050888, 17865671  
17325413, 14010183, 18554871, 17080436, 16613964, 17761775, 16721594  
17588480, 17551709, 17344412, 18681862, 15979965, 13609098, 18139690  
17501491, 17239687, 17752121, 17602269, 18203835, 17297939, 17313525  
16731148, 17811456, 14133975, 17600719, 17385178, 17571306, 16450169  
17655634, 18094246, 17892268, 17165204, 17011832, 17648596, 16785708  
17477958, 16180763, 16220077, 17465741, 17174582, 18522509, 16069901  
16285691, 17323222, 18180390, 17393915, 16875449, 18096714, 17238511
```

Version 11.2.0.4.v1

Version 11.2.0.4.v1 includes the following:

- Oracle PSU 11.2.0.4.1
- [Creating and dropping directories in the main data storage space \(p. 1095\)](#)

Baseline: Oracle database patch set update 11.2.0.4.1 (released January 2014)

Bugs fixed:

```
17432124, 16850630, 17551709, 13944971, 17811447, 13866822, 17811429,  
16069901  
16721594, 17443671, 17478514, 17612828, 17610798, 17239687, 17501491  
17446237, 16450169, 17811438, 17288409, 17811456, 12905058, 17088068  
16285691, 17332800
```

PostgreSQL on Amazon RDS

Amazon RDS supports DB instances running several versions of PostgreSQL. You can create DB instances and DB snapshots, point-in-time restores and backups. DB instances running PostgreSQL support Multi-AZ deployments, read replicas, Provisioned IOPS, and can be created inside a VPC. You can also use Secure Socket Layer (SSL) to connect to a DB instance running PostgreSQL.

Before creating a DB instance, you should complete the steps in the [Setting up for Amazon RDS \(p. 67\)](#) section of this guide.

You can use any standard SQL client application to run commands for the instance from your client computer. Such applications include *pgAdmin*, a popular Open Source administration and development tool for PostgreSQL, or *psql*, a command line utility that is part of a PostgreSQL installation. To deliver a managed service experience, Amazon RDS doesn't provide host access to DB instances, and it restricts access to certain system procedures and tables that require advanced privileges. Amazon RDS supports access to databases on a DB instance using any standard SQL client application. Amazon RDS doesn't allow direct host access to a DB instance by using Telnet or Secure Shell (SSH).

Amazon RDS for PostgreSQL is compliant with many industry standards. For example, you can use Amazon RDS for PostgreSQL databases to build HIPAA-compliant applications and to store healthcare-related information, including protected health information (PHI) under a completed Business Associate Agreement (BAA) with AWS. Amazon RDS for PostgreSQL also meets Federal Risk and Authorization Management Program (FedRAMP) security requirements. Amazon RDS for PostgreSQL has received a FedRAMP Joint Authorization Board (JAB) Provisional Authority to Operate (P-ATO) at the FedRAMP HIGH Baseline within the AWS GovCloud (US) Regions. For more information on supported compliance standards, see [AWS cloud compliance](#).

To import PostgreSQL data into a DB instance, follow the information in the [Importing data into PostgreSQL on Amazon RDS \(p. 1548\)](#) section.

Topics

- [Common management tasks for PostgreSQL on Amazon RDS \(p. 1454\)](#)
- [Working with the database preview environment \(p. 1457\)](#)
- [Limitations for PostgreSQL DB instances \(p. 1460\)](#)
- [Supported PostgreSQL database versions \(p. 1461\)](#)
- [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#)
- [Some supported PostgreSQL features \(p. 1499\)](#)
- [Connecting to a DB instance running the PostgreSQL database engine \(p. 1508\)](#)
- [Security with RDS for PostgreSQL \(p. 1513\)](#)
- [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#)
- [Upgrading a PostgreSQL DB snapshot engine version \(p. 1542\)](#)
- [Working with PostgreSQL read replicas in Amazon RDS \(p. 1544\)](#)
- [Importing data into PostgreSQL on Amazon RDS \(p. 1548\)](#)
- [Exporting data from an RDS for PostgreSQL DB instance to Amazon S3 \(p. 1568\)](#)
- [Common DBA tasks for PostgreSQL \(p. 1578\)](#)

Common management tasks for PostgreSQL on Amazon RDS

The following are the common management tasks you perform with an Amazon RDS for PostgreSQL DB instance, with links to relevant documentation for each task.

Task area	Relevant documentation
Setting up Amazon RDS for first-time use There are prerequisites you must complete before you create your DB instance. For example, DB instances are created by default with a firewall that prevents access to it. You therefore must create a security group with the correct IP addresses and network configuration to access the DB instance.	Setting up for Amazon RDS (p. 67)
Understanding Amazon RDS DB instances If you are creating a DB instance for production purposes, you should understand how instance classes, storage types, and Provisioned IOPS work in Amazon RDS.	DB instance classes (p. 7) Amazon RDS storage types (p. 40) Provisioned IOPS SSD storage (p. 42)
Finding supported PostgreSQL versions Amazon RDS supports several versions of PostgreSQL.	Supported PostgreSQL database versions (p. 1461)
Setting up high availability and failover support A production DB instance should use Multi-AZ deployments. Multi-AZ deployments provide increased availability, data durability, and fault tolerance for DB instances.	High availability (Multi-AZ) for Amazon RDS (p. 53)
Understanding the Amazon Virtual Private Cloud (VPC) network If your AWS account has a default VPC, then your DB instance is automatically created inside the default VPC. In some cases, your account might not have a default VPC, and you might want the DB instance in a VPC. In these cases, create the VPC and subnet groups before you create the DB instance.	Determining whether you are using the EC2-VPC or EC2-Classic platform (p. 1718) Working with a DB instance in a VPC (p. 1727)
Importing data into Amazon RDS PostgreSQL You can use several different tools to import data into your PostgreSQL DB instance on Amazon RDS.	Importing data into PostgreSQL on Amazon RDS (p. 1548)
Setting up read-only read replicas (primary and standbys) PostgreSQL on Amazon RDS supports read replicas in both the same AWS Region and in a different AWS Region from the primary instance.	Working with read replicas (p. 278) Working with PostgreSQL read replicas in Amazon RDS (p. 1544) Creating a read replica in a different AWS Region (p. 290)

Task area	Relevant documentation
Understanding security groups <p>By default, DB instances are created with a firewall that prevents access to them. You therefore must create a security group with the correct IP addresses and network configuration to access the DB instance.</p> <p>In general, if your DB instance is on the EC2-Classic platform, you need to create a DB security group. If your DB instance is on the EC2-VPC platform, you need to create a VPC security group.</p>	Determining whether you are using the EC2-VPC or EC2-Classic platform (p. 1718) Controlling access with security groups (p. 1699)
Setting up parameter groups and features <p>If your DB instance is going to require specific database parameters, you should create a parameter group before you create the DB instance.</p>	Working with DB parameter groups (p. 228)
Performing common DBA tasks for PostgreSQL <p>Some of the more common tasks for PostgreSQL DBAs include:</p> <ul style="list-style-type: none"> • Creating roles (p. 1578) • Managing PostgreSQL database access (p. 1579) • Working with PostgreSQL parameters (p. 1579) • Working with PostgreSQL autovacuum on Amazon RDS (p. 1593) • Audit logging for a PostgreSQL DB instance (p. 1588) • Working with the PostGIS extension (p. 1602) • Using pgBadger for log analysis with PostgreSQL (p. 1590) • Using a custom DNS server for outbound network access (p. 1605) 	Common DBA tasks for PostgreSQL (p. 1578)
Connecting to your PostgreSQL DB instance <p>After creating a security group and associating it to a DB instance, you can connect to the DB instance using any standard SQL client application such as pgadmin III.</p>	Connecting to a DB instance running the PostgreSQL database engine (p. 1508) Using SSL with a PostgreSQL DB instance (p. 1513)
Backing up and restoring your DB instance <p>You can configure your DB instance to take automated backups, or take manual snapshots, and then restore instances from the backups or snapshots.</p>	Backing up and restoring an Amazon RDS DB instance (p. 327)
Monitoring the activity and performance of your DB instance <p>You can monitor a PostgreSQL DB instance by using CloudWatch Amazon RDS metrics, events, and enhanced monitoring.</p>	Viewing DB instance metrics (p. 548) Viewing Amazon RDS events (p. 503)

Task area	Relevant documentation
Upgrading the PostgreSQL database version You can do both major and minor version upgrades for your PostgreSQL DB instance.	Upgrading the PostgreSQL DB engine for Amazon RDS (p. 1533) Choosing a major version upgrade for PostgreSQL (p. 1534)
Working with log files You can access the log files for your PostgreSQL DB instance.	PostgreSQL database log files (p. 534)
Understanding the best practices for PostgreSQL DB instances Find some of the best practices for working with PostgreSQL on Amazon RDS.	Best practices for working with PostgreSQL (p. 137)

Working with the database preview environment

When you create a DB instance in Amazon RDS, you know that the PostgreSQL version it's based on has been tested and is fully supported by Amazon. The PostgreSQL community releases new versions and new extensions continuously. You can try out new PostgreSQL versions and extensions before they are fully supported. To do that, you can create a new DB instance in the Database Preview Environment.

DB instances in the Database Preview Environment are similar to DB instances in a production environment. However, keep in mind several important factors:

- All DB instances are deleted 60 days after you create them, along with any backups and snapshots.
- You can only create a DB instance in a virtual private cloud (VPC) based on the Amazon VPC service.
- You can only create M6g, M5, T3, R6g, and R5 instance types. For more information about RDS instance classes, see [DB instance classes \(p. 7\)](#).
- You can only use General Purpose SSD and Provisioned IOPS SSD storage.
- You can't get help from AWS Support with DB instances. You can post your questions in the [RDS database preview environment forum](#).
- You can't copy a snapshot of a DB instance to a production environment.
- You can use both single-AZ and multi-AZ deployments.
- You can use standard PostgreSQL dump and load functions to export databases from or import databases to the Database Preview Environment.

Topics

- [Features not supported in the preview environment \(p. 1457\)](#)
- [PostgreSQL extensions supported in the preview environment \(p. 1457\)](#)
- [Creating a new DB instance in the preview environment \(p. 1459\)](#)

Features not supported in the preview environment

The following features are not available in the preview environment:

- Cross-region snapshot copy
- Cross-region read replicas
- Extensions not in the following table of supported extensions

PostgreSQL extensions supported in the preview environment

The PostgreSQL extensions supported in the Database Preview Environment are listed in the following table.

Extension	Version
amcheck	1.2
aws_commons	1.0
aws_s3	1.0

Extension	Version
bloom	1.0
btree_gin	1.3
btree_gist	1.5
citext	1.6
cube	1.4
dblink	1.2
dict_int	1.0
dict_xsyn	1.0
earthdistance	1.1
fuzzystrmatch	1.1
hstore	1.7
hstore_plper	1.0
intagg	1.1
intarray	1.3
ip4r	2.4
isn	1.2
jsonb_plperl	1.0
ltree	1.2
pageinspect	1.8
pg_buffercache	1.3
pg_freespacemap	1.2
pg_prewarm	1.2
pg_similarity	1.0
pg_stat_statements	1.8
pg_transport	1.0
pg_trgm	1.5
pg_visibility	1.2
pgcrypto	1.3
pgrouting	3.0.0
pgrowlocks	1.2
pgstattuple	1.5

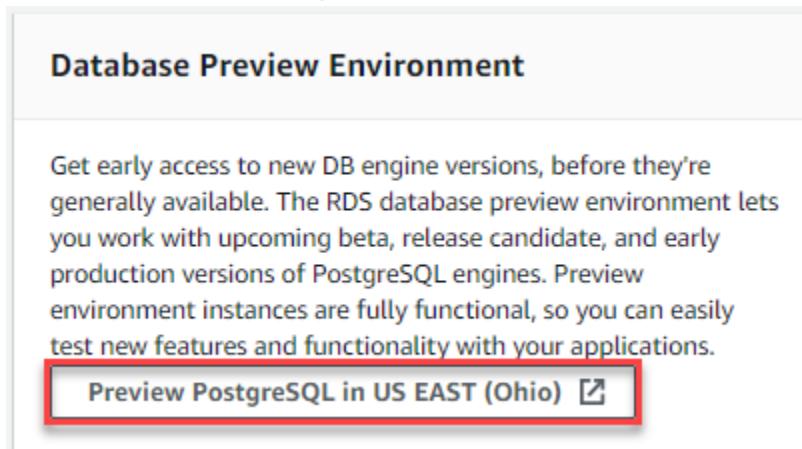
Extension	Version
pgtap	1.1.0
plperl	1.0
plpgsql	1.0
plprofiler	4.1
pltcl	1.0
postgres_fdw	1.0
prefix	1.2.0
sslinfo	1.2
tablefunc	1.0
test_parser	1.0
tsm_system_rows	1.0
tsm_system_time	1.0
unaccent	1.1
uuid-ossp	1.1

Creating a new DB instance in the preview environment

Use the following procedure to create a DB instance in the preview environment.

To create a DB instance in the preview environment

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose **Dashboard** from the navigation pane.
3. Choose **Switch to database preview environment**.



You also can navigate directly to the [Database preview environment](#).

Note

If you want to create an instance in the Database Preview Environment with the API or CLI, the endpoint is `rds-preview.us-east-2.amazonaws.com`.

4. Continue with the procedure as described in [Console \(p. 141\)](#).

Limitations for PostgreSQL DB instances

The following is a list of limitations for PostgreSQL on Amazon RDS:

- You can have up to 40 PostgreSQL DB instances.
- For storage limits, see [Amazon RDS DB instance storage \(p. 40\)](#).
- Amazon RDS reserves up to 3 connections for system maintenance. If you specify a value for the user connections parameter, you need to add 3 to the number of connections that you expect to use.

Supported PostgreSQL database versions

Amazon RDS supports DB instances running several editions of PostgreSQL. You can specify any currently supported PostgreSQL version when creating a new DB instance. You can specify the major version (such as PostgreSQL 10), and any supported minor version for the specified major version. If no version is specified, Amazon RDS defaults to a supported version, typically the most recent version. If a major version is specified but a minor version is not, Amazon RDS defaults to a recent release of the major version you have specified.

To see a list of supported versions, as well as defaults for newly created DB instances, use the `describe-db-engine-versions` AWS CLI command. For example, to display the default PostgreSQL engine version, use the following command:

```
aws rds describe-db-engine-versions --default-only --engine postgres
```

Topics

- [PostgreSQL 13 versions \(p. 1461\)](#)
- [PostgreSQL 12 versions \(p. 1462\)](#)
- [PostgreSQL 11 versions \(p. 1463\)](#)
- [PostgreSQL 10 versions \(p. 1466\)](#)
- [PostgreSQL 9.6 versions \(p. 1471\)](#)
- [PostgreSQL 9.5 versions \(p. 1476\)](#)

PostgreSQL 13 versions

Minor versions

- [PostgreSQL version 13.2 on Amazon RDS \(p. 1461\)](#)
- [PostgreSQL version 13.1 on Amazon RDS \(p. 1461\)](#)

PostgreSQL version 13.2 on Amazon RDS

PostgreSQL version 13.2 is now available on Amazon RDS. PostgreSQL contains several improvements that were announced in [PostgreSQL 13.2](#).

This version also added the following new extensions:

1. The `aws_lambda` extension version 1.0. For more information, see [Invoking an AWS Lambda function from an RDS for PostgreSQL DB instance \(p. 1618\)](#).
2. The `pg_bigm` extension version 1.2.

For information on all extensions, see [PostgreSQL version 13 extensions supported on Amazon RDS \(p. 1483\)](#).

PostgreSQL version 13.1 on Amazon RDS

PostgreSQL version 13.1 is now available on Amazon RDS. PostgreSQL contains several improvements that were announced in [PostgreSQL 13.0](#) and [PostgreSQL 13.1](#).

This version added the `bool_plperl` extension version 1.0.

For information on all extensions, see [PostgreSQL version 13 extensions supported on Amazon RDS \(p. 1483\)](#).

PostgreSQL 12 versions

Minor versions

- [PostgreSQL version 12.6 on Amazon RDS \(p. 1462\)](#)
- [PostgreSQL version 12.5 on Amazon RDS \(p. 1462\)](#)
- [PostgreSQL version 12.4 on Amazon RDS \(p. 1462\)](#)
- [PostgreSQL version 12.3 on Amazon RDS \(p. 1463\)](#)
- [PostgreSQL version 12.2 on Amazon RDS \(p. 1463\)](#)

PostgreSQL version 12.6 on Amazon RDS

PostgreSQL version 12.6 is now available on Amazon RDS. PostgreSQL version 12.6 contains several improvements that were announced for PostgreSQL release [12.6](#).

This version also includes the following changes:

1. The `aws_lambda` extension version 1.0 is added. For more information, see [Invoking an AWS Lambda function from an RDS for PostgreSQL DB instance \(p. 1618\)](#).
2. The `pg_bigm` extension version 1.2 is added.
3. The [PostGIS \(p. 1604\)](#) extension is updated to version 3.0.2.

For information on all extensions, see [PostgreSQL version 12 extensions supported on Amazon RDS \(p. 1486\)](#).

PostgreSQL version 12.5 on Amazon RDS

PostgreSQL version 12.5 is now available on Amazon RDS. PostgreSQL version 12.5 contains several improvements that were announced for PostgreSQL release [12.5](#).

This version also includes the following changes:

1. Added the `pg_partman` extension version 4.4.0. For more information, see [Managing PostgreSQL partitions with the pg_partman extension \(p. 1614\)](#).
2. Added the `pg_cron` extension version 1.3.0. For more information, see [Scheduling maintenance with the PostgreSQL pg_cron extension \(p. 1607\)](#).

For information on all extensions, see [PostgreSQL version 12 extensions supported on Amazon RDS \(p. 1486\)](#).

PostgreSQL version 12.4 on Amazon RDS

PostgreSQL version 12.4 is now available on Amazon RDS. PostgreSQL version 12.4 contains several improvements that were announced for PostgreSQL release [12.4](#).

This version also includes the following changes:

1. Added the `pg_proctab` extension version 0.0.9
2. Added the `rdkit` extension version 3.8

3. Upgraded the `aws_s3` extension to version 1.1.
4. Upgraded the `pglogical` extension to version 2.3.2
5. Upgraded the `wal2json` extension to version 2.3

For information on all extensions, see [PostgreSQL version 12 extensions supported on Amazon RDS \(p. 1486\)](#).

PostgreSQL version 12.3 on Amazon RDS

PostgreSQL version 12.3 is now available on Amazon RDS. PostgreSQL version 12.3 contains several improvements that were announced for PostgreSQL release [12.3](#).

This version also includes the following changes:

1. Upgraded the `pg_hint_plan` extension to version 1.3.5.
2. Upgraded the `pglogical` extension to version 2.3.1.

For information on all extensions, see [PostgreSQL version 12 extensions supported on Amazon RDS \(p. 1486\)](#).

PostgreSQL version 12.2 on Amazon RDS

PostgreSQL version 12.2 is now available on Amazon RDS. PostgreSQL version 12.2 contains several improvements that were announced for PostgreSQL releases [12.0](#), [12.1](#), and [12.2](#).

For information on all extensions, see [PostgreSQL version 12 extensions supported on Amazon RDS \(p. 1486\)](#).

PostgreSQL 11 versions

Minor versions

- [PostgreSQL version 11.11 on Amazon RDS \(p. 1463\)](#)
- [PostgreSQL version 11.10 on Amazon RDS \(p. 1464\)](#)
- [PostgreSQL version 11.9 on Amazon RDS \(p. 1464\)](#)
- [PostgreSQL version 11.8 on Amazon RDS \(p. 1464\)](#)
- [PostgreSQL version 11.7 on Amazon RDS \(p. 1464\)](#)
- [PostgreSQL version 11.6 on Amazon RDS \(p. 1464\)](#)
- [PostgreSQL version 11.5 on Amazon RDS \(p. 1465\)](#)
- [PostgreSQL version 11.4 on Amazon RDS \(p. 1465\)](#)
- [PostgreSQL version 11.2 on Amazon RDS \(p. 1465\)](#)
- [PostgreSQL version 11.1 on Amazon RDS \(p. 1465\)](#)

PostgreSQL version 11.11 on Amazon RDS

PostgreSQL version 11.11 is now available on Amazon RDS. PostgreSQL version 11.11 contains several improvements that were announced for PostgreSQL release [11.11](#).

This version also added the following new extension:

1. The `pg_bigm` extension version 1.2.

For information on all extensions, see [PostgreSQL version 11.x extensions supported on Amazon RDS \(p. 1489\)](#).

PostgreSQL version 11.10 on Amazon RDS

PostgreSQL version 11.10 is now available on Amazon RDS. PostgreSQL version 11.10 contains several improvements that were announced for PostgreSQL release [11.10](#).

For information on all extensions, see [PostgreSQL version 11.x extensions supported on Amazon RDS \(p. 1489\)](#).

PostgreSQL version 11.9 on Amazon RDS

PostgreSQL version 11.9 is now available on Amazon RDS. PostgreSQL version 11.9 contains several improvements that were announced for PostgreSQL release [11.9](#).

This version also includes the following changes:

1. Added the `aws_s3` extension version 1.1
2. Added the `pg_proctab` extension version 0.0.9
3. Upgraded the `pgaudit` extension to version 1.3.1.
4. Upgraded the `pglogical` extension to version 2.2.2
5. Added the `rdkit` extension version 3.8

For information on all extensions, see [PostgreSQL version 11.x extensions supported on Amazon RDS \(p. 1489\)](#).

PostgreSQL version 11.8 on Amazon RDS

PostgreSQL version 11.8 contains several bug fixes for issues in release 11.7. For more information on the fixes in PostgreSQL 11.8, see the [PostgreSQL 11.8 documentation](#).

This version also includes the following change:

1. Upgraded the `pg_hint_plan` extension to version 1.3.5.

For information on all extensions, see [PostgreSQL version 11.x extensions supported on Amazon RDS \(p. 1489\)](#).

PostgreSQL version 11.7 on Amazon RDS

PostgreSQL version 11.7 contains several bug fixes for issues in release 11.6. For more information on the fixes in PostgreSQL 11.7, see the [PostgreSQL 11.7 documentation](#).

PostgreSQL version 11.6 on Amazon RDS

PostgreSQL version 11.6 contains several bug fixes for issues in release 11.5. For more information on the fixes in PostgreSQL 11.6, see the [PostgreSQL documentation](#).

This version also includes the following changes:

1. Upgraded the `pgTAP` extension to version 1.1.0.
2. Added the `plprofiler` extension.

3. Added to `shared_preload_libraries` support for `pg_prewarm` to start automatically.

PostgreSQL version 11.5 on Amazon RDS

PostgreSQL version 11.5 contains several bug fixes for issues in release 11.4. For more information on the fixes in PostgreSQL 11.5, see the [PostgreSQL documentation](#).

This version also includes the following changes:

- A new extension `pg_transport` is added.
- The extension `aws_s3` has been updated to support virtual-hosted style requests. For more information, see [Amazon S3 path deprecation plan – The rest of the story](#).
- The `PostGIS` extension is updated to version 2.5.2.

PostgreSQL version 11.4 on Amazon RDS

This release contains an important security fix and also bug fixes and improvements done by the PostgreSQL community. For more information on the security fix, see the [PostgreSQL community announcement](#) and the security fix CVE-2019-10164.

With this release, the `pg_hint_plan` extension has been updated to version 1.3.4.

For more information on the fixes in PostgreSQL 11.4, see the [PostgreSQL documentation](#).

PostgreSQL version 11.2 on Amazon RDS

PostgreSQL version 11.2 contains several bug fixes for issues in release 11.1. For more information on the fixes in PostgreSQL 11.2, see the [PostgreSQL documentation](#).

This version also includes the following changes:

- A new `pgTAP` extension version 1.0.
- Support for Amazon S3 import. For more information, see [Importing Amazon S3 data into an RDS for PostgreSQL DB instance \(p. 1552\)](#).
- Multiple major version upgrade is available to PostgreSQL 11.2 from certain previous PostgreSQL versions. For more information, see [Choosing a major version upgrade for PostgreSQL \(p. 1534\)](#).

For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 11.1 on Amazon RDS

PostgreSQL version 11.1 contains several improvements that were announced in [PostgreSQL 11.1 released!](#) This version includes SQL stored procedures that enable embedded transactions within a procedure. This version also includes major improvements to partitioning and parallelism and many useful performance improvements. For example, by using a non-null constant for a column default, you can now use an `ALTER TABLE` command to add a column without causing a table rewrite.

PostgreSQL version 11.1 contains several bug fixes for issues in release 11. For complete details, see the [PostgreSQL release 11.1 documentation](#). Some changes in this version include the following:

- Partitioning – Partitioning improvements include support for hash partitioning, enabling creation of a default partition, and dynamic row movement to another partition based on the key column update.
- Performance – Performance improvements include parallelism while creating indexes, materialized views, hash joins, and sequential scans to make the operations perform better.
- Stored procedures – SQL stored procedures now added support embedded transactions.
- Support for Just-In-Time (JIT) capability – RDS for PostgreSQL 11 instances are created with JIT capability, speeding evaluation of expressions. To enable JIT capability, set the `jit` parameter to 1 in the PostgreSQL parameter group for the database.
- Segment size – The write-ahead logging (WAL) segment size has been changed from 16 MB to 64 MB.
- Autovacuum improvements – To provide valuable logging, the parameter `rds.force_autovacuum_logging` is ON by default in conjunction with the `log_autovacuum_min_duration` parameter set to 10 seconds. To increase autovacuum effectiveness, the values for the `autovacuum_max_workers` and `autovacuum_vacuum_cost_limit` parameters are computed based on host memory capacity to provide larger default values.
- Improved transaction timeout – The parameter `idle_in_transaction_session_timeout` is set to 12 hours. Any session that has been idle more than 12 hours is terminated.
- Performance metrics – The `pg_stat_statements` extension is included in `shared_preload_libraries` by default. This avoids having to reboot the instance immediately after creation. However, this functionality still requires you to run the statement `CREATE EXTENSION pg_stat_statements;`. Also, `track_io_timing` is enabled by default to add more granular data to `pg_stat_statements`.
- The `tsearch2` extension is no longer supported – If your application uses `tsearch2` functions, update it to use the equivalent functions provided by the core PostgreSQL engine. For more information about the `tsearch2` extension, see [PostgreSQL tsearch2](#).
- The `chkpass` extension is no longer supported – For more information about the `chkpass` extension, see [PostgreSQL chkpass](#).
- Extension updates for RDS for PostgreSQL 11.1 include the following:
 - `pgaudit` is updated to 1.3.0.
 - `pg_hint_plan` is updated to 1.3.2.
 - `pglogical` is updated to 2.2.1.
 - `plcoffee` is updated to 2.3.8.
 - `plv8` is updated to 2.3.8.
 - `PostGIS` is updated to 2.5.1.
 - `prefix` is updated to 1.2.8.
 - `wal2json` is updated to hash 9e962bad.

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL 10 versions

Minor versions

- [PostgreSQL version 10.16 on Amazon RDS \(p. 1467\)](#)
- [PostgreSQL version 10.15 on Amazon RDS \(p. 1467\)](#)
- [PostgreSQL version 10.14 on Amazon RDS \(p. 1467\)](#)
- [PostgreSQL version 10.13 on Amazon RDS \(p. 1467\)](#)
- [PostgreSQL version 10.12 on Amazon RDS \(p. 1468\)](#)
- [PostgreSQL version 10.11 on Amazon RDS \(p. 1468\)](#)

- [PostgreSQL version 10.10 on Amazon RDS \(p. 1468\)](#)
- [PostgreSQL version 10.9 on Amazon RDS \(p. 1468\)](#)
- [PostgreSQL version 10.7 on Amazon RDS \(p. 1468\)](#)
- [PostgreSQL version 10.6 on Amazon RDS \(p. 1468\)](#)
- [PostgreSQL version 10.5 on Amazon RDS \(p. 1469\)](#)
- [PostgreSQL version 10.4 on Amazon RDS \(p. 1469\)](#)
- [PostgreSQL version 10.3 on Amazon RDS \(p. 1470\)](#)
- [PostgreSQL version 10.1 on Amazon RDS \(p. 1470\)](#)

PostgreSQL version 10.16 on Amazon RDS

PostgreSQL version 10.16 is now available on Amazon RDS. PostgreSQL version 10.16 contains several improvements that were announced for PostgreSQL release [10.16](#).

For information on all extensions, see [PostgreSQL version 10.x extensions supported on Amazon RDS \(p. 1491\)](#).

PostgreSQL version 10.15 on Amazon RDS

PostgreSQL version 10.15 is now available on Amazon RDS. PostgreSQL version 10.15 contains several improvements that were announced for PostgreSQL release [10.15](#).

For information on all extensions, see [PostgreSQL version 10.x extensions supported on Amazon RDS \(p. 1491\)](#).

PostgreSQL version 10.14 on Amazon RDS

PostgreSQL version 10.14 is now available on Amazon RDS. PostgreSQL version 10.14 contains several improvements that were announced for PostgreSQL release [10.14](#).

This version also includes the following changes:

1. Added the `aws_s3` extension version 1.1. For more information, see [Exporting data from an RDS for PostgreSQL DB instance to Amazon S3 \(p. 1568\)](#).
2. Upgraded the `pgaudit` extension to version 1.2.1
3. Upgraded the `pglogical` extension to version 2.2.2
4. Upgraded the `wal2json` extension to version 2.3

For information on all extensions, see [PostgreSQL version 10.x extensions supported on Amazon RDS \(p. 1491\)](#).

PostgreSQL version 10.13 on Amazon RDS

PostgreSQL version 10.13 contains several bug fixes for issues in release 10.12. For more information on the fixes in PostgreSQL 10.13, see the [PostgreSQL 10.13 documentation](#).

This version also includes the following change:

1. Upgraded the `pg_hint_plan` extension to version 1.3.5.

For information on all extensions, see [PostgreSQL version 10.x extensions supported on Amazon RDS \(p. 1491\)](#).

PostgreSQL version 10.12 on Amazon RDS

PostgreSQL version 10.12 contains several bug fixes for issues in release 10.11. For more information on the fixes in PostgreSQL 10.12, see the [PostgreSQL 10.12 documentation](#).

PostgreSQL version 10.11 on Amazon RDS

PostgreSQL version 10.11 contains several bug fixes for issues in release 10.10. For more information on the fixes in PostgreSQL 10.11, see the [PostgreSQL documentation](#). Changes in this version include the following:

1. Added the `plprofiler` extension.

PostgreSQL version 10.10 on Amazon RDS

PostgreSQL version 10.10 contains several bug fixes for issues in release 10.9. For more information on the fixes in PostgreSQL 10.10, see the [PostgreSQL documentation](#). Changes in this version include the following:

1. The `aws_s3` extension is updated to support virtual-hosted style requests. For more information, see [Amazon S3 path deprecation plan – The rest of the story](#).
2. The `PostGIS` extension is updated to version 2.5.2.

PostgreSQL version 10.9 on Amazon RDS

This release contains an important security fix and also bug fixes and improvements done by the PostgreSQL community. For more information on the security fix, see the [PostgreSQL community announcement and security fix CVE-2019-10164](#).

With this release, the `pg_hint_plan` extension has been updated to version 1.3.3.

For more information on the fixes in PostgreSQL 10.9, see the [PostgreSQL documentation](#).

PostgreSQL version 10.7 on Amazon RDS

PostgreSQL version 10.7 contains several bug fixes for issues in release 10.6. For more information on the fixes in 10.7, see the [PostgreSQL documentation](#).

This version also includes the following changes:

- Support for Amazon S3 import. For more information, see [Importing Amazon S3 data into an RDS for PostgreSQL DB instance \(p. 1552\)](#).
- Multiple major version upgrade is available to PostgreSQL 10.7 from certain previous PostgreSQL versions. For more information, see [Choosing a major version upgrade for PostgreSQL \(p. 1534\)](#).

For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

PostgreSQL version 10.6 on Amazon RDS

PostgreSQL version 10.6 contains several bug fixes for issues in release 10.5. For more information on the fixes in PostgreSQL 10.6, see the [PostgreSQL documentation](#).

This version also includes the following changes:

- A new `rds.restrict_password_commands` parameter and a new `rds_password` role have been introduced. When the `rds.restrict_password_commands` parameter is enabled, only users who have the `rds_password` role can make user password and password expiration changes. By restricting password-related operations to a limited set of roles, you can implement policies such as password complexity requirements from the client side. The `rds.restrict_password_commands` parameter is static, so it requires a database restart to change it. For more information, see [Restricting password management \(p. 1593\)](#).
- The logical decoding plugin `wal2json` has been updated to commit `9e962ba`.

For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

Note

Amazon RDS for PostgreSQL has announced the removal of the `tsearch2` extension in the next major release. We encourage customers still using pre-8.3 text search to migrate to the equivalent built-in features. For more information about migrating, see the [PostgreSQL documentation](#).

PostgreSQL version 10.5 on Amazon RDS

PostgreSQL version 10.5 contains several bug fixes for issues in release 10.4. For more information on the fixes in 10.5, see the [PostgreSQL documentation](#).

This version also includes the following changes:

- Support for the `pglogical` extension version 2.2.0. Prerequisites for using this extension are the same as the prerequisites for using logical replication for PostgreSQL as described in [Logical replication for PostgreSQL on Amazon RDS \(p. 1502\)](#).
- Support for the `pg_similarity` extension version 1.0.
- Support for the `pageinspect` extension version 1.6.
- Support for the `libprotobuf` extension version 1.3.0 for the PostGIS component.
- An update for the `pg_hint_plan` extension to version 1.3.1.
- An update for the `wal2json` extension to version `01c5c1e`.

For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 10.4 on Amazon RDS

PostgreSQL version 10.4 contains several bug fixes for issues in release 10.3. For more information on the fixes in 10.4, see the [PostgreSQL documentation](#).

This version also includes the following changes:

- Support for PostgreSQL 10 Logical Replication using the native publication and subscription framework. RDS for PostgreSQL databases can function as both publishers and subscribers. You can specify replication to other PostgreSQL databases at the database-level or at the table-level. With

logical replication, the publisher and subscriber databases need not be physically identical (block-to-block) to each other. This allows for use cases such as data consolidation, data distribution, and data replication across different database versions for 10.4 and above. For more details, refer to [Logical replication for PostgreSQL on Amazon RDS \(p. 1502\)](#).

- The temporary file size limitation is user-configurable. You require the `rds_superuser` role to modify the `temp_file_limit` parameter.
- Update of the GDAL library, which is used by the PostGIS extension. See [Working with the PostGIS extension \(p. 1602\)](#).
- Update of the `ip4r` extension to version 2.1.1.
- Update of the `pg_repack` extension to version 1.4.3. See [Working with the pg_repack extension \(p. 1590\)](#).
- Update of the `plv8` extension to version 2.1.2.

For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

Note

The `tsearch2` extension is to be removed in the next major release. We encourage customers still using pre-8.3 text search to migrate to the equivalent built-in features. For more information about migrating, see the [PostgreSQL documentation](#).

PostgreSQL version 10.3 on Amazon RDS

PostgreSQL version 10.3 contains several bug fixes for issues in release 10. For more information on the fixes in 10.3, see the [PostgreSQL documentation](#).

Version 2.1.0 of `plv8` is now available. If you use `plv8` and upgrade PostgreSQL to a new `plv8` version, you immediately take advantage of the new extension but the catalog metadata doesn't reflect this fact. For the steps to synchronize your catalog metadata with the new version of `plv8`, see [Upgrading plv8 \(p. 1500\)](#).

For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 10.1 on Amazon RDS

PostgreSQL version 10.1 contains several bug fixes for issues in release 10. For more information on the fixes in 10.1, see the [PostgreSQL documentation](#) and the [PostgreSQL 10 community announcement](#).

For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

PostgreSQL version 10.1 includes the following changes:

- **Declarative table partitioning** – PostgreSQL 10 adds table partitioning to SQL syntax and native tuple routing.
- **Parallel queries** – When you create a new PostgreSQL 10.1 instance, parallel queries are enabled for the `default.postgres10` parameter group. The parameter `max_parallel_workers_per_gather` is set to 2 by default, but you can modify it to support your specific workload requirements.

- **Support for the international components for unicode (ICU)** – You can use the ICU library to provide explicitly versioned collations. Amazon RDS for PostgreSQL 10.1 is compiled with ICU version 60.2. For more information about ICU implementation in PostgreSQL, see [Collation support](#).
- **Huge pages** – Huge pages is a feature of the Linux kernel that uses multiple page size capabilities of modern hardware architectures. Amazon RDS for PostgreSQL supports huge pages with a global configuration parameter. When you create a new PostgreSQL 10.1 instance with RDS, the `huge_pages` parameter is set to "on" for the `default.postgres10` parameter group. You can modify this setting to support your specific workload requirements.
- Extension **plv8 update** – plv8 is a procedural language that you can use to write functions in JavaScript that you can then call from SQL. This release of PostgreSQL supports version 2.1.0 of plv8.
- **Renaming of xlog and location** – In PostgreSQL version 10 the abbreviation "xlog" has changed to "wal", and the term "location" has changed to "lsn". For more information, see <https://www.postgresql.org/docs/10/static/release-10.html#id-1.11.6.8.4>.
- **tsearch2 extension** – Amazon RDS continues to provide the `tsearch2` extension in PostgreSQL version 10, but is to remove it in the next major version release. If your application uses `tsearch2` functions update it to use the equivalent functions the core engine provides. For more information see [tsearch2](#) in the PostgreSQL documentation.

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL 9.6 versions

Minor versions

- [PostgreSQL version 9.6.21 on Amazon RDS \(p. 1471\)](#)
- [PostgreSQL version 9.6.20 on Amazon RDS \(p. 1472\)](#)
- [PostgreSQL version 9.6.19 on Amazon RDS \(p. 1472\)](#)
- [PostgreSQL version 9.6.18 on Amazon RDS \(p. 1472\)](#)
- [PostgreSQL version 9.6.17 on Amazon RDS \(p. 1472\)](#)
- [PostgreSQL version 9.6.16 on Amazon RDS \(p. 1472\)](#)
- [PostgreSQL version 9.6.15 on Amazon RDS \(p. 1472\)](#)
- [PostgreSQL version 9.6.14 on Amazon RDS \(p. 1473\)](#)
- [PostgreSQL version 9.6.12 on Amazon RDS \(p. 1473\)](#)
- [PostgreSQL version 9.6.11 on Amazon RDS \(p. 1473\)](#)
- [PostgreSQL version 9.6.10 on Amazon RDS \(p. 1473\)](#)
- [PostgreSQL version 9.6.9 on Amazon RDS \(p. 1473\)](#)
- [PostgreSQL version 9.6.8 on Amazon RDS \(p. 1474\)](#)
- [PostgreSQL version 9.6.6 on Amazon RDS \(p. 1474\)](#)
- [PostgreSQL version 9.6.5 on Amazon RDS \(p. 1474\)](#)
- [PostgreSQL version 9.6.3 on Amazon RDS \(p. 1475\)](#)
- [PostgreSQL version 9.6.2 on Amazon RDS \(p. 1475\)](#)
- [PostgreSQL version 9.6.1 on Amazon RDS \(p. 1475\)](#)

PostgreSQL version 9.6.21 on Amazon RDS

PostgreSQL version 9.6.21 is now available on Amazon RDS. PostgreSQL version 9.6.21 contains several improvements that were announced for PostgreSQL release [9.6.21](#).

For information on all extensions, see [PostgreSQL version 9.6.x extensions supported on Amazon RDS \(p. 1494\)](#).

PostgreSQL version 9.6.20 on Amazon RDS

PostgreSQL version 9.6.20 is now available on Amazon RDS. PostgreSQL version 9.6.20 contains several improvements that were announced for PostgreSQL release [9.6.20](#).

For information on all extensions, see [PostgreSQL version 9.6.x extensions supported on Amazon RDS \(p. 1494\)](#).

PostgreSQL version 9.6.19 on Amazon RDS

PostgreSQL version 9.6.19 is now available on Amazon RDS. PostgreSQL version 9.6.19 contains several improvements that were announced for PostgreSQL release [9.6.19](#).

This version also includes the following changes:

1. Upgraded the `pgaudit` extension to version 1.1.2
2. Upgraded the `pglogical` extension to version 2.2.2
3. Upgraded the `wal2json` extension to version 2.3

For information on all extensions, see [PostgreSQL version 9.6.x extensions supported on Amazon RDS \(p. 1494\)](#).

PostgreSQL version 9.6.18 on Amazon RDS

PostgreSQL version 9.6.18 contains several bug fixes for issues in release 9.6.17. For more information on the fixes in PostgreSQL 9.6.18, see the [PostgreSQL 9.6.18 documentation](#).

This version also includes the following change:

1. Upgraded the `pg_hint_plan` extension to version 1.2.6.

For information on all extensions, see [PostgreSQL version 9.6.x extensions supported on Amazon RDS \(p. 1494\)](#).

PostgreSQL version 9.6.17 on Amazon RDS

PostgreSQL version 9.6.17 contains several bug fixes for issues in release 9.6.16. For more information on the fixes in PostgreSQL 9.6.17, see the [PostgreSQL 9.6.17 documentation](#).

PostgreSQL version 9.6.16 on Amazon RDS

PostgreSQL version 9.6.16 contains several bug fixes for issues in release 9.6.15. For more information on the fixes in PostgreSQL 9.6.16, see the [PostgreSQL documentation](#).

PostgreSQL version 9.6.15 on Amazon RDS

PostgreSQL version 9.6.15 contains several bug fixes for issues in release 9.6.14. For more information on the fixes in PostgreSQL 9.6.15, see the [PostgreSQL documentation](#).

The PostGIS extension is updated to version 2.5.2.

PostgreSQL version 9.6.14 on Amazon RDS

This release contains bug fixes and improvements done by the PostgreSQL community.

With this release, the `pg_hint_plan` extension has been updated to version 1.2.5.

For more information on the fixes in PostgreSQL 9.6.14, see the [PostgreSQL documentation](#).

PostgreSQL version 9.6.12 on Amazon RDS

PostgreSQL version 9.6.12 contains several bug fixes for issues in release 9.6.11. For more information on the fixes in 9.6.12, see the [PostgreSQL documentation](#).

For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

PostgreSQL version 9.6.11 on Amazon RDS

PostgreSQL version 9.6.11 contains several bug fixes for issues in release 9.6.10. For more information on the fixes in PostgreSQL 9.6.11, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

With this version, the logical decoding plugin `wal2json` has been updated to commit 9e962ba.

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 9.6.10 on Amazon RDS

PostgreSQL version 9.6.10 contains several bug fixes for issues in release 9.6.9. For more information on the fixes in 9.6.10, see the [PostgreSQL documentation](#).

This version includes the following changes:

- Support for the `pglogical` extension version 2.2.0. Prerequisites for using this extension are the same as the prerequisites for using logical replication for PostgreSQL as described in [Logical replication for PostgreSQL on Amazon RDS \(p. 1502\)](#).
- Support for the `pg_similarity` extension version 2.2.0.
- An update for the `wal2json` extension to version 01c5c1e.
- An update for the `pg_hint_plan` extension to version 1.2.3.

For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 9.6.9 on Amazon RDS

PostgreSQL version 9.6.9 contains several bug fixes for issues in release 9.6.8. For more information on the fixes in 9.6.9, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

This version includes the following changes:

- The temporary file size limitation is user-configurable. You require the **rds_superuser** role to modify the `temp_file_limit` parameter.
- Update of the `GDAL` library, which is used by the PostGIS extension. See [Working with the PostGIS extension \(p. 1602\)](#).
- Update of the `ip4r` extension to version 2.1.1.
- Update of the `pgaudit` extension to version 1.1.1. See [Working with the pgaudit extension \(p. 1588\)](#).

Update of the `pg_repack` extension to version 1.4.3. See [Working with the pg_repack extension \(p. 1590\)](#).
- Update of the `plv8` extension to version 2.1.2.

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 9.6.8 on Amazon RDS

PostgreSQL version 9.6.8 contains several bug fixes for issues in release 9.6.6. For more information on the fixes in 9.6.8, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 9.6.6 on Amazon RDS

PostgreSQL version 9.6.6 contains several bug fixes for issues in release 9.6.5. For more information on the fixes in 9.6.6, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

This version includes the following features:

- Supports the `orafce` extension, version 3.6.1. This extension contains functions that are native to commercial databases, and can be helpful if you are porting a commercial database to PostgreSQL. For more information about using `orafce` with Amazon RDS, see [Working with the orafce extension \(p. 1591\)](#).
- Supports the `prefix` extension, version 1.2.6. This extension provides an operator for text prefix searches. For more information about `prefix`, see the [prefix project on GitHub](#).
- Supports version 2.3.4 of PostGIS, version 2.4.2 of pgrouting, and an updated version of wal2json.

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 9.6.5 on Amazon RDS

PostgreSQL version 9.6.5 contains several bug fixes for issues in release 9.6.4. For more information on the fixes in 9.6.5, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

This version also includes support for the `pgrouting`, `postgresql-hll` extensions, and the `decoder_raw` optional extension.

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 9.6.3 on Amazon RDS

PostgreSQL version 9.6.3 contains several new features and bug fixes. This version includes the following features:

- Supports the extension `pg_repack` version 1.4.0. You can use this extension to remove bloat from tables and indexes. For more information on using `pg_repack` with Amazon RDS, see [Working with the pg_repack extension \(p. 1590\)](#).
- Supports the extension `pgaudit` version 1.1.0. This extension provides detailed session and object audit logging. For more information on using `pgaudit` with Amazon RDS, see [Working with the pgaudit extension \(p. 1588\)](#).
- Supports `wal2json`, an output plugin for logical decoding.
- Supports the `auto_explain` extension. You can use this extension to log execution plans of slow statements automatically. The following example shows how to use `auto_explain` from within an Amazon RDS PostgreSQL session:

```
LOAD '$libdir/plugins/auto_explain';
```

For more information on using `auto_explain`, see the [PostgreSQL documentation](#).

PostgreSQL version 9.6.2 on Amazon RDS

PostgreSQL version 9.6.2 contains several new features and bug fixes. The new version also includes the following extension versions:

- PostGIS version 2.3.2
- `pg_freespacemap` version 1.1—Provides a way to examine the free space map (FSM). This extension provides an overloaded function called `pg_freespace`. The functions show the value recorded in the free space map for a given page, or for all pages in the relation.
- `pg_hint_plan` version 1.1.3—Provides control of execution plans by using hinting phrases at the beginning of SQL statements.
- `log_fdw` version 1.0—Using this extension from Amazon RDS, you can load and query your database engine log from within the database. For more information, see [Using the log_fdw extension \(p. 1499\)](#).
- With this version release, you can now edit the `max_worker_processes` parameter in a DB parameter group.

PostgreSQL version 9.6.2 on Amazon RDS also supports altering enum values. For more information, see [ALTER ENUM for PostgreSQL \(p. 1506\)](#).

For more information on the fixes in 9.6.2, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

PostgreSQL version 9.6.1 on Amazon RDS

PostgreSQL version 9.6.1 contains several new features and improvements. For more information about the fixes and improvements in PostgreSQL 9.6.1, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#). For information about performing parallel queries and phrase searching using Amazon RDS for PostgreSQL 9.6.1, see the [AWS database blog](#).

PostgreSQL version 9.6.1 includes the following changes:

- **Parallel query processing:** Supports parallel processing of large read-only queries, allowing sequential scans, hash joins, nested loops, and aggregates to be run in parallel. By default, parallel query processing is not enabled. To enable parallel query processing, set the parameter `max_parallel_workers_per_gather` to a value larger than zero.
- **Updated postgres_fdw extension:** Supports remote JOINs, SORTs, UPDATEs, and DELETE operations.
- **plv8 update:** Provides version 1.5.3 of the plv8 language.
- **PostGIS version update:** Supports POSTGIS="2.3.0 r15146" GEOS="3.5.0-CAPI-1.9.0 r4084" PROJ="Rel. 4.9.2, 08 September 2015" GDAL="GDAL 2.1.1, released 2016/07/07" LIBXML="2.9.1" LIBJSON="0.12" RASTER
- **Vacuum improvement:** Avoids scanning pages unnecessarily during vacuum freeze operations.
- **Full-text search support for phrases:** Supports the ability to specify a phrase-search query in tsquery input using the new operators <> and <N>.
- **Two new extensions are supported:**
 - `bloom`, an index access method based on [Bloom filters](#)
 - `pg_visibility`, which provides a means for examining the visibility map and page-level visibility information of a table.
- With the release of version 9.6.2, you can now edit the `max_worker_processes` parameter in a PostgreSQL version 9.6.1 DB parameter group.

You can create a new PostgreSQL 9.6.1 database instance using the AWS Management Console, AWS CLI, or RDS API. You can also upgrade an existing PostgreSQL 9.5 instance to version 9.6.1 using major version upgrade. If you want to upgrade a DB instance from version 9.4 to 9.6, you must perform a point-and-click upgrade to the next major version first. Each upgrade operation involves a short period of unavailability for your DB instance.

PostgreSQL 9.5 versions

Minor versions

- [PostgreSQL version 9.5.25 on Amazon RDS \(p. 1477\)](#)
- [PostgreSQL version 9.5.24 on Amazon RDS \(p. 1477\)](#)
- [PostgreSQL version 9.5.23 on Amazon RDS \(p. 1477\)](#)
- [PostgreSQL version 9.5.22 on Amazon RDS \(p. 1477\)](#)
- [PostgreSQL version 9.5.21 on Amazon RDS \(p. 1477\)](#)
- [PostgreSQL version 9.5.20 on Amazon RDS \(p. 1477\)](#)
- [PostgreSQL version 9.5.19 on Amazon RDS \(p. 1477\)](#)
- [PostgreSQL version 9.5.18 on Amazon RDS \(p. 1478\)](#)
- [PostgreSQL version 9.5.16 on Amazon RDS \(p. 1478\)](#)
- [PostgreSQL version 9.5.15 on Amazon RDS \(p. 1478\)](#)
- [PostgreSQL version 9.5.14 on Amazon RDS \(p. 1478\)](#)
- [PostgreSQL version 9.5.13 on Amazon RDS \(p. 1478\)](#)
- [PostgreSQL version 9.5.12 on Amazon RDS \(p. 1479\)](#)
- [PostgreSQL version 9.5.10 on Amazon RDS \(p. 1479\)](#)
- [PostgreSQL version 9.5.9 on Amazon RDS \(p. 1479\)](#)
- [PostgreSQL version 9.5.7 on Amazon RDS \(p. 1479\)](#)

- [PostgreSQL version 9.5.6 on Amazon RDS \(p. 1479\)](#)
- [PostgreSQL version 9.5.4 on Amazon RDS \(p. 1480\)](#)
- [PostgreSQL version 9.5.2 on Amazon RDS \(p. 1480\)](#)

PostgreSQL version 9.5.25 on Amazon RDS

PostgreSQL version 9.5.25 is now available on Amazon RDS. PostgreSQL version 9.5.25 contains several improvements that were announced for PostgreSQL release [9.5.25](#).

For information on all extensions, see [PostgreSQL version 9.5.x extensions supported on Amazon RDS \(p. 1496\)](#).

PostgreSQL version 9.5.24 on Amazon RDS

PostgreSQL version 9.5.24 is now available on Amazon RDS. PostgreSQL version 9.5.24 contains several improvements that were announced for PostgreSQL release [9.5.24](#).

For information on all extensions, see [PostgreSQL version 9.5.x extensions supported on Amazon RDS \(p. 1496\)](#).

PostgreSQL version 9.5.23 on Amazon RDS

PostgreSQL version 9.5.23 is now available on Amazon RDS. PostgreSQL version 9.5.23 contains several improvements that were announced for PostgreSQL release [9.5.23](#).

For information on all extensions, see [PostgreSQL version 9.5.x extensions supported on Amazon RDS \(p. 1496\)](#).

PostgreSQL version 9.5.22 on Amazon RDS

PostgreSQL version 9.5.22 contains several bug fixes for issues in release 9.5.21. For more information on the fixes in PostgreSQL 9.5.22, see the [PostgreSQL 9.5.22 documentation](#).

This version also includes the following change:

1. Upgraded the `pg_hint_plan` extension to version 1.1.9.

For information on all extensions, see [PostgreSQL version 9.5.x extensions supported on Amazon RDS \(p. 1496\)](#).

PostgreSQL version 9.5.21 on Amazon RDS

PostgreSQL version 9.5.21 contains several bug fixes for issues in release 9.5.20. For more information on the fixes in PostgreSQL 9.5.21, see the [PostgreSQL 9.5.21 documentation](#).

PostgreSQL version 9.5.20 on Amazon RDS

PostgreSQL version 9.5.20 contains several bug fixes for issues in release 9.5.19. For more information on the fixes in PostgreSQL 9.5.20, see the [PostgreSQL documentation](#).

PostgreSQL version 9.5.19 on Amazon RDS

PostgreSQL version 9.5.19 contains several bug fixes for issues in release 9.5.18. For more information on the fixes in PostgreSQL 9.5.19, see the [PostgreSQL documentation](#).

The PostGIS extension is updated to version 2.5.2.

PostgreSQL version 9.5.18 on Amazon RDS

This release contains bug fixes and improvements done by the PostgreSQL community.

With this release, the pg_hint_plan extension has been updated to version 1.1.8.

For more information on the fixes in PostgreSQL 9.5.18, see the [PostgreSQL documentation](#).

PostgreSQL version 9.5.16 on Amazon RDS

PostgreSQL version 9.5.16 contains several bug fixes for issues in release 9.5.15. For more information on the fixes in 9.5.16, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 9.5.15 on Amazon RDS

PostgreSQL version 9.5.15 contains several bug fixes for issues in release 9.5.14. For more information on the fixes in 9.5.15, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 9.5.14 on Amazon RDS

PostgreSQL version 9.5.14 contains several bug fixes for issues in release 9.5.13. For more information on the fixes in 9.5.14, see the [PostgreSQL documentation](#).

For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 9.5.13 on Amazon RDS

PostgreSQL version 9.5.13 contains several bug fixes for issues in release 9.5.12. For more information on the fixes in 9.5.13, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

This version includes the following extension updates:

- Update of the pgaudit extension to version 1.0.6. See [Working with the pgaudit extension \(p. 1588\)](#).
- Update of the pg_hint_plan extension to version 1.1.5.
- Update of the plv8 extension to version 2.1.2.

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 9.5.12 on Amazon RDS

PostgreSQL version 9.5.12 contains several bug fixes for issues in release 9.5.10. For more information on the fixes in 9.5.12, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

For the complete list of extensions supported by Amazon RDS for PostgreSQL, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

PostgreSQL version 9.5.10 on Amazon RDS

PostgreSQL version 9.5.10 contains several bug fixes for issues in version 9.5.9. For more information on the fixes in 9.5.10, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

PostgreSQL version 9.5.9 on Amazon RDS

PostgreSQL version 9.5.9 contains several bug fixes for issues in version 9.5.8. For more information on the fixes in 9.5.9, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

PostgreSQL version 9.5.7 on Amazon RDS

PostgreSQL version 9.5.7 contains several new features and bug fixes. This version includes the following features:

- Supports the extension `pgaudit` version 1.0.5. This extension provides detailed session and object audit logging. For more information on using `pgaudit` with Amazon RDS, see [Working with the pgaudit extension \(p. 1588\)](#).
- Supports `wal2json`, an output plugin for logical decoding.
- Supports the `auto_explain` extension. You can use this extension to log execution plans of slow statements automatically. The following example shows how to use `auto_explain` from within an Amazon RDS PostgreSQL session.

```
LOAD '$libdir/plugins/auto_explain';
```

For more information on using `auto_explain`, see the [PostgreSQL documentation](#).

PostgreSQL version 9.5.6 on Amazon RDS

PostgreSQL version 9.5.6 contains several new features and bug fixes. The new version also includes the following extension versions:

- PostGIS version 2.2.5
- `pg_freespacemap` version 1.1—Provides a way to examine the free space map (FSM). This extension provides an overloaded function called `pg_freespace`. This function shows the value recorded in the free space map for a given page, or for all pages in the relation.
- `pg_hint_plan` version 1.1.3—Provides control of execution plans by using hinting phrases at the beginning of SQL statements.

PostgreSQL version 9.5.6 on Amazon RDS also supports altering enum values. For more information, see [ALTER ENUM for PostgreSQL \(p. 1506\)](#).

For more information on the fixes in 9.5.6, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

PostgreSQL version 9.5.4 on Amazon RDS

PostgreSQL version 9.5.4 contains several fixes to issue found in previous versions. For more information on the fixes in 9.5.4, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

PostgreSQL supports the streaming of WAL changes using logical replication decoding. Amazon RDS supports logical replication for PostgreSQL version 9.5.4 and higher. For more information about PostgreSQL logical replication on Amazon RDS, see [Logical replication for PostgreSQL on Amazon RDS \(p. 1502\)](#).

Beginning with PostgreSQL version 9.5.4 for Amazon RDS, the command ALTER USER WITH BYPASSRLS is supported.

PostgreSQL versions 9.5.4 and later support event triggers, and Amazon RDS supports event triggers for these versions. You can use the master user account can be used to create, modify, rename, and delete event triggers. Event triggers are at the DB instance level, so they can apply to all databases on an instance. For more information about PostgreSQL event triggers on Amazon RDS, see [Event triggers for PostgreSQL on Amazon RDS \(p. 1504\)](#).

PostgreSQL version 9.5.2 on Amazon RDS

PostgreSQL version 9.5.2 contains several fixes to issues found in previous versions. For more information on the features in 9.5.2, see the [PostgreSQL documentation](#). For information on upgrading the engine version for your PostgreSQL DB instance, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#).

PostgreSQL version 9.5.2 doesn't support the db.m1 or db.m2 DB instance classes. If you need to upgrade a DB instance running PostgreSQL version 9.4 to version 9.5.2 to one of these instance classes, you need to scale compute. To do that, you need a comparable db.t2 or db.m3 DB instance class before you can upgrade a DB instance running PostgreSQL version 9.4 to version 9.5.2. For more information on DB instance classes, see [DB instance classes \(p. 7\)](#).

Native PostgreSQL version 9.5.2 introduced the command ALTER USER WITH BYPASSRLS.

This release includes updates from previous versions, including the following:

- **CVE-2016-2193:** Fixes an issue where a query plan might be reused for more than one ROLE in the same session. Reusing a query plan can cause the query to use the wrong set of Row Level Security (RLS) policies.
- **CVE-2016-3065:** Fixes a server crash bug triggered by using pageinspect with BRIN index pages. Because an attacker might be able to expose a few bytes of server memory, this crash is being treated as a security issue.

Major enhancements in RDS for PostgreSQL 9.5 include the following:

- UPSERT: Allow INSERTs that would generate constraint conflicts to be turned into UPDATEs or ignored
- Add the GROUP BY analysis features GROUPING SETS, CUBE, and ROLLUP
- Add row-level security control
- Create mechanisms for tracking the progress of replication, including methods for identifying the origin of individual changes during logical replication

- Add Block Range Indexes (BRIN)
- Add substantial performance improvements for sorting
- Add substantial performance improvements for multi-CPU machines
- PostGIS 2.2.2 - To use this latest version of PostGIS, use the ALTER EXTENSION UPDATE statement to update after you upgrade to version 9.5.2. Example:

```
ALTER EXTENSION POSTGIS UPDATE TO '2.2.2'
```
- Improved visibility of autovacuum sessions by allowing the rds_superuser account to view autovacuum sessions in pg_stat_activity. For example, you can identify and terminate an autovacuum session that is blocking a command from running, or running slower than a manually issued vacuum command.

RDS for PostgreSQL version 9.5.2 includes the following new extensions:

- **address_standardizer** – A single-line address parser that takes an input address and normalizes it based on a set of rules stored in a table, helper lex, and gaz tables.
- **hstore_plperl** – Provides transforms for the hstore type for PL/Perl.
- **tsm_system_rows** – Provides the table sampling method SYSTEM_ROWS, which can be used in the TABLESAMPLE clause of a SELECT command.
- **tsm_system_time** – Provides the table sampling method SYSTEM_TIME, which can be used in the TABLESAMPLE clause of a SELECT command.

PostgreSQL extensions supported on Amazon RDS

RDS for PostgreSQL supports many PostgreSQL extensions. The PostgreSQL community sometimes refers to these as modules. Extensions expand on the functionality provided by the PostgreSQL engine. You can find a list of extensions supported by Amazon RDS in the default DB parameter group for that PostgreSQL version. You can also see the current extensions list using `psql` by showing the `rds.extensions` parameter as in the following example.

```
SHOW rds.extensions;
```

Note

Parameters added in a minor version release might display inaccurately when using the `rds.extensions` parameter in `psql`.

The following sections show the extensions supported by Amazon RDS for the major PostgreSQL versions.

Contents

- [Restricting installation of PostgreSQL extensions \(p. 1482\)](#)
- [PostgreSQL version 13 extensions supported on Amazon RDS \(p. 1483\)](#)
 - [PostgreSQL trusted extensions \(p. 1485\)](#)
- [PostgreSQL version 12 extensions supported on Amazon RDS \(p. 1486\)](#)
- [PostgreSQL version 11.x extensions supported on Amazon RDS \(p. 1489\)](#)
- [PostgreSQL version 10.x extensions supported on Amazon RDS \(p. 1491\)](#)
- [PostgreSQL version 9.6.x extensions supported on Amazon RDS \(p. 1494\)](#)
- [PostgreSQL version 9.5.x extensions supported on Amazon RDS \(p. 1496\)](#)

Restricting installation of PostgreSQL extensions

You can restrict which extensions can be installed on a PostgreSQL DB instance. To do so, set the `rds.allowed_extensions` parameter to a string of comma-separated extension names. Only these extensions can then be installed in the PostgreSQL DB instance.

The default string for the `rds.allowed_extensions` parameter is `'*'`, which means that any extension available for the engine version can be installed. Changing the `rds.allowed_extensions` parameter does not require a database restart because it's a dynamic parameter.

The PostgreSQL DB instance engine must be one of the following versions for you to use the `rds.allowed_extensions` parameter:

- PostgreSQL 13.2 or a later minor version
- PostgreSQL 12.6 or a later minor version

To see which extension installations are allowed, use the following `psql` command.

```
postgres=>SHOW rds.allowed_extensions;
rds.allowed_extensions
-----
*
```

If an extension was installed prior to it being left out of the list in the `rds.allowed_extensions` parameter, the extension can still be used normally, and commands such as `ALTER EXTENSION` and

`DROP EXTENSION` will continue to work. However, after an extension is restricted, `CREATE EXTENSION` commands for the restricted extension will fail.

Installation of extension dependencies with `CREATE EXTENSION CASCADE` are also restricted. The extension and its dependencies must be specified in `rds.allowed_extensions`. If an extension dependency installation fails, the entire `CREATE EXTENSION CASCADE` statement will fail.

If an extension is not included with the `rds.allowed_extensions` parameter, you will see an error such as the following if you try to install it.

```
ERROR: permission denied to create extension "extension-name"  
HINT: This extension is not specified in "rds.allowed_extensions".
```

PostgreSQL version 13 extensions supported on Amazon RDS

The following table shows PostgreSQL extensions for PostgreSQL version 13 that are currently supported on Amazon RDS. For more information on PostgreSQL extensions, see [Packaging related objects into an extension](#).

Extension	13.1	13.2
address_standardizer	3.0.2	3.0.2
address_standardizer_data_us	3.0.2	3.0.2
amcheck	1.2	1.2
aws_commons (p. 1560)	1.1	1.1
aws_lambda (p. 1618)	NA	1.0
aws_s3.table_import_from_s3 (p. 1560) aws_s3.query_export_to_s3 (p. 1574)		1.1
bloom	1.0	1.0
bool_plperl	1.0	1.0
btree_gin	1.3	1.3
btree_gist	1.5	1.5
citext	1.6	1.6
cube	1.4	1.4
dblink	1.2	1.2
dict_int	1.0	1.0
dict_xsyn	1.0	1.0
earthdistance	1.1	1.1
fuzzystrmatch	1.1	1.1
hll	2.15	2.15

Extension	13.1	13.2
hstore	1.7	1.7
hstore_plperl	1.0	1.0
ICU module	60.2	60.2
intagg	1.1	1.1
intarray	1.3	1.3
ip4r	2.4	2.4
isn	1.2	1.2
jsonb_plperl	1.0	1.0
log_fdw (p. 1499)	1.2	1.2
ltree	1.2	1.2
orafce	3.13.4	3.13.4
pageinspect	1.8	1.8
pg_bigm	NA	1.2
pg_buffercache	1.3	1.3
pg_cron (p. 1607)	1.3.0	1.3.0
pg_freespacemap	1.2	1.2
pg_hint_plan	1.3.7	1.3.7
pg_partman (p. 1614)	4.4.0	4.4.0
pg_prewarm	1.2	1.2
pg_proctab	0.0.9	0.0.9
pg_repack	1.4.6	1.4.6
pg_similarity	1.0	1.0
pg_stat_statements	1.8	1.8
pg_transport (p. 1563)	1.0	1.0
pg_trgm	1.5	1.5
pg_visibility	1.2	1.2
pgaudit	1.5	1.5
pgcrypto	1.3	1.3
pglogical	2.3.3	2.3.3
pgrouting	3.1.0	3.1.0
pgrowlocks	1.2	1.2

Extension	13.1	13.2
pgstattuple	1.5	1.5
pgTAP	1.1.0	1.1.0
plcoffee	2.3.15	2.3.15
plls	2.3.15	2.3.15
plperl	1.0	1.0
plpgsql	1.0	1.0
plprofiler	4.1	4.1
pltcl	1.0	1.0
plv8 (p. 1500)	2.3.15	2.3.15
PostGIS (p. 1602)	3.0.2	3.0.2
postgis_raster	3.0.2	3.0.2
postgis_tiger_geocoder	3.0.2	3.0.2
postgis_topology	3.0.2	3.0.2
postgres_fdw	1.0	1.0
prefix	1.2.0	1.2.0
rdkit	3.8	3.8
sslinfo	1.2	1.2
tablefunc	1.0	1.0
test_parser	1.0	1.0
tsm_system_rows	1.0	1.0
tsm_system_time	1.0	1.0
unaccent	1.1	1.1
uuid-ossp	1.1	1.1
wal2json	2.3	2.3

PostgreSQL trusted extensions

To install most PostgreSQL extensions requires `rds_superuser` privileges. PostgreSQL 13 introduced [trusted extensions](#), which reduces the need to grant `rds_superuser` privileges to regular users. With this feature, users can install many extensions if they have the `CREATE` privilege on the current database instead of requiring the `rds_superuser` role. For more information, see the SQL [CREATE EXTENSION](#) command in the PostgreSQL documentation.

The following lists the extensions that can be installed by a user who has the `CREATE` privilege on the current database and do not require the `rds_superuser` role:

- [bool_plperl](#)
- [btree_gin](#)
- [btree_gist](#)
- [citext](#)
- [cube](#)
- [dict_int](#)
- [fuzzystrmatch](#)
- [hstore](#)
- [intarray](#)
- [isn](#)
- [jsonb_plperl](#)
- [ltree](#)
- [pg_trgm](#)
- [pgcrypto](#)
- [plperl](#)
- [plpgsql](#)
- [pltcl](#)
- [tablefunc](#)
- [tsm_system_rows](#)
- [tsm_system_time](#)
- [unaccent](#)
- [uuid-ossp](#)

PostgreSQL version 12 extensions supported on Amazon RDS

The following table shows PostgreSQL extensions for PostgreSQL version 12 that are currently supported on Amazon RDS. For more information on PostgreSQL extensions, see [Packaging related objects into an extension](#).

Extension	12.2	12.3	12.4	12.5	12.6
address_standardizer	3.0.0	3.0.0	3.0.0	3.0.0	3.0.2
address_standardizer_data_us	3.0.0	3.0.0	3.0.0	3.0.0	3.0.2
amcheck	1.2	1.2	1.2	1.2	1.2
aws_commons (p. 1560)	1.0	1.0	1.0	1.0	1.0
aws_lambda (p. 1618)	NA	NA	NA	NA	1.0
aws_s3.table_import_from_s3 (p. 1060) aws_s3.query_export_to_s3 (p. 1574)		1.0	1.1	1.1	1.1
bloom	1.0	1.0	1.0	1.0	1.0
btree_gin	1.3	1.3	1.3	1.3	1.3
btree_gist	1.5	1.5	1.5	1.5	1.5

Extension	12.2	12.3	12.4	12.5	12.6
citext	1.6	1.6	1.6	1.6	1.6
cube	1.4	1.4	1.4	1.4	1.4
dblink	1.2	1.2	1.2	1.2	1.2
dict_int	1.0	1.0	1.0	1.0	1.0
dict_xsyn	1.0	1.0	1.0	1.0	1.0
earthdistance	1.1	1.1	1.1	1.1	1.1
fuzzystrmatch	1.1	1.1	1.1	1.1	1.1
hll	2.14	2.14	2.14	2.14	2.14
hstore	1.6	1.6	1.6	1.6	1.6
hstore_plperl	1.0	1.0	1.0	1.0	1.0
ICU module	60.2	60.2	60.2	60.2	60.2
intagg	1.1	1.1	1.1	1.1	1.1
intarray	1.2	1.2	1.2	1.2	1.2
ip4r	2.4	2.4	2.4	2.4	2.4
isn	1.2	1.2	1.2	1.2	1.2
jsonb_plperl	1.0	1.0	1.0	1.0	1.0
log_fdw (p. 1499)	1.1	1.1	1.1	1.1	1.1
ltree	1.1	1.1	1.1	1.1	1.1
orafce	3.8	3.8	3.8	3.8	3.8
pageinspect	1.7	1.7	1.7	1.7	1.7
pg_bigm	NA	NA	NA	NA	1.2
pg_buffercache	1.3	1.3	1.3	1.3	1.3
pg_cron (p. 1607)	NA	NA	NA	1.3.0	1.3.0
pg_freespacemap	1.2	1.2	1.2	1.2	1.2
pg_hint_plan	1.3.4	1.3.5	1.3.5	1.3.5	1.3.5
pg_partman (p. 1614)	NA	NA	NA	4.4.0	4.4.0
pg_prewarm	1.2	1.2	1.2	1.2	1.2
pg_proctab	NA	NA	0.0.9	0.0.9	0.0.9
pg_repack	1.4.5	1.4.5	1.4.5	1.4.5	1.4.5
pg_similarity	1.0	1.0	1.0	1.0	1.0
pg_stat_statements	1.7	1.7	1.7	1.7	1.7

Extension	12.2	12.3	12.4	12.5	12.6
pg_transport (p. 1563)	1.0	1.0	1.0	1.0	1.0
pg_trgm	1.4	1.4	1.4	1.4	1.4
pg_visibility	1.2	1.2	1.2	1.2	1.2
pgaudit	1.4	1.4	1.4	1.4	1.4
pgcrypto	1.3	1.3	1.3	1.3	1.3
pglogical	2.3.0	2.3.1	2.3.2	2.3.2	2.3.2
pgrouting	3.0.0	3.0.0	3.0.0	3.0.0	3.0.0
pgrowlocks	1.2	1.2	1.2	1.2	1.2
pgstattuple	1.5	1.5	1.5	1.5	1.5
pgTAP	1.1.0	1.1.0	1.1.0	1.1.0	1.1.0
plcoffee	2.3.14	2.3.14	2.3.14	2.3.14	2.3.14
plls	2.3.14	2.3.14	2.3.14	2.3.14	2.3.14
plperl	1.0	1.0	1.0	1.0	1.0
plpgsql	1.0	1.0	1.0	1.0	1.0
plprofiler	4.1	4.1	4.1	4.1	4.1
pltcl	1.0	1.0	1.0	1.0	1.0
plv8 (p. 1500)	2.3.14	2.3.14	2.3.14	2.3.14	2.3.14
PostGIS (p. 1602)	3.0.0	3.0.0	3.0.0	3.0.0	3.0.2
postgis_raster	3.0.0	3.0.0	3.0.0	3.0.0	3.0.2
postgis_tiger_geocoder	3.0.0	3.0.0	3.0.0	3.0.0	3.0.2
postgis_topology	3.0.0	3.0.0	3.0.0	3.0.0	3.0.2
postgres_fdw	1.0	1.0	1.0	1.0	1.0
prefix	1.2.0	1.2.0	1.2.0	1.2.0	1.2.0
rdkit	NA	NA	3.8	3.8	3.8
sslinfo	1.2	1.2	1.2	1.2	1.2
tablefunc	1.0	1.0	1.0	1.0	1.0
test_parser	1.0	1.0	1.0	1.0	1.0
tsm_system_rows	1.0	1.0	1.0	1.0	1.0
tsm_system_time	1.0	1.0	1.0	1.0	1.0
unaccent	1.1	1.1	1.1	1.1	1.1
uuid-ossp	1.1	1.1	1.1	1.1	1.1

Extension	12.2	12.3	12.4	12.5	12.6
wal2json	2.1	2.1	2.3	2.3	2.3

PostgreSQL version 11.x extensions supported on Amazon RDS

The following tables show PostgreSQL extensions for PostgreSQL version 11.x that are currently supported by PostgreSQL on Amazon RDS. "N/A" indicates that the extension is not available for that PostgreSQL version. For more information on PostgreSQL extensions, see [Packaging related objects into an extension](#).

Extension	11.1	11.2	11.4	11.5	11.6	11.7	11.8	11.9	11.10	11.11
address_standardizer	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1
address_standardizer_data	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1
aws_s3.table_import_from_s3 (p. 1574)	N/A	1.1	1.1	1.1						
aws_s3.query_export_to_s3 (p. 1574)										
amcheck	yes									
auto_explain	yes									
bloom	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
btree_gin	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
btree_gist	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
citext	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
cube	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
dblink	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
decoder_raw	yes									
dict_int	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
dict_xsyn	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
earthdistance	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
fuzzystrmatch	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
hstore	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
hstore_plperl	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
ICU module	60.2	60.2	60.2	60.2	60.2	60.2	60.2	60.2	60.2	60.2
intagg	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
intarray	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
ip4r	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3

Extension	11.1	11.2	11.4	11.5	11.6	11.7	11.8	11.9	11.10	11.11
isn	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
log_fdw (p. 1499)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
libprotobuf	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0
ltree	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
orafce	3.7	3.7	3.7	3.7	3.7	3.8	3.8	3.8	3.8	3.8
pageinspect	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6
pg_bigm	NA	1.2								
pg_buffercache	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
pg_freespacemap	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pg_hint_plan	1.3.2	1.3.2	1.3.4	1.3.4	1.3.4	1.3.4	1.3.5	1.3.5	1.3.5	1.3.5
pg_prewarm	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pg_proctab	N/A	0.0.9	0.0.9	0.0.9						
pg_repack	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4
pg_similarity	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pg_stat_statements	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6
pg_transport	N/A	N/A	N/A	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pg_trgm	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
pg_visibility	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pgaudit	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.1	1.3.1	1.3.1
pgcrypto	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
pglogical	2.2.1	2.2.1	2.2.1	2.2.1	2.2.1	2.2.1	2.2.1	2.2.2	2.2.2	2.2.2
pgrowlocks	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pgrouting	2.6.1	2.6.1	2.6.1	2.6.1	2.6.1	2.6.1	2.6.1	2.6.1	2.6.1	2.6.1
pgstattuple	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
pgTAP	N/A	1.0	1.0	1.0	1.1.0	1.1.0	1.1.0	1.1.0	1.1.0	1.1.0
plcoffee	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8
plls	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8
plperl	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
plpgsql	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
plprofiler	N/A	N/A	N/A	N/A	4.1	4.1	4.1	4.1	4.1	4.1
pltcl	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Extension	11.1	11.2	11.4	11.5	11.6	11.7	11.8	11.9	11.10	11.11
plv8 (p. 1500)	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8	2.3.8
PostGIS (p. 1602)	2.5.1	2.5.1	2.5.1	2.5.2	2.5.2	2.5.2	2.5.2	2.5.2	2.5.2	2.5.2
postgis_tiger_geocoder	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1
postgis_topology	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1
postgres_fdw	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
postgresql-hll	2.11	2.11	2.11	2.11	2.11	2.11	2.11	2.11	2.11	2.11
prefix	1.2.8	1.2.8	1.2.8	1.2.8	1.2.8	1.2.8	1.2.8	1.2.8	1.2.8	1.2.8
rdkit	N/A	N/A	N/A	N/A	N/A	N/A	N/A	3.8	3.8	3.8
sslinfo	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
tablefunc	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
test_decoding	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
test_parser	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tsm_system_rows	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tsm_system_time	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
unaccent	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
uuid-ossp	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
wal2json	Commit hash 9e962bae962bae962bae962bae962bad	Commit hash 9e962bae962bae962bae962bad	Commit hash 9e962bae962bae962bad	Commit hash 9e962bae962bad	Commit hash 9e962bad	Commit hash 9e962bad	2.1	2.1	2.3	2.3

PostgreSQL version 10.x extensions supported on Amazon RDS

The following tables show PostgreSQL extensions for PostgreSQL version 10 that are currently supported by PostgreSQL on Amazon RDS. "N/A" indicates that the extension is not available for that PostgreSQL version. For more information on PostgreSQL extensions, see [Packaging related objects into an extension](#).

Extension	10.1	10.3	10.4	10.5	10.6	10.7	10.9	10.10	10.11	10.12	10.13	10.14	10.15	10.16
address_standardizer	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2
address_standardizer_data	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2
amcheck	N/A	yes												
auto_explain	yes													
aws_s3 (p. 1552)	N/A	1.1	1.1	1.1										

Extension	10.1	10.3	10.4	10.5	10.6	10.7	10.9	10.10	10.11	10.12	10.13	10.14	10.15	10.16
bloom	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
btree_gin	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
btree_gist	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
chkpass	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
citext	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
cube	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
dblink	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
decoder_raw	yes													
dict_int	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
dict_xsyn	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
earthdistance	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
fuzzystrmatch	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
hstore	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
hstore_plperl	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
ICU module	60.2	60.2	60.2	60.2	60.2	60.2	60.2	60.2	60.2	60.2	60.2	60.2	60.2	60.2
intagg	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
intarray	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
ip4r	2.0	2.0	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1
isn	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
log_fdw (p. 1499)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
libprotobuf	N/A	N/A	N/A	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0	1.3.0
ltree	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
orafce	3.6.1	3.6.1	3.6.1	3.6.1	3.6.1	3.6.1	3.6.1	3.6.1	3.6.1	3.6.1	3.8	3.8	3.8	3.8
pgaudit	1.2.0	1.2.0	1.2.0	1.2.0	1.2.0	1.2.0	1.2.0	1.2.0	1.2.0	1.2.0	1.2.0	1.2.1	1.2.1	1.2.1
pg_buffercache	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
pg_freespacemap	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pg_hint_plan	1.3.0	1.3.0	1.3.0	1.3.1	1.3.1	1.3.1	1.3.3	1.3.3	1.3.3	1.3.3	1.3.5	1.3.5	1.3.5	1.3.5
pg_prewarm	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pg_repack	1.4.2	1.4.2	1.4.3	1.4.3	1.4.3	1.4.3	1.4.3	1.4.3	1.4.3	1.4.3	1.4.3	1.4.3	1.4.3	1.4.3
pg_similarity	N/A	N/A	N/A	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pg_stat_statements	1.5	1.5	1.5	1.5	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6

Amazon Relational Database Service User Guide

PostgreSQL version 10.x extensions supported on Amazon RDS

Extension	10.1	10.3	10.4	10.5	10.6	10.7	10.9	10.10	10.11	10.12	10.13	10.14	10.15	10.16		
wal2json	Comm hash 5352c552c552c04c5c1e962ba962ba962ba962ba	2.1	2.3	2.3	2.3											

The `tsearch2` extension is deprecated in version 10. The `tsearch2` extension was removed from PostgreSQL version 11.1 on Amazon RDS (p. 1465).

PostgreSQL version 9.6.x extensions supported on Amazon RDS

The following tables show PostgreSQL extensions for PostgreSQL version 9.6.x that are currently supported by PostgreSQL on Amazon RDS. "N/A" indicates that the extension is not available for that PostgreSQL version. For more information on PostgreSQL extensions, see [Packaging related objects into an extension](#).

Extension	9.6.1	9.6.2	9.6.3	9.6.5	9.6.6	9.6.8	9.6.9	9.6.1	9.6.1	9.6.1	9.6.1	9.6.1	9.6.1	9.6.1	9.6.1	9.6.2	9.6.21
address	2.3.0	2.3.2	2.3.2	2.3.2	2.3.2	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4
address	2.3.0	2.3.2	2.3.2	2.3.2	2.3.2	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4
auto_explain	N/A	N/A	N/A	yes													
bloom	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
btree_gist	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
btree_gist	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
chkpass	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
citext	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
cube	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
dblink	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
decode	N/A	yes															
dict_int	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
dict_xid	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
earthdistance	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
fuzzystrmatch	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
hstore	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
hstore	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
intagg	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
intarray	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
ip4r	2.0	2.0	2.0	2.0	2.0	2.0	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1	2.1.1

Amazon Relational Database Service User Guide

PostgreSQL version 9.6.x extensions supported on Amazon RDS

Extension	9.6.1	9.6.2	9.6.3	9.6.5	9.6.6	9.6.8	9.6.9	9.6.1	9.6.1	9.6.1	9.6.1	9.6.1	9.6.1	9.6.1	9.6.1	9.6.1	9.6.1	9.6.2	9.6.21
prefix	N/A	N/A	N/A	N/A	1.2.6	1.2.6	1.2.6	1.2.6	1.2.6	1.2.6	1.2.6	1.2.6	1.2.6	1.2.6	1.2.6	1.2.6	1.2.6	1.2.6	1.2.6
sslinfo	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
tablefunc	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
test_decodings	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
test_parser	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tsearch	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tsm_syslogem_10aws	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tsm_syslogem_10one	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
unaccent	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
uuid-ossp	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
wal2json	N/A	N/A	Common	Common	Common	Common	Common	Common	Common	Common	Common	Common	Common	Common	Common	Common	version	version	version
			hash	hash	hash	hash	hash	hash	hash	hash	hash	hash	hash	hash	hash	hash	2.1	2.1	2.3
			28284695ab695ab5952c5352c64c5c9e962ba962ba962ba962ba962ba														2.3	2.3	

PostgreSQL version 9.5.x extensions supported on Amazon RDS

The following tables show PostgreSQL extensions for PostgreSQL version 9.5.x that are currently supported by PostgreSQL on Amazon RDS. "N/A" indicates that the extension is not available for that PostgreSQL version. For more information on PostgreSQL extensions, see [Packaging related objects into an extension](#).

Extension	9.5.2	9.5.4	9.5.6	9.5.7	9.5.9	9.5.1	9.5.1	9.5.1	9.5.1	9.5.1	9.5.1	9.5.1	9.5.1	9.5.2	9.5.2	9.5.2	9.5.2	9.5.25
address	2.2	2.2	2.2	2.2	2.2	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5
address	2.2	2.2	2.2	2.2	2.2	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5	2.2.5
auto_explain	N/A	yes	yes	yes	yes	yes												
bloom	N/A																	
btree_gin	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
btree_gist	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
chkpass	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
citext	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
cube	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
dblink	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1

Extension	9.5.2	9.5.4	9.5.6	9.5.7	9.5.9	9.5.1	9.5.1	9.5.1	9.5.1	9.5.1	9.5.1	9.5.1	9.5.1	9.5.2	9.5.2	9.5.2	9.5.25
dict_int	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
dict_xid	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
earthdistance	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
fuzzystrmatch	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
hstore	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
hstore	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
intagg	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
intarray	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
ip4r	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
isn	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
log_fcnames	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A							
ltree	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pgaudit	N/A	N/A	N/A	1.0.5	1.0.5	1.0.5	1.0.5	1.0.6	1.0.6	1.0.6	1.0.6	1.0.6	1.0.6	1.0.6	1.0.6	1.0.6	1.0.6
pg_buffercache	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pg_fsync	N/A	N/A	N/A	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pg_hints	N/A	N/A	1.1.3	1.1.3	1.1.3	1.1.3	1.1.3	1.1.5	1.1.5	1.1.5	1.1.5	1.1.8	1.1.8	1.1.8	1.1.8	1.1.9	1.1.9
pg_notify	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pg_prewarm	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pg_statements	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
pg_trgm	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pg_visibility	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A							
pgcrypt	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pgrowlocks	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pgstatuple	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
plcoffee	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0
plls	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0
plperl	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
plpgsql	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pltcl	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Some supported PostgreSQL features

Amazon RDS supports many of the most common PostgreSQL extensions and features.

Topics

- [Using the log_fdw extension \(p. 1499\)](#)
- [Upgrading plv8 \(p. 1500\)](#)
- [Logical replication for PostgreSQL on Amazon RDS \(p. 1502\)](#)
- [Event triggers for PostgreSQL on Amazon RDS \(p. 1504\)](#)
- [Huge pages for Amazon RDS for PostgreSQL \(p. 1505\)](#)
- [Tablespaces for PostgreSQL on Amazon RDS \(p. 1505\)](#)
- [Autovacuum for PostgreSQL on Amazon RDS \(p. 1506\)](#)
- [RAM disk for the stats_temp_directory \(p. 1506\)](#)
- [ALTER ENUM for PostgreSQL \(p. 1506\)](#)

Using the log_fdw extension

The `log_fdw` extension is new for Amazon RDS for PostgreSQL version 9.6.2 and later. Using this extension, you can access your database engine log using a SQL interface. In addition to viewing the `stderr` log files that are generated by default on RDS, you can view CSV logs (set the `log_destination` parameter to `csvlog`) and build foreign tables with the data neatly split into several columns.

This extension introduces two new functions that make it easy to create foreign tables for database logs:

- `list_postgres_log_files()` – Lists the files in the database log directory and the file size in bytes.
- `create_foreign_table_for_log_file(table_name text, server_name text, log_file_name text)` – Builds a foreign table for the specified file in the current database.

All functions created by `log_fdw` are owned by `rds_superuser`. Members of the `rds_superuser` role can grant access to these functions to other database users.

The following example shows how to use the `log_fdw` extension.

To use the log_fdw extension

1. Get the `log_fdw` extension.

```
postgres=> CREATE EXTENSION log_fdw;
CREATE EXTENSION
```

2. Create the log server as a foreign data wrapper.

```
postgres=> CREATE SERVER log_server FOREIGN DATA WRAPPER log_fdw;
CREATE SERVER
```

3. Select all from a list of log files.

```
postgres=> SELECT * from list_postgres_log_files() order by 1;
```

A sample response is as follows.

file_name	file_size_bytes
postgresql.log.2016-08-09-22.csv	1111
postgresql.log.2016-08-09-23.csv	1172
postgresql.log.2016-08-10-00.csv	1744
postgresql.log.2016-08-10-01.csv	1102
(4 rows)	

4. Create a table with a single 'log_entry' column for non-CSV files.

```
postgres=> SELECT create_foreign_table_for_log_file('my_postgres_error_log',
    'log_server', 'postgresql.log.2016-08-09-22.csv');
```

A sample response is as follows.

```
-----  
(1 row)
```

5. Select a sample of the log file. The following code retrieves the log time and error message description.

```
postgres=> SELECT log_time, message from my_postgres_error_log order by 1;
```

A sample response is as follows.

log_time	message
Tue Aug 09 15:45:18.172 2016 PDT	ending log output to stderr
Tue Aug 09 15:45:18.175 2016 PDT	database system was interrupted; last known up at 2016-08-09 22:43:34 UTC
Tue Aug 09 15:45:18.223 2016 PDT	checkpoint record is at 0/90002E0
Tue Aug 09 15:45:18.223 2016 PDT	redo record is at 0/90002A8; shutdown FALSE
Tue Aug 09 15:45:18.223 2016 PDT	next transaction ID: 0/1879; next OID: 24578
Tue Aug 09 15:45:18.223 2016 PDT	next MultiXactId: 1; next MultiXactOffset: 0
Tue Aug 09 15:45:18.223 2016 PDT	oldest unfrozen transaction ID: 1822, in database 1
(7 rows)	

Upgrading plv8

If you use [plv8](#) and upgrade PostgreSQL to a new plv8 version, you immediately take advantage of the new extension. Take the following steps to synchronize your catalog metadata with the new version of

plv8. These steps are optional, but we highly recommended that you complete them to avoid metadata mismatch warnings.

To synchronize your catalog metadata with a new version of plv8

1. Verify that you need to update. To do this, run the following command while connected to your instance.

```
select * from pg_available_extensions where name in
('plv8','plls','plcoffee');
```

If your results contain values for an installed version that is a lower number than the default version, continue with this procedure to update your extensions.

For example, the following result set indicates that you should update.

name	default_version	installed_version	comment
plls	2.1.0	1.5.3	PL/LiveScript (v8) trusted procedural language
plcoffee	2.1.0	1.5.3	PL/CoffeeScript (v8) trusted procedural language
plv8	2.1.0	1.5.3	PL/JavaScript (v8) trusted procedural language

(3 rows)

2. Take a snapshot of your instance as a precaution, because the upgrade drops all your plv8 functions. You can continue with the following steps while the snapshot is being created.

For steps to create a snapshot see, [Creating a DB snapshot \(p. 346\)](#)

3. Get a count of the number of plv8 functions in your DB instance so you can validate that they are all in place after the upgrade.

The following code returns the number of functions written in plv8, plcoffee, or plls.

```
select proname, nspname, lanname
from pg_proc p, pg_language l, pg_namespace n
where p.prolang = l.oid
and n.oid = p.pronamespace
and lanname in ('plv8','plcoffee','plls');
```

4. Use pg_dump to create a schema-only dump file.

The following code creates a file on your client machine in the /tmp directory.

```
./pg_dump -Fc --schema-only -U master postgres > /tmp/test.dmp
```

This example uses the following options:

- -FC "format custom"
- --schema-only "will only dump commands necessary to create schema (functions in our case)"
- -U "rds master username"
- database "the database name in our instance"

For more information on pg_dump, see the [pg_dump](#) page in the PostgreSQL documentation.

5. Extract the "CREATE FUNCTION" DDL statement that is present in the dump file.

The following code extracts the DDL statement needed to create the functions. You use this in subsequent steps to recreate the functions. The code uses the `grep` command to extract the statements to a file.

```
./pg_restore -l /tmp/test.dmp | grep FUNCTION > /tmp/function_list/
```

For more information on `pg_restore` see, [pg_restore](#).

6. Drop the functions and extensions.

The following code drops any plv8 based objects. The cascade option ensures that any dependent are dropped.

```
drop extension plv8 cascade;
```

If your PostgreSQL instance contains objects based on plcoffee or pll, repeat this step for those extensions.

7. Create the extensions.

The following code creates the plv8, plcoffee, and pll extensions.

```
create extension plv8;  
  
create extension plcoffee;  
  
create extension pll;
```

8. Create the functions using the dump file and "driver" file.

The following code recreates the functions that you extracted previously.

```
./pg_restore -U master -d postgres -Fc -L /tmp/function_list /tmp/test.dmp
```

9. Verify your functions count.

Validate that your functions have all been recreated by running the following code statement.

```
select * from pg_available_extensions where name in  
('plv8','pll','plcoffee');
```

Note

The plv8 version 2 adds the following extra row to your result set:

proname	nspname	lanname
plv8_version	pg_catalog	plv8

Logical replication for PostgreSQL on Amazon RDS

Beginning with PostgreSQL version 10.4, RDS supports the publication and subscription SQL Syntax for PostgreSQL 10 Logical Replication.

To enable logical replication for an Amazon RDS for PostgreSQL DB instance

1. The AWS user account requires the `rds_superuser` role to perform logical replication for the PostgreSQL database on Amazon RDS.
2. Set the `rds.logical_replication` static parameter to 1.

3. Modify the inbound rules of the security group for the publisher instance (production) to allow the subscriber instance (replica) to connect. This is usually done by including the IP address of the subscriber in the security group.
4. Restart the DB instance for the changes to the static parameter `rds.logical_replication` to take effect.

For more information on PostgreSQL logical replication, see the [PostgreSQL documentation](#).

Topics

- [Logical decoding and logical replication \(p. 1503\)](#)
- [Working with logical replication slots \(p. 1503\)](#)

Logical decoding and logical replication

RDS for PostgreSQL supports the streaming of WAL changes using logical replication slots. Amazon RDS supports logical decoding for a PostgreSQL DB instance version 9.5.4 and higher. You can set up logical replication slots on your instance and stream database changes through these slots to a client such as `pg_recvlogical`. Logical replication slots are created at the database level and support replication connections to a single database.

The most common clients for PostgreSQL logical replication are the AWS Database Migration Service or a custom-managed host on an Amazon EC2 instance. The logical replication slot knows nothing about the receiver of the stream, and there is no requirement that the target be a replica database. If you set up a logical replication slot and don't read from the slot, data can be written and quickly fill up your DB instance's storage.

PostgreSQL logical replication and logical decoding on Amazon RDS are enabled with a parameter, a replication connection type, and a security role. The client for logical decoding can be any client that is capable of establishing a replication connection to a database on a PostgreSQL DB instance.

To enable logical decoding for an Amazon RDS for PostgreSQL DB instance

1. The user account requires the `rds_superuser` role to enable logical replication. The user account also requires the `rds_replication` role to grant permissions to manage logical slots and to stream data using logical slots.
2. Set the `rds.logical_replication` static parameter to 1. As part of applying this parameter, we also set the parameters `wal_level`, `max_wal_senders`, `max_replication_slots`, and `max_connections`. These parameter changes can increase WAL generation, so you should only set the `rds.logical_replication` parameter when you are using logical slots.
3. Reboot the DB instance for the static `rds.logical_replication` parameter to take effect.
4. Create a logical replication slot as explained in the next section. This process requires that you specify a decoding plugin. Currently we support the `test_decoding` and `wal2json` output plugins that ship with PostgreSQL.

For more information on PostgreSQL logical decoding, see the [PostgreSQL documentation](#).

Working with logical replication slots

You can use SQL commands to work with logical slots. For example, the following command creates a logical slot named `test_slot` using the default PostgreSQL output plugin `test_decoding`.

```
SELECT * FROM pg_create_logical_replication_slot('test_slot', 'test_decoding');
slot_name      | xlog_position
-----+-----
```

```
regression_slot | 0/16B1970
(1 row)
```

To list logical slots, use the following command.

```
SELECT * FROM pg_replication_slots;
```

To drop a logical slot, use the following command.

```
SELECT pg_drop_replication_slot('test_slot');
pg_drop_replication_slot
-----
(1 row)
```

For more examples on working with logical replication slots, see [Logical decoding examples](#) in the PostgreSQL documentation.

After you create the logical replication slot, you can start streaming. The following example shows how logical decoding is controlled over the streaming replication protocol. This uses the program `pg_recvlogical`, which is included in the PostgreSQL distribution. This requires that client authentication is set up to allow replication connections.

```
pg_recvlogical -d postgres --slot test_slot -U master
--host sg-postgresql1.c6c8mresaghgv0.us-west-2.rds.amazonaws.com
-f - --start
```

To see the contents of the `pg_show_replication_origin_status` view, query the `pg_show_replication_origin_status()` function.

```
SELECT * FROM pg_show_replication_origin_status();
local_id | external_id | remote_lsn | local_lsn
-----+-----+-----+
(0 rows)
```

Event triggers for PostgreSQL on Amazon RDS

PostgreSQL versions 9.5.4 and later support event triggers, and Amazon RDS supports event triggers for these versions. The master user account can be used to create, modify, rename, and delete event triggers. Event triggers are at the DB instance level, so they can apply to all databases on an instance.

For example, the following code creates an event trigger that prints the current user at the end of every DDL command.

```
CREATE OR REPLACE FUNCTION raise_notice_func()
RETURNS event_trigger
LANGUAGE plpgsql AS
$$
BEGIN
    RAISE NOTICE 'In trigger function: %', current_user;
END;
$$;

CREATE EVENT TRIGGER event_trigger_1
ON ddl_command_end
EXECUTE PROCEDURE raise_notice_func();
```

For more information about PostgreSQL event triggers, see [Event triggers](#) in the PostgreSQL documentation.

There are several limitations to using PostgreSQL event triggers on Amazon RDS. These include:

- You cannot create event triggers on read replicas. You can, however, create event triggers on a read replica source. The event triggers are then copied to the read replica. The event triggers on the read replica don't fire on the read replica when changes are pushed from the source. However, if the read replica is promoted, the existing event triggers fire when database operations occur.
- To perform a major version upgrade to a PostgreSQL DB instance that uses event triggers, you must delete the event triggers before you upgrade the instance.

Huge pages for Amazon RDS for PostgreSQL

Amazon RDS for PostgreSQL supports multiple page sizes for PostgreSQL versions 9.5.6 and later, and 9.6.2 and later. This support includes 4 K and 2 MB page sizes.

Huge pages reduce overhead when using large contiguous chunks of memory. You allocate huge pages for your application by using calls to *mmap* or SYSV shared memory. You enable huge pages on an Amazon RDS for PostgreSQL database by using the `huge_pages` parameter. Set this parameter to "on" to enable huge pages.

For PostgreSQL versions 10 and above, huge pages are enabled for all instance classes. For PostgreSQL versions below 10, huge pages are enabled by default for `db.r4.*`, `db.m4.16xlarge`, and `db.m5.*` instance classes. For other instance classes, huge pages are disabled by default.

When you set the `huge_pages` parameter to "on," Amazon RDS uses huge pages based on the available shared memory. If the DB instance is unable to use huge pages due to shared memory constraints, Amazon RDS prevents the instance from starting and sets the status of the DB instance to an incompatible parameters state. In this case, you can set the `huge_pages` parameter to "off" to allow Amazon RDS to start the DB instance.

The `shared_buffers` parameter is key to setting the shared memory pool that is required for using huge pages. The default value for the `shared_buffers` parameter is set to a percentage of the total 8K pages available for that instance's memory. When you use huge pages, those pages are allocated in the huge pages collocated together. Amazon RDS puts a DB instance into an incompatible parameters state if the shared memory parameters are set to require more than 90 percent of the DB instance memory. For more information about setting shared memory for PostgreSQL, see the [PostgreSQL documentation](#).

Note

Huge pages are not supported for the `db.m1`, `db.m2`, and `db.m3` DB instance classes.

Tablespaces for PostgreSQL on Amazon RDS

PostgreSQL on Amazon RDS supports tablespaces for compatibility. Because all storage is on a single logical volume, you can't use tablespaces for IO splitting or isolation. Our benchmarks and experience indicate that a single logical volume is the best setup for most use cases.

If you specify a file name when you create a tablespace, the path prefix is `/rdsdbdata/db/base/` `tablespace`. The following example places tablespace files in `/rdsdbdata/db/base/tablespace/` `data`.

```
CREATE TABLESPACE act_data
  OWNER dbadmin
```

```
LOCATION '/data';
```

Autovacuum for PostgreSQL on Amazon RDS

The PostgreSQL autovacuum feature is turned on by default for new PostgreSQL DB instances. Autovacuum is optional, but we highly recommend that you do not turn autovacuum off. For more information on using autovacuum with Amazon RDS for PostgreSQL, see [Working with PostgreSQL autovacuum on Amazon RDS \(p. 1593\)](#).

RAM disk for the stats_temp_directory

The Amazon RDS for PostgreSQL parameter, `rds.pg_stat_ramdisk_size`, can be used to specify the system memory allocated to a RAM disk for storing the PostgreSQL `stats_temp_directory`. The RAM disk parameter is available for all PostgreSQL versions on Amazon RDS.

Under certain workloads, setting this parameter can improve performance and decrease IO requirements. For more information about the `stats_temp_directory`, see [the PostgreSQL documentation..](#)

To enable a RAM disk for your `stats_temp_directory`, set the `rds.pg_stat_ramdisk_size` parameter to a non-zero value in the parameter group used by your DB instance. The parameter value is in MB. You must reboot the DB instance before the change takes effect.

For example, the following AWS CLI command sets the RAM disk parameter to 256 MB.

```
aws rds modify-db-parameter-group \
--db-parameter-group-name pg-95-ramdisk-testing \
--parameters "ParameterName=rds.pg_stat_ramdisk_size, ParameterValue=256,
ApplyMethod=pending-reboot"
```

After you reboot, run the following command to see the status of the `stats_temp_directory`:

```
postgres=>show stats_temp_directory;
```

The command should return the following:

```
stats_temp_directory
-----
/rdsdbramdisk/pg_stat_tmp
(1 row)
```

ALTER ENUM for PostgreSQL

Amazon RDS for PostgreSQL versions 9.6.2 and 9.5.6 and later support the ability to alter enumerations. This feature is not available in other versions on Amazon RDS.

The following code shows an example of altering an enum value.

```
postgres=> CREATE TYPE rainbow AS ENUM ('red', 'orange', 'yellow', 'green', 'blue',
'purple');
```

```
CREATE TYPE
postgres=> CREATE TABLE t1 (colors rainbow);
CREATE TABLE
postgres=> INSERT INTO t1 VALUES ('red'), ('orange');
INSERT 0 2
postgres=> SELECT * from t1;
colors
-----
red
orange
(2 rows)
postgres=> ALTER TYPE rainbow RENAME VALUE 'red' TO 'crimson';
ALTER TYPE
postgres=> SELECT * from t1;
colors
-----
crimson
orange
(2 rows)
```

Connecting to a DB instance running the PostgreSQL database engine

After Amazon RDS provisions your DB instance, you can use any standard SQL client application to connect to the instance. To list the details of an Amazon RDS DB instance, you can use the AWS Management Console, the AWS CLI [describe-db-instances](#) command, or the Amazon RDS API [DescribeDBInstances](#) operation. You need the following information to connect:

- The host or host name for the DB instance, for example:

```
myinstance.123456789012.us-east-1.rds.amazonaws.com
```

- The port on which the DB instance is listening. For example, the default PostgreSQL port is 5432.
- The user name and password for the DB instance.

Following are two ways to connect to a PostgreSQL DB instance. The first example uses pgAdmin, a popular open-source administration and development tool for PostgreSQL. The second example uses psql, a command line utility that is part of a PostgreSQL installation.

Topics

- [Using pgAdmin to connect to a PostgreSQL DB instance \(p. 1508\)](#)
- [Using psql to connect to a PostgreSQL DB instance \(p. 1510\)](#)
- [Troubleshooting connections to your PostgreSQL instance \(p. 1511\)](#)

Using pgAdmin to connect to a PostgreSQL DB instance

You can use the open-source tool pgAdmin to connect to a PostgreSQL DB instance.

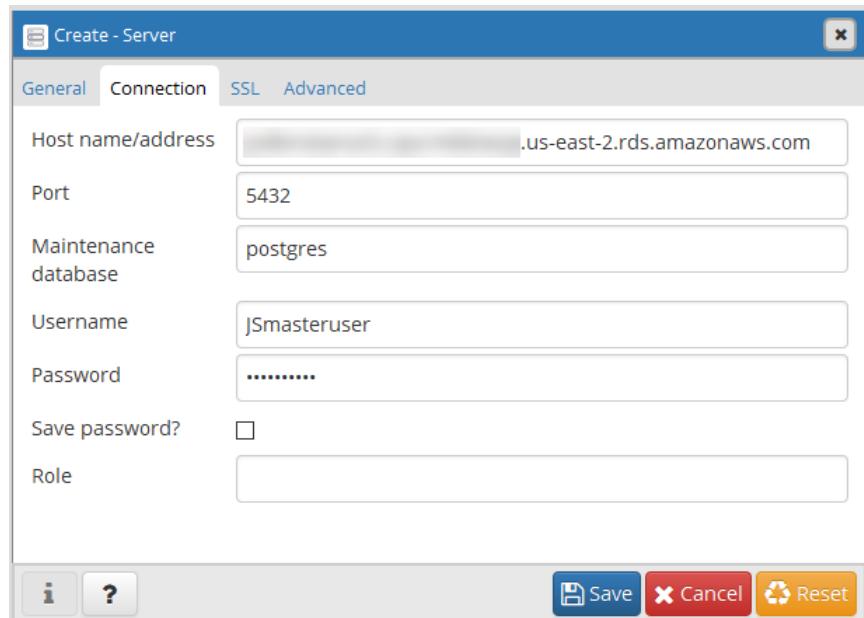
To connect to a PostgreSQL DB instance using pgAdmin

1. Find the endpoint (DNS name) and port number for your DB Instance.
 - a. Open the RDS console and then choose **Databases** to display a list of your DB instances.
 - b. Choose the PostgreSQL DB instance name to display its details.
 - c. On the **Connectivity & security** tab, copy the endpoint. Also, note the port number. You need both the endpoint and the port number to connect to the DB instance.

The screenshot shows the AWS RDS 'Summary' page for a database named 'database-1'. The 'Connectivity & security' tab is selected. The 'Endpoint' field is highlighted with a red oval.

DB identifier	database-1
Role	Instance
Connectivity & security	
Endpoint	database-1.us-west-1.rds.amazonaws.com
Port	5432

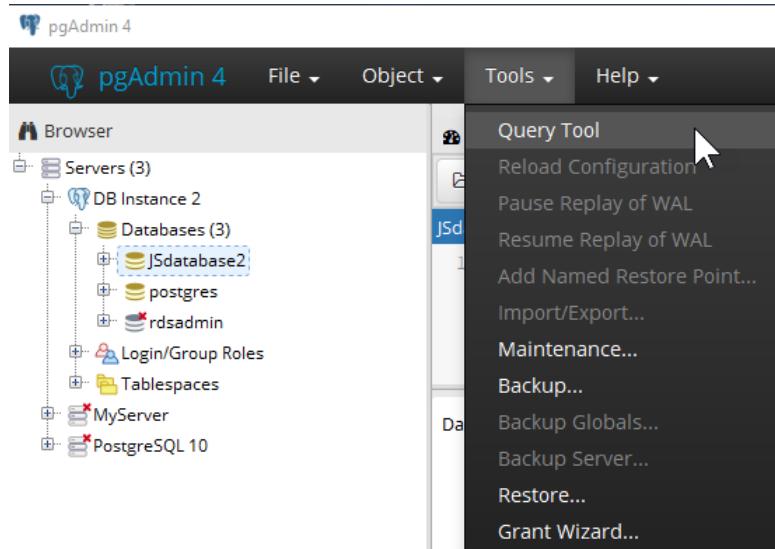
2. Install pgAdmin from <http://www.pgadmin.org/>. You can download and use pgAdmin without having a local instance of PostgreSQL on your client computer.
3. Launch the pgAdmin application on your client computer.
4. On the **Dashboard** tab, choose **Add New Server**.
5. In the **Create - Server** dialog box, type a name on the **General** tab to identify the server in pgAdmin.
6. On the **Connection** tab, type the following information from your DB instance:
 - For **Host**, type the endpoint, for example `mypostgresql.c6c8dntfzzhgv0.us-east-2.rds.amazonaws.com`.
 - For **Port**, type the assigned port.
 - For **Username**, type the user name that you entered when you created the DB instance.
 - For **Password**, type the password that you entered when you created the DB instance.



7. Choose **Save**.

If you have any problems connecting, see [Troubleshooting connections to your PostgreSQL instance \(p. 1511\)](#).

8. To access a database in the pgAdmin browser, expand **Servers**, the DB instance, and **Databases**. Choose the DB instance's database name.



9. To open a panel where you can enter SQL commands, choose **Tools**, **Query Tool**.

Using psql to connect to a PostgreSQL DB instance

You can use a local instance of the psql command line utility to connect to a PostgreSQL DB instance. You need either PostgreSQL or the psql client installed on your client computer. To connect to your PostgreSQL DB instance using psql, you need to provide host information and access credentials.

Use one of the following formats to connect to a PostgreSQL DB instance on Amazon RDS. When you connect, you're prompted for a password. For batch jobs or scripts, use the `--no-password` option. This option is set for the entire session.

Note

A connection attempt with `--no-password` fails when the server requires password authentication and a password is not available from other sources. For more information, see the [psql documentation](#).

If this is the first time you are connecting to this DB instance, try using the default database name `postgres` for the `--dbname` option.

For Unix, use the following format.

```
psql \
  --host=<DB instance endpoint> \
  --port=<port> \
  --username=<master username> \
  --password \
  --dbname=<database name>
```

For Windows, use the following format.

```
psql ^
  --host=<DB instance endpoint> ^
  --port=<port> ^
  --username=<master username> ^
  --password ^
  --dbname=<database name>
```

For example, the following command connects to a database called `mypgdb` on a PostgreSQL DB instance called `mypostgresql` using fictitious credentials.

```
psql --host=mypostgresql.c6c8mwvfdgv0.us-west-2.rds.amazonaws.com --port=5432 --
username=awsuser --password --dbname=mypgdb
```

Troubleshooting connections to your PostgreSQL instance

Topics

- [Error – FATAL: database `name` does not exist \(p. 1511\)](#)
- [Error – Could not connect to server: Connection timed out \(p. 1511\)](#)
- [Errors with security group access rules \(p. 1512\)](#)

Error – FATAL: database `name` does not exist

If when trying to connect you receive an error like `FATAL: database "name" does not exist`, try using the default database name `postgres` for the `--dbname` option.

Error – Could not connect to server: Connection timed out

If you can't connect to the DB instance, the most common error is `Could not connect to server: Connection timed out`. If you receive this error, check the following:

- Check that the host name used is the DB instance endpoint and that the port number used is correct.
- Make sure that the DB instance's public accessibility is set to **Yes** to allow external connections. To modify the **Public access** setting, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
- Check that the security group assigned to the DB instance has rules to allow access through any firewall your connection might go through. For example, if the DB instance was created using the default port of 5432, your company might have firewall rules blocking connections to that port from external company devices.

To fix this, modify the DB instance to use a different port. Also, make sure that the security group applied to the DB instance allows connections to the new port. To modify the **Database port** setting, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

- See also [Errors with security group access rules \(p. 1512\)](#).

Errors with security group access rules

By far the most common connection problem is with the security group's access rules assigned to the DB instance. If you used the default DB security group when you created the DB instance, the security group likely didn't have access rules that allow you to access the instance.

For the connection to work, the security group you assigned to the DB instance at its creation must allow access to the DB instance. For example, if the DB instance was created in a VPC, it must have a VPC security group that authorizes connections. Check if the DB instance was created using a security group that doesn't authorize connections from the device or Amazon EC2 instance where the application is running.

You can add or edit an inbound rule in the security group. For **Source**, choosing **My IP** allows access to the DB instance from the IP address detected in your browser. For more information, see [Provide access to your DB instance in your VPC by creating a security group \(p. 70\)](#).

Alternatively, if the DB instance was created outside of a VPC, it must have a database security group that authorizes those connections.

For more information about Amazon RDS security groups, see [Controlling access with security groups \(p. 1699\)](#).

Security with RDS for PostgreSQL

Security with RDS for PostgreSQL includes the following topics.

Topics

- [Using SSL with a PostgreSQL DB instance \(p. 1513\)](#)
- [Updating applications to connect to PostgreSQL DB instances using new SSL/TLS certificates \(p. 1516\)](#)
- [Using Kerberos authentication with Amazon RDS for PostgreSQL \(p. 1520\)](#)

Using SSL with a PostgreSQL DB instance

Amazon RDS supports Secure Socket Layer (SSL) encryption for PostgreSQL DB instances. Using SSL, you can encrypt a PostgreSQL connection between your applications and your PostgreSQL DB instances. You can also force all connections to your PostgreSQL DB instance to use SSL.

Amazon RDS for PostgreSQL supports Transport Layer Security (TLS) versions 1.1 and 1.2. Amazon RDS doesn't enforce TLS connections so they must be enforced from your application.

For general information about SSL support and PostgreSQL databases, see [SSL support](#) in the PostgreSQL documentation. For information about using an SSL connection over JDBC, see [Configuring the client](#) in the PostgreSQL documentation.

SSL support is available in all AWS Regions for PostgreSQL. Amazon RDS creates an SSL certificate for your PostgreSQL DB instance when the instance is created. If you enable SSL certificate verification, then the SSL certificate includes the DB instance endpoint as the Common Name (CN) for the SSL certificate to guard against spoofing attacks.

Topics

- [Connecting to a PostgreSQL DB instance over SSL \(p. 1513\)](#)
- [Requiring an SSL connection to a PostgreSQL DB instance \(p. 1514\)](#)
- [Determining the SSL connection status \(p. 1514\)](#)
- [SSL cipher suites in RDS for PostgreSQL \(p. 1515\)](#)

Connecting to a PostgreSQL DB instance over SSL

To connect to a PostgreSQL DB instance over SSL

1. Download the certificate.

For information about downloading certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

2. Import the certificate into your operating system.

For sample scripts that import certificates, see [Sample script for importing certificates into your trust store \(p. 1642\)](#).

3. Connect to your PostgreSQL DB instance over SSL.

When you connect using SSL, your client can choose whether to verify the certificate chain. If your connection parameters specify `sslmode=verify-ca` or `sslmode=verify-full`, then your client requires the RDS CA certificates to be in their trust store or referenced in the connection URL. This requirement is to verify the certificate chain that signs your database certificate.

When a client, such as psql or JDBC, is configured with SSL support, the client first tries to connect to the database with SSL by default. If the client can't connect with SSL, it reverts to connecting without SSL. The default `sslmode` mode used is different between libpq-based clients (such as psql) and JDBC. The libpq-based clients default to `prefer`, and JDBC clients default to `verify-full`.

Use the `sslrootcert` parameter to reference the certificate, for example `sslrootcert=rds-ssl-ca-cert.pem`.

The following is an example of using psql to connect to a PostgreSQL DB instance.

```
$ psql -h testpg.cdhmuqifdpib.us-east-1.rds.amazonaws.com -p 5432 \
  "dbname=testpg user=testuser sslrootcert=rds-ca-2019-root.pem sslmode=verify-full"
```

Requiring an SSL connection to a PostgreSQL DB instance

You can require that connections to your PostgreSQL DB instance use SSL by using the `rds.force_ssl` parameter. By default, the `rds.force_ssl` parameter is set to 0 (off). You can set the `rds.force_ssl` parameter to 1 (on) to require SSL for connections to your DB instance. Updating the `rds.force_ssl` parameter also sets the PostgreSQL `ssl` parameter to 1 (on) and modifies your DB instance's `pg_hba.conf` file to support the new SSL configuration.

You can set the `rds.force_ssl` parameter value by updating the parameter group for your DB instance. If the parameter group for your DB instance isn't the default one, and the `ssl` parameter is already set to 1 when you set `rds.force_ssl` to 1, you don't need to reboot your DB instance. Otherwise, you must reboot your DB instance for the change to take effect. For more information on parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

When the `rds.force_ssl` parameter is set to 1 for a DB instance, you see output similar to the following when you connect, indicating that SSL is now required:

```
$ psql postgres -h SOMEHOST.amazonaws.com -p 8192 -U someuser
. . .
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.

postgres=>
```

Determining the SSL connection status

The encrypted status of your connection is shown in the logon banner when you connect to the DB instance:

```
Password for user master:
psql (10.3)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.

postgres=>
```

You can also load the `sslinfo` extension and then call the `ssl_is_used()` function to determine if SSL is being used. The function returns `t` if the connection is using SSL, otherwise it returns `f`.

```
postgres=> create extension sslinfo;
CREATE EXTENSION

postgres=> select ssl_is_used();
 ssl_is_used
-----
 t
(1 row)
```

You can use the `select ssl_cipher()` command to determine the SSL cipher:

```
postgres=> select ssl_cipher();
ssl_cipher
-----
DHE-RSA-AES256-SHA
(1 row)
```

If you enable `set rds.force_ssl` and restart your instance, non-SSL connections are refused with the following message:

```
$ export PGSSLMODE=disable
$ psql postgres -h SOMEHOST.amazonaws.com -p 8192 -U someuser
psql: FATAL: no pg_hba.conf entry for host "host.ip", user "someuser", database "postgres",
      SSL off
$
```

For information about the `sslmode` option, see [Database connection control functions](#) in the PostgreSQL documentation.

SSL cipher suites in RDS for PostgreSQL

The PostgreSQL configuration parameter `ssl_ciphers` specifies the categories of cipher suites that are allowed for SSL connections. The following table lists the default cipher suites used in RDS for PostgreSQL.

PostgreSQL engine version	Cipher suites
13	HIGH:!aNULL:!3DES
12	HIGH:!aNULL:!3DES
11.4 and higher minor versions	HIGH:MEDIUM:+3DES:!aNULL:!RC4
11.1, 11.2	HIGH:MEDIUM:+3DES:!aNULL
10.9 and higher minor versions	HIGH:MEDIUM:+3DES:!aNULL:!RC4
10.7 and lower minor versions	HIGH:MEDIUM:+3DES:!aNULL
9.6.14 and higher minor versions	HIGH:MEDIUM:+3DES:!aNULL:!RC4
9.6.12 and lower minor versions	HIGH:MEDIUM:+3DES:!aNULL

Updating applications to connect to PostgreSQL DB instances using new SSL/TLS certificates

As of September 19, 2019, Amazon RDS has published new Certificate Authority (CA) certificates for connecting to your RDS DB instances using Secure Socket Layer or Transport Layer Security (SSL/TLS). Following, you can find information about updating your applications to use the new certificates.

This topic can help you to determine whether any client applications use SSL/TLS to connect to your DB instances. If they do, you can further check whether those applications require certificate verification to connect.

Note

Some applications are configured to connect to PostgreSQL DB instances only if they can successfully verify the certificate on the server.

For such applications, you must update your client application trust stores to include the new CA certificates.

After you update your CA certificates in the client application trust stores, you can rotate the certificates on your DB instances. We strongly recommend testing these procedures in a development or staging environment before implementing them in your production environments.

For more information about certificate rotation, see [Rotating your SSL/TLS certificate \(p. 1636\)](#). For more information about downloading certificates, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#). For information about using SSL/TLS with PostgreSQL DB instances, see [Using SSL with a PostgreSQL DB instance \(p. 1513\)](#).

Topics

- [Determining whether applications are connecting to PostgreSQL DB instances using SSL \(p. 1516\)](#)
- [Determining whether a client requires certificate verification in order to connect \(p. 1517\)](#)
- [Updating your application trust store \(p. 1517\)](#)
- [Using SSL/TLS connections for different types of applications \(p. 1518\)](#)

Determining whether applications are connecting to PostgreSQL DB instances using SSL

Check the DB instance configuration for the value of the `rds.force_ssl` parameter. By default, the `rds.force_ssl` parameter is set to 0 (off). If the `rds.force_ssl` parameter is set to 1 (on), clients are required to use SSL/TLS for connections. For more information about parameter groups, see [Working with DB parameter groups \(p. 228\)](#).

If you are using RDS PostgreSQL version 9.5 or later major version and `rds.force_ssl` is not set to 1 (on), query the `pg_stat_ssl` view to check connections using SSL. For example, the following query returns only SSL connections and information about the clients using SSL.

```
select datname, usename, ssl, client_addr from pg_stat_ssl inner join pg_stat_activity on pg_stat_ssl.pid = pg_stat_activity.pid where ssl is true and usename<>'rdsadmin';
```

Only rows using SSL/TLS connections are displayed with information about the connection. The following is sample output.

```
datname | username | ssl | client_addr
-----+-----+-----+
benchdb | pgadmin | t   | 53.95.6.13
postgres | pgadmin | t   | 53.95.6.13
(2 rows)
```

This query displays only the current connections at the time of the query. The absence of results doesn't indicate that no applications are using SSL connections. Other SSL connections might be established at a different time.

Determining whether a client requires certificate verification in order to connect

When a client, such as psql or JDBC, is configured with SSL support, the client first tries to connect to the database with SSL by default. If the client can't connect with SSL, it reverts to connecting without SSL. The default `sslmode` mode used is different between libpq-based clients (such as psql) and JDBC. The libpq-based clients default to `prefer`, where JDBC clients default to `verify-full`. The certificate on the server is verified only when `sslrootcert` is provided with `sslmode` set to `require`, `verify-ca`, or `verify-full`. An error is thrown if the certificate is invalid.

Use `PGSSLROOTCERT` to verify the certificate with the `PGSSLMODE` environment variable, with `PGSSLMODE` set to `require`, `verify-ca`, or `verify-full`.

```
PGSSLMODE=require PGSSLROOTCERT=/fullpath/rds-ca-2019-root.pem psql -h
pgdbidentifier.cxxxxxxxxx.us-east-2.rds.amazonaws.com -U masteruser -d postgres
```

Use the `sslrootcert` argument to verify the certificate with `sslmode` in connection string format, with `sslmode` set to `require`, `verify-ca`, or `verify-full` to verify the certificate.

```
psql "host=pgdbidentifier.cxxxxxxxxx.us-east-2.rds.amazonaws.com sslmode=require
sslrootcert=/full/path/rds-ca-2019-root.pem user=masteruser dbname=postgres"
```

For example, in the preceding case, if you are using an invalid root certificate, then you see an error similar to the following on your client.

```
psql: SSL error: certificate verify failed
```

Updating your application trust store

For information about updating the trust store for PostgreSQL applications, see [Secure TCP/IP connections with SSL](#) in the PostgreSQL documentation.

Note

When you update the trust store, you can retain older certificates in addition to adding the new certificates.

Updating your application trust store for JDBC

You can update the trust store for applications that use JDBC for SSL/TLS connections.

To update the trust store for JDBC applications

1. Download the 2019 root certificate that works for all AWS Regions and put the file in your trust store directory.

For information about downloading the root certificate, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

2. Convert the certificate to .der format using the following command.

```
openssl x509 -outform der -in rds-ca-2019-root.pem -out rds-ca-2019-root.der
```

Replace the file name with the one that you downloaded.

3. Import the certificate into the key store using the following command.

```
keytool -import -alias rds-root -keystore clientkeystore -file rds-ca-2019-root.der
```

4. Confirm that the key store was updated successfully.

```
keytool -list -v -keystore clientkeystore.jks
```

Enter the key store password when you are prompted for it.

Your output should contain the following:

```
rds-root, date, trustedCertEntry,  
Certificate fingerprint (SHA1):  
D4:0D:DB:29:E3:75:0D:FF:A6:71:C3:14:0B:BF:5F:47:8D:1C:80:96  
# This fingerprint should match the output from the below command  
openssl x509 -fingerprint -in rds-ca-2019-root.pem -noout
```

Using SSL/TLS connections for different types of applications

The following provides information about using SSL/TLS connections for different types of applications:

- **psql**

The client is invoked from the command line by specifying options either as a connection string or as environment variables. For SSL/TLS connections, the relevant options are `sslmode` (environment variable `PGSSLMODE`), `sslrootcert` (environment variable `PGSSLROOTCERT`).

For the complete list of options, see [Parameter key words](#) in the PostgreSQL documentation. For the complete list of environment variables, see [Environment variables](#) in the PostgreSQL documentation.

- **pgAdmin**

This browser-based client is a more user-friendly interface for connecting to a PostgreSQL database.

For information about configuring connections, see the [pgAdmin documentation](#).

- **JDBC**

JDBC enables database connections with Java applications.

For general information about connecting to a PostgreSQL database with JDBC, see [Connecting to the database](#) in the PostgreSQL documentation. For information about connecting with SSL/TLS, see [Configuring the client](#) in the PostgreSQL documentation.

- **Python**

A popular Python library for connecting to PostgreSQL databases is `psycopg2`.

For information about using `psycopg2`, see the [psycopg2 documentation](#). For a short tutorial on how to connect to a PostgreSQL database, see [Psycopg2 tutorial](#). You can find information about the options the `connect` command accepts in [The psycopg2 module content](#).

Important

After you have determined that your database connections use SSL/TLS and have updated your application trust store, you can update your database to use the rds-ca-2019 certificates. For instructions, see step 3 in [Updating your CA certificate by modifying your DB instance \(p. 1636\)](#).

Using Kerberos authentication with Amazon RDS for PostgreSQL

You can use Kerberos authentication to authenticate users when they connect to your DB instance running PostgreSQL. In this case, your DB instance works with AWS Directory Service for Microsoft Active Directory to enable Kerberos authentication. AWS Directory Service for Microsoft Active Directory is also called AWS Managed Microsoft AD.

You create an AWS Managed Microsoft AD directory to store user credentials. You then provide to your PostgreSQL DB instance the Active Directory's domain and other information. When users authenticate with the PostgreSQL DB instance, authentication requests are forwarded to the AWS Managed Microsoft AD directory.

Keeping all of your credentials in the same directory can save you time and effort. You have a centralized place for storing and managing credentials for multiple DB instances. Using a directory can also improve your overall security profile.

You can also access credentials from your own on-premises Microsoft Active Directory. To do so you create a trusting domain relationship so that the AWS Managed Microsoft AD directory trusts your on-premises Microsoft Active Directory. In this way, your users can access your PostgreSQL instances with the same Windows single sign-on (SSO) experience as when they access workloads in your on-premises network.

Topics

- [Availability of Kerberos authentication \(p. 1520\)](#)
- [Overview of Kerberos authentication for PostgreSQL DB instances \(p. 1521\)](#)
- [Setting up Kerberos authentication for PostgreSQL DB instances \(p. 1522\)](#)
- [Managing a DB instance in a Domain \(p. 1530\)](#)
- [Connecting to PostgreSQL with Kerberos authentication \(p. 1531\)](#)

Availability of Kerberos authentication

Amazon RDS supports Kerberos authentication for PostgreSQL DB instances in the following AWS Regions:

Region name	Region
US East (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1

Region name	Region
Canada (Central)	ca-central-1
China (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
South America (São Paulo)	sa-east-1
AWS GovCloud (US-East)	gov-us-east-1
AWS GovCloud (US-West)	gov-us-west-1

Overview of Kerberos authentication for PostgreSQL DB instances

To set up Kerberos authentication for a PostgreSQL DB instance, take the following steps, described in more detail later:

1. Use AWS Managed Microsoft AD to create an AWS Managed Microsoft AD directory. You can use the AWS Management Console, the AWS CLI, or the AWS Directory Service API to create the directory. Make sure to open the relevant outbound ports on the directory security group so that the directory can communicate with the instance.
2. Create a role that provides Amazon RDS access to make calls to your AWS Managed Microsoft AD directory. To do so, create an AWS Identity and Access Management (IAM) role that uses the managed IAM policy `AmazonRDSDirectoryServiceAccess`.

For the IAM role to allow access, the AWS Security Token Service (AWS STS) endpoint must be activated in the correct AWS Region for your AWS account. AWS STS endpoints are active by default in all AWS Regions, and you can use them without any further actions. For more information, see [Activating and deactivating AWS STS in an AWS Region](#) in the *IAM User Guide*.

3. Create and configure users in the AWS Managed Microsoft AD directory using the Microsoft Active Directory tools. For more information about creating users in your Active Directory, see [Manage users and groups in AWS Managed Microsoft AD](#) in the *AWS Directory Service Administration Guide*.
4. If you plan to locate the directory and the DB instance in different AWS accounts or virtual private clouds (VPCs), configure VPC peering. For more information, see [What is VPC peering?](#) in the *Amazon VPC Peering Guide*.
5. Create or modify a PostgreSQL DB instance either from the console, CLI, or RDS API using one of the following methods:
 - [Creating an Amazon RDS DB instance \(p. 141\)](#)
 - [Modifying an Amazon RDS DB instance \(p. 250\)](#)
 - [Restoring from a DB snapshot \(p. 349\)](#)
 - [Restoring a DB instance to a specified time \(p. 389\)](#)

You can locate the instance in the same Amazon Virtual Private Cloud (VPC) as the directory or in a different AWS account or VPC. When you create or modify the PostgreSQL DB instance, do the following:

- Provide the domain identifier (d-* identifier) that was generated when you created your directory.
 - Provide the name of the IAM role that you created.
 - Ensure that the DB instance security group can receive inbound traffic from the directory security group.
6. Use the RDS master user credentials to connect to the PostgreSQL DB instance. Create the user in PostgreSQL to be identified externally. Externally identified users can log in to the PostgreSQL DB instance using Kerberos authentication.

Setting up Kerberos authentication for PostgreSQL DB instances

You use AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) to set up Kerberos authentication for a PostgreSQL DB instance. To set up Kerberos authentication, take the following steps.

Topics

- [Step 1: Create a directory using AWS Managed Microsoft AD \(p. 1522\)](#)
- [Step 2: \(Optional\) create a trust for an on-premises Active Directory \(p. 1525\)](#)
- [Step 3: Create an IAM role for Amazon RDS to access the AWS Directory Service \(p. 1526\)](#)
- [Step 4: Create and configure users \(p. 1527\)](#)
- [Step 5: Enable cross-VPC traffic between the directory and the DB instance \(p. 1527\)](#)
- [Step 6: Create or modify a PostgreSQL DB instance \(p. 1528\)](#)
- [Step 7: Create Kerberos authentication PostgreSQL logins \(p. 1529\)](#)
- [Step 8: Configure a PostgreSQL client \(p. 1529\)](#)

Step 1: Create a directory using AWS Managed Microsoft AD

AWS Directory Service creates a fully managed Active Directory in the AWS Cloud. When you create an AWS Managed Microsoft AD directory, AWS Directory Service creates two domain controllers and DNS servers for you. The directory servers are created in different subnets in a VPC. This redundancy helps make sure that your directory remains accessible even if a failure occurs.

When you create an AWS Managed Microsoft AD directory, AWS Directory Service performs the following tasks on your behalf:

- Sets up an Active Directory within your VPC.
- Creates a directory administrator account with the user name `Admin` and the specified password. You use this account to manage your directory.

Important

Make sure to save this password. AWS Directory Service doesn't store this password, and it can't be retrieved or reset.

- Creates a security group for the directory controllers. The security group must permit communication with the PostgreSQL DB instance.

When you launch AWS Directory Service for Microsoft Active Directory, AWS creates an Organizational Unit (OU) that contains all of your directory's objects. This OU, which has the NetBIOS name that you entered when you created your directory, is located in the domain root. The domain root is owned and managed by AWS.

The **Admin** account that was created with your AWS Managed Microsoft AD directory has permissions for the most common administrative activities for your OU:

- Create, update, or delete users
- Add resources to your domain such as file or print servers, and then assign permissions for those resources to users in your OU
- Create additional OUs and containers
- Delegate authority
- Restore deleted objects from the Active Directory Recycle Bin
- Run Active Directory and Domain Name Service (DNS) modules for Windows PowerShell on the Active Directory Web Service

The **Admin** account also has rights to perform the following domain-wide activities:

- Manage DNS configurations (add, remove, or update records, zones, and forwarders)
- View DNS event logs
- View security event logs

To create a directory with AWS Managed Microsoft AD

1. In the [AWS Directory Service console](#) navigation pane, choose **Directories**, and then choose **Set up directory**.
2. Choose **AWS Managed Microsoft AD**. AWS Managed Microsoft AD is the only option currently supported for use with Amazon RDS.
3. Choose **Next**.
4. On the **Enter directory information** page, provide the following information:

Edition

Choose the edition that meets your requirements.

Directory DNS name

The fully qualified name for the directory, such as **corp.example.com**.

Directory NetBIOS name

An optional short name for the directory, such as **CORP**.

Directory description

An optional description for the directory.

Admin password

The password for the directory administrator. The directory creation process creates an administrator account with the user name **Admin** and this password.

The directory administrator password can't include the word "admin." The password is case-sensitive and must be 8–64 characters in length. It must also contain at least one character from three of the following four categories:

- Lowercase letters (a–z)
- Uppercase letters (A–Z)
- Numbers (0–9)
- Nonalphanumeric characters (~!@#\$%^&*_+=`|\{}{};:"'<>,.?/)

Confirm password

Retype the administrator password.

Important

Make sure that you save this password. AWS Directory Service doesn't store this password, and it can't be retrieved or reset.

5. Choose **Next**.
6. On the **Choose VPC and subnets** page, provide the following information:

VPC

Choose the VPC for the directory. You can create the PostgreSQL DB instance in this same VPC or in a different VPC.

Subnets

Choose the subnets for the directory servers. The two subnets must be in different Availability Zones.

7. Choose **Next**.
8. Review the directory information. If changes are needed, choose **Previous** and make the changes. When the information is correct, choose **Create directory**.

Review & create

Review

Directory type	VPC
Microsoft AD	vpc-8b6b78e9 ([REDACTED])
Directory DNS name	Subnets
corp.example.com	subnet-75128d10 ([REDACTED], us-east-1a) subnet-f51665dd ([REDACTED], us-east-1b)
Directory NetBIOS name	
CORP	
Directory description	
My directory	

Pricing

Edition	Free trial eligible Learn more
Standard	30-day limited trial

~USD [REDACTED] *

* Includes two domain controllers, USD [REDACTED] /mo for each additional domain controller.

Cancel

Previous

Create directory

It takes several minutes for the directory to be created. When it has been successfully created, the **Status** value changes to **Active**.

To see information about your directory, choose the directory ID in the directory listing. Make a note of the **Directory ID** value. You need this value when you create or modify your PostgreSQL DB instance.

The screenshot shows the AWS Directory Service interface. At the top, there's a breadcrumb navigation: Directory Service > Directories > d-90670a8d36. Below this, a modal window displays 'Directory details' for the selected directory. The modal has two buttons at the top right: 'Reset user password' and a close button. The directory details are listed in a table:

Directory type	VPC	Status
Microsoft AD	vpc-6594f31c [edit]	<input checked="" type="checkbox"/> Active
Edition	Subnets	Last updated
Standard	subnet-7d36a227 [edit] subnet-a2ab49c6 [edit]	Tuesday, January 7, 2020
Directory ID	Availability zones	Launch time
d-90670a8d36	us-east-1c, us-east-1d	Tuesday, January 7, 2020
Directory DNS name	DNS address	
corp.example.com	[redacted]	
Directory NetBIOS name		
CORP		
Description - Edit		
My directory		

At the bottom of the modal, there are four tabs: Application management (which is active), Scale & share, Networking & security, and Maintenance.

Step 2: (Optional) create a trust for an on-premises Active Directory

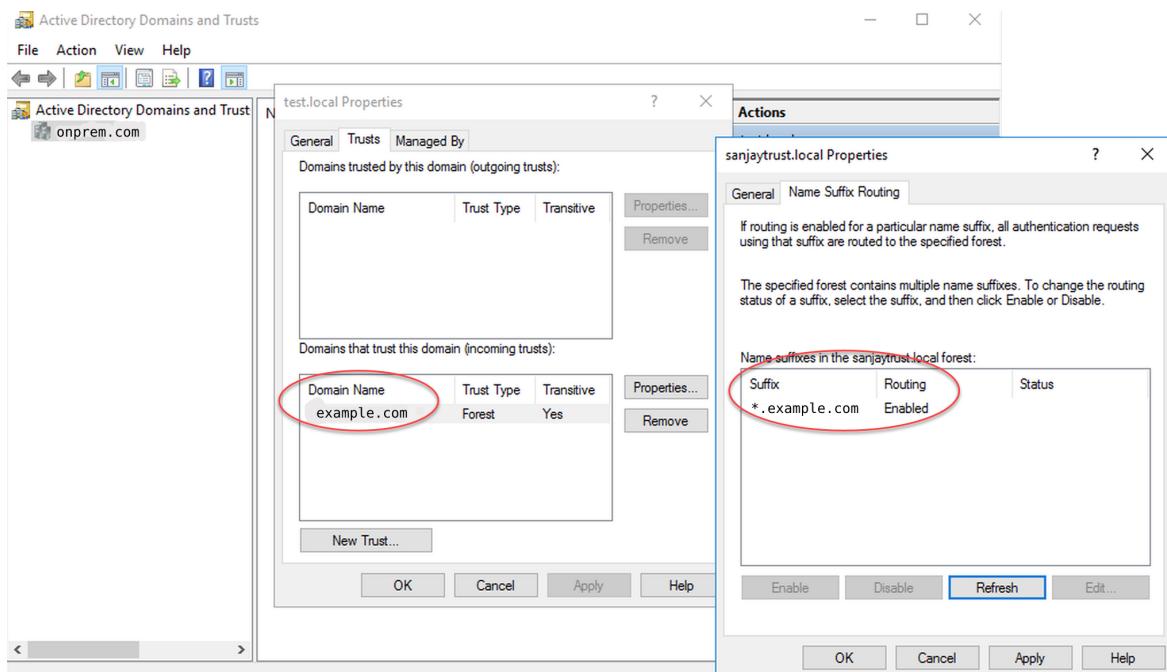
If you don't plan to use your own on-premises Microsoft Active Directory, skip to [Step 3: Create an IAM role for Amazon RDS to access the AWS Directory Service \(p. 1526\)](#).

To get Kerberos authentication using your on-premises Active Directory, you need to create a trusting domain relationship using a forest trust between your on-premises Microsoft Active Directory and the AWS Managed Microsoft AD directory (created in [Step 1: Create a directory using AWS Managed Microsoft AD \(p. 1522\)](#)). The trust can be one-way, where the AWS Managed Microsoft AD directory trusts the on-premises Microsoft Active Directory. The trust can also be two-way, where both Active Directories trust each other. For more information about setting up trusts using AWS Directory Service, see [When to create a trust relationship in the AWS Directory Service Administration Guide](#).

Note

If you use an on-premises Microsoft Active Directory, DB instance endpoints can't be used by Windows clients.

Make sure that your on-premises Microsoft Active Directory domain name includes a DNS suffix routing that corresponds to the newly created trust relationship. The following screenshot shows an example.



Step 3: Create an IAM role for Amazon RDS to access the AWS Directory Service

For Amazon RDS to call AWS Directory Service for you, an IAM role that uses the managed IAM policy `AmazonRDSDirectoryServiceAccess` is required. This role allows Amazon RDS to make calls to AWS Directory Service.

When a DB instance is created using the AWS Management Console and the console user has the `iam:CreateRole` permission, the console creates this role automatically. In this case, the role name is `rds-directoryservice-kerberos-access-role`. Otherwise, create the IAM role manually. Choose **RDS** and then **RDS - Directory Service**. Attach the AWS managed policy `AmazonRDSDirectoryServiceAccess` to this role.

For more information about creating IAM roles for a service, see [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.

Note

The IAM role used for Windows Authentication for RDS for Microsoft SQL Server can't be used for Amazon RDS for PostgreSQL.

Optionally, you can create policies with the required permissions instead of using the managed IAM policy `AmazonRDSDirectoryServiceAccess`. In this case, the IAM role must have the following IAM trust policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
    ]
}
```

The role must also have the following IAM role policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Step 4: Create and configure users

You can create users by using the Active Directory Users and Computers tool. This is one of the Active Directory Domain Services and Active Directory Lightweight Directory Services tools. In this case, *users* are individual people or entities who have access to your directory.

To create users in an AWS Directory Service directory, you must be connected to a Windows-based Amazon EC2 instance. Also, this EC2 instance must be a member of the AWS Directory Service directory. At the same time, you must be logged in as a user that has privileges to create users. For more information, see [Create a user in the AWS Directory Service Administration Guide](#).

Step 5: Enable cross-VPC traffic between the directory and the DB instance

If you plan to locate the directory and the DB instance in the same VPC, skip this step and move on to [Step 6: Create or modify a PostgreSQL DB instance \(p. 1528\)](#).

If you plan to locate the directory and the DB instance in different VPCs, configure cross-VPC traffic using VPC peering or [AWS Transit Gateway](#).

The following procedure enables traffic between VPCs using VPC peering. Follow the instructions in [What is VPC peering?](#) in the *Amazon Virtual Private Cloud Peering Guide*.

To enable cross-VPC traffic using VPC peering

1. Set up appropriate VPC routing rules to ensure that network traffic can flow both ways.
2. Ensure that the DB instance security group can receive inbound traffic from the directory security group.
3. Ensure that there is no network access control list (ACL) rule to block traffic.

If a different AWS account owns the directory, you must share the directory.

To share the directory between AWS accounts

1. Start sharing the directory with the AWS account that the DB instance will be created in by following the instructions in [Tutorial: Sharing your AWS Managed Microsoft AD directory for seamless EC2 Domain-join](#) in the *AWS Directory Service Administration Guide*.

2. Sign in to the AWS Directory Service console using the account for the DB instance, and ensure that the domain has the **SHARED** status before proceeding.
3. While signed into the AWS Directory Service console using the account for the DB instance, note the **Directory ID** value. You use this directory ID to join the DB instance to the domain.

Step 6: Create or modify a PostgreSQL DB instance

Create or modify a PostgreSQL DB instance for use with your directory. You can use the console, CLI, or RDS API to associate a DB instance with a directory. You can do this in one of the following ways:

- Create a new PostgreSQL DB instance using the console, the [create-db-instance](#) CLI command, or the [CreateDBInstance](#) RDS API operation. For instructions, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- Modify an existing PostgreSQL DB instance using the console, the [modify-db-instance](#) CLI command, or the [ModifyDBInstance](#) RDS API operation. For instructions, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
- Restore a PostgreSQL DB instance from a DB snapshot using the console, the [restore-db-instance-from-db-snapshot](#) CLI command, or the [RestoreDBInstanceFromDBSnapshot](#) RDS API operation. For instructions, see [Restoring from a DB snapshot \(p. 349\)](#).
- Restore a PostgreSQL DB instance to a point-in-time using the console, the [restore-db-instance-to-point-in-time](#) CLI command, or the [RestoreDBInstanceToPointInTime](#) RDS API operation. For instructions, see [Restoring a DB instance to a specified time \(p. 389\)](#).

Kerberos authentication is only supported for PostgreSQL DB instances in a VPC. The DB instance can be in the same VPC as the directory, or in a different VPC. The DB instance must use a security group that allows ingress and egress within the directory's VPC so the DB instance can communicate with the directory.

Console

When you use the console to create, modify, or restore a DB instance, choose **Password and Kerberos authentication** in the **Database authentication** section. Then choose **Browse Directory**. Select the directory or choose **Create a new directory** to use the Directory Service.

AWS CLI

When you use the AWS CLI, the following parameters are required for the DB instance to be able to use the directory that you created:

- For the `--domain` parameter, use the domain identifier ("d-*" identifier) generated when you created the directory.
- For the `--domain-iam-role-name` parameter, use the role you created that uses the managed IAM policy `AmazonRDSDirectoryServiceAccess`.

For example, the following CLI command modifies a DB instance to use a directory.

```
aws rds modify-db-instance --db-instance-identifier mydbinstance --domain d-Directory-ID --domain-iam-role-name role-name
```

Important

If you modify a DB instance to enable Kerberos authentication, reboot the DB instance after making the change.

Step 7: Create Kerberos authentication PostgreSQL logins

Use the RDS master user credentials to connect to the PostgreSQL DB instance as you do with any other DB instance. The DB instance is joined to the AWS Managed Microsoft AD domain. Thus, you can provision PostgreSQL logins and users from the Microsoft Active Directory users and groups in your domain. To manage database permissions, you grant and revoke standard PostgreSQL permissions to these logins.

To allow an Active Directory user to authenticate with PostgreSQL, use the RDS master user credentials. You use these credentials to connect to the PostgreSQL DB instance as you do with any other DB instance. After you're logged in, create an externally authenticated user in PostgreSQL and grant the `rds_ad` role to this user.

```
CREATE USER "username@CORP.EXAMPLE.COM" WITH LOGIN;  
GRANT rds_ad TO "username@CORP.EXAMPLE.COM";
```

Replace `username` with the user name and include the domain name in uppercase. Users (both humans and applications) from your domain can now connect to the RDS PostgreSQL instance from a domain-joined client machine using Kerberos authentication.

Note that a database user can use either Kerberos or IAM authentication but not both, so this user can't also have the `rds_iam` role. This also applies to nested memberships. For more information, see [IAM database authentication for MySQL and PostgreSQL \(p. 1660\)](#).

Step 8: Configure a PostgreSQL client

To configure a PostgreSQL client, take the following steps:

- Create a `krb5.conf` file (or equivalent) to point to the domain.
- Verify that traffic can flow between the client host and AWS Directory Service. Use a network utility such as Netcat for the following:
 - Verify traffic over DNS for port 53.
 - Verify traffic over TCP/UDP for port 53 and for Kerberos, which includes ports 88 and 464 for AWS Directory Service.
- Verify that traffic can flow between the client host and the DB instance over the database port. For example, use `psql` to connect and access the database.

The following is sample `krb5.conf` content for AWS Managed Microsoft AD.

```
[libdefaults]  
default_realm = EXAMPLE.COM  
[realms]  
EXAMPLE.COM = {  
    kdc = example.com  
    admin_server = example.com  
}  
[domain_realm]  
.example.com = EXAMPLE.COM  
example.com = EXAMPLE.COM
```

The following is sample `krb5.conf` content for an on-premises Microsoft Active Directory.

```
[libdefaults]  
default_realm = EXAMPLE.COM  
[realms]  
EXAMPLE.COM = {
```

```
kdc = example.com
admin_server = example.com
}
ONPREM.COM = {
    kdc = onprem.com
    admin_server = onprem.com
}
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
.onprem.com = ONPREM.COM
onprem.com = ONPREM.COM
.rds.amazonaws.com = EXAMPLE.COM
.amazonaws.com.cn = EXAMPLE.COM
.amazon.com = EXAMPLE.COM
```

Managing a DB instance in a Domain

You can use the console, the CLI, or the RDS API to manage your DB instance and its relationship with your Microsoft Active Directory. For example, you can associate an Active Directory to enable Kerberos authentication. You can also remove the association for an Active Directory to disable Kerberos authentication. You can also move a DB instance to be externally authenticated by one Microsoft Active Directory to another.

For example, using the CLI, you can do the following:

- To reattempt enabling Kerberos authentication for a failed membership, use the [modify-db-instance](#) CLI command. Specify the current membership's directory ID for the --domain option.
- To disable Kerberos authentication on a DB instance, use the [modify-db-instance](#) CLI command. Specify none for the --domain option.
- To move a DB instance from one domain to another, use the [modify-db-instance](#) CLI command. Specify the domain identifier of the new domain for the --domain option.

Understanding Domain membership

After you create or modify your DB instance, it becomes a member of the domain. You can view the status of the domain membership in the console or by running the [describe-db-instances](#) CLI command. The status of the DB instance can be one of the following:

- **kerberos-enabled** – The DB instance has Kerberos authentication enabled.
- **enabling-kerberos** – AWS is in the process of enabling Kerberos authentication on this DB instance.
- **pending-enable-kerberos** – Enabling Kerberos authentication is pending on this DB instance.
- **pending-maintenance-enable-kerberos** – AWS will attempt to enable Kerberos authentication on the DB instance during the next scheduled maintenance window.
- **pending-disable-kerberos** – Disabling Kerberos authentication is pending on this DB instance.
- **pending-maintenance-disable-kerberos** – AWS will attempt to disable Kerberos authentication on the DB instance during the next scheduled maintenance window.
- **enable-kerberos-failed** – A configuration problem prevented AWS from enabling Kerberos authentication on the DB instance. Correct the configuration problem before reissuing the command to modify the DB instance.
- **disabling-kerberos** – AWS is in the process of disabling Kerberos authentication on this DB instance.

A request to enable Kerberos authentication can fail because of a network connectivity issue or an incorrect IAM role. In some cases, the attempt to enable Kerberos authentication might fail when you

create or modify a DB instance. If so, make sure that you are using the correct IAM role, then modify the DB instance to join the domain.

Note

Only Kerberos authentication with RDS for PostgreSQL sends traffic to the domain's DNS servers. All other DNS requests are treated as outbound network access on your DB instances running PostgreSQL. For more information about outbound network access with RDS for PostgreSQL, see [Using a custom DNS server for outbound network access \(p. 1605\)](#).

Connecting to PostgreSQL with Kerberos authentication

You can connect to PostgreSQL with Kerberos authentication with the pgAdmin interface or with a command line interface such as psql. For more information about connecting, see [Connecting to a DB instance running the PostgreSQL database engine \(p. 1508\)](#).

pgAdmin

To use pgAdmin to connect to PostgreSQL with Kerberos authentication, take the following steps:

1. Launch the pgAdmin application on your client computer.
2. On the **Dashboard** tab, choose **Add New Server**.
3. In the **Create - Server** dialog box, enter a name on the **General** tab to identify the server in pgAdmin.
4. On the **Connection** tab, enter the following information from your RDS for PostgreSQL database:
 - For **Host**, enter the endpoint. Use a format such as *PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com*.

If you're using an on-premises Microsoft Active Directory from a Windows client, then you need to connect using a specialized endpoint. Instead of using the Amazon domain `rds.amazonaws.com` in the host endpoint, use the domain name of the AWS Managed Active Directory.

For example, suppose that the domain name for the AWS Managed Active Directory is `corp.example.com`. Then for **Host**, use the format *PostgreSQL-endpoint.AWS-Region.corp.example.com*.

- For **Port**, enter the assigned port.
- For **Maintenance database**, enter the name of the initial database to which the client will connect.
- For **Username**, enter the user name that you entered for Kerberos authentication in [Step 7: Create Kerberos authentication PostgreSQL logins \(p. 1529\)](#).

5. Choose **Save**.

PsSql

To use psql to connect to PostgreSQL with Kerberos authentication, take the following steps:

1. At a command prompt, run the following command.

```
kinit username
```

Replace `username` with the user name. At the prompt, enter the password stored in the Microsoft Active Directory for the user.

2. If the PostgreSQL DB instance is using a publicly accessible VPC, put a private IP address for your DB instance endpoint in your /etc/hosts file on the EC2 client. For example, the following commands obtain the private IP address and then put it in the /etc/hosts file.

```
% dig +short PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com
;; Truncated, retrying in TCP mode.
```

```
ec2-34-210-197-118.AWS-Region.compute.amazonaws.com.  
34.210.197.118  
% echo " 34.210.197.118 PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com" >> /etc/hosts
```

If you're using an on-premises Microsoft Active Directory from a Windows client, then you need to connect using a specialized endpoint. Instead of using the Amazon domain `rds.amazonaws.com` in the host endpoint, use the domain name of the AWS Managed Active Directory.

For example, suppose that the domain name for your AWS Managed Active Directory is `corp.example.com`. Then use the format `PostgreSQL-endpoint.AWS-Region.corp.example.com` for the endpoint and put it in the `/etc/hosts` file.

```
% echo " 34.210.197.118 PostgreSQL-endpoint.AWS-Region.corp.example.com" >> /etc/hosts
```

3. Use the following `psql` command to log in to a PostgreSQL DB instance that is integrated with Active Directory.

```
psql -U username@CORP.EXAMPLE.COM -p 5432 -h PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com postgres
```

To log in to the PostgreSQL DB cluster from a Windows client using an on-premises Active Directory, use the following `psql` command with the domain name from the previous step (`corp.example.com`):

```
psql -U username@CORP.EXAMPLE.COM -p 5432 -h PostgreSQL-endpoint.AWS-Region.corp.example.com postgres
```

Upgrading the PostgreSQL DB engine for Amazon RDS

There are two types of upgrades you can manage for your PostgreSQL DB instance:

- Operating system updates – Occasionally, Amazon RDS might need to update the underlying operating system of your DB instance to apply security fixes or OS changes. You can decide when Amazon RDS applies OS updates by using the RDS console, AWS Command Line Interface (AWS CLI), or RDS API. For more information about OS updates, see [Applying updates for a DB instance \(p. 266\)](#).
- Database engine upgrades – When Amazon RDS supports a new version of a database engine, you can upgrade your DB instances to the new version.

When Amazon RDS supports a new version of a database engine, you can upgrade your DB instances to the new version. There are two kinds of upgrades for PostgreSQL DB instances: major version upgrades and minor version upgrades.

Major version upgrades

Major version upgrades can contain database changes that are not backward-compatible with existing applications. As a result, you must manually perform major version upgrades of your DB instances. You can initiate a major version upgrade by modifying your DB instance. However, before you perform a major version upgrade, we recommend that you follow the steps described in [Choosing a major version upgrade for PostgreSQL \(p. 1534\)](#). During a major version upgrade, Amazon RDS also upgrades all of your in-Region read replicas along with the primary DB instance.

Minor version upgrades

In contrast, *minor version upgrades* include only changes that are backward-compatible with existing applications. You can initiate a minor version upgrade manually by modifying your DB instance. Or you can enable the **Auto minor version upgrade** option when creating or modifying a DB instance. Doing so means that your DB instance is automatically upgraded after Amazon RDS tests and approves the new version. If your PostgreSQL DB instance is using read replicas, you must upgrade all of the read replicas before the minor version upgrade of the source instance. For more details, see [Automatic minor version upgrades for PostgreSQL \(p. 1540\)](#). For information about manually performing a minor version upgrade, see [Manually upgrading the engine version \(p. 271\)](#).

Topics

- [Overview of upgrading PostgreSQL \(p. 1533\)](#)
- [PostgreSQL version numbers \(p. 1534\)](#)
- [Choosing a major version upgrade for PostgreSQL \(p. 1534\)](#)
- [How to perform a major version upgrade \(p. 1536\)](#)
- [Automatic minor version upgrades for PostgreSQL \(p. 1540\)](#)
- [Upgrading PostgreSQL extensions \(p. 1540\)](#)

Overview of upgrading PostgreSQL

To safely upgrade your DB instances, Amazon RDS uses the `pg_upgrade` utility described in the [PostgreSQL documentation](#).

Amazon RDS takes two DB snapshots during the upgrade process if your backup retention period is greater than 0. The first DB snapshot is of the DB instance before any upgrade changes have been made.

If the upgrade doesn't work for your databases, you can restore this snapshot to create a DB instance running the old version. The second DB snapshot is taken after the upgrade completes.

Note

Amazon RDS takes DB snapshots during the upgrade process only if you have set the backup retention period for your DB instance to a number greater than 0. To change your backup retention period, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

When you upgrade the primary DB instance, all the in-Region read replicas are also automatically upgraded. After the upgrade workflow starts, the replica instances wait for the `pg_upgrade` to complete successfully on the primary DB instance. Then the primary instance upgrade waits for the replica instance upgrades to complete. You experience an outage until the upgrade is complete. If you have any older replicas from earlier major versions they aren't upgraded.

If your DB instance is in a Multi-AZ deployment, both the primary writer DB instance and standby DB instances are upgraded. The writer and standby DB instances are upgraded at the same time.

After an upgrade is complete, you can't revert to the previous version of the database engine. If you want to return to the previous version, restore the DB snapshot that was taken before the upgrade to create a new DB instance.

PostgreSQL version numbers

The version numbering sequence for the PostgreSQL database engine is as follows:

- For PostgreSQL versions 10 and later, the engine version number is in the form *major.minor*. The major version number is the integer part of the version number. The minor version number is the fractional part of the version number.

A major version upgrade increases the integer part of the version number, such as upgrading from *10.minor* to *11.minor*.

- For PostgreSQL versions earlier than 10, the engine version number is in the form *major.major.minor*. The major engine version number is both the integer and the first fractional part of the version number. For example, 9.6 is a major version. The minor version number is the third part of the version number. For example, for version 9.6.12, the 12 is the minor version number.

A major version upgrade increases the major part of the version number. For example, an upgrade from 9.6.12 to 10.11 is a major version upgrade, where 9.6 and 10 are the major version numbers.

Choosing a major version upgrade for PostgreSQL

Major version upgrades can contain database changes that are not backward-compatible with previous versions of the database. This functionality can cause your existing applications to stop working correctly.

As a result, Amazon RDS doesn't apply major version upgrades automatically. To perform a major version upgrade, you modify your DB instance manually. Make sure that you thoroughly test any upgrade to verify that your applications work correctly before applying the upgrade to your production DB instances. When you do a PostgreSQL major version upgrade, we recommend that you follow the steps described in [How to perform a major version upgrade \(p. 1536\)](#).

You can upgrade a PostgreSQL database to its next major version. From some PostgreSQL database versions, you can skip to a higher major version when upgrading. If your upgrade skips a major version, the read replicas are also upgraded to that target major version. The following table lists the source PostgreSQL database versions and the associated target major versions available for upgrading.

Note

Upgrade targets are enabled to a higher version released at the same time as the source minor version or later.

If a database uses the `PostGIS` extension, you can't skip major versions for some source to target combinations. For these circumstances, upgrade to a recent minor version, then upgrade to PostgreSQL 12, and finally upgrade to your desired target version.

The `pgRouting` extension isn't supported for an upgrade that skips a major version to versions 11.x. A major version is skipped when the upgrade goes from versions 9.4.x, 9.5.x, or 9.6.x to versions 11.x. You can drop the `pgRouting` extension and then add it again after an upgrade. The `tsearch2` and `chkpass` extensions aren't supported in PostgreSQL 11 or later. If you are upgrading to version 11.x, drop these extensions before the upgrade.

Current source version	Newest upgrade target	Preferred major upgrade targets				
12.6, 12.5, 12.4, 12.3, 12.2	13.2 (p. 1461)	13.2 (p. 1461)				
11.11	13.2 (p. 1461)	13.2 (p. 1461)	12.6 (p. 1462)			
11.10	13.1 (p. 1461)	13.1 (p. 1461)	12.6 (p. 1462)	11.11 (p. 1463)		
11.9, 11.8, 11.7, 11.6, 11.5, 11.4, 11.2, 11.1	12.6 (p. 1462)		12.6 (p. 1462)	11.11 (p. 1463)		
10.16	13.2 (p. 1461)	13.2 (p. 1461)	12.6 (p. 1462)	11.11 (p. 1463)		
10.15	13.1 (p. 1461)	13.1 (p. 1461)	12.5 (p. 1462)	11.11 (p. 1463)		
10.14	12.4 (p. 1462)		12.4 (p. 1462)	11.11 (p. 1463)		
10.13	12.3 (p. 1463)		12.3 (p. 1463)	11.11 (p. 1463)		
10.12	12.2 (p. 1463)		12.2 (p. 1463)	11.11 (p. 1463)		
10.11, 10.10, 10.9, 10.7, 10.6, 10.5, 10.4, 10.3, 10.1	11.11 (p. 1463)			11.11 (p. 1463)		
9.6.21	13.2 (p. 1461)	13.2 (p. 1461)	12.6 (p. 1462)	11.11 (p. 1463)	10.16 (p. 1467)	
9.6.20	13.1 (p. 1461)	13.1 (p. 1461)	12.5 (p. 1462)	11.10 (p. 1464)	10.16 (p. 1467)	
9.6.19	12.4 (p. 1462)		12.4 (p. 1462)	11.9 (p. 1464)	10.16 (p. 1467)	
9.6.18	12.3 (p. 1463)		12.3 (p. 1463)	11.8 (p. 1464)	10.16 (p. 1467)	
9.6.17	12.2 (p. 1463)		12.2 (p. 1463)	11.7 (p. 1464)	10.16 (p. 1467)	
9.6.16	11.6 (p. 1464)			11.6 (p. 1464)	10.16 (p. 1467)	
9.6.15	11.5 (p. 1465)			11.5 (p. 1465)	10.16 (p. 1467)	
9.6.14	11.4 (p. 1465)			11.4 (p. 1465)	10.16 (p. 1467)	
9.6.12	11.2 (p. 1465)			11.2 (p. 1465)	10.16 (p. 1467)	
9.6.11	11.1 (p. 1465)			11.1 (p. 1465)	10.16 (p. 1467)	

Current source version	Newest upgrade target	Preferred major upgrade targets					
9.6.10, 9.6.9, 9.6.8, 9.6.6, 9.6.5, 9.6.3, 9.6.2, 9.6.1	10.16 (p. 1467)				10.16 (p. 1467)		
9.5.25	12.6 (p. 1462)		12.6 (p. 1462)	11.11 (p. 1463)	10.16 (p. 1467)	9.6.21 (p. 1471)	
9.5.24	12.5 (p. 1462)		12.5 (p. 1462)	11.10 (p. 1464)	10.16 (p. 1467)	9.6.21 (p. 1471)	
9.5.23	12.4 (p. 1462)		12.4 (p. 1462)	11.9 (p. 1464)	10.16 (p. 1467)	9.6.21 (p. 1471)	
9.5.22	12.3 (p. 1463)		12.3 (p. 1463)	11.8 (p. 1464)	10.16 (p. 1467)	9.6.21 (p. 1471)	
9.5.21	12.2 (p. 1463)		12.2 (p. 1463)	11.7 (p. 1464)	10.16 (p. 1467)	9.6.21 (p. 1471)	
9.5.20	11.6 (p. 1464)			11.6 (p. 1464)	10.16 (p. 1467)	9.6.21 (p. 1471)	
9.5.19	11.5 (p. 1465)			11.5 (p. 1465)	10.16 (p. 1467)	9.6.21 (p. 1471)	
9.5.18	11.4 (p. 1465)			11.4 (p. 1465)	10.16 (p. 1467)	9.6.21 (p. 1471)	
9.5.16	11.2 (p. 1465)			11.2 (p. 1465)	10.16 (p. 1467)	9.6.21 (p. 1471)	
9.5.15	11.1 (p. 1465)			11.1 (p. 1465)	10.16 (p. 1467)	9.6.21 (p. 1471)	
9.5.14, 9.5.13, 9.5.12, 9.5.10, 9.5.9, 9.5.9, 9.5.7, 9.5.6, 9.5.4	9.6.21 (p. 1471)						9.6.21 (p. 1471)

To get a list of all valid upgrade targets for a current source version, use the [describe-db-engine-versions](#) CLI command. For example:

```
export REGION=AWS-Region
export ENDPOINT=https://rds.AWS-Region.amazonaws.com

aws rds describe-db-engine-versions --engine postgres --region $REGION --endpoint
$ENDPOINT --output text --query '*[?ValidUpgradeTarget[?IsMajorVersionUpgrade==`true`].
{EngineVersion:EngineVersion}' --engine-version DB-current-version
```

How to perform a major version upgrade

We recommend the following process when upgrading an Amazon RDS PostgreSQL DB instance:

1. **Have a version-compatible parameter group ready** – If you are using a custom parameter group, you have two options. You can specify a default parameter group for the new DB engine version. Or you can create your own custom parameter group for the new DB engine version.

If you associate a new parameter group with a DB instance, reboot the database after the upgrade completes. If the instance needs to be rebooted to apply the parameter group changes, the instance's parameter group status shows `pending-reboot`. You can view an instance's parameter group status in the console or by using a [describe-db-instances](#) command, such as [describe-db-instances](#).

2. **Check for unsupported DB instance classes** – Check that your database's instance class is compatible with the PostgreSQL version you are upgrading to. For more information, see [Supported DB engines for DB instance classes \(p. 8\)](#).
3. **Check for unsupported usage:**
 - **Prepared transactions** – Commit or roll back all open prepared transactions before attempting an upgrade.

You can use the following query to verify that there are no open prepared transactions on your instance.

```
SELECT count(*) FROM pg_catalog.pg_prepared_xacts;
```

- **Reg* data types** – Remove all uses of the *reg** data types before attempting an upgrade. Except for `regtype` and `regclass`, you can't upgrade the *reg** data types. The `pg_upgrade` utility can't persist this data type, which is used by Amazon RDS to do the upgrade.

To verify that there are no uses of unsupported *reg** data types, use the following query for each database.

```
SELECT count(*) FROM pg_catalog.pg_class c, pg_catalog.pg_namespace n,
pg_catalog.pg_attribute a
WHERE c.oid = a.attrelid
AND NOT a.attisdropped
AND a.atttypid IN ('pg_catalog.regproc'::pg_catalog.regtype,
'pg_catalog.regprocedure'::pg_catalog.regtype,
'pg_catalog.regoper'::pg_catalog.regtype,
'pg_catalog.regoperator'::pg_catalog.regtype,
'pg_catalog.regconfig'::pg_catalog.regtype,
'pg_catalog.regdictionary'::pg_catalog.regtype)
AND c.relnamespace = n.oid
AND n.nspname NOT IN ('pg_catalog', 'information_schema');
```

4. **Handle read replicas** – An upgrade also upgrades the in-Region read replicas along with the primary instance.

You can't upgrade read replicas separately. If you could, it could lead to situations where the primary and replica instances have different PostgreSQL major versions. However, replica upgrades might increase downtime on the primary instance. To prevent a replica upgrade, promote the replica to a standalone instance or delete it before starting the upgrade process.

The upgrade process recreates the replica's parameter group based on the replica instance's current parameter group. You can apply a custom parameter group to a replica only after the upgrade completes by using the [modify-db-parameter-group](#) CLI command.

Read replicas on the virtual private cloud (VPC) platform are upgraded but replicas on the EC2-Classic platform aren't upgraded. Any EC2-Classic replicas are left in the replication terminated state after the upgrade process completes. To move a DB instance from the EC2-Classic platform into a VPC, see [Moving a DB instance not in a VPC into a VPC \(p. 1735\)](#). For more information, see [Working with PostgreSQL read replicas in Amazon RDS \(p. 1544\)](#).

5. **Perform a backup** – We recommend that you perform a backup before performing the major version upgrade so that you have a known restore point for your database. If your backup retention period is greater than 0, the upgrade process creates DB snapshots of your DB instance before and after upgrading. To change your backup retention period, see [Modifying an Amazon RDS DB instance \(p. 250\)](#). To perform a backup manually, see [Creating a DB snapshot \(p. 346\)](#).
6. **Upgrade certain extensions before the major version upgrade** – If you plan to skip a major version with the upgrade, you need to update certain extensions *before* performing the major version

upgrade. Upgrading from versions 9.4.x, 9.5.x, or 9.6.x to versions 11.x skip a major version. The extensions to update include:

- address_standardizer
- address_standardizer_data_us
- postGIS
- postgis_tiger_geocoder
- postgis_topology

Run the following command for each extension you are using.

```
ALTER EXTENSION PostgreSQL-extension UPDATE TO 'new-version'
```

For more information, see [Upgrading PostgreSQL extensions \(p. 1540\)](#).

7. **Drop certain extensions before the major version upgrade** – An upgrade that skips a major version to version 11.x doesn't support updating the pgRouting extension. Upgrading from versions 9.4.x, 9.5.x, or 9.6.x to versions 11.x skip a major version. It's safe to drop the pgRouting extension and then reinstall it to a compatible version after the upgrade. For the extension versions you can update to, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

The tsearch2 and chkpass extensions are no longer supported for PostgreSQL versions 11 or later. If you are upgrading to version 11.x, drop the tsearch2, and chkpass extensions before the upgrade.

8. **Drop unknown data types** – Drop unknown data types depending on the target version.

PostgreSQL version 10 stopped supporting the unknown data type. If a version 9.6 database uses the unknown data type, an upgrade to a version 10 shows an error message such as the following:

```
Database instance is in a state that cannot be upgraded: PreUpgrade checks failed:  
The instance could not be upgraded because the 'unknown' data type is used in user  
tables.  
Please remove all usages of the 'unknown' data type and try again."
```

To find the unknown data type in your database so you can remove the offending column or change it to a supported data type, use the following SQL:

```
SELECT DISTINCT data_type FROM information_schema.columns WHERE data_type ILIKE  
'unknown';
```

9. **Perform an upgrade dry run** – We highly recommend testing a major version upgrade on a duplicate of your production database before attempting the upgrade on your production database. To create a duplicate test instance, you can either restore your database from a recent snapshot or do a point-in-time restore of your database to its latest restorable time. For more information, see [Restoring from a snapshot \(p. 350\)](#) or [Restoring a DB instance to a specified time \(p. 389\)](#). For details on performing the upgrade, see [Manually upgrading the engine version \(p. 271\)](#).

During the major version upgrade, the public and template1 databases and the public schema in every database on the instance are temporarily renamed. These objects appear in the logs with their original name and a random string appended. The string is appended so that custom settings such as locale and owner are preserved during the major version upgrade. After the upgrade completes, the objects are renamed back to their original names.

Note

During the major version upgrade process, you can't do a point-in-time restore of your instance. After Amazon RDS performs the upgrade, it takes an automatic backup of the instance. You can perform a point-in-time restore to times before the upgrade began and after the automatic backup of your instance has completed.

10If an upgrade fails with precheck procedure errors, resolve the issues – During the major version upgrade process, Amazon RDS for PostgreSQL first runs a precheck procedure to identify any issues that might cause the upgrade to fail. The precheck procedure checks all potential incompatible conditions across all databases in the instance.

If the precheck encounters an issue, it creates a log event indicating the upgrade precheck failed. The precheck process details are in an upgrade log named `pg_upgrade_precheck.log` for all the databases of a DB instance. Amazon RDS appends a timestamp to the file name. For more information about viewing logs, see [Accessing Amazon RDS database log files \(p. 504\)](#).

If a replica upgrade fails at precheck, replication on the failed replica is broken and the replica is put in the terminated state. Delete the replica and recreate a new replica based on the upgraded primary instance.

Resolve all of the issues identified in the precheck log and then retry the major version upgrade. The following is an example of a precheck log.

```
-----  
Upgrade could not be run on Wed Apr 4 18:30:52 2018  
-----  
The instance could not be upgraded from 9.6.11 to 10.6 for the following reasons.  
Please take appropriate action on databases that have usage incompatible with the  
requested major engine version upgrade and try the upgrade again.  
  
* There are uncommitted prepared transactions. Please commit or rollback all prepared  
transactions.* One or more role names start with 'pg_'. Rename all role names that start  
with 'pg_'.  
  
* The following issues in the database 'my"million$db' need to be corrected before  
upgrading:** The ["line","reg*"] data types are used in user tables. Remove all usage of  
these data types.  
** The database name contains characters that are not supported by RDS for PostgreSQL.  
Rename the database.  
** The database has extensions installed that are not supported on the target database  
version. Drop the following extensions from your database: ["tsearch2"].  
  
* The following issues in the database 'mydb' need to be corrected before upgrading:**  
The database has views or materialized views that depend on 'pg_stat_activity'. Drop the  
views.
```

11If a replica upgrade fails while upgrading the database, resolve the issue – A failed replica is placed in the incompatible-restore state and replication is terminated on the DB instance. Delete the replica and recreate a new replica based on the upgraded primary instance.

A replica upgrade might fail for the following reasons:

- It was unable to catch up with the primary instance even after a wait time.
- It was in a terminal or incompatible lifecycle state such as storage-full, incompatible-restore, and so on.
- When the primary instance upgrade started, there was a separate minor version upgrade running on the replica.
- The replica instance used incompatible parameters.
- The replica instance was unable to communicate with the primary instance to synchronize the data folder.

12Upgrade your production instance – When the dry-run major version upgrade is successful, you should be able to upgrade your production database with confidence. For more information, see [Manually upgrading the engine version \(p. 271\)](#).

After the major version upgrade is complete, we recommend the following:

- Run the `ANALYZE` operation to refresh the `pg_statistic` table.
- A PostgreSQL upgrade doesn't upgrade any PostgreSQL extensions. To upgrade extensions, see [Upgrading PostgreSQL extensions \(p. 1540\)](#).
- Optionally, use Amazon RDS to view two logs that the `pg_upgrade` utility produces. These are `pg_upgrade_internal.log` and `pg_upgrade_server.log`. Amazon RDS appends a timestamp to the file name for these logs. You can view these logs as you can any other log. For more information, see [Accessing Amazon RDS database log files \(p. 504\)](#).

You can also upload the upgrade logs to Amazon CloudWatch Logs. For more information, see [Publishing PostgreSQL logs to Amazon CloudWatch Logs \(p. 537\)](#).

- To verify that everything works as expected, test your application on the upgraded database with a similar workload. After the upgrade is verified, you can delete this test instance.

Automatic minor version upgrades for PostgreSQL

If you enable the **Auto minor version upgrade** option when creating or modifying a DB instance, you can have your DB instance automatically upgraded.

For each RDS for PostgreSQL major version, one minor version is designated by RDS as the automatic upgrade version. After a minor version has been tested and approved by Amazon RDS, the minor version upgrade occurs automatically during your maintenance window. RDS doesn't automatically set newer released minor versions as the automatic upgrade version. Before RDS designates a newer automatic upgrade version, several criteria are considered, such as the following:

- Known security issues
- Bugs in the PostgreSQL community version
- Overall fleet stability since the minor version was released

You can use the following AWS CLI command and script to determine the current automatic upgrade minor versions.

```
aws rds describe-db-engine-versions --engine postgres | grep -A 1 AutoUpgrade| grep -A 2 true |grep PostgreSQL | sort --unique | sed -e 's/"Description": "//g'
```

Note

If no results are returned, there is no automatic minor version upgrade available and scheduled.

A PostgreSQL DB instance is automatically upgraded during your maintenance window if the following criteria are met:

- The DB instance has the **Auto minor version upgrade** option enabled.
- The DB instance is running a minor DB engine version that is less than the current automatic upgrade minor version.

For more information, see [Automatically upgrading the minor engine version \(p. 273\)](#).

Note

A PostgreSQL upgrade doesn't upgrade PostgreSQL extensions. To upgrade extensions, see [Upgrading PostgreSQL extensions \(p. 1540\)](#).

Upgrading PostgreSQL extensions

A PostgreSQL engine upgrade doesn't upgrade most PostgreSQL extensions. To update an extension after a version upgrade, use the `ALTER EXTENSION UPDATE` command.

Note

If you are running the PostGIS extension in your Amazon RDS PostgreSQL DB instance, make sure that you follow the [PostGIS upgrade instructions](#) in the PostGIS documentation before you update the extension.

To upgrade an extension, use the following command.

```
ALTER EXTENSION extension_name UPDATE TO 'new_version'
```

For the list of supported versions of PostgreSQL extensions, see [PostgreSQL extensions supported on Amazon RDS \(p. 1482\)](#).

To list your currently installed extensions, use the PostgreSQL [pg_extension](#) catalog in the following command.

```
SELECT * FROM pg_extension;
```

To view a list of the specific extension versions that are available for your installation, use the PostgreSQL [pg_available_extension_versions](#) view in the following command.

```
SELECT * FROM pg_available_extension_versions;
```

Upgrading a PostgreSQL DB snapshot engine version

With Amazon RDS, you can create a storage volume DB snapshot of your PostgreSQL DB instance. When you create a DB snapshot, the snapshot is based on the engine version used by your Amazon RDS instance. In addition to upgrading the DB engine version of your DB instance, you can also upgrade the engine version for your DB snapshots.

After restoring a DB snapshot upgraded to a new engine version, make sure to test that the upgrade was successful. For more information about a major version upgrade, see [Upgrading the PostgreSQL DB engine for Amazon RDS \(p. 1533\)](#). To learn how to restore a DB snapshot, see [Restoring from a DB snapshot \(p. 349\)](#).

You can upgrade manual DB snapshots that are either encrypted or not encrypted.

For the list of engine versions that are available for upgrading a DB snapshot, see [Upgrading the PostgreSQL DB engine for Amazon RDS](#).

Note

- The DB snapshot must be from the same AWS Region as the account.
- You can't upgrade DB snapshots that are copied within region, copied across regions, or shared across accounts.
- You can't upgrade automated DB snapshots that are created during the automated backup process.

Console

To upgrade a DB snapshot

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. Choose the snapshot that you want to upgrade.
4. For **Actions**, choose **Upgrade snapshot**. The **Upgrade snapshot** page appears.
5. Choose the **New engine version** to upgrade to.
6. Choose **Save changes** to upgrade the snapshot.

During the upgrade process, all snapshot actions are disabled for this DB snapshot. Also, the DB snapshot status changes from **available** to **upgrading**, and then changes to **active** upon completion. If the DB snapshot can't be upgraded because of snapshot corruption issues, the status changes to **unavailable**. You can't recover the snapshot from this state.

Note

If the DB snapshot upgrade fails, the snapshot is rolled back to the original state with the original version.

AWS CLI

To upgrade a DB snapshot to a new database engine version, use the AWS CLI [modify-db-snapshot](#) command.

Parameters

- **--db-snapshot-identifier** – The identifier of the DB snapshot to upgrade. The identifier must be a unique Amazon Resource Name (ARN). For more information, see [Working with Amazon Resource Names \(ARNs\) in Amazon RDS \(p. 309\)](#).
- **--engine-version** – The engine version to upgrade the DB snapshot to.

Example

For Linux, macOS, or Unix:

```
aws rds modify-db-snapshot \
--db-snapshot-identifier my_db_snapshot \
--engine-version new_version
```

For Windows:

```
aws rds modify-db-snapshot ^
--db-snapshot-identifier my_db_snapshot ^
--engine-version new_version
```

RDS API

To upgrade a DB snapshot to a new database engine version, call the Amazon RDS API [ModifyDBSnapshot](#) operation.

- **DBSnapshotIdentifier** – The identifier of the DB snapshot to upgrade. The identifier must be a unique Amazon Resource Name (ARN). For more information, see [Working with Amazon Resource Names \(ARNs\) in Amazon RDS \(p. 309\)](#).
- **EngineVersion** – The engine version to upgrade the DB snapshot to.

Working with PostgreSQL read replicas in Amazon RDS

You usually use read replicas to configure replication between Amazon RDS DB instances. For general information about read replicas, see [Working with read replicas \(p. 278\)](#).

This section contains specific information about working with read replicas on PostgreSQL.

Topics

- [Read replica configuration with PostgreSQL \(p. 1544\)](#)
- [Monitoring PostgreSQL read replicas \(p. 1545\)](#)
- [Read replica limitations with PostgreSQL \(p. 1545\)](#)
- [Replication interruptions with PostgreSQL read replicas \(p. 1545\)](#)
- [Troubleshooting a PostgreSQL read replica problem \(p. 1546\)](#)

Read replica configuration with PostgreSQL

Amazon RDS PostgreSQL uses PostgreSQL native streaming replication to create a read-only copy of a source DB instance. This read replica (a "standby" in PostgreSQL terms) DB instance is an asynchronously created physical replication of the source DB instance. It's created by a special connection that transmits write ahead log (WAL) data between the source DB instance and the read replica where PostgreSQL asynchronously streams database changes as they are made.

PostgreSQL uses a "replication" role to perform streaming replication. The role is privileged, but can't be used to modify any data. PostgreSQL uses a single process for handling replication.

Before a DB instance can serve as a source DB instance, you must enable automatic backups on the source DB instance by setting the backup retention period to a value other than 0.

Creating a PostgreSQL read replica doesn't require an outage for the source DB instance. Amazon RDS sets the necessary parameters and permissions for the source DB instance and the read replica without any service interruption. A snapshot is taken of the source DB instance, and this snapshot becomes the read replica. No outage occurs when you delete a read replica.

You can create up to five read replicas from one source DB instance. For replication to operate effectively, each read replica should have the same amount of compute and storage resources as the source DB instance. If you scale the source DB instance, also scale the read replicas.

Amazon RDS overrides any incompatible parameters on a read replica if it prevents the read replica from starting. For example, suppose that the `max_connections` parameter value is higher on the source DB instance than on the read replica. In that case, Amazon RDS updates the parameter on the read replica to be the same value as that on the source DB instance.

PostgreSQL DB instances use a secure connection that you can encrypt by setting the `ssl` parameter to 1 for both the source and the read replica instances.

You can create a read replica from either single-AZ or Multi-AZ DB instance deployments. You use Multi-AZ deployments to improve the durability and availability of critical data, but you can't use the Multi-AZ secondary to serve read-only queries. Instead, you can create read replicas from high-traffic Multi-AZ DB instances to offload read-only queries. If the source instance of a Multi-AZ deployment fails over to the secondary, any associated read replicas automatically switch to use the secondary (now primary) as their replication source. For more information, see [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).

You can create a read replica as a Multi-AZ DB instance. Amazon RDS creates a standby of your replica in another Availability Zone for failover support for the replica. Creating your read replica as a Multi-AZ DB instance is independent of whether the source database is a Multi-AZ DB instance.

If you use the [postgres_fdw](#) extension to access data from a remote server, the read replica will also have access to the remote server. For more information about using `postgres_fdw`, see [Accessing external data with the postgres_fdw extension \(p. 1592\)](#).

Monitoring PostgreSQL read replicas

For PostgreSQL read replicas, you can monitor replication lag in Amazon CloudWatch by viewing the Amazon RDS `ReplicaLag` metric. The `ReplicaLag` metric reports the value of `SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS replica_lag`.

Read replica limitations with PostgreSQL

The following are limitations for PostgreSQL read replicas:

- Each PostgreSQL read replicas is read-only. You can't make a writable read replica.
- You can't create a read replica from another read replica. Thus, you can't create cascading read replicas.
- You can promote a PostgreSQL read replica to be a new source DB instance. However, the read replica doesn't become the new source DB instance automatically. The read replica, when promoted, stops receiving WAL communications and is no longer a read-only instance. You must set up any replication you intend to have going forward because the promoted read replica is now a new source DB instance.
- If no user transactions are occurring on the source DB instance, a PostgreSQL read replica reports a replication lag of up to five minutes.
- You can't create physical replication slots in PostgreSQL.
- You can't enable automated backups on PostgreSQL read replicas.

Replication interruptions with PostgreSQL read replicas

In several situations, a PostgreSQL source DB instance can unintentionally break replication with a read replica. These situations include the following:

- The `max_wal_senders` parameter is set too low to provide enough data to the number of read replicas. This situation causes replication to stop.
- The PostgreSQL parameter `wal_keep_segments` dictates how many WAL files are kept to provide data to the read replicas. The parameter value specifies the number of logs to keep. If you set the parameter value too low, you can cause a read replica to fall so far behind that streaming replication stops. In this case, Amazon RDS reports a replication error and begins recovery on the read replica by replaying the source DB instance's archived WAL logs. This recovery process continues until the read replica has caught up enough to continue streaming replication. For more information, see [Troubleshooting a PostgreSQL read replica problem \(p. 1546\)](#).
- Starting with PostgreSQL 13, a database restart isn't required for read replicas if the source DB instance IP address changes. For PostgreSQL versions older than 13, changes to the source DB instance IP address require a read replica reboot. IP address changes include a DB instance name change or a DB instance class change.

When the WAL stream that provides data to a read replica is broken, PostgreSQL switches into recovery mode to restore the read replica by using archived WAL files. When this process is complete, PostgreSQL attempts to re-establish streaming replication.

Troubleshooting a PostgreSQL read replica problem

PostgreSQL uses replication slots for cross-Region replication, so the process for troubleshooting same-region replication problems and cross-Region replication problems is different.

Troubleshooting PostgreSQL read replica problems within an AWS Region

The PostgreSQL parameter, `wal_keep_segments`, dictates how many write ahead log (WAL) files are kept to provide data to the read replicas. The parameter value specifies the number of logs to keep. If you set the parameter value too low, you can cause a read replica to fall so far behind that streaming replication stops. In this case, Amazon RDS reports a replication error and begins recovery on the read replica by replaying the source DB instance's archived WAL logs. This recovery process continues until the read replica has caught up enough to continue streaming replication.

The PostgreSQL log on the read replica shows when Amazon RDS is recovering a read replica that is this state by replaying archived WAL files.

```
2014-11-07 19:01:10 UTC::@[23180]:DEBUG: switched WAL source from archive to stream after
failure 2014-11-07 19:01:10 UTC::@[11575]:LOG: started streaming WAL from primary at
1A/D3000000 on timeline 1 2014-11-07 19:01:10 UTC::@[11575]:FATAL: could not receive
data from WAL stream: ERROR: requested WAL segment 000000010000001A000000D3 has already been
removed 2014-11-07 19:01:10 UTC::@[23180]:DEBUG: could not restore file "00000002.history" from archive: return code 0 2014-11-07 19:01:15 UTC::@[23180]:DEBUG: switched WAL source from stream to archive after failure
recovering 000000010000001A000000D3 2014-11-07 19:01:16 UTC::@[23180]:LOG: restored log file "000000010000001A000000D3"
from archive
```

After a certain amount of time, Amazon RDS replays enough archived WAL files on the replica to catch up and allow the read replica to begin streaming again. At this point, PostgreSQL resumes streaming and writes a similar line to the following to the log file.

```
2014-11-07 19:41:36 UTC::@[24714]:LOG: started streaming WAL from primary at 1B/B6000000
on timeline 1
```

You can determine how many WAL files you should keep by looking at the checkpoint information in the log. The PostgreSQL log shows the following information at each checkpoint. By looking at the "# recycled" transaction log files of these log statements, you can understand how many transaction files will be recycled during a time range and use this information to tune the `wal_keep_segments` parameter.

```
2014-11-07 19:59:35 UTC::@[26820]:LOG: checkpoint complete: wrote 376 buffers (0.2%); 0 transaction log file(s) added, 0 removed, 1 recycled; write=35.681 s, sync=0.013 s, total=35.703 s; sync files=10, longest=0.013 s, average=0.001 s
```

For example, suppose that the PostgreSQL log shows that 35 files are recycled from the "checkpoint completed" log statements within a 5-minute time frame. In that case, we know that with this usage pattern a read replica relies on 35 transaction files in five minutes. A read replica can't survive five minutes in a nonstreaming state if the source DB instance is set to the default `wal_keep_segments` parameter value of 32.

Troubleshooting PostgreSQL read replica problems across AWS Regions

PostgreSQL version 9.5.2 uses physical replication slots to manage write ahead log (WAL) retention on the source DB instance. For each cross-Region read replica instance, Amazon RDS creates and associates a physical replication slot. You can use two Amazon CloudWatch metrics, Oldest Replication Slot Lag and Transaction Logs Disk Usage, to see how far behind the most lagging replica is in terms of WAL data received and to see how much storage is being used for WAL data. The Transaction Logs Disk Usage value can substantially increase when a cross-Region read replica is lagging significantly.

If the workload on your DB instance generates a large amount of WAL data, you might need to change the DB instance class of your source DB instance and read replica. In that case, you change it to one with high (10 Gbps) network performance for the replica to keep up. The Amazon CloudWatch metric Transaction Logs Generation can help you understand the rate at which your workload is generating WAL data.

To determine the status of a cross-Region read replica, you can query pg_replication_slots on the source instance, as in the following example:

```
postgres=# select * from pg_replication_slots;

          slot_name           | plugin | slot_type | datoid | database |
active | active_pid | xmin | catalog_xmin | restart_lsn
-----+-----+-----+-----+-----+
rds_us_east_1_db_uzwlholddgpb1ksce6hgw4nkte |       | physical |        |          | t
| 12598 |       |           | 4E/95000060
(1 row)
```

Importing data into PostgreSQL on Amazon RDS

Suppose that you have an existing PostgreSQL deployment that you want to move to Amazon RDS. The complexity of your task depends on the size of your database and the types of database objects that you're transferring. For example, consider a database that contains datasets on the order of gigabytes, along with stored procedures and triggers. Such a database is going to be more complicated than a simple database with only a few megabytes of test data and no triggers or stored procedures.

We recommend that you use native PostgreSQL database migration tools under the following conditions:

- You have a homogeneous migration, where you are migrating from a database with the same database engine as the target database.
- You are migrating an entire database.
- The native tools allow you to migrate your system with minimal downtime.

In most other cases, performing a database migration using AWS Database Migration Service (AWS DMS) is the best approach. AWS DMS can migrate databases without downtime and, for many database engines, continue ongoing replication until you are ready to switch over to the target database. You can migrate to either the same database engine or a different database engine using AWS DMS. If you are migrating to a different database engine than your source database, you can use the AWS Schema Conversion Tool (AWS SCT). You use AWS SCT to migrate schema objects that are not migrated by AWS DMS. For more information about AWS DMS, see [What is AWS Database Migration Service?](#)

Modify your DB parameter group to include the following settings *for your import only*. You should test the parameter settings to find the most efficient settings for your DB instance size. You also need to revert back to production values for these parameters after your import completes.

Modify your DB instance settings to the following:

- Disable DB instance backups (set backup_retention to 0).
- Disable Multi-AZ.

Modify your DB parameter group to include the following settings. You should only use these settings when importing data. You should test the parameter settings to find the most efficient settings for your DB instance size. You also need to revert back to production values for these parameters after your import completes.

Parameter	Recommended value when importing	Description
<code>maintenance_work_mem</code>	524288, 1048576, 2097152, or 4194304 (in KB). These settings are comparable to 512 MB, 1 GB, 2 GB, and 4 GB.	The value for this setting depends on the size of your host. This parameter is used during CREATE INDEX statements and each parallel command can use this much memory. Calculate the best value so that you don't set this value so high that you run out of memory.
<code>checkpoint_segments</code>	256	The value for this setting consumes more disk space, but gives you less contention on a write ahead log (WAL). This setting is only supported for PostgreSQL versions 9.5 and earlier. For versions 9.6 and later, use <code>max_wal_size</code> .

Parameter	Recommended value when importing	Description
max_wal_size	256 (for version 9.6), 4096 (for versions 10 and later)	Maximum size to let the WAL grow during automatic checkpoints. Increasing this parameter can increase the amount of time needed for crash recovery. This parameter replaces <code>checkpoint_segments</code> for PostgreSQL 9.6 and later. For PostgreSQL version 9.6, this value is in 16 MB units. For later versions, the value is in 1 MB units. For example, in version 9.6, 128 means 128 chunks that are each 16 MB in size. In version 12.4, 2048 means 2048 chunks that are each 1 MB in size.
checkpoint_timeout	1800	The value for this setting allows for less frequent WAL rotation.
synchronous_commit	Off	Disable this setting to speed up writes. Turning this parameter off can increase the risk of data loss in the event of a server crash (do not turn off FSYNC).
wal_buffers	8192	This value is in 8 KB units. This again helps your WAL generation speed
autovacuum	Off	Disable the PostgreSQL auto vacuum parameter while you are loading data so that it doesn't use resources

Use the `pg_dump -Fc` (compressed) or `pg_restore -j` (parallel) commands with these settings.

Note

The PostgreSQL command `pg_dumpall` requires `super_user` permissions that are not granted when you create a DB instance, so it cannot be used for importing data.

Topics

- [Importing a PostgreSQL database from an Amazon EC2 instance \(p. 1549\)](#)
- [Using the \copy command to import data to a table on a PostgreSQL DB instance \(p. 1551\)](#)
- [Importing Amazon S3 data into an RDS for PostgreSQL DB instance \(p. 1552\)](#)
- [Transporting PostgreSQL databases between DB instances \(p. 1563\)](#)

Importing a PostgreSQL database from an Amazon EC2 instance

If you have data in a PostgreSQL server on an Amazon EC2 instance and want to move it to a PostgreSQL DB instance, you can use the following process. The following list shows the steps to take. Each step is discussed in more detail in the following sections.

1. Create a file using `pg_dump` that contains the data to be loaded
2. Create the target DB instance
3. Use `psql` to create the database on the DB instance and load the data

4. Create a DB snapshot of the DB instance

Step 1: Create a file using pg_dump that contains the data to load

The `pg_dump` utility uses the `COPY` command to create a schema and data dump of a PostgreSQL database. The dump script generated by `pg_dump` loads data into a database with the same name and recreates the tables, indexes, and foreign keys. You can use the `pg_restore` command and the `-d` parameter to restore the data to a database with a different name.

Before you create the data dump, you should query the tables to be dumped to get a row count so you can confirm the count on the target DB instance.

The following command creates a dump file called `mydb2dump.sql` for a database called `mydb2`.

```
prompt>pg_dump dbname=mydb2 -f mydb2dump.sql
```

Step 2: Create the target DB instance

Create the target PostgreSQL DB instance using either the Amazon RDS console, AWS CLI, or API. Create the instance with the backup retention setting set to 0 and disable Multi-AZ. Doing so allows faster data import. You must create a database on the instance before you can dump the data. The database can have the same name as the database that is contained the dumped data. Alternatively, you can create a database with a different name. In this case, you use the `pg_restore` command and the `-d` parameter to restore the data into the newly named database.

For example, the following commands can be used to dump, restore, and rename a database.

```
pg_dump -Fc -v -h [endpoint of instance] -U [master username] [database] > [database].dump
createdb [new database name]
pg_restore -v -h [endpoint of instance] -U [master username] -d [new database
name] [database].dump
```

Step 3: Use psql to create the database on the DB instance and load data

You can use the same connection you used to run the `pg_dump` command to connect to the target DB instance and recreate the database. Using `psql`, you can use the master user name and master password to create the database on the DB instance.

The following example uses `psql` and a dump file named `mydb2dump.sql` to create a database called `mydb2` on a PostgreSQL DB instance called `mypginstance`:

For Linux, macOS, or Unix:

```
psql \
-f mydb2dump.sql \
--host mypginstance.c6c8mntzhgv0.us-west-2.rds.amazonaws.com \
--port 8199 \
--username myawsuser \
--password password \
--dbname mydb2
```

For Windows:

```
psql ^  
-f mydb2dump.sql ^  
--host mypginstance.c6c8mntzhgv0.us-west-2.rds.amazonaws.com ^  
--port 8199 ^  
--username myawsuser ^  
--password password ^  
--dbname mydb
```

Step 4: Create a DB snapshot of the DB instance

Once you have verified that the data was loaded into your DB instance, we recommend that you create a DB snapshot of the target PostgreSQL DB instance. DB snapshots are complete backups of your DB instance that can be used to restore your DB instance to a known state. A DB snapshot taken immediately after the load protects you from having to load the data again in case of a mishap. You can also use such a snapshot to seed new DB instances. For information about creating a DB snapshot, see [Creating a DB snapshot \(p. 346\)](#).

Using the \copy command to import data to a table on a PostgreSQL DB instance

You can run the `\copy` command from the `psql` prompt to import data into a table on a PostgreSQL DB instance. The table must already exist on the DB instance. For more information on the `\copy` command, see the [PostgreSQL documentation](#).

Note

The `\copy` command doesn't provide confirmation of actions, such as a count of rows inserted. PostgreSQL does provide error messages if the copy command fails due to an error.

Create a .csv file from the data in the source table, log on to the target database on the PostgreSQL instance using `psql`, and then run the following command. This example uses *source-table* as the source table name, *source-table.csv* as the .csv file, and *target-db* as the target database:

```
target-db=> \copy source-table from 'source-table.csv' with DELIMITER ',';
```

You can also run the following command from your client computer command prompt. This example uses *source-table* as the source table name, *source-table.csv* as the .csv file, and *target-db* as the target database:

For Linux, macOS, or Unix:

```
$psql target-db \  
-U <admin user> \  
-p <port> \  
-h <DB instance name> \  
-c "\copy source-table from 'source-table.csv' with DELIMITER ','"
```

For Windows:

```
$psql target-db ^  
-U <admin user> ^  
-p <port> ^  
-h <DB instance name> ^
```

```
-c "\copy source-table from 'source-table.csv' with DELIMITER ',' "
```

Importing Amazon S3 data into an RDS for PostgreSQL DB instance

You can import data from Amazon S3 into a table belonging to an RDS for PostgreSQL DB instance. To do this, you use the `aws_s3` PostgreSQL extension that Amazon RDS provides.

Note

To import from Amazon S3 into RDS for PostgreSQL, your database must be running PostgreSQL version 10.7 or later.

For more information on storing data with Amazon S3, see [Create a bucket](#) in the *Amazon Simple Storage Service Getting Started Guide*. For instructions on how to upload a file to an Amazon S3 bucket, see [Add an object to a bucket](#) in the *Amazon Simple Storage Service Getting Started Guide*.

Topics

- [Overview of importing Amazon S3 data \(p. 1552\)](#)
- [Setting up access to an Amazon S3 bucket \(p. 1553\)](#)
- [Using the `aws_s3.table_import_from_s3` function to import Amazon S3 data \(p. 1558\)](#)
- [Function reference \(p. 1560\)](#)

Overview of importing Amazon S3 data

To import data stored in an Amazon S3 bucket to a PostgreSQL database table, follow these steps.

To import S3 data into Amazon RDS

1. Install the required PostgreSQL extensions. These include the `aws_s3` and `aws_commons` extensions. To do so, start `psql` and use the following command.

```
psql=> CREATE EXTENSION aws_s3 CASCADE;
NOTICE: installing required extension "aws_commons"
```

The `aws_s3` extension provides the [aws_s3.table_import_from_s3 \(p. 1560\)](#) function that you use to import Amazon S3 data. The `aws_commons` extension provides additional helper functions.

2. Identify the database table and Amazon S3 file to use.

The [aws_s3.table_import_from_s3 \(p. 1560\)](#) function requires the name of the PostgreSQL database table that you want to import data into. The function also requires that you identify the Amazon S3 file to import. To provide this information, take the following steps.

- a. Identify the PostgreSQL database table to put the data in. For example, the following is a sample `t1` database table used in the examples for this topic.

```
psql=> CREATE TABLE t1 (col1 varchar(80), col2 varchar(80), col3 varchar(80));
```

- b. Get the following information to identify the Amazon S3 file that you want to import:

- Bucket name – A *bucket* is a container for Amazon S3 objects or files.
- File path – The file path locates the file in the Amazon S3 bucket.
- AWS Region – The AWS Region is the location of the Amazon S3 bucket. For example, if the S3 bucket is in the US East (N. Virginia) Region, use `us-east-1`. For a listing of AWS Region names and associated values, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

To find how to get this information, see [View an object](#) in the *Amazon Simple Storage Service Getting Started Guide*. You can confirm the information by using the AWS CLI command `aws s3 cp`. If the information is correct, this command downloads a copy of the Amazon S3 file.

```
aws s3 cp s3://sample_s3_bucket/sample_file_path ./
```

- c. Use the [aws_commons.create_s3_uri \(p. 1562\)](#) function to create an `aws_commons._s3_uri_1` structure to hold the Amazon S3 file information. You provide this `aws_commons._s3_uri_1` structure as a parameter in the call to the [aws_s3.table_import_from_s3 \(p. 1560\)](#) function.

For a psql example, see the following.

```
psql=> SELECT aws_commons.create_s3_uri(  
      'sample_s3_bucket',  
      'sample.csv',  
      'us-east-1'  
) AS s3_uri \gset
```

3. Provide permission to access the Amazon S3 file.

To import data from an Amazon S3 file, give the RDS for PostgreSQL DB instance permission to access the Amazon S3 bucket the file is in. To do this, you use either an AWS Identity and Access Management (IAM) role or security credentials. For more information, see [Setting up access to an Amazon S3 bucket \(p. 1553\)](#).

4. Import the Amazon S3 data by calling the `aws_s3.table_import_from_s3` function.

After you complete the previous preparation tasks, use the [aws_s3.table_import_from_s3 \(p. 1560\)](#) function to import the Amazon S3 data. For more information, see [Using the aws_s3.table_import_from_s3 function to import Amazon S3 data \(p. 1558\)](#).

Setting up access to an Amazon S3 bucket

To import data from an Amazon S3 file, give the RDS for PostgreSQL DB instance permission to access the Amazon S3 bucket the file is in. You provide access to an Amazon S3 bucket in one of two ways, as described in the following topics.

Topics

- [Using an IAM role to access an Amazon S3 bucket \(p. 1553\)](#)
- [Using security credentials to access an Amazon S3 bucket \(p. 1557\)](#)
- [Troubleshooting access to Amazon S3 \(p. 1557\)](#)

Using an IAM role to access an Amazon S3 bucket

Before you load data from an Amazon S3 file, give your RDS for PostgreSQL DB instance permission to access the Amazon S3 bucket the file is in. This way, you don't have to manage additional credential information or provide it in the [aws_s3.table_import_from_s3 \(p. 1560\)](#) function call.

To do this, create an IAM policy that provides access to the Amazon S3 bucket. Create an IAM role and attach the policy to the role. Then assign the IAM role to your DB instance.

To give an RDS for PostgreSQL DB instance access to Amazon S3 through an IAM role

1. Create an IAM policy.

This policy provides the bucket and object permissions that allow your RDS for PostgreSQL DB instance to access Amazon S3.

Include in the policy the following required actions to allow the transfer of files from an Amazon S3 bucket to Amazon RDS:

- s3:GetObject
- s3>ListBucket

Include in the policy the following resources to identify the Amazon S3 bucket and objects in the bucket. This shows the Amazon Resource Name (ARN) format for accessing Amazon S3.

- arn:aws:s3:::*your-s3-bucket*
- arn:aws:s3:::*your-s3-bucket*/*

For more information on creating an IAM policy for Amazon RDS for PostgreSQL, see [Creating and using an IAM policy for IAM database access \(p. 1664\)](#). See also [Tutorial: Create and attach your first customer managed policy](#) in the *IAM User Guide*.

The following AWS CLI command creates an IAM policy named `rds-s3-import-policy` with these options. It grants access to a bucket named `your-s3-bucket`.

Note

After you create the policy, note the Amazon Resource Name (ARN) of the policy. You need the ARN for a subsequent step when you attach the policy to an IAM role.

Example

For Linux, macOS, or Unix:

```
aws iam create-policy \
--policy-name rds-s3-import-policy \
--policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "s3import",
            "Action": [
                "s3:GetObject",
                "s3>ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::your-s3-bucket",
                "arn:aws:s3:::your-s3-bucket/*"
            ]
        }
    ]
}'
```

For Windows:

```
aws iam create-policy ^
--policy-name rds-s3-import-policy ^
--policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
"Sid": "s3import",
"Action": [
    "s3:GetObject",
    "s3>ListBucket"
],
"Effect": "Allow",
"Resource": [
    "arn:aws:s3:::your-s3-bucket",
    "arn:aws:s3:::your-s3-bucket/*"
]
}
}'
```

2. Create an IAM role.

You do this so Amazon RDS can assume this IAM role on your behalf to access your Amazon S3 buckets. For more information, see [Creating a role to delegate permissions to an IAM user](#) in the *IAM User Guide*.

The following example shows using the AWS CLI command to create a role named `rds-s3-import-role`.

Example

For Linux, macOS, or Unix:

```
aws iam create-role \
--role-name rds-s3-import-role \
--assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}'
```

For Windows:

```
aws iam create-role ^
--role-name rds-s3-import-role ^
--assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}'
```

3. Attach the IAM policy that you created to the IAM role that you created.

The following AWS CLI command attaches the policy created earlier to the role named `rds-s3-import-role`. Replace `your-policy-arn` with the policy ARN that you noted in an earlier step.

Example

For Linux, macOS, or Unix:

```
aws iam attach-role-policy \
--policy-arn your-policy-arn \
--role-name rds-s3-import-role
```

For Windows:

```
aws iam attach-role-policy ^
--policy-arn your-policy-arn ^
--role-name rds-s3-import-role
```

4. Add the IAM role to the DB instance.

You do so by using the AWS Management Console or AWS CLI, as described following.

Note

Also, be sure the database you use doesn't have any restrictions noted in [Importing Amazon S3 data into an RDS for PostgreSQL DB instance \(p. 1552\)](#).

Console

To add an IAM role for a PostgreSQL DB instance using the console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose the PostgreSQL DB instance name to display its details.
3. On the **Connectivity & security** tab, in the **Manage IAM roles** section, choose the role to add under **Add IAM roles to this instance**.
4. Under **Feature**, choose **s3Import**.
5. Choose **Add role**.

AWS CLI

To add an IAM role for a PostgreSQL DB instance using the CLI

- Use the following command to add the role to the PostgreSQL DB instance named `my-db-instance`. Replace `your-role-arn` with the role ARN that you noted in a previous step. Use `s3Import` for the value of the `--feature-name` option.

Example

For Linux, macOS, or Unix:

```
aws rds add-role-to-db-instance \
--db-instance-identifier my-db-instance \
--feature-name s3Import \
--role-arn your-role-arn \
--region your-region
```

For Windows:

```
aws rds add-role-to-db-instance ^
--db-instance-identifier my-db-instance ^
--feature-name s3Import ^
--role-arn your-role-arn ^
--region your-region
```

RDS API

To add an IAM role for a PostgreSQL DB instance using the Amazon RDS API, call the [AddRoleToDBInstance](#) operation.

Using security credentials to access an Amazon S3 bucket

If you prefer, you can use security credentials to provide access to an Amazon S3 bucket instead of providing access with an IAM role. To do this, use the `credentials` parameter in the [aws_s3.table_import_from_s3 \(p. 1560\)](#) function call.

The `credentials` parameter is a structure of type `aws_commons._aws_credentials_1`, which contains AWS credentials. Use the [aws_commons.create_aws_credentials \(p. 1562\)](#) function to set the access key and secret key in an `aws_commons._aws_credentials_1` structure, as shown following.

```
psql=> SELECT aws_commons.create_aws_credentials(
  'sample_access_key', 'sample_secret_key', '')
AS creds \gset
```

After creating the `aws_commons._aws_credentials_1` structure, use the [aws_s3.table_import_from_s3 \(p. 1560\)](#) function with the `credentials` parameter to import the data, as shown following.

```
psql=> SELECT aws_s3.table_import_from_s3(
  't', '', '(format csv)',
  :'s3_uri',
  :'creds'
);
```

Or you can include the [aws_commons.create_aws_credentials \(p. 1562\)](#) function call inline within the `aws_s3.table_import_from_s3` function call.

```
psql=> SELECT aws_s3.table_import_from_s3(
  't', '', '(format csv)',
  :'s3_uri',
  aws_commons.create_aws_credentials('sample_access_key', 'sample_secret_key', '')
);
```

Troubleshooting access to Amazon S3

If you encounter connection problems when attempting to import Amazon S3 file data, see the following for recommendations:

- [Troubleshooting Amazon RDS identity and access \(p. 1689\)](#)
- [Troubleshooting Amazon S3](#)
- [Troubleshooting Amazon S3 and IAM](#)

Using the aws_s3.table_import_from_s3 function to import Amazon S3 data

Import your Amazon S3 data by calling the [aws_s3.table_import_from_s3 \(p. 1560\)](#) function.

Note

The following examples use the IAM role method for providing access to the Amazon S3 bucket. Thus, the `aws_s3.table_import_from_s3` function calls don't include credential parameters.

The following shows a typical PostgreSQL example using `psql`.

```
psql=> SELECT aws_s3.table_import_from_s3(
    't1',
    '',
    '(format csv)',
    :'s3_uri'
);
```

The parameters are the following:

- `t1` – The name for the table in the PostgreSQL DB instance to copy the data into.
- `''` – An optional list of columns in the database table. You can use this parameter to indicate which columns of the S3 data go in which table columns. If no columns are specified, all the columns are copied to the table. For an example of using a column list, see [Importing an Amazon S3 file that uses a custom delimiter \(p. 1558\)](#).
- `(format csv)` – PostgreSQL COPY arguments. The copy process uses the arguments and format of the [PostgreSQL COPY command](#). In the preceding example, the `COPY` command uses the comma-separated value (CSV) file format to copy the data.
- `s3_uri` – A structure that contains the information identifying the Amazon S3 file. For an example of using the [aws_commons.create_s3_uri \(p. 1562\)](#) function to create an `s3_uri` structure, see [Overview of importing Amazon S3 data \(p. 1552\)](#).

The return value is text. For the full reference of this function, see [aws_s3.table_import_from_s3 \(p. 1560\)](#).

The following examples show how to specify different kinds of files when importing Amazon S3 data.

Topics

- [Importing an Amazon S3 file that uses a custom delimiter \(p. 1558\)](#)
- [Importing an Amazon S3 compressed \(gzip\) file \(p. 1559\)](#)
- [Importing an encoded Amazon S3 file \(p. 1559\)](#)

Importing an Amazon S3 file that uses a custom delimiter

The following example shows how to import a file that uses a custom delimiter. It also shows how to control where to put the data in the database table using the `column_list` parameter of the [aws_s3.table_import_from_s3 \(p. 1560\)](#) function.

For this example, assume that the following information is organized into pipe-delimited columns in the Amazon S3 file.

```
1|foo1|bar1|elephant1
2|foo2|bar2|elephant2
3|foo3|bar3|elephant3
4|foo4|bar4|elephant4
```

...

To import a file that uses a custom delimiter

1. Create a table in the database for the imported data.

```
psql=> CREATE TABLE test (a text, b text, c text, d text, e text);
CREATE TABLE
```

2. Use the following form of the [aws_s3.table_import_from_s3 \(p. 1560\)](#) function to import data from the Amazon S3 file.

You can include the [aws_commons.create_s3_uri \(p. 1562\)](#) function call inline within the `aws_s3.table_import_from_s3` function call to specify the file.

```
psql=> SELECT aws_s3.table_import_from_s3(
    'test',
    'a,b,d,e',
    'DELIMITER ''|''',
    aws_commons.create_s3_uri('sampleBucket', 'pipeDelimitedSampleFile', 'us-east-2')
);
```

The data is now in the table in the following columns.

```
psql=> SELECT * FROM test;
a | b | c | d | e
---+---+---+---+---
1 | foo1 | | bar1 | elephant1
2 | foo2 | | bar2 | elephant2
3 | foo3 | | bar3 | elephant3
4 | foo4 | | bar4 | elephant4
```

Importing an Amazon S3 compressed (gzip) file

The following example shows how to import a file from Amazon S3 that is compressed with gzip.

Ensure that the file contains the following Amazon S3 metadata:

- Key: Content-Encoding
- Value: gzip

For more about adding these values to Amazon S3 metadata, see [How do I add metadata to an S3 object?](#) in the *Amazon Simple Storage Service Console User Guide*.

Import the gzip file into your RDS for PostgreSQL DB instance as shown following.

```
psql=> CREATE TABLE test_gzip(id int, a text, b text, c text, d text);
CREATE TABLE
psql=> SELECT aws_s3.table_import_from_s3(
    'test_gzip', '', '(format csv)',
    'myS3Bucket', 'test-data.gz', 'us-east-2'
);
```

Importing an encoded Amazon S3 file

The following example shows how to import a file from Amazon S3 that has Windows-1252 encoding.

```
psql=> SELECT aws_s3.table_import_from_s3(
  'test_table', '', 'encoding ''WIN1252'''',
  aws_commons.create_s3_uri('sampleBucket', 'SampleFile', 'us-east-2')
);
```

Function reference

Functions

- [aws_s3.table_import_from_s3 \(p. 1560\)](#)
- [aws_commons.create_s3_uri \(p. 1562\)](#)
- [aws_commons.create_aws_credentials \(p. 1562\)](#)

aws_s3.table_import_from_s3

Imports Amazon S3 data into an Amazon RDS table. The `aws_s3` extension provides the `aws_s3.table_import_from_s3` function. The return value is text.

Syntax

The required parameters are `table_name`, `column_list` and `options`. These identify the database table and specify how the data is copied into the table.

You can also use the following parameters:

- The `s3_info` parameter specifies the Amazon S3 file to import. When you use this parameter, access to Amazon S3 is provided by an IAM role for the PostgreSQL DB instance.

```
aws_s3.table_import_from_s3 (
    table_name text,
    column_list text,
    options text,
    s3_info aws_commons._s3_uri_1
)
```

- The `credentials` parameter specifies the credentials to access Amazon S3. When you use this parameter, you don't use an IAM role.

```
aws_s3.table_import_from_s3 (
    table_name text,
    column_list text,
    options text,
    s3_info aws_commons._s3_uri_1,
    credentials aws_commons._aws_credentials_1
)
```

Parameters

table_name

A required text string containing the name of the PostgreSQL database table to import the data into.

column_list

A required text string containing an optional list of the PostgreSQL database table columns in which to copy the data. If the string is empty, all columns of the table are used. For an example, see [Importing an Amazon S3 file that uses a custom delimiter \(p. 1558\)](#).

options

A required text string containing arguments for the PostgreSQL `COPY` command. These arguments specify how the data is to be copied into the PostgreSQL table. For more details, see the [PostgreSQL COPY documentation](#).

s3_info

An `aws_commons._s3_uri_1` composite type containing the following information about the S3 object:

- `bucket` – The name of the Amazon S3 bucket containing the file.
- `file_path` – The Amazon S3 file name including the path of the file.
- `region` – The AWS Region that the file is in. For a listing of AWS Region names and associated values, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

credentials

An `aws_commons._aws_credentials_1` composite type containing the following credentials to use for the import operation:

- Access key
- Secret key
- Session token

For information about creating an `aws_commons._aws_credentials_1` composite structure, see [aws_commons.create_aws_credentials \(p. 1562\)](#).

Alternate syntax

To help with testing, you can use an expanded set of parameters instead of the `s3_info` and `credentials` parameters. Following are additional syntax variations for the `aws_s3.table_import_from_s3` function:

- Instead of using the `s3_info` parameter to identify an Amazon S3 file, use the combination of the `bucket`, `file_path`, and `region` parameters. With this form of the function, access to Amazon S3 is provided by an IAM role on the PostgreSQL DB instance.

```
aws_s3.table_import_from_s3 (
    table_name text,
    column_list text,
    options text,
    bucket text,
    file_path text,
    region text
)
```

- Instead of using the `credentials` parameter to specify Amazon S3 access, use the combination of the `access_key`, `session_key`, and `session_token` parameters.

```
aws_s3.table_import_from_s3 (
    table_name text,
    column_list text,
    options text,
    bucket text,
    file_path text,
    region text,
    access_key text,
    secret_key text,
    session_token text
)
```

Alternate parameters

bucket

A text string containing the name of the Amazon S3 bucket that contains the file.

file_path

A text string containing the Amazon S3 file name including the path of the file.

region

A text string containing the AWS Region that the file is in. For a listing of AWS Region names and associated values, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

access_key

A text string containing the access key to use for the import operation. The default is NULL.

secret_key

A text string containing the secret key to use for the import operation. The default is NULL.

session_token

(Optional) A text string containing the session key to use for the import operation. The default is NULL.

aws_commons.create_s3_uri

Creates an `aws_commons._s3_uri_1` structure to hold Amazon S3 file information. Use the results of the `aws_commons.create_s3_uri` function in the `s3_info` parameter of the [aws_s3.table_import_from_s3 \(p. 1560\)](#) function.

Syntax

```
aws_commons.create_s3_uri(  
    bucket text,  
    file_path text,  
    region text  
)
```

Parameters

bucket

A required text string containing the Amazon S3 bucket name for the file.

file_path

A required text string containing the Amazon S3 file name including the path of the file.

region

A required text string containing the AWS Region that the file is in. For a listing of AWS Region names and associated values, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

aws_commons.create_aws_credentials

Sets an access key and secret key in an `aws_commons._aws_credentials_1` structure. Use the results of the `aws_commons.create_aws_credentials` function in the `credentials` parameter of the [aws_s3.table_import_from_s3 \(p. 1560\)](#) function.

Syntax

```
aws_commons.create_aws_credentials(  
    access_key text,  
    secret_key text,  
    session_token text  
)
```

Parameters

access_key

A required text string containing the access key to use for importing an Amazon S3 file. The default is NULL.

secret_key

A required text string containing the secret key to use for importing an Amazon S3 file. The default is NULL.

session_token

An optional text string containing the session token to use for importing an Amazon S3 file. The default is NULL. If you provide an optional *session_token*, you can use temporary credentials.

Transporting PostgreSQL databases between DB instances

By using PostgreSQL Transportable Databases for Amazon RDS, you can transport a PostgreSQL database between two DB instances. This provides an extremely fast method of migrating large databases between separate DB instances. To transport databases using this method, your DB instances must both run the same major version of PostgreSQL.

To use transportable databases, install the `pg_transport` extension. This extension provides a physical transport mechanism to move each database. By streaming the database files with minimal processing, physical transport moves data much faster than traditional dump and load processes and takes minimal downtime. PostgreSQL transportable databases use a pull model where the destination DB instance imports the database from the source DB instance.

Note

PostgreSQL transportable databases are available in RDS for PostgreSQL versions 10.10 and later, and 11.5 and later.

Topics

- [Limitations for using PostgreSQL transportable databases \(p. 1563\)](#)
- [Setting up to transport PostgreSQL databases \(p. 1564\)](#)
- [Transporting a PostgreSQL database using the `transport.import_from_server` function \(p. 1565\)](#)
- [What happens during database transport \(p. 1565\)](#)
- [`transport.import_from_server` function reference \(p. 1566\)](#)
- [Configuration parameters for the `pg_transport` extension \(p. 1566\)](#)

Limitations for using PostgreSQL transportable databases

Transportable databases have the following limitations:

- **Read replicas** – You can't use transportable databases on read replicas or parent instances of read replicas.
- **Unsupported column types** – You can't use the `reg` data types in any database tables that you plan to transport with this method. These types depend on system catalog object IDs (OIDs), which often change during transport.
- **Tablespaces** – All source database objects must be in the default `pg_default` tablespace.
- **Compatibility** – Both the source and destination DB instances must run the same major version of PostgreSQL.

Before transport begins, the `transport.import_from_server` function compares the source and destination DB instances to ensure database compatibility. This includes verifying PostgreSQL major version compatibility. Also, the function verifies that the destination DB instance likely has enough space to receive the source database. The function performs several additional checks to make sure that the transport is smooth.

- **Extensions** – The only extension that you can install on the source DB instance during transport is `pg_transport`.
- **Roles and ACLs** – The source database's access privileges and ownership information aren't carried over to the destination database. All database objects are created and owned by the local destination user of the transport.
- **Concurrent transports** – You can run up to 32 total transports at the same time on a DB instance, including both imports and exports. To define the worker processes used for each transport, use the `pg_transport.work_mem` and `pg_transport.num_workers` parameters. To accommodate concurrent transports, you might need to increase the `max_worker_processes` parameter quite a bit. For more information, see [Configuration parameters for the pg_transport extension \(p. 1566\)](#).

Setting up to transport PostgreSQL databases

To prepare to transport a PostgreSQL database from one DB instance to another, take the following steps.

To set up for transporting a PostgreSQL database

1. Make sure that the source DB instance's security group allows inbound traffic from the destination DB instance. This is required because the destination DB instance starts the database transport with an import call to the source DB instance. For information about how to use security groups, see [Controlling access with security groups \(p. 1699\)](#).
2. For both the source and destination DB instances, add `pg_transport` to the `shared_preload_libraries` parameter for each parameter group. The `shared_preload_libraries` parameter is static and requires a database restart for changes to take effect. For information about parameter groups, see [Working with DB parameter groups \(p. 228\)](#).
3. For both the source and destination DB instances, install the required `pg_transport` PostgreSQL extension.

To do so, start `psql` as a user with the `rds_superuser` role for each DB instance, and then run the following command.

```
psql=> CREATE EXTENSION pg_transport;
```

Transporting a PostgreSQL database using the `transport.import_from_server` function

After you complete the process described in [Setting up to transport PostgreSQL databases \(p. 1564\)](#), you can start the transport. To do so, run the `transport.import_from_server` (p. 1566) function on the destination DB instance.

Note

Both the destination user for transport and the source user for the connection must be members of the `rds_superuser` role.

The destination DB instance can't already contain a database with the same name as the source database to be transported, or the transport fails.

The following shows an example transport.

```
SELECT transport.import_from_server(
    'source-db-instance-endpoint',
    'source-db-instance-port',
    'source-db-instance-user',
    'source-user-password',
    'source-database-name',
    'destination-user-password',
    false);
```

This function requires that you provide database user passwords. Thus, we recommend that you change the passwords of the user roles you used after transport is complete. Or, you can use SQL bind variables to create temporary user roles. Use these temporary roles for the transport and then discard the roles afterwards.

For details of the `transport.import_from_server` function and its parameters, see [transport.import_from_server function reference \(p. 1566\)](#).

What happens during database transport

The `transport.import_from_server` function creates the in-transit database on the destination DB instance. The in-transit database is inaccessible on the destination DB instance for the duration of the transport.

When transport begins, all current sessions on the source database are ended. Any databases other than the source database on the source DB instance aren't affected by the transport.

The source database is put into a special read-only mode. While it's in this mode, you can connect to the source database and run read-only queries. However, write-enabled queries and some other types of commands are blocked. Only the specific source database that is being transported is affected by these restrictions.

During transport, you can't restore the destination DB instance to a point in time. This is because the transport isn't transactional and doesn't use the PostgreSQL write-ahead log to record changes. If the destination DB instance has automatic backups enabled, a backup is automatically taken after transport completes. Point-in-time restores are available for times after the backup finishes.

If the transport fails, the `pg_transport` extension attempts to undo all changes to the source and destination DB instances. This includes removing the destination's partially transported database. Depending on the type of failure, the source database might continue to reject write-enabled queries. If this happens, use the following command to allow write-enabled queries.

```
ALTER DATABASE my-database SET default_transaction_read_only = false;
```

transport.import_from_server function reference

The `transport.import_from_server` function transports a PostgreSQL database by importing it from a source DB instance to a destination DB instance. It does this by using a physical database connection transport mechanism.

Syntax

```
transport.import_from_server(  
    host text,  
    port int,  
    username text,  
    password text,  
    database text,  
    local_password text,  
    dry_run bool  
)
```

Return Value

None.

Parameters

You can find descriptions of the `transport.import_from_server` function parameters in the following table.

Parameter	Description
<code>host</code>	The endpoint of the source DB instance.
<code>port</code>	An integer representing the port of the source DB instance. PostgreSQL DB instances often use port 5432.
<code>username</code>	The user of the source DB instance. This user must be a member of the <code>rds_superuser</code> role.
<code>password</code>	The user password of the source DB instance.
<code>database</code>	The name of the database in the source DB instance to transport.
<code>local_password</code>	The local password of the current user for the destination DB instance. This user must be a member of the <code>rds_superuser</code> role.
<code>dry_run</code>	An optional Boolean value specifying whether to perform a dry run. The default is <code>false</code> , which means the transport proceeds. To confirm compatibility between the source and destination DB instances without performing the actual transport, set <code>dry_run</code> to <code>true</code> .

Example

For an example, see [Transporting a PostgreSQL database using the `transport.import_from_server` function \(p. 1565\)](#).

Configuration parameters for the pg_transport extension

Use the following parameters to configure the `pg_transport` extension behavior.

```
SET pg_transport.num_workers = integer;
SET pg_transport.work_mem = kilobytes;
SET pg_transport.timing = Boolean;
```

You can find descriptions of these parameters in the following table.

Parameter	Description
<code>pg_transport.num_workers</code>	<p>The number of workers to use for a physical transport. The default is 3. Valid values are 1–32. Even large transports typically reach their maximum throughput with fewer than 8 workers.</p> <p>During transport, the <code>pg_transport.num_workers</code> setting on the destination DB instance is used on both the destination and source DB instances.</p> <p>A related parameter is the PostgreSQL <code>max_worker_processes</code> parameter. The transport process creates several background worker processes. Thus, your setting for the <code>pg_transport.num_workers</code> parameter might require you to set the <code>max_worker_processes</code> parameter significantly higher on both the source and destination DB instances.</p> <p>We recommend that you set <code>max_worker_processes</code> on both the source and destination DB instances to at least three times the destination DB instance's setting for the <code>pg_transport.num_workers</code> parameter. Add a few more to provide nontransport background worker processes.</p> <p>For more information about the <code>max_worker_processes</code> parameter, see the PostgreSQL documentation about Asynchronous behavior.</p>
<code>pg_transport.timing</code>	<p>A Boolean value that specifies whether to report timing information during the transport. The default is <code>true</code>. Valid values are <code>true</code> to report timing information and <code>false</code> to disable the reporting of timing information.</p> <p>We don't recommend that you set this parameter to <code>false</code>. Disabling <code>pg_transport.timing</code> significantly reduces your ability to track the progress of transports.</p>
<code>pg_transport.work_mem</code>	<p>The maximum amount of memory to allocate for each worker. The default is 131,072 kilobytes (KB). The minimum value is 64 megabytes (65,536 KB). Valid values are in kilobytes (KBs) as binary base-2 units, where 1 KB = 1,024 bytes.</p> <p>The transport might use less memory than is specified in this parameter. Even large transports typically reach their maximum throughput with less than 256 MB (262,144 KB) of memory per worker.</p>

Exporting data from an RDS for PostgreSQL DB instance to Amazon S3

You can query data from an RDS for PostgreSQL DB instance and export it directly into files stored in an Amazon S3 bucket. To do this, you use the `aws_s3` PostgreSQL extension that Amazon RDS provides.

For more information on storing data with Amazon S3, see [Create a bucket](#) in the *Amazon Simple Storage Service Getting Started Guide*.

Note

You can save DB snapshot data to Amazon S3 using the AWS Management Console, AWS CLI, or Amazon RDS API. For more information, see [Exporting DB snapshot data to Amazon S3 \(p. 373\)](#).

Topics

- [Overview of exporting data to Amazon S3 \(p. 1568\)](#)
- [Verify that your PostgreSQL version supports exports \(p. 1569\)](#)
- [Specifying the Amazon S3 file path to export to \(p. 1569\)](#)
- [Setting up access to an Amazon S3 bucket \(p. 1570\)](#)
- [Exporting query data using the `aws_s3.query_export_to_s3` function \(p. 1572\)](#)
- [Function reference \(p. 1574\)](#)

Overview of exporting data to Amazon S3

To export data stored in an RDS PostgreSQL database to an Amazon S3 bucket, use the following procedure.

To export Amazon RDS PostgreSQL data to S3

1. Ensure that your version of Amazon RDS PostgreSQL supports Amazon S3 exports. Currently, exports are supported for PostgreSQL 10.14, 11.9, 12.4 and later.
2. Install the required PostgreSQL extensions. These include the `aws_s3` and `aws_commons` extensions. To do so, start `psql` and use the following commands.

```
CREATE EXTENSION IF NOT EXISTS aws_s3 CASCADE;
```

The `aws_s3` extension provides the [aws_s3.query_export_to_s3 \(p. 1574\)](#) function that you use to export data to Amazon S3. The `aws_commons` extension is included to provide additional helper functions.

3. Identify an Amazon S3 file path to use for exporting data. For details about this process, see [Specifying the Amazon S3 file path to export to \(p. 1569\)](#).
4. Provide permission to access the Amazon S3 bucket.

To export data to an Amazon S3 file, give the RDS for PostgreSQL DB instance permission to access the Amazon S3 bucket that the export will use for storage. Doing this includes the following steps:

1. Create an IAM policy that provides access to an Amazon S3 bucket that you want to export to.
2. Create an IAM role.
3. Attach the policy you created to the role you created.
4. Add this IAM role to your DB instance.

For details about this process, see [Setting up access to an Amazon S3 bucket \(p. 1570\)](#).

5. Identify a database query to get the data. Export the query data by calling the `aws_s3.query_export_to_s3` function.

After you complete the preceding preparation tasks, use the `aws_s3.query_export_to_s3` (p. 1574) function to export query results to Amazon S3. For details about this process, see [Exporting query data using the aws_s3.query_export_to_s3 function \(p. 1572\)](#).

Verify that your PostgreSQL version supports exports

Currently, Amazon S3 exports are supported for PostgreSQL 10.14, 11.9, and 12.4 and later. You can also verify support by using the `describe-db-engine-versions` command. The following example verify support for version 10.14.

```
aws rds describe-db-engine-versions --region us-east-1 \
--engine postgres --engine-version 10.14 | grep s3Export
```

If the output includes the string "s3Export", then the engine supports Amazon S3 exports. Otherwise, the engine doesn't support them.

Specifying the Amazon S3 file path to export to

Specify the following information to identify the location in Amazon S3 where you want to export data to:

- Bucket name – A *bucket* is a container for Amazon S3 objects or files.

For more information on storing data with Amazon S3, see [Create a bucket](#) and [View an object](#) in the [Amazon Simple Storage Service Getting Started Guide](#).

- File path – The file path identifies where the export is stored in the Amazon S3 bucket. The file path consists of the following:
 - An optional path prefix that identifies a virtual folder path.
 - A file prefix that identifies one or more files to be stored. Larger exports are stored in multiple files, each with a maximum size of approximately 6 GB. The additional file names have the same file prefix but with `_partXX` appended. The `XX` represents 2, then 3, and so on.

For example, a file path with an `exports` folder and a `query-1-export` file prefix is `/exports/query-1-export`.

- AWS Region (optional) – The AWS Region where the Amazon S3 bucket is located. If you don't specify an AWS Region value, then Amazon RDS saves your files into Amazon S3 in the same AWS Region as the exporting DB instance.

Note

Currently, the AWS Region must be the same as the region of the exporting DB instance.

For a listing of AWS Region names and associated values, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

To hold the Amazon S3 file information about where the export is to be stored, you can use the `aws_commons.create_s3_uri` (p. 1576) function to create an `aws_commons._s3_uri_1` composite structure as follows.

```
| psql=> SELECT aws_commons.create_s3_uri(
```

```
'sample-bucket',
'samplefilepath',
'us-west-2'
) AS s3_uri_1 \gset
```

You later provide this `s3_uri_1` value as a parameter in the call to the [aws_s3.query_export_to_s3 \(p. 1574\)](#) function. For examples, see [Exporting query data using the aws_s3.query_export_to_s3 function \(p. 1572\)](#).

Setting up access to an Amazon S3 bucket

To export data to Amazon S3, give your PostgreSQL DB instance permission to access the Amazon S3 bucket that the files are to go in.

To do this, use the following procedure.

To give a PostgreSQL DB instance access to Amazon S3 through an IAM role

1. Create an IAM policy.

This policy provides the bucket and object permissions that allow your PostgreSQL DB instance to access Amazon S3.

As part of creating this policy, take the following steps:

- a. Include in the policy the following required actions to allow the transfer of files from your PostgreSQL DB instance to an Amazon S3 bucket:
 - `s3:PutObject`
 - `s3:AbortMultipartUpload`
- b. Include the Amazon Resource Name (ARN) that identifies the Amazon S3 bucket and objects in the bucket. The ARN format for accessing Amazon S3 is: `arn:aws:s3:::your-s3-bucket/*`

For more information on creating an IAM policy for Amazon RDS for PostgreSQL, see [Creating and using an IAM policy for IAM database access \(p. 1664\)](#). See also [Tutorial: Create and attach your first customer managed policy](#) in the *IAM User Guide*.

The following AWS CLI command creates an IAM policy named `rds-s3-export-policy` with these options. It grants access to a bucket named `your-s3-bucket`.

Warning

We recommend that you set up your database within a private VPC that has endpoint policies configured for accessing specific buckets. For more information, see [Using endpoint policies for Amazon S3](#) in the Amazon VPC User Guide.

We strongly recommend that you do not create a policy with all-resource access. This access can pose a threat for data security. If you create a policy that gives `S3:PutObject` access to all resources using "`Resource": "*"`, then a user with export privileges can export data to all buckets in your account. In addition, the user can export data to *any publicly writable bucket within your AWS Region*.

After you create the policy, note the Amazon Resource Name (ARN) of the policy. You need the ARN for a subsequent step when you attach the policy to an IAM role.

```
aws iam create-policy --policy-name rds-s3-export-policy --policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
        {
```

```
        "Sid": "s3export",
        "Action": [
            "S3:PutObject"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::your-s3-bucket/*"
        ]
    }
}'
```

2. Create an IAM role.

You do this so Amazon RDS can assume this IAM role on your behalf to access your Amazon S3 buckets. For more information, see [Creating a role to delegate permissions to an IAM user](#) in the *IAM User Guide*.

The following example shows using the AWS CLI command to create a role named `rds-s3-export-role`.

```
aws iam create-role --role-name rds-s3-export-role --assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}'
```

3. Attach the IAM policy that you created to the IAM role that you created.

The following AWS CLI command attaches the policy created earlier to the role named `rds-s3-export-role`. Replace `your-policy-arn` with the policy ARN that you noted in an earlier step.

```
aws iam attach-role-policy --policy-arn your-policy-arn --role-name rds-s3-export-role
```

4. Add the IAM role to the DB instance. You do so by using the AWS Management Console or AWS CLI, as described following.

Console

To add an IAM role for a PostgreSQL DB instance using the console

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose the PostgreSQL DB instance name to display its details.
3. On the **Connectivity & security** tab, in the **Manage IAM roles** section, choose the role to add under **Add IAM roles to this instance**.
4. Under **Feature**, choose **s3Export**.
5. Choose **Add role**.

AWS CLI

To add an IAM role for a PostgreSQL DB instance using the CLI

- Use the following command to add the role to the PostgreSQL DB instance named `my-db-instance`. Replace `your-role-arn` with the role ARN that you noted in a previous step. Use `s3Export` for the value of the `--feature-name` option.

Example

For Linux, macOS, or Unix:

```
aws rds add-role-to-db-instance \  
  --db-instance-identifier my-db-instance \  
  --feature-name s3Export \  
  --role-arn your-role-arn \  
  --region your-region
```

For Windows:

```
aws rds add-role-to-db-instance ^  
  --db-instance-identifier my-db-instance ^  
  --feature-name s3Export ^  
  --role-arn your-role-arn ^  
  --region your-region
```

Exporting query data using the `aws_s3.query_export_to_s3` function

Export your PostgreSQL data to Amazon S3 by calling the [aws_s3.query_export_to_s3](#) (p. 1574) function.

Topics

- [Prerequisites](#) (p. 1572)
- [Calling aws_s3.query_export_to_s3](#) (p. 1573)
- [Exporting to a CSV file that uses a custom delimiter](#) (p. 1574)
- [Exporting to a binary file with encoding](#) (p. 1574)
- [Troubleshooting access to Amazon S3](#) (p. 1574)

Prerequisites

Before you use the `aws_s3.query_export_to_s3` function, be sure to complete the following prerequisites:

- Install the required PostgreSQL extensions as described in [Overview of exporting data to Amazon S3](#) (p. 1568).
- Determine where to export your data to Amazon S3 as described in [Specifying the Amazon S3 file path to export to](#) (p. 1569).
- Make sure that the DB instance has export access to Amazon S3 as described in [Setting up access to an Amazon S3 bucket](#) (p. 1570).

The examples following use a database table called sample_table. These examples export the data into a bucket called sample-bucket. The example table and data are created with the following SQL statements in psql.

```
psql=> CREATE TABLE sample_table (bid bigint PRIMARY KEY, name varchar(80));
psql=> INSERT INTO sample_table (bid, name) VALUES (1, 'Monday'), (2, 'Tuesday'), (3, 'Wednesday');
```

Calling aws_s3.query_export_to_s3

The following shows the basic ways of calling the [aws_s3.query_export_to_s3 \(p. 1574\)](#) function.

These examples use the variable s3_uri_1 to identify a structure that contains the information identifying the Amazon S3 file. Use the [aws_commons.create_s3_uri \(p. 1576\)](#) function to create the structure.

```
psql=> SELECT aws_commons.create_s3_uri(
    'sample-bucket',
    'samplefilepath',
    'us-west-2'
) AS s3_uri_1 \gset
```

Although the parameters vary for the following two aws_s3.query_export_to_s3 function calls, the results are the same for these examples. All rows of the sample_table table are exported into a bucket called sample-bucket.

```
psql=> SELECT * FROM aws_s3.query_export_to_s3('SELECT * FROM sample_table', :'s3_uri_1');
psql=> SELECT * FROM aws_s3.query_export_to_s3('SELECT * FROM sample_table', :'s3_uri_1',
    options := 'format text');
```

The parameters are described as follows:

- 'SELECT * FROM sample_table' – The first parameter is a required text string containing an SQL query. The PostgreSQL engine runs this query. The results of the query are copied to the S3 bucket identified in other parameters.
- ':s3_uri_1' – This parameter is a structure that identifies the Amazon S3 file. This example uses a variable to identify the previously created structure. You can instead create the structure by including the `aws_commons.create_s3_uri` function call inline within the `aws_s3.query_export_to_s3` function call as follows.

```
SELECT * from aws_s3.query_export_to_s3('select * from sample_table',
    aws_commons.create_s3_uri('sample-bucket', 'samplefilepath', 'us-west-2')
);
```

- `options := 'format text'` – The `options` parameter is an optional text string containing PostgreSQL COPY arguments. The copy process uses the arguments and format of the [PostgreSQL COPY command](#).

If the file specified doesn't exist in the Amazon S3 bucket, it's created. If the file already exists, it's overwritten. The syntax for accessing the exported data in Amazon S3 is the following.

```
s3-region://bucket-name[/path-prefix]/file-prefix
```

Larger exports are stored in multiple files, each with a maximum size of approximately 6 GB. The additional file names have the same file prefix but with `_partXX` appended. The `XX` represents 2, then 3, and so on. For example, suppose that you specify the path where you store data files as the following.

```
s3-us-west-2://my-bucket/my-prefix
```

If the export has to create three data files, the Amazon S3 bucket contains the following data files.

```
s3-us-west-2://my-bucket/my-prefix
s3-us-west-2://my-bucket/my-prefix_part2
s3-us-west-2://my-bucket/my-prefix_part3
```

For the full reference for this function and additional ways to call it, see [aws_s3.query_export_to_s3 \(p. 1574\)](#). For more about accessing files in Amazon S3, see [View an object in the Amazon Simple Storage Service Getting Started Guide](#).

Exporting to a CSV file that uses a custom delimiter

The following example shows how to call the [aws_s3.query_export_to_s3 \(p. 1574\)](#) function to export data to a file that uses a custom delimiter. The example uses arguments of the [PostgreSQL COPY](#) command to specify the comma-separated value (CSV) format and a colon (:) delimiter.

```
SELECT * from aws_s3.query_export_to_s3('select * from basic_test', :'s3_uri_1',
options :='format csv, delimiter $$:$>');
```

Exporting to a binary file with encoding

The following example shows how to call the [aws_s3.query_export_to_s3 \(p. 1574\)](#) function to export data to a binary file that has Windows-1253 encoding.

```
SELECT * from aws_s3.query_export_to_s3('select * from basic_test', :'s3_uri_1',
options :='format binary, encoding WIN1253');
```

Troubleshooting access to Amazon S3

If you encounter connection problems when attempting to export data to Amazon S3, see the following for recommendations:

- [Troubleshooting Amazon RDS identity and access \(p. 1689\)](#)
- [Troubleshooting Amazon S3 in the Amazon Simple Storage Service Developer Guide](#).
- [Troubleshooting Amazon S3 and IAM in the IAM User Guide](#).

Function reference

Functions

- [aws_s3.query_export_to_s3 \(p. 1574\)](#)
- [aws_commons.create_s3_uri \(p. 1576\)](#)

aws_s3.query_export_to_s3

Exports a PostgreSQL query result to an Amazon S3 bucket. The `aws_s3` extension provides the `aws_s3.query_export_to_s3` function.

The two required parameters are `query` and `s3_info`. These define the query to be exported and identify the Amazon S3 bucket to export to. An optional parameter called `options` provides for defining various export parameters. For examples of using the `aws_s3.query_export_to_s3` function, see [Exporting query data using the aws_s3.query_export_to_s3 function \(p. 1572\)](#).

Syntax

```
aws_s3.query_export_to_s3(  
    query text,  
    s3_info aws_commons._s3_uri_1,  
    options text  
)
```

Input parameters

`query`

A required text string containing an SQL query that the PostgreSQL engine runs. The results of this query are copied to an S3 bucket identified in the `s3_info` parameter.

`s3_info`

An `aws_commons._s3_uri_1` composite type containing the following information about the S3 object:

- `bucket` – The name of the Amazon S3 bucket to contain the file.
- `file_path` – The Amazon S3 file name and path.
- `region` – The AWS Region that the bucket is in. For a listing of AWS Region names and associated values, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

Currently, this value must be the same AWS Region as that of the exporting DB instance. The default is the AWS Region of the exporting DB instance.

To create an `aws_commons._s3_uri_1` composite structure, see the [aws_commons.create_s3_uri \(p. 1576\)](#) function.

`options`

An optional text string containing arguments for the PostgreSQL `COPY` command. These arguments specify how the data is to be copied when exported. For more details, see the [PostgreSQL COPY documentation](#).

Alternate input parameters

To help with testing, you can use an expanded set of parameters instead of the `s3_info` parameter. Following are additional syntax variations for the `aws_s3.query_export_to_s3` function.

Instead of using the `s3_info` parameter to identify an Amazon S3 file, use the combination of the `bucket`, `file_path`, and `region` parameters.

```
aws_s3.query_export_to_s3(  
    query text,  
    bucket text,  
    file_path text,  
    region text,  
    options text  
)
```

query

A required text string containing an SQL query that the PostgreSQL engine runs. The results of this query are copied to an S3 bucket identified in the `s3_info` parameter.

bucket

A required text string containing the name of the Amazon S3 bucket that contains the file.

file_path

A required text string containing the Amazon S3 file name including the path of the file.

region

An optional text string containing the AWS Region that the bucket is in. For a listing of AWS Region names and associated values, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

Currently, this value must be the same AWS Region as that of the exporting DB instance. The default is the AWS Region of the exporting DB instance.

options

An optional text string containing arguments for the PostgreSQL `COPY` command. These arguments specify how the data is to be copied when exported. For more details, see the [PostgreSQL COPY documentation](#).

Output parameters

```
aws_s3.query_export_to_s3(
    OUT rows_uploaded bigint,
    OUT files_uploaded bigint,
    OUT bytes_uploaded bigint
)
```

rows_uploaded

The number of table rows that were successfully uploaded to Amazon S3 for the given query.

files_uploaded

The number of files uploaded to Amazon S3. Files are created in sizes of approximately 6 GB. Each additional file created has `_partXX` appended to the name. The `XX` represents 2, then 3, and so on as needed.

bytes_uploaded

The total number of bytes uploaded to Amazon S3.

Examples

```
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'sample-
bucket', 'sample-filepath');
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'sample-
bucket', 'sample-filepath','us-west-2');
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'sample-
bucket', 'sample-filepath','us-west-2','format text');
```

aws_commons.create_s3_uri

Creates an `aws_commons._s3_uri_1` structure to hold Amazon S3 file information. You use the results of the `aws_commons.create_s3_uri` function in the `s3_info`

parameter of the [aws_s3.query_export_to_s3 \(p. 1574\)](#) function. For an example of using the `aws_commons.create_s3_uri` function, see [Specifying the Amazon S3 file path to export to \(p. 1569\)](#).

Syntax

```
aws_commons.create_s3_uri(  
    bucket text,  
    file_path text,  
    region text  
)
```

Input parameters

bucket

A required text string containing the Amazon S3 bucket name for the file.

file_path

A required text string containing the Amazon S3 file name including the path of the file.

region

A required text string containing the AWS Region that the file is in. For a listing of AWS Region names and associated values, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

Common DBA tasks for PostgreSQL

This section describes the Amazon RDS implementations of some common DBA tasks for DB instances running the PostgreSQL database engine. To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances, and it restricts access to certain system procedures and tables that require advanced privileges.

For information about working with PostgreSQL log files on Amazon RDS, see [PostgreSQL database log files \(p. 534\)](#).

Topics

- [Creating roles \(p. 1578\)](#)
- [Managing PostgreSQL database access \(p. 1579\)](#)
- [Working with PostgreSQL parameters \(p. 1579\)](#)
- [Audit logging for a PostgreSQL DB instance \(p. 1588\)](#)
- [Working with the pgaudit extension \(p. 1588\)](#)
- [Working with the pg_repack extension \(p. 1590\)](#)
- [Using pgBadger for log analysis with PostgreSQL \(p. 1590\)](#)
- [Viewing the contents of pg_config \(p. 1590\)](#)
- [Working with the orafce extension \(p. 1591\)](#)
- [Accessing external data with the postgres_fdw extension \(p. 1592\)](#)
- [Restricting password management \(p. 1593\)](#)
- [Working with PostgreSQL autovacuum on Amazon RDS \(p. 1593\)](#)
- [Working with the PostGIS extension \(p. 1602\)](#)
- [Using a custom DNS server for outbound network access \(p. 1605\)](#)
- [Scheduling maintenance with the PostgreSQL pg_cron extension \(p. 1607\)](#)
- [Managing PostgreSQL partitions with the pg_partman extension \(p. 1614\)](#)
- [Invoking an AWS Lambda function from an RDS for PostgreSQL DB instance \(p. 1618\)](#)

Creating roles

When you create a DB instance, the master user system account that you create is assigned to the `rds_superuser` role. The `rds_superuser` role is a predefined Amazon RDS role similar to the PostgreSQL superuser role (customarily named `postgres` in local instances), but with some restrictions. As with the PostgreSQL superuser role, the `rds_superuser` role has the most privileges for your DB instance. You should not assign this role to users unless they need the most access to the DB instance.

The `rds_superuser` role can do the following:

- Add extensions that are available for use with Amazon RDS. For more information, see [Some supported PostgreSQL features \(p. 1499\)](#) and the [PostgreSQL documentation](#).
- Manage tablespaces, including creating and deleting them. For more information, see [Tablespaces for PostgreSQL on Amazon RDS \(p. 1505\)](#) and the [Tablespaces](#) section in the PostgreSQL documentation.
- View all users not assigned the `rds_superuser` role using the `pg_stat_activity` command and stop their connections using the `pg_terminate_backend` and `pg_cancel_backend` commands.
- Grant and revoke the `rds_replication` role for all roles that are not the `rds_superuser` role. For more information, see the [GRANT](#) section in the PostgreSQL documentation.

The following example shows how to create a user and then grant the user the `rds_superuser` role. User-defined roles, such as `rds_superuser`, have to be granted.

```
create role testuser with password 'testuser' login;
grant rds_superuser to testuser;
```

Managing PostgreSQL database access

In Amazon RDS for PostgreSQL, you can manage which users have privileges to connect to which databases. In other PostgreSQL environments, you sometimes perform this kind of management by modifying the `pg_hba.conf` file. In Amazon RDS, you can use database grants instead.

New databases in PostgreSQL are always created with a default set of privileges. The default privileges allow `PUBLIC` (all users) to connect to the database and to create temporary tables while connected.

To control which users are allowed to connect to a given database in Amazon RDS, first revoke the default `PUBLIC` privileges. Then grant back the privileges on a more granular basis. The following example code shows how.

```
psql> revoke all on database <database-name> from public;
psql> grant connect, temporary on database <database-name> to <user/role name>;
```

For more information about privileges in PostgreSQL databases, see the [GRANT command](#) in the PostgreSQL documentation.

Working with PostgreSQL parameters

PostgreSQL parameters that you set for a local PostgreSQL instance in the `postgresql.conf` file are maintained in the DB parameter group for your DB instance. If you create a DB instance using the default parameter group, the parameter settings are in the parameter group called `default.postgres9.6`.

When you create a DB instance, the parameters in the associated DB parameter group are loaded. You can modify parameter values by changing values in the parameter group. You can also change parameter values, if you have the security privileges to do so, by using the `ALTER DATABASE`, `ALTER ROLE`, and `SET` commands. You can't use the command line `postgres` command or the `env PGOPTIONS` command, because you have no access to the host.

Keeping track of PostgreSQL parameter settings can occasionally be difficult. Use the following command to list current parameter settings and the default value.

```
select name, setting, boot_val, reset_val, unit
from pg_settings
order by name;
```

For an explanation of the output values, see the [pg_settings topic](#) in the PostgreSQL documentation.

If you set the memory settings too large for `max_connections` or `shared_buffers`, you will prevent the PostgreSQL instance from starting up. Some parameters use units that you might not be familiar with; for example, `shared_buffers` sets the number of 8-KB shared memory buffers used by the server.

The following error is written to the `postgres.log` file when the instance is attempting to start up, but incorrect parameter settings are preventing it from starting.

```
2013-09-18 21:13:15 UTC::@[8097]:FATAL:  could not map anonymous shared
memory: Cannot allocate memory
2013-09-18 21:13:15 UTC::@[8097]:HINT:  This error usually means that
PostgreSQL's request for a shared memory segment exceeded available memory or
swap space. To reduce the request size (currently 3514134274048 bytes), reduce
PostgreSQL's shared memory usage, perhaps by reducing shared_buffers or
```

`max_connections.`

There are two types of PostgreSQL parameters, static and dynamic. Static parameters require that the DB instance be rebooted before they are applied. Dynamic parameters can be applied immediately. The following table shows parameters that you can modify for a PostgreSQL DB instance and each parameter's type.

Parameter name	Apply_Type	Description
<code>application_name</code>	Dynamic	Sets the application name to be reported in statistics and logs.
<code>array_nulls</code>	Dynamic	Enables input of NULL elements in arrays.
<code>authentication_timeout</code>	Dynamic	Sets the maximum allowed time to complete client authentication.
<code>autovacuum</code>	Dynamic	Starts the autovacuum subprocess.
<code>autovacuum_analyze_scale_factor</code>	Dynamic	Number of tuple inserts, updates, or deletes before analyze as a fraction of reltuples.
<code>autovacuum_analyze_threshold</code>	Dynamic	Minimum number of tuple inserts, updates, or deletes before analyze.
<code>autovacuum_naptime</code>	Dynamic	Time to sleep between autovacuum runs.
<code>autovacuum_vacuum_cost_delay</code>	Dynamic	Vacuum cost delay, in milliseconds, for autovacuum.
<code>autovacuum_vacuum_cost_limit</code>	Dynamic	Vacuum cost amount available before napping, for autovacuum.
<code>autovacuum_vacuum_scale_factor</code>	Dynamic	Number of tuple updates or deletes before vacuum as a fraction of reltuples.
<code>autovacuum_vacuum_threshold</code>	Dynamic	Minimum number of tuple updates or deletes before vacuum.
<code>backslash_quote</code>	Dynamic	Sets whether a backslash (\) is allowed in string literals.
<code>bgwriter_delay</code>	Dynamic	Background writer sleep time between rounds.
<code>bgwriter_lru_maxpages</code>	Dynamic	Background writer maximum number of LRU pages to flush per round.
<code>bgwriter_lru_multiplier</code>	Dynamic	Multiple of the average buffer usage to free per round.
<code>bytea_output</code>	Dynamic	Sets the output format for bytes.
<code>check_function_bodies</code>	Dynamic	Checks function bodies during CREATE FUNCTION.
<code>checkpoint_completion_target</code>	Dynamic	Time spent flushing dirty buffers during checkpoint, as a fraction of the checkpoint interval.
<code>checkpoint_segments</code>	Dynamic	Sets the maximum distance in log segments between automatic write-ahead log (WAL) checkpoints.
<code>checkpoint_timeout</code>	Dynamic	Sets the maximum time between automatic WAL checkpoints.

Parameter name	Apply_Type	Description
checkpoint_warning	Dynamic	Enables warnings if checkpoint segments are filled more frequently than this.
client_encoding	Dynamic	Sets the client's character set encoding.
client_min_messages	Dynamic	Sets the message levels that are sent to the client.
commit_delay	Dynamic	Sets the delay in microseconds between transaction commit and flushing WAL to disk.
commit_siblings	Dynamic	Sets the minimum concurrent open transactions before performing commit_delay.
constraint_exclusion	Dynamic	Enables the planner to use constraints to optimize queries.
cpu_index_tuple_cost	Dynamic	Sets the planner's estimate of the cost of processing each index entry during an index scan.
cpu_operator_cost	Dynamic	Sets the planner's estimate of the cost of processing each operator or function call.
cpu_tuple_cost	Dynamic	Sets the planner's estimate of the cost of processing each tuple (row).
cursor_tuple_fraction	Dynamic	Sets the planner's estimate of the fraction of a cursor's rows that will be retrieved.
datestyle	Dynamic	Sets the display format for date and time values.
deadlock_timeout	Dynamic	Sets the time to wait on a lock before checking for deadlock.
debug_pretty_print	Dynamic	Indents parse and plan tree displays.
debug_print_parse	Dynamic	Logs each query's parse tree.
debug_print_plan	Dynamic	Logs each query's execution plan.
debug_print_rewritten	Dynamic	Logs each query's rewritten parse tree.
default_statistics_target	Dynamic	Sets the default statistics target.
default_tablespace	Dynamic	Sets the default tablespace to create tables and indexes in.
default_transaction_deferrable	Dynamic	Sets the default deferrable status of new transactions.
default_transaction_isolation	Dynamic	Sets the transaction isolation level of each new transaction.
default_transaction_read_only	Dynamic	Sets the default read-only status of new transactions.
default_with_oids	Dynamic	Creates new tables with OIDs by default.
effective_cache_size	Dynamic	Sets the planner's assumption about the size of the disk cache.

Parameter name	Apply_Type	Description
<code>effective_io_concurrency</code>	Dynamic	Number of simultaneous requests that can be handled efficiently by the disk subsystem.
<code>enable_bitmapscan</code>	Dynamic	Enables the planner's use of bitmap-scan plans.
<code>enable_hashagg</code>	Dynamic	Enables the planner's use of hashed aggregation plans.
<code>enable_hashjoin</code>	Dynamic	Enables the planner's use of hash join plans.
<code>enable_indexscan</code>	Dynamic	Enables the planner's use of index-scan plans.
<code>enable_material</code>	Dynamic	Enables the planner's use of materialization.
<code>enable_mergejoin</code>	Dynamic	Enables the planner's use of merge join plans.
<code>enable_nestloop</code>	Dynamic	Enables the planner's use of nested-loop join plans.
<code>enable_seqscan</code>	Dynamic	Enables the planner's use of sequential-scan plans.
<code>enable_sort</code>	Dynamic	Enables the planner's use of explicit sort steps.
<code>enable_tidscan</code>	Dynamic	Enables the planner's use of TID scan plans.
<code>escape_string_warning</code>	Dynamic	Warns about backslash (\) escapes in ordinary string literals.
<code>extra_float_digits</code>	Dynamic	Sets the number of digits displayed for floating-point values.
<code>fromCollapse_limit</code>	Dynamic	Sets the FROM-list size beyond which subqueries are not collapsed.
<code>fsync</code>	Dynamic	Forces synchronization of updates to disk.
<code>full_page_writes</code>	Dynamic	Writes full pages to WAL when first modified after a checkpoint.
<code>geqo</code>	Dynamic	Enables genetic query optimization.
<code>geqo_effort</code>	Dynamic	GEQO: effort is used to set the default for other GEQO parameters.
<code>geqo_generations</code>	Dynamic	GEQO: number of iterations of the algorithm.
<code>geqo_pool_size</code>	Dynamic	GEQO: number of individuals in the population.
<code>geqo_seed</code>	Dynamic	GEQO: seed for random path selection.
<code>geqo_selection_bias</code>	Dynamic	GEQO: selective pressure within the population.
<code>geqo_threshold</code>	Dynamic	Sets the threshold of FROM items beyond which GEQO is used.
<code>gin_fuzzy_search_limit</code>	Dynamic	Sets the maximum allowed result for exact search by GIN.
<code>hot_standby_feedback</code>	Dynamic	Determines whether a hot standby sends feedback messages to the primary or upstream standby.

Parameter name	Apply_Type	Description
intervalstyle	Dynamic	Sets the display format for interval values.
joinCollapse_limit	Dynamic	Sets the FROM-list size beyond which JOIN constructs are not flattened.
lc_messages	Dynamic	Sets the language in which messages are displayed.
lc_monetary	Dynamic	Sets the locale for formatting monetary amounts.
lc_numeric	Dynamic	Sets the locale for formatting numbers.
lc_time	Dynamic	Sets the locale for formatting date and time values.
log_autovacuum_min_duration	Dynamic	Sets the minimum running time above which autovacuum actions will be logged.
log_checkpoints	Dynamic	Logs each checkpoint.
log_connections	Dynamic	Logs each successful connection.
log_disconnections	Dynamic	Logs end of a session, including duration.
log_duration	Dynamic	Logs the duration of each completed SQL statement.
log_error_verbosity	Dynamic	Sets the verbosity of logged messages.
log_executor_stats	Dynamic	Writes executor performance statistics to the server log.
log_filename	Dynamic	Sets the file name pattern for log files.
log_hostname	Dynamic	Logs the host name in the connection logs.
log_lock_waits	Dynamic	Logs long lock waits.
log_min_duration_statement	Dynamic	Sets the minimum running time above which statements will be logged.
log_min_error_statement	Dynamic	Causes all statements generating an error at or above this level to be logged.
log_min_messages	Dynamic	Sets the message levels that are logged.
log_parser_stats	Dynamic	Writes parser performance statistics to the server log.
log_planner_stats	Dynamic	Writes planner performance statistics to the server log.
log_rotation_age	Dynamic	Automatic log file rotation will occur after N minutes.
log_rotation_size	Dynamic	Automatic log file rotation will occur after N kilobytes.
log_statement	Dynamic	Sets the type of statements logged.
log_statement_stats	Dynamic	Writes cumulative performance statistics to the server log.
log_temp_files	Dynamic	Logs the use of temporary files larger than this number of kilobytes.

Parameter name	Apply_Type	Description
<code>maintenance_work_mem</code>	Dynamic	Sets the maximum memory to be used for maintenance operations.
<code>max_stack_depth</code>	Dynamic	Sets the maximum stack depth, in kilobytes.
<code>max_standby_archive_delay</code>	Dynamic	Sets the maximum delay before canceling queries when a hot standby server is processing archived WAL data.
<code>max_standby_streaming_delay</code>	Dynamic	Sets the maximum delay before canceling queries when a hot standby server is processing streamed WAL data.
<code>max_wal_size</code>	Static	Sets the WAL size that triggers the checkpoint. For PostgreSQL version 9.6 and earlier, <code>max_wal_size</code> is in units of 16 MB. For PostgreSQL version 10 and later, <code>max_wal_size</code> is in units of 1 MB.
<code>min_wal_size</code>	Static	Sets the minimum size to shrink the WAL to. For PostgreSQL version 9.6 and earlier, <code>min_wal_size</code> is in units of 16 MB. For PostgreSQL version 10 and later, <code>min_wal_size</code> is in units of 1 MB.
<code>quote_all_identifiers</code>	Dynamic	Adds quotes ("") to all identifiers when generating SQL fragments.
<code>random_page_cost</code>	Dynamic	Sets the planner's estimate of the cost of a non-sequentially fetched disk page.
<code>rds.adaptive_autovacuum</code>	Dynamic	Automatically tunes the autovacuum parameters whenever the transaction ID thresholds are exceeded.
<code>rds.log_retention_period</code>	Dynamic	Sets log retention such that Amazon RDS deletes PostgreSQL logs that are older than N minutes.
<code>rds.restrict_password_commands</code>	Static	Restricts who can manage passwords to users with the <code>rds_password</code> role. Set this parameter to 1 to enable password restriction. The default is 0.
<code>search_path</code>	Dynamic	Sets the schema search order for names that are not schema-qualified.
<code>seq_page_cost</code>	Dynamic	Sets the planner's estimate of the cost of a sequentially fetched disk page.
<code>session_replication_role</code>	Dynamic	Sets the sessions behavior for triggers and rewrite rules.
<code>sql_inheritance</code>	Dynamic	Causes subtables to be included by default in various commands.
<code>ssl_renegotiation_limit</code>	Dynamic	Sets the amount of traffic to send and receive before renegotiating the encryption keys.
<code>standard_conforming_strings</code>	Dynamic	Causes ... strings to treat backslashes literally.
<code>statement_timeout</code>	Dynamic	Sets the maximum allowed duration of any statement.

Parameter name	Apply_Type	Description
<code>synchronize_seqscans</code>	Dynamic	Enables synchronized sequential scans.
<code>synchronous_commit</code>	Dynamic	Sets the current transactions synchronization level.
<code>tcp_keepalives_count</code>	Dynamic	Maximum number of TCP keepalive retransmits.
<code>tcp_keepalives_idle</code>	Dynamic	Time between issuing TCP keepalives.
<code>tcp_keepalives_interval</code>	Dynamic	Time between TCP keepalive retransmits.
<code>temp_buffers</code>	Dynamic	Sets the maximum number of temporary buffers used by each session.
<code>temp_tablespaces</code>	Dynamic	Sets the tablespaces to use for temporary tables and sort files.
<code>timezone</code>	Dynamic	Sets the time zone for displaying and interpreting time stamps.
<code>track_activities</code>	Dynamic	Collects information about running commands.
<code>track_counts</code>	Dynamic	Collects statistics on database activity.
<code>track_functions</code>	Dynamic	Collects function-level statistics on database activity.
<code>track_io_timing</code>	Dynamic	Collects timing statistics on database I/O activity.
<code>transaction_deferrable</code>	Dynamic	Indicates whether to defer a read-only serializable transaction until it can be started with no possible serialization failures.
<code>transaction_isolation</code>	Dynamic	Sets the current transactions isolation level.
<code>transaction_read_only</code>	Dynamic	Sets the current transactions read-only status.
<code>transform_null_equals</code>	Dynamic	Treats <code>expr=NULL</code> as <code>expr IS NULL</code> .
<code>update_process_title</code>	Dynamic	Updates the process title to show the active SQL command.
<code>vacuum_cost_delay</code>	Dynamic	Vacuum cost delay in milliseconds.
<code>vacuum_cost_limit</code>	Dynamic	Vacuum cost amount available before napping.
<code>vacuum_cost_page_dirty</code>	Dynamic	Vacuum cost for a page dirtied by vacuum.
<code>vacuum_cost_page_hit</code>	Dynamic	Vacuum cost for a page found in the buffer cache.
<code>vacuum_cost_page_miss</code>	Dynamic	Vacuum cost for a page not found in the buffer cache.
<code>vacuum_defer_cleanup_age</code>	Dynamic	Number of transactions by which vacuum and hot cleanup should be deferred, if any.
<code>vacuum_freeze_min_age</code>	Dynamic	Minimum age at which vacuum should freeze a table row.
<code>vacuum_freeze_table_age</code>	Dynamic	Age at which vacuum should scan a whole table to freeze tuples.
<code>wal_writer_delay</code>	Dynamic	WAL writer sleep time between WAL flushes.

Parameter name	Apply_Type	Description
<code>work_mem</code>	Dynamic	Sets the maximum memory to be used for query workspaces.
<code>xmlbinary</code>	Dynamic	Sets how binary values are to be encoded in XML.
<code>xmloption</code>	Dynamic	Sets whether XML data in implicit parsing and serialization operations is to be considered as documents or content fragments.
<code>autovacuum_freeze_max_age</code>	Static	Age at which to autovacuum a table to prevent transaction ID wraparound.
<code>autovacuum_max_workers</code>	Static	Sets the maximum number of simultaneously running autovacuum worker processes.
<code>max_connections</code>	Static	Sets the maximum number of concurrent connections.
<code>max_files_per_process</code>	Static	Sets the maximum number of simultaneously open files for each server process.
<code>max_locks_per_transaction</code>	Static	Sets the maximum number of locks per transaction.
<code>max_pred_locks_per_transaction</code>	Static	Sets the maximum number of predicate locks per transaction.
<code>max_prepared_transactions</code>	Static	Sets the maximum number of simultaneously prepared transactions.
<code>shared_buffers</code>	Static	Sets the number of shared memory buffers used by the server.
<code>ssl</code>	Static	Enables SSL connections.
<code>temp_file_limit</code>	Static	Sets the maximum size in KB to which the temporary files can grow.
<code>track_activity_query_size</code>	Static	Sets the size reserved for <code>pg_stat_activity.current_query</code> , in bytes.
<code>wal_buffers</code>	Static	Sets the number of disk-page buffers in shared memory for WAL.

Amazon RDS uses the default PostgreSQL units for all parameters. The following table shows the PostgreSQL default unit and value for each parameter.

Parameter name	Unit
<code>effective_cache_size</code>	8 KB
<code>segment_size</code>	8 KB
<code>shared_buffers</code>	8 KB
<code>temp_buffers</code>	8 KB
<code>wal_buffers</code>	8 KB

Parameter name	Unit
wal_segment_size	8 KB
log_rotation_size	KB
log_temp_files	KB
maintenance_work_mem	KB
max_stack_depth	KB
ssl_renegotiation_limit	KB
temp_file_limit	KB
work_mem	KB
log_rotation_age	minutes
autovacuum_vacuum_cost_delay	ms
bgwriter_delay	ms
deadlock_timeout	ms
lock_timeout	ms
log_autovacuum_min_duration	ms
log_min_duration_statement	ms
max_standby_archive_delay	ms
max_standby_streaming_delay	ms
statement_timeout	ms
vacuum_cost_delay	ms
wal_receiver_timeout	ms
wal_sender_timeout	ms
wal_writer_delay	ms
archive_timeout	s
authentication_timeout	s
autovacuum_naptime	s
checkpoint_timeout	s
checkpoint_warning	s
post_auth_delay	s
pre_auth_delay	s
tcp_keepalives_idle	s
tcp_keepalives_interval	s

Parameter name	Unit
wal_receiver_status_interval	s

Audit logging for a PostgreSQL DB instance

There are several parameters you can set to log activity that occurs on your PostgreSQL DB instance. These parameters include the following:

- The `log_statement` parameter can be used to log user activity in your PostgreSQL database. For more information, see [PostgreSQL database log files \(p. 534\)](#).
- The `rds.force_admin_logging_level` parameter logs actions by the RDS internal user (`rdsadmin`) in the databases on the DB instance, and writes the output to the PostgreSQL error log. Allowed values are disabled, debug5, debug4, debug3, debug2, debug1, info, notice, warning, error, log, fatal, and panic. The default value is disabled.
- The `rds.force_autovacuum_logging_level` parameter logs autovacuum worker operations in all databases on the DB instance, and writes the output to the PostgreSQL error log. Allowed values are disabled, debug5, debug4, debug3, debug2, debug1, info, notice, warning, error, log, fatal, and panic. The default value is disabled. The Amazon RDS recommended setting for `rds.force_autovacuum_logging_level`: is LOG. Set `log_autovacuum_min_duration` to a value from 1000 or 5000. Setting this value to 5,000 writes activity to the log that takes more than 5 seconds and shows "vacuum skipped" messages. For more information on this parameter, see [Best practices for working with PostgreSQL \(p. 137\)](#).

Working with the pgaudit extension

The `pgaudit` extension provides detailed session and object audit logging for Amazon RDS for PostgreSQL version 9.6.3 and later and version 9.5.7 version and later. You can enable session auditing or object auditing using this extension.

With session auditing, you can log audit events from various sources and includes the fully qualified command text when available. For example, you can use session auditing to log all READ statements that connect to a database by setting `pgaudit.log` to `READ`.

With object auditing, you can refine the audit logging to work with specific commands. For example, you can specify that you want audit logging for READ operations on a specific number of tables.

To use object based logging with the pgaudit extension

1. Create a database role called `rds_pgaudit` using the following command.

```
CREATE ROLE rds_pgaudit;
```

2. Modify the parameter group that is associated with your DB instance to do the following:

- Use the shared preload libraries that contain `pgaudit`.
- Set `pgaudit.role` to the role `rds_pgaudit`.

The following commands modify a custom parameter group.

```
aws rds modify-db-parameter-group \
--db-parameter-group-name rds-parameter-group-96 \
```

```
--parameters
"ParameterName=pgaudit.role,ParameterValue=rds_pgaudit,ApplyMethod=pending-reboot" \
--region us-west-2

aws rds modify-db-parameter-group \
--db-parameter-group-name rds-parameter-group-96 \
--parameters
"ParameterName=shared_preload_libraries,ParameterValue=pgaudit,ApplyMethod=pending-
reboot" \
--region us-west-2
```

3. Reboot the instance so that the DB instance picks up the changes to the parameter group.

```
aws rds reboot-db-instance \
--db-instance-identifier rds-test-instance \
--region us-west-2
```

4. Run the following command to confirm that pgaudit has been initialized.

```
SHOW shared_preload_libraries;

shared_preload_libraries
-----
rdsutils,pgaudit
(1 row)
```

5. Run the following command to create the pgaudit extension.

```
CREATE EXTENSION pgaudit;
```

6. Run the following command to confirm `pgaudit.role` is set to `rds_pgaudit`.

```
SHOW pgaudit.role;

pgaudit.role
-----
rds_pgaudit
```

To test the audit logging, run several commands that you have chosen to audit. For example, you might run the following commands.

```
CREATE TABLE t1 (id int);
GRANT SELECT ON t1 TO rds_pgaudit;
SELECT * FROM t1;

id
-----
(0 rows)
```

The database logs should contain an entry similar to the following.

```
...
2017-06-12 19:09:49 UTC:...:rds_test@postgres:[11701]:LOG: AUDIT:
OBJECT,1,1,READ,SELECT,TABLE,public.t1,select * from t1;
...
```

For information on viewing the logs, see [Accessing Amazon RDS database log files \(p. 504\)](#).

Working with the pg_repack extension

You can use the `pg_repack` extension to remove bloat from tables and indexes. This extension is supported on Amazon RDS for PostgreSQL versions 9.6.3 and later. For more information on the `pg_repack` extension, see the [GitHub project documentation](#).

To use the pg_repack extension

1. Install the `pg_repack` extension on your Amazon RDS for PostgreSQL DB instance by running the following command.

```
CREATE EXTENSION pg_repack;
```

2. Use the `pg_repack` client utility to connect to a database. Use a database role that has `rds_superuser` privileges to connect to the database. In the following connection example, the `rds_test` role has `rds_superuser` privileges, and the database endpoint used is `rds-test-instance.cw7jjfgdr4on8.us-west-2.rds.amazonaws.com`.

```
pg_repack -h rds-test-instance.cw7jjfgdr4on8.us-west-2.rds.amazonaws.com -U rds_test -k postgres
```

Connect using the `-k` option. The `-a` option is not supported.

3. The response from the `pg_repack` client provides information on the tables on the DB instance that are repacked.

```
INFO: repacking table "pgbench_tellers"  
INFO: repacking table "pgbench_accounts"  
INFO: repacking table "pgbench_branches"
```

Using pgBadger for log analysis with PostgreSQL

You can use a log analyzer such as `pgbadger` to analyze PostgreSQL logs. The `pgbadger` documentation states that the `%l` pattern (log line for session/process) should be a part of the prefix. However, if you provide the current rds log_line_prefix as a parameter to `pgbadger` it should still produce a report.

For example, the following command correctly formats an Amazon RDS for PostgreSQL log file dated 2014-02-04 using `pgbadger`.

```
./pgbadger -p '%t:%r:%u@%d:[%p]:' postgresql.log.2014-02-04-00
```

Viewing the contents of pg_config

In PostgreSQL version 9.6.1, you can see the compile-time configuration parameters of the currently installed version of PostgreSQL using the new view `pg_config`. You can use the view by calling the `pg_config` function as shown in the following sample.

```
select * from pg_config();
   name    |      setting
-----
+-----+
-----+
 BINDIR | /rdsdbbin/postgres-9.6.1.R1/bin
```

```

DOCDIR           | /rdsdbbin/postgres-9.6.1.R1/share/doc
HTMLDIR          | /rdsdbbin/postgres-9.6.1.R1/share/doc
INCLUDEDIR       | /rdsdbbin/postgres-9.6.1.R1/include
PKGINCLUDEDIR   | /rdsdbbin/postgres-9.6.1.R1/include
INCLUDEDIR-SERVER | /rdsdbbin/postgres-9.6.1.R1/include/server
LIBDIR           | /rdsdbbin/postgres-9.6.1.R1/lib
PKGLIBDIR        | /rdsdbbin/postgres-9.6.1.R1/lib
LOCALEDIR        | /rdsdbbin/postgres-9.6.1.R1/share/locale
MANDIR           | /rdsdbbin/postgres-9.6.1.R1/share/man
SHAREDIR         | /rdsdbbin/postgres-9.6.1.R1/share
SYSCONFDIR      | /rdsdbbin/postgres-9.6.1.R1/etc
PGXS             | /rdsdbbin/postgres-9.6.1.R1/lib/pgxs/src/makefiles/pgxs.mk
CONFIGURE        | '--prefix=/rdsdbbin/postgres-9.6.1.R1' '--with-openssl' '--with-perl'
                  '--with-tcl' '--with-ossp-uuid' '--with-libxml' '--with-libraries=/rdsdbbin
                  /postgres-9.6.1.R1/lib' '--with-includes=/rdsdbbin/postgres-9.6.1.R1/include' '--enable-
                  debug'
CC                | gcc
CPPFLAGS          | -D_GNU_SOURCE -I/usr/include/libxml2 -I/rdsdbbin/postgres-9.6.1.R1/
include
CFLAGS            | -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-
statement
-Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-
aliasing -fwrapv -fexcess-precision=standard -g -O2
CFLAGS_SL         | -fpic
LDFLAGS           | -L../../src/common -L/rdsdbbin/postgres-9.6.1.R1/lib -Wl,--as-needed -
Wl,
-rpath,'/rdsdbbin/postgres-9.6.1.R1/lib',--enable-new-dtags
LDFLAGS_EX        |
LDFLAGS_SL        |
LIBS              | -lpgcommon -lpgport -lxmll2 -lssl -lcrypto -lz -lreadline -lrt -lcrypt
-lldl -lm
VERSION           | PostgreSQL 9.6.1
(23 rows)

```

If you attempt to access the view directly, the request fails.

```

select * from pg_config;
ERROR: permission denied for relation pg_config

```

Working with the orafce extension

The `orafce` extension provides functions that are common in commercial databases, and can make it easier for you to port a commercial database to PostgreSQL. Amazon RDS for PostgreSQL versions 9.6.6 and later support this extension. For more information about `orafce`, see the [orafce project on GitHub](#).

Note

Amazon RDS for PostgreSQL doesn't support the `utl_file` package that is part of the `orafce` extension. This is because the `utl_file` schema functions provide read and write operations on operating-system text files, which requires superuser access to the underlying host.

To use the `orafce` extension

1. Connect to the DB instance with the master user name that you used to create the DB instance.

Note

If you want to enable `orafce` on a different database in the same instance, use the `/c dbname` `psql` command to change from the primary database after initiating the connection.

2. Enable the `orafce` extension with the `CREATE EXTENSION` statement.

```
CREATE EXTENSION orafce;
```

3. Transfer ownership of the oracle schema to the `rds_superuser` role with the `ALTER SCHEMA` statement.

```
ALTER SCHEMA oracle OWNER TO rds_superuser;
```

Note

If you want to see the list of owners for the oracle schema, use the `\dn` psql command.

Accessing external data with the `postgres_fdw` extension

You can access data in a table on a remote database server with the `postgres_fdw` extension. If you set up a remote connection from your PostgreSQL DB instance, access is also available to your read replica.

To use `postgres_fdw` to access a remote database server

1. Install the `postgres_fdw` extension.

```
CREATE EXTENSION postgres_fdw;
```

2. Create a foreign data server using `CREATE SERVER`.

```
CREATE SERVER foreign_server
FOREIGN DATA WRAPPER postgres_fdw
OPTIONS (host 'xxx.xx.xxx.xx', port '5432', dbname 'foreign_db');
```

3. Create a user mapping to identify the role to be used on the remote server.

```
CREATE USER MAPPING FOR local_user
SERVER foreign_server
OPTIONS (user 'foreign_user', password 'password');
```

4. Create a table that maps to the table on the remote server.

```
CREATE FOREIGN TABLE foreign_table (
    id integer NOT NULL,
    data text)
SERVER foreign_server
OPTIONS (schema_name 'some_schema', table_name 'some_table');
```

Restricting password management

You can restrict who can manage database user passwords to a special role. By doing this, you can have more control over password management on the client side.

You enable restricted password management with the static parameter `rds.restrict_password_commands` and use a role called `rds_password`. When the parameter `rds.restrict_password_commands` is set to 1, only users that are members of the `rds_password` role can run certain SQL commands. The restricted SQL commands are commands that modify database user passwords and password expiration time.

To use restricted password management, your DB instance must be running Amazon RDS for PostgreSQL 10.6 or higher. Because the `rds.restrict_password_commands` parameter is static, changing this parameter requires a database restart.

When a database has restricted password management enabled, if you try to run restricted SQL commands you get the following error: ERROR: must be a member of `rds_password` to alter passwords.

Following are some examples of SQL commands that are restricted when restricted password management is enabled.

```
postgres=> CREATE ROLE myrole WITH PASSWORD 'mypassword';
postgres=> CREATE ROLE myrole WITH PASSWORD 'mypassword' VALID UNTIL '2020-01-01';
postgres=> ALTER ROLE myrole WITH PASSWORD 'mypassword' VALID UNTIL '2020-01-01';
postgres=> ALTER ROLE myrole WITH PASSWORD 'mypassword';
postgres=> ALTER ROLE myrole VALID UNTIL '2020-01-01';
postgres=> ALTER ROLE myrole RENAME TO myrole2;
```

Some `ALTER ROLE` commands that include `RENAME TO` might also be restricted. They might be restricted because renaming a PostgreSQL role that has an MD5 password clears the password.

The `rds_superuser` role has membership for the `rds_password` role by default, and you can't change this. You can give other roles membership for the `rds_password` role by using the `GRANT` SQL command. We recommend that you give membership to `rds_password` to only a few roles that you use solely for password management. These roles require the `CREATEROLE` attribute to modify other roles.

Make sure that you verify password requirements such as expiration and needed complexity on the client side. We recommend that you restrict password-related changes by using your own client-side utility. This utility should have a role that is a member of `rds_password` and has the `CREATEROLE` role attribute.

Working with PostgreSQL autovacuum on Amazon RDS

We strongly recommend that you use the autovacuum feature for PostgreSQL databases to maintain the health of your PostgreSQL DB instance. Autovacuum automates the start of the `VACUUM` and the `ANALYZE` commands. Autovacuum checks for tables that have had a large number of inserted, updated, or deleted tuples. Autovacuum then reclaims storage by removing obsolete data or tuples from the PostgreSQL database.

Autovacuum is enabled by default for all new Amazon RDS for PostgreSQL DB instances, and the related autovacuum configuration parameters are appropriately set by default. Because our defaults are somewhat generic, you can benefit from tuning parameters to your specific workload. The following section can help you perform the needed autovacuum tuning.

Topics

- Allocating memory for autovacuum (p. 1594)
- Reducing the likelihood of transaction ID wraparound (p. 1594)
- Determining if the tables in your database need vacuuming (p. 1595)
- Determining which tables are currently eligible for autovacuum (p. 1596)
- Determining if autovacuum is currently running and for how long (p. 1597)
- Performing a manual vacuum freeze (p. 1598)
- Reindexing a table when autovacuum is running (p. 1600)
- Other parameters that affect autovacuum (p. 1600)
- Setting table-level autovacuum parameters (p. 1601)
- Autovacuum logging (p. 1601)

Allocating memory for autovacuum

One of the most important parameters influencing autovacuum performance is the `maintenance_work_mem` parameter. This parameter determines how much memory that you allocate for autovacuum to use to scan a database table and to hold all the row IDs that are going to be vacuumed. If you set the value of the `maintenance_work_mem` parameter too low, the vacuum process might have to scan the table multiple times to complete its work. Such multiple scans can have a negative impact on performance.

When doing calculations to determine the `maintenance_work_mem` parameter value, keep in mind two things:

- The default unit is kilobytes (KB) for this parameter.
- The `maintenance_work_mem` parameter works in conjunction with the `autovacuum_max_workers` parameter. If you have many small tables, allocate more `autovacuum_max_workers` and less `maintenance_work_mem`. If you have large tables (say, larger than 100 GB), allocate more memory and fewer worker processes. You need to have enough memory allocated to succeed on your biggest table. Each `autovacuum_max_workers` can use the memory you allocate. Thus, you should make sure the combination of worker processes and memory equal the total memory that you want to allocate.

In general terms, for large hosts set the `maintenance_work_mem` parameter to a value between one and two gigabytes (between 1,048,576 and 2,097,152 KB). For extremely large hosts, set the parameter to a value between two and four gigabytes (between 2,097,152 and 4,194,304 KB). The value you set for this parameter should depend on the workload. Amazon RDS has updated its default for this parameter to be kilobytes calculated as follows:

```
GREATEST({DBInstanceClassMemory/63963136*1024}, 65536).
```

Reducing the likelihood of transaction ID wraparound

In some cases, parameter group settings related to autovacuum might not be aggressive enough to prevent transaction ID wraparound. To address this, Amazon RDS for PostgreSQL provides a mechanism that adapts the autovacuum parameter values automatically. *Adaptive autovacuum parameter tuning* is a feature for RDS for PostgreSQL. A detailed explanation of `TransactionID wraparound` is found in the PostgreSQL documentation.

Adaptive autovacuum parameter tuning is enabled by default for RDS for PostgreSQL instances with the dynamic parameter `rds.adaptive_autovacuum` set to ON. We strongly recommend that you keep this enabled. However, to turn off adaptive autovacuum parameter tuning, set the `rds.adaptive_autovacuum` parameter to 0 or OFF.

Transaction ID wraparound is still possible even when RDS tunes the autovacuum parameters. We encourage you to implement an Amazon CloudWatch alarm for transaction ID wraparound. For more information, see the blog post [Implement an early warning system for transaction ID wraparound in Amazon RDS for PostgreSQL](#).

With adaptive autovacuum parameter tuning enabled, RDS will begin adjusting autovacuum parameters when the CloudWatch metric `MaximumUsedTransactionIDs` reaches the value of the `autovacuum_freeze_max_age` parameter or 500,000,000, whichever is greater.

RDS continues to adjust parameters for autovacuum if a table continues to trend toward transaction ID wraparound. Each of these adjustments dedicates more resources to autovacuum to avoid wraparound. RDS updates the following autovacuum-related parameters:

- `autovacuum_vacuum_cost_delay`
- `autovacuum_vacuum_cost_limit`
- `autovacuum_work_mem`
- `autovacuum_naptime`

RDS modifies these parameters only if the new value makes autovacuum more aggressive. The parameters are modified in memory on the DB instance. The values in the parameter group aren't changed. To view the current in-memory settings, use the PostgreSQL `SHOW` SQL command.

Whenever RDS modifies any of these autovacuum parameters, it generates an event for the affected DB instance that is visible on the AWS Management Console (<https://console.aws.amazon.com/rds/>) and through the RDS API. After the `MaximumUsedTransactionIDs` CloudWatch metric returns below the threshold, RDS resets the autovacuum related parameters in memory back to the values specified in the parameter group and generates another event corresponding to this change.

Determining if the tables in your database need vacuuming

You can use the following query to show the number of unvacuumed transactions in a database. The `datfrozenxid` column of a database's `pg_database` row is a lower bound on the normal transaction IDs appearing in that database. This column is the minimum of the per-table `realfrozenxid` values within the database.

```
SELECT datname, age(datfrozenxid) FROM pg_database ORDER BY age(datfrozenxid) desc limit 20;
```

For example, the results of running the preceding query might be the following.

datname	age
mydb	1771757888
template0	1721757888
template1	1721757888
rdsadmin	1694008527
postgres	1693881061

(5 rows)

When the age of a database reaches 2 billion transaction IDs, transaction ID (XID) wraparound occurs and the database becomes read-only. This query can be used to produce a metric and run a few times a day. By default, autovacuum is set to keep the age of transactions to no more than 200,000,000 ([autovacuum_freeze_max_age](#)).

A sample monitoring strategy might look like this:

- Set the `autovacuum_freeze_max_age` value to 200 million transactions.

- If a table reaches 500 million unvacuumed transactions, that triggers a low-severity alarm. This isn't an unreasonable value, but it can indicate that autovacuum isn't keeping up.
- If a table ages to 1 billion, this should be treated as an alarm to take action on. In general, you want to keep ages closer to `autovacuum_freeze_max_age` for performance reasons. We recommend you investigate using the recommendations that follow.
- If a table reaches 1.5 billion unvacuumed transactions, that triggers a high-severity alarm. Depending on how quickly your database uses transaction IDs, this alarm can indicate that the system is running out of time to run autovacuum. In this case, we recommend you resolve this immediately.

If a table is constantly breaching these thresholds, you need to modify your autovacuum parameters further. By default, using VACUUM manually (which has cost-based delays disabled) is more aggressive than using the default autovacuum, but it is also more intrusive to the system as a whole.

We recommend the following:

- Be aware and enable a monitoring mechanism so that you are aware of the age of your oldest transactions.

For information on creating a process that warns you about transaction ID wraparound, see the AWS Database Blog post [Implement an early warning system for transaction ID wraparound in Amazon RDS for PostgreSQL](#).

- For busier tables, perform a manual vacuum freeze regularly during a maintenance window, in addition to relying on autovacuum. For information on performing a manual vacuum freeze, see [Performing a manual vacuum freeze \(p. 1598\)](#).

Determining which tables are currently eligible for autovacuum

Often, it is one or two tables in need of vacuuming. Tables whose `realfrozenxid` value is greater than the number of transactions in `autovacuum_freeze_max_age` are always targeted by autovacuum. Otherwise, if the number of tuples made obsolete since the last VACUUM exceeds the "vacuum threshold", the table is vacuumed.

The [autovacuum threshold](#) is defined as:

```
Vacuum-threshold = vacuum-base-threshold + vacuum-scale-factor * number-of-tuples
```

While you are connected to your database, run the following query to see a list of tables that autovacuum sees as eligible for vacuuming:

```

WITH vbt AS (SELECT setting AS autovacuum_vacuum_threshold FROM
pg_settings WHERE name = 'autovacuum_vacuum_threshold')
     , vsf AS (SELECT setting AS autovacuum_vacuum_scale_factor FROM
pg_settings WHERE name = 'autovacuum_vacuum_scale_factor')
     , fma AS (SELECT setting AS autovacuum_freeze_max_age FROM
pg_settings WHERE name = 'autovacuum_freeze_max_age')
     , sto AS (select opt_oid, split_part(setting, '=', 1) as param,
split_part(setting, '=', 2) as value from (select oid opt_oid,
unnest(reloptions) setting from pg_class) opt)
SELECT
    '''||ns.nspname||'.'||c.relname||''' as relation
    , pg_size_pretty(pg_table_size(c.oid)) as table_size
    , age(realfrozenxid) as xid_age
    , coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
autovacuum_freeze_max_age
    , (coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float)
```

```

+ coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) *
c.reltuples) as autovacuum_vacuum_tuples
    , n_dead_tup as dead_tuples
FROM pg_class c join pg_namespace ns on ns.oid = c.relnamespace
join pg_stat_all_tables stat on stat.relid = c.oid
join vbt on (1=1) join vsf on (1=1) join fma on (1=1)
left join sto cvbt on cvbt.param = 'autovacuum_vacuum_threshold' and
c.oid = cvbt.opt_oid
left join sto csvf on csvf.param = 'autovacuum_vacuum_scale_factor' and
c.oid = csvf.opt_oid
left join sto cfma on cfma.param = 'autovacuum_freeze_max_age' and
c.oid = cfma.opt_oid
WHERE c.relkind = 'r' and nspname <> 'pg_catalog'
and (
    age(relfrozenxid) >= coalesce(cfma.value::float,
autovacuum_freeze_max_age::float)
    or
    coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
coalesce(csvf.value::float,autovacuum_vacuum_scale_factor::float) *
c.reltuples <= n_dead_tup
    -- or 1 = 1
)
ORDER BY age(relfrozenxid) DESC LIMIT 50;

```

Determining if autovacuum is currently running and for how long

If you need to manually vacuum a table, you need to determine if autovacuum is currently running. If it is, you might need to adjust parameters to make it run more efficiently, or terminate autovacuum so you can manually run VACUUM.

Use the following query to determine if autovacuum is running, how long it has been running, and if it is waiting on another session.

If you are using RDS for PostgreSQL 9.6+ or higher, use this query:

```

SELECT datname, usename, pid, state, wait_event, current_timestamp - xact_start AS
xact_runtime, query
FROM pg_stat_activity
WHERE upper(query) LIKE '%VACUUM%'
ORDER BY xact_start;

```

After running the query, you should see output similar to the following.

datname	usename	pid	state	wait_event	xact_runtime	query
mydb	rdsadmin	16473	active		33 days 16:32:11.600656	autovacuum: VACUUM ANALYZE public.mytable1 (to prevent wraparound)
mydb	rdsadmin	22553	active		14 days 09:15:34.073141	autovacuum: VACUUM ANALYZE public.mytable2 (to prevent wraparound)
mydb	rdsadmin	41909	active		3 days 02:43:54.203349	autovacuum: VACUUM ANALYZE public.mytable3
mydb	rdsadmin	618	active		00:00:00	SELECT datname, usename, pid, state, wait_event, current_timestamp - xact_start AS xact_runtime, query+
						FROM pg_stat_activity

```

|           |           |           |           |
| WHERE query
+           +           +           +           +
|           |           |           |           |
| ORDER BY
xact_start;
+

```

If you are using an Amazon RDS for PostgreSQL version less than 9.6, use the following query.

```

SELECT datname, usename, pid, waiting, current_timestamp - xact_start AS xact_runtime,
       query
FROM pg_stat_activity
WHERE upper(query) LIKE '%VACUUM%'
ORDER BY xact_start;

```

After running the query, you should see output similar to the following.

datname	username	pid	waiting	xact_runtime	query
mydb	rdsadmin	16473	f	33 days 16:32:11.600656	autovacuum: VACUUM ANALYZE public.mytable1 (to prevent wraparound)
mydb	rdsadmin	22553	f	14 days 09:15:34.073141	autovacuum: VACUUM ANALYZE public.mytable2 (to prevent wraparound)
mydb	rdsadmin	41909	f	3 days 02:43:54.203349	autovacuum: VACUUM ANALYZE public.mytable3
mydb	rdsadmin	618	f	00:00:00	SELECT datname, usename, pid, waiting, current_timestamp - xact_start AS xact_runtime, query+
%					FROM pg_stat_activity
					+ WHERE query like '%VACUUM'
					+ ORDER BY xact_start;
					+

Several issues can cause a long-running autovacuum session (that is, multiple days long). The most common issue is that your [maintenance_work_mem](#) parameter value is set too low for the size of the table or rate of updates.

We recommend that you use the following formula to set the [maintenance_work_mem](#) parameter value.

```
GREATEST({DBInstanceClassMemory/63963136*1024},65536)
```

Short running autovacuum sessions can also indicate problems:

- It can indicate that there aren't enough `autovacuum_max_workers` for your workload. In this case, you need to indicate the number of workers.
- It can indicate that there is an index corruption (autovacuum crashes and restart on the same relation but make no progress). In this case, run a manual vacuum freeze verbose `__table__` to see the exact cause.

Performing a manual vacuum freeze

You might want to perform a manual vacuum on a table that has a vacuum process already running. This is useful if you have identified a table with an age approaching 2 billion transactions (or above any threshold you are monitoring).

The following steps are a guideline, and there are several variations to the process. For example, during testing, suppose that you find that the `maintenance_work_mem` parameter value was set too small and that you need to take immediate action on a table. However, perhaps you don't want to bounce the instance at the moment. Using the queries in previous sections, you determine which table is the problem and notice a long running autovacuum session. You know that you need to change the `maintenance_work_mem` parameter setting, but you also need to take immediate action and vacuum the table in question. The following procedure shows what to do in this situation:

To manually perform a vacuum freeze

1. Open two sessions to the database containing the table you want to vacuum. For the second session, use "screen" or another utility that maintains the session if your connection is dropped.
2. In session one, get the PID of the autovacuum session running on the table.

Run the following query to get the PID of the autovacuum session.

```
SELECT datname, username, pid, current_timestamp - xact_start
AS xact_runtime, query
FROM pg_stat_activity WHERE upper(query) LIKE '%VACUUM%' ORDER BY
xact_start;
```

3. In session two, calculate the amount of memory you need for this operation. In this example, we determine that we can afford to use up to 2 GB of memory for this operation, so we set `maintenance_work_mem` for the current session to 2 GB.

```
set maintenance_work_mem='2 GB';
SET
```

4. In session two, issue a `vacuum freeze verbose` command for the table. The verbose setting is useful because, although there is no progress report for this in PostgreSQL currently, you can see activity.

```
\timing on
Timing is on.
vacuum freeze verbose pgbench_branches;
```

```
INFO:  vacuuming "public.pgbench_branches"
INFO:  index "pgbench_branches_pkey" now contains 50 row versions in 2 pages
DETAIL:  0 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO:  index "pgbench_branches_test_index" now contains 50 row versions in 2 pages
DETAIL:  0 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO:  "pgbench_branches": found 0 removable, 50 nonremovable row versions
      in 43 out of 43 pages
DETAIL:  0 dead row versions cannot be removed yet.
There were 9347 unused item pointers.
0 pages are entirely empty.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
VACUUM
Time: 2.765 ms
```

5. In session one, if autovacuum was blocking, you see in `pg_stat_activity` that waiting is "T" for your vacuum session. In this case, you need to terminate the autovacuum process as follows.

```
SELECT pg_terminate_backend('the_pid');
```

6. At this point, your session begins. It's important to note that autovacuum restarts immediately because this table is probably the highest on its list of work. Initiate your `vacuum freeze verbose` command in session 2 and then terminate the autovacuum process in session 1.

Reindexing a table when autovacuum is running

If an index has become corrupt, autovacuum continues to process the table and fails. If you attempt a manual vacuum in this situation, you will receive an error message similar to the following:

```
mydb=# vacuum freeze pgbench_branches;
ERROR: index "pgbench_branches_test_index" contains unexpected
      zero page at block 30521
HINT: Please REINDEX it.
```

When the index is corrupted and autovacuum is attempting to run against the table, you contend with an already running autovacuum session. When you issue a "`REINDEX`" command, you take out an exclusive lock on the table. Write operations are blocked, and also reads that use that specific index.

To reindex a table when autovacuum is running on the table

1. Open two sessions to the database containing the table you want to vacuum. For the second session, use "screen" or another utility that maintains the session if your connection is dropped.
2. In session one, get the PID of the autovacuum session running on the table.

Run the following query to get the PID of the autovacuum session.

```
SELECT datname, usename, pid, current_timestamp - xact_start
AS xact_runtime, query
FROM pg_stat_activity WHERE upper(query) like '%VACUUM%' ORDER BY
xact_start;
```

3. In session two, issue the reindex command.

```
\timing on
Timing is on.
reindex index pgbench_branches_test_index;
REINDEX
Time: 9.966 ms
```

4. In session one, if autovacuum was blocking, you see in `pg_stat_activity` that waiting is "T" for your vacuum session. In this case, you will need to terminate the autovacuum process.

```
select pg_terminate_backend('the_pid');
```

5. At this point, your session begins. It's important to note that autovacuum restarts immediately because this table is probably the highest on its list of work. Initiate your command in session 2 and then terminate the autovacuum process in session 1.

Other parameters that affect autovacuum

The following query shows the values of some of the parameters that directly affect autovacuum and its behavior. The [autovacuum parameters](#) are described fully in the PostgreSQL documentation.

```
SELECT name, setting, unit, short_desc
FROM pg_settings
```

```
WHERE name IN (
    'autovacuum_max_workers',
    'autovacuum_analyze_scale_factor',
    'autovacuum_naptime',
    'autovacuum_analyze_threshold',
    'autovacuum_analyze_scale_factor',
    'autovacuum_vacuum_threshold',
    'autovacuum_vacuum_scale_factor',
    'autovacuum_vacuum_threshold',
    'autovacuum_vacuum_cost_delay',
    'autovacuum_vacuum_cost_limit',
    'vacuum_cost_limit',
    'autovacuum_freeze_max_age',
    'maintenance_work_mem',
    'vacuum_freeze_min_age');
```

While these all affect autovacuum, some of the most important ones are:

- [maintenance_work_mem](#)
- [autovacuum_freeze_max_age](#)
- [autovacuum_max_workers](#)
- [autovacuum_vacuum_cost_delay](#)
- [Autovacuum_vacuum_cost_limit](#)

Setting table-level autovacuum parameters

Autovacuum-related [storage parameters](#) can be set at a table level, which can be better than altering the behavior of the entire database. For large tables, you might need to set aggressive settings and you might not want to make autovacuum behave that way for all tables.

The following query shows which tables currently have table-level options in place.

```
SELECT relname, reloptions
FROM pg_class
WHERE reloptions IS NOT null;
```

An example where this might be useful is on tables that are much larger than the rest of your tables. Suppose that you have one 300-GB table and 30 other tables less than 1 GB. In this case, you might set some specific parameters for your large table so you don't alter the behavior of your entire system.

```
ALTER TABLE mytable set (autovacuum_vacuum_cost_delay=0);
```

Doing this disables the cost-based autovacuum delay for this table at the expense of more resource usage on your system. Normally, autovacuum pauses for autovacuum_vacuum_cost_delay each time autovacuum_cost_limit is reached. You can find more details in the PostgreSQL documentation about [cost-based vacuuming](#).

Autovacuum logging

By default, the *postgresql.log* doesn't contain information about the autovacuum process. You can see output in the PostgreSQL error log from the autovacuum worker operations by setting the `rds.force_autovacuum_logging_level` parameter. Allowed values are `disabled`, `debug5`, `debug4`, `debug3`, `debug2`, `debug1`, `info`, `notice`, `warning`, `error`, `log`, `fatal`, and `panic`. The default value is `disabled` because the other allowable values can add significant amount of information to your logs.

We recommend that you set the value of the `rds.force_autovacuum_logging_level` parameter to warning and that you set the `log_autovacuum_min_duration` parameter to a value from 1,000 to 5,000 milliseconds. If you set this value to 5,000, Amazon RDS writes any activity to the log that takes more than five seconds. It also shows "vacuum skipped" messages when application locking is causing autovacuum to intentionally skip tables. If you are troubleshooting a problem and need more detail, you can use a different logging level value, such as `debug1` or `debug3`. Use these debug parameters for a short period of time because these settings produce extremely verbose content written to the error log file. For more information about these debug settings, see the [PostgreSQL documentation](#).

Note

PostgreSQL allows the `rds_superuser` account to view autovacuum sessions in `pg_stat_activity`. For example, you can identify and end an autovacuum session that is blocking a command from running, or running slower than a manually issued `vacuum` command.

Working with the PostGIS extension

PostGIS is an extension to PostgreSQL for storing and managing spatial information. If you are not familiar with PostGIS, see [PostGIS.net](#).

You need to perform some setup before you can use the PostGIS extension. The following list shows what you need to do. Each step is described in greater detail later in this section.

Topics

- [Step 1: Connect to the DB instance using the user name used to create the DB instance \(p. 1602\)](#)
- [Step 2: Load the PostGIS extensions \(p. 1602\)](#)
- [Step 3: Transfer ownership of the extensions to the rds_superuser role \(p. 1603\)](#)
- [Step 4: Transfer ownership of the objects to the rds_superuser role \(p. 1603\)](#)
- [Step 5: Test the extensions \(p. 1603\)](#)
- [PostGIS extension versions \(p. 1604\)](#)

Step 1: Connect to the DB instance using the user name used to create the DB instance

First, you connect to the DB instance using the user name that was used to create the DB instance. That name is automatically assigned the `rds_superuser` role. You need the `rds_superuser` role that is needed to do the remaining steps.

The following example uses `SELECT` to show you the current user. In this case, the current user should be the user name you chose when creating the DB instance.

```
SELECT CURRENT_USER;
current_user
-----
myawsuser
(1 row)
```

Step 2: Load the PostGIS extensions

Use `CREATE EXTENSION` statements to load the PostGIS extensions. You must also load the `extension` extension. You can then use the `\dn` command to list the owners of the PostGIS schemas.

```
CREATE EXTENSION postgis;
CREATE EXTENSION fuzzystrmatch;
CREATE EXTENSION postgis_tiger_geocoder;
```

```
CREATE EXTENSION postgis_topology;
\dn
   List of schemas
   Name      | Owner
   public    | myawsuser
   tiger     | rdsadmin
   tiger_data | rdsadmin
   topology   | rdsadmin
(4 rows)
```

Step 3: Transfer ownership of the extensions to the rds_superuser role

Use the ALTER SCHEMA statements to transfer ownership of the schemas to the `rds_superuser` role.

```
ALTER SCHEMA tiger OWNER TO rds_superuser;
ALTER SCHEMA tiger_data OWNER TO rds_superuser;
ALTER SCHEMA topology OWNER TO rds_superuser;
\dn
   List of schemas
   Name      | Owner
   public    | myawsuser
   tiger     | rds_superuser
   tiger_data | rds_superuser
   topology   | rds_superuser
(4 rows)
```

Step 4: Transfer ownership of the objects to the rds_superuser role

Use the following function to transfer ownership of the PostGIS objects to the `rds_superuser` role. Run the following statement from the psql prompt to create the function.

```
CREATE FUNCTION exec(text) returns text language plpgsql volatile AS $$ BEGIN EXECUTE $1;
  RETURN $1; END; $$;
```

Next, run this query to run the exec function that in turn runs the statements and alters the permissions.

```
SELECT exec('ALTER TABLE ' || quote_ident(s.nspname) || '.' || quote_ident(s.relname) || '
  OWNER TO rds_superuser;');
FROM (
  SELECT nspname, relname
  FROM pg_class c JOIN pg_namespace n ON (c.relnamespace = n.oid)
  WHERE nspname in ('tiger','topology') AND
    relkind IN ('r','S','v') ORDER BY relkind = 'S')
s;
```

Step 5: Test the extensions

Add `tiger` to your search path using the following command.

```
SET search_path=public,tiger;
```

Test `tiger` by using the following SELECT statement.

```
SELECT na.address, na.streetname, na.streettypeabbrev, na.zip
FROM normalize_address('1 Devonshire Place, Boston, MA 02109') AS na;
address | streetname | streettypeabbrev | zip
-----+-----+-----+
1 | Devonshire | Pl | 02109
(1 row)
```

Test topology by using the following SELECT statement.

```
SELECT topology.createtopology('my_new_topo',26986,0.5);
createtopology
-----
1
(1 row)
```

PostGIS extension versions

The following table shows the PostGIS versions that ship with the RDS for PostgreSQL versions.

PostgreSQL version	PostGIS version
13.2, 13.1	3.0.2
12.6	3.0.2
12.5, 12.4, 12.3, 12.2	3.0.0
11.11, 11.10, 11.9, 11.8, 11.7, 11.6, 11.5	2.5.2
11.4, 11.2, 11.1	2.5.1
10.16, 10.15, 10.14, 10.13, 10.12, 10.11, 10.10	2.5.2
10.9, 10.7, 10.6, 10.5, 10.4	2.4.4
10.3, 10.1	2.4.2
9.6.21, 9.6.20, 9.6.19, 9.6.18, 9.6.17, 9.6.16, 9.6.15	2.5.2
9.6.14, 9.6.12, 9.6.11, 9.6.10, 9.6.9	2.3.7
9.6.8, 9.6.6	2.3.4
9.6.5, 9.6.3, 9.6.2	2.3.2
9.6.1	2.3.0
9.5.25, 9.5.24, 9.5.23, 9.5.22, 9.5.21, 9.5.20, 9.5.19	2.5.2
9.5.18, 9.5.16, 9.5.15, 9.5.14, 9.5.13, 9.5.12, 9.5.10, 9.5.9, 9.5.7, 9.5.6	2.2.5
9.5.4, 9.5.2	2.2.2

Note

PostgreSQL 10.5 added support for the libprotobuf extension version 1.3.0 to the PostGIS component.

Using a custom DNS server for outbound network access

Amazon RDS for PostgreSQL supports outbound network access on your DB instances and allows Domain Name Service (DNS) resolution from a custom DNS server owned by the customer. You can resolve only fully qualified domain names from your Amazon RDS DB instance through your custom DNS server.

Topics

- [Enabling custom DNS resolution \(p. 1605\)](#)
- [Disabling custom DNS resolution \(p. 1605\)](#)
- [Setting up a custom DNS server \(p. 1605\)](#)

Enabling custom DNS resolution

To enable DNS resolution in your customer VPC, associate a custom DB parameter group to your RDS for PostgreSQL instance, turn on the `rds.custom_dns_resolution` parameter by setting it to 1, and then restart the DB instance for the changes to take place.

Disabling custom DNS resolution

To disable DNS resolution in your customer VPC, turn off the `rds.custom_dns_resolution` parameter of your custom DB parameter group by setting it to 0, then restart the DB instance for the changes to take place.

Setting up a custom DNS server

After you set up your custom DNS name server, it takes up to 30 minutes to propagate the changes to your DB instance. After the changes are propagated to your DB instance, all outbound network traffic requiring a DNS lookup queries your DNS server over port 53.

Note

If you don't set up a custom DNS server, and `rds.custom_dns_resolution` is set to 1, hosts are resolved using a Route 53 private zone. For more information, see [Working with private hosted zones](#).

To set up a custom DNS server for your Amazon RDS for PostgreSQL DB instance

1. From the DHCP options set attached to your VPC, set the `domain-name-servers` option to the IP address of your DNS name server. For more information, see [DHCP options sets](#).

Note

The `domain-name-servers` option accepts up to four values, but your Amazon RDS DB instance uses only the first value.

2. Ensure that your DNS server can resolve all lookup queries, including public DNS names, Amazon EC2 private DNS names, and customer-specific DNS names. If the outbound network traffic contains any DNS lookups that your DNS server can't handle, your DNS server must have appropriate upstream DNS providers configured.
3. Configure your DNS server to produce User Datagram Protocol (UDP) responses of 512 bytes or less.
4. Configure your DNS server to produce Transmission Control Protocol (TCP) responses of 1024 bytes or less.
5. Configure your DNS server to allow inbound traffic from your Amazon RDS DB instances over port 53. If your DNS server is in an Amazon VPC, the VPC must have a security group that contains

inbound rules that allow UDP and TCP traffic on port 53. If your DNS server is not in an Amazon VPC, it must have appropriate firewall settings to allow UDP and TCP inbound traffic on port 53.

For more information, see [Security groups for your VPC](#) and [Adding and removing rules](#).

6. Configure the VPC of your Amazon RDS DB instance to allow outbound traffic over port 53. Your VPC must have a security group that contains outbound rules that allow UDP and TCP traffic on port 53.

For more information, see [Security groups for your VPC](#) and [Adding and removing rules](#).

7. The routing path between the Amazon RDS DB instance and the DNS server has to be configured correctly to allow DNS traffic.

If the Amazon RDS DB instance and the DNS server are not in the same VPC, a peering connection has to be set up between them. For more information, see [What is VPC peering?](#)

Scheduling maintenance with the PostgreSQL pg_cron extension

You can use the PostgreSQL pg_cron extension to schedule maintenance commands within a PostgreSQL database. For a complete description, see [What is pg_cron?](#) in the pg_cron documentation.

The pg_cron extension is supported on Amazon RDS for PostgreSQL engine versions 12.5 and higher.

Topics

- [Enabling the pg_cron extension \(p. 1607\)](#)
- [Granting permissions to pg_cron \(p. 1607\)](#)
- [Cron job to manually vacuum a table \(p. 1608\)](#)
- [Cron job to purge the pg_cron history \(p. 1609\)](#)
- [Disabling logging of pg_cron history \(p. 1609\)](#)
- [Scheduling a cron job for a database other than postgres \(p. 1609\)](#)
- [The pg_cron reference \(p. 1610\)](#)

Enabling the pg_cron extension

Enable the pg_cron extension as follows:

1. Modify the parameter group associated with your DB instance and add pg_cron to the shared_preload_libraries parameter value. This change requires a DB instance restart to take effect. For more information, see [Modifying parameters in a DB parameter group \(p. 232\)](#).
2. After the DB instance has restarted, run the following command using an account that has the rds_superuser permissions.

```
CREATE EXTENSION pg_cron;
```

3. Either use the default settings, or schedule jobs to run in other databases within your PostgreSQL DB instance. The pg_cron scheduler is set in the default PostgreSQL database named `postgres`. The pg_cron objects are created in this `postgres` database and all scheduling actions run in this database.

To schedule jobs to run in other databases within your PostgreSQL DB instance, see the example in [Scheduling a cron job for a database other than postgres \(p. 1609\)](#).

Granting permissions to pg_cron

As the `rds_superuser` role, you can create the pg_cron extension and then grant permissions to other users. For other users to be able to schedule jobs, grant them permissions to objects in the cron schema.

Important

We recommend that you grant permissions to the cron schema sparingly.

To grant others permission to the cron schema, run the following command.

```
postgres=> GRANT USAGE ON SCHEMA cron TO other-user;
```

This permission provides `other-user` with access to the cron schema to schedule and unschedule cron jobs. However, for the cron jobs to run successfully, the user also needs permission to access the objects in the cron jobs. If the user doesn't have permission, the job fails and errors such as the

following appears in the `postgresql.log`. In this example, the user doesn't have permission to access the `pgbench_accounts` table.

```
2020-12-08 16:41:00 UTC::@[30647]:ERROR: permission denied for table pgbench_accounts
2020-12-08 16:41:00 UTC::@[30647]:STATEMENT: update pgbench_accounts set abalance =
abalance + 1
2020-12-08 16:41:00 UTC::@[27071]:LOG: background worker "pg_cron" (PID 30647) exited with
exit code 1
```

Other messages in the `cron.job_run_details` table appear like the following.

```
postgres=> select jobid, username, status, return_message, start_time from
cron.job_run_details where status = 'failed';
jobid | username | status | return_message | start_time
-----+-----+-----+-----+
143 | unprivuser | failed | ERROR: permission denied for table pgbench_accounts |
2020-12-08 16:41:00.036268+00
143 | unprivuser | failed | ERROR: permission denied for table pgbench_accounts |
2020-12-08 16:40:00.050844+00
143 | unprivuser | failed | ERROR: permission denied for table pgbench_accounts |
2020-12-08 16:42:00.175644+00
143 | unprivuser | failed | ERROR: permission denied for table pgbench_accounts |
2020-12-08 16:43:00.069174+00
143 | unprivuser | failed | ERROR: permission denied for table pgbench_accounts |
2020-12-08 16:44:00.059466+00
(5 rows)
```

For more information, see [The pg_cron tables \(p. 1612\)](#).

Cron job to manually vacuum a table

Autovacuum handles vacuum maintenance for most cases. For more information, see [Working with PostgreSQL autovacuum on Amazon RDS \(p. 1593\)](#).

However, you might want to manually vacuum a specific table at a time of your choosing. Following is an example of using the `cron.schedule` function to set up a job to use `VACUUM FREEZE` on a specific table every day at 22:00 (GMT).

```
SELECT cron.schedule('manual vacuum', '0 22 * * *', 'VACUUM FREEZE pgbench_accounts');
schedule
-----
1
(1 row)
```

After the preceding example runs, you can check the history in the `cron.job_run_details` table as follows.

```
postgres=> select * from cron.job_run_details;
jobid | runid | job_pid | database | username | command | status | return_message | start_time | end_time
-----+-----+-----+-----+-----+-----+-----+-----+
1 | 1 | 3395 | postgres | adminuser | vacuum freeze pgbench_accounts | succeeded | VACUUM |
2020-12-04 21:10:00.050386+00 | 2020-12-04 21:10:00.072028+00
(1 row)
```

Following is an example of viewing the history in the cron.job_run_details table to investigate why a job failed.

```
postgres=> select * from cron.job_run_details where status = 'failed';
   jobid | runid | job_pid | database | username | command | status | return_message | start_time | end_time
-----+-----+-----+-----+-----+-----+-----+-----+
      5 | 4 | 30339 | postgres | adminuser | vacuum freeze pgbench_account | failed | ERROR: relation "pgbench_account" does not exist | 2020-12-04 21:48:00.015145+00 | 2020-12-04 21:48:00.029567+00
(1 row)
```

For more information, see [The pg_cron tables \(p. 1612\)](#).

Cron job to purge the pg_cron history

The cron.job_run_details table contains a history of cron jobs that can become very large over time. We recommend that you schedule a job that purges this table. For example, keeping a week's worth of entries might be sufficient for troubleshooting purposes.

The following example uses the [cron.schedule \(p. 1611\)](#) function to schedule a job that runs every day at midnight to purge the cron.job_run_details table. The job keeps only the last seven days. Use your rds_superuser account to schedule the job such as the following.

```
SELECT cron.schedule('0 0 * * *', $$DELETE
    FROM cron.job_run_details
    WHERE end_time < now() - interval '7 days'$$);
```

For more information, see [The pg_cron tables \(p. 1612\)](#).

Disabling logging of pg_cron history

To completely disable writing anything to the cron.job_run_details table, modify the parameter group associated with the DB instance and set the cron.log_run parameter to off. If you do this, the pg_cron extension no longer writes to the table and produces errors only in the postgresql.log file. For more information, see [Modifying parameters in a DB parameter group \(p. 232\)](#).

Use the following command to check the value of the cron.log_run parameter.

```
postgres=> SHOW cron.log_run;
```

For more information, see [The pg_cron parameters \(p. 1610\)](#).

Scheduling a cron job for a database other than postgres

The metadata for pg_cron is all held in the PostgreSQL default database named `postgres`. Because background workers are used for running the maintenance cron jobs, you can schedule a job in any of your databases within the RDS DB instance:

1. In the cron database, schedule the job as you normally do using the [cron.schedule \(p. 1611\)](#).

```
postgres=> SELECT cron.schedule('database1 manual vacuum', '29 03 * * *', 'vacuum freeze
    test_table');
```

2. As a user with the `rds_superuser` role, update the database column for the job that you just created so that it runs in another database within your RDS DB instance.

```
postgres=> UPDATE cron.job SET database = 'database1' WHERE jobid = 106;
```

3. Verify by querying the `cron.job` table.

```
postgres=> select * from cron.job;
   jobid | schedule | command | nodename | nodeport | database | username | active | jobname
-----+-----+-----+-----+-----+-----+-----+-----+
 106 | 29 03 * * * | vacuum freeze test_table | localhost | 8192 | database1 | adminuser | t | database1 manual vacuum
 1 | 59 23 * * * | vacuum freeze pgbench_accounts | localhost | 8192 | postgres | adminuser | t | manual vacuum
(2 rows)
```

Note

In some situations, you might add a cron job that you intend to run on a different database. In such cases, the job might try to run in the default database (`postgres`) before you update the correct database column. If the user name has permissions, the job successfully runs in the default database.

The pg_cron reference

You can use the following parameters, functions, and tables with the pg_cron extension. For more information, see [What is pg_cron?](#) in the pg_cron documentation.

Topics

- [The pg_cron parameters \(p. 1610\)](#)
- [cron.schedule function \(p. 1611\)](#)
- [cron.unschedule function \(p. 1612\)](#)
- [The pg_cron tables \(p. 1612\)](#)

The pg_cron parameters

Following is the list of parameters to control the pg_cron extension behavior.

Parameter	Description
<code>cron.database_name</code>	The database in which pg_cron metadata is kept.
<code>cron.host</code>	The hostname to connect to PostgreSQL. You can't modify this value.
<code>cron.log_run</code>	Log all the jobs that run into the <code>job_run_details</code> table. Values are <code>on</code> or <code>off</code> . For more information, see The pg_cron tables (p. 1612) .
<code>cron.log_statement</code>	Log all cron statements before running them. Values are <code>on</code> or <code>off</code> .

Parameter	Description
<code>cron.max_running_jobs</code>	The maximum number of jobs that can run concurrently.
<code>cron.use_background_workers</code>	Use background workers instead of client sessions. You can't modify this value.

You can use the following SQL command to display these parameters and their values.

```
postgres=> SELECT name, setting, short_desc FROM pg_settings WHERE name LIKE 'cron.%' ORDER BY name;
```

cron.schedule function

This function schedules a cron job. The job is initially scheduled in the default `postgres` database. The function returns a bigint value representing the job identifier. To schedule jobs to run in other databases within your PostgreSQL DB instance, see the example in [Scheduling a cron job for a database other than postgres \(p. 1609\)](#).

The function has two syntax formats.

Syntax

```
cron.schedule (job_name,
               schedule,
               command
               );
cron.schedule (schedule,
               command
               );
```

Parameters

Parameter	Description
<code>job_name</code>	The name of the cron job.
<code>schedule</code>	Text indicating the schedule for the cron job. The format is the standard cron format.
<code>command</code>	Text of the command to run.

Examples

```
postgres=> SELECT cron.schedule ('test','0 10 * * *', 'VACUUM pgbench_history');
schedule
-----
        145
(1 row)

postgres=> SELECT cron.schedule ('0 15 * * *', 'VACUUM pgbench_accounts');
schedule
-----
        146
(1 row)
```

cron.unschedule function

This function deletes a cron job. You can either pass in the `job_name` or the `job_id`. A policy makes sure that you are the owner to remove the schedule for the job. The function returns a Boolean indicating success or failure.

The function has the following syntax formats.

Syntax

```
cron.unschedule (job_id);
cron.unschedule (job_name);
```

Parameters

Parameter	Description
<code>job_id</code>	A job identifier that was returned from the <code>cron.schedule</code> function when the cron job was scheduled.
<code>job_name</code>	The name of a cron job that was scheduled with the <code>cron.schedule</code> function.

Examples

```
postgres=> select cron.unschedule(108);
unschedule
-----
t
(1 row)

postgres=> select cron.unschedule('test');
unschedule
-----
t
(1 row)
```

The pg_cron tables

The following tables are created and used to schedule the cron jobs and record how the jobs completed.

Table	Description
<code>cron.job</code>	<p>Contains the metadata about each scheduled job. Most interactions with this table should be done by using the <code>cron.schedule</code> and <code>cron.unschedule</code> functions.</p> <p>Note We don't recommend giving update or insert privileges directly to this table. Doing so would allow the user to update the <code>username</code> column to run as <code>rds-superuser</code>.</p>

Table	Description
cron.job_run_details	<p>Contains historic information about past scheduled job executions. This is useful to investigate the status, return messages, and start and end time from the job execution.</p> <p>Note To prevent this table from growing indefinitely, purge it on a regular basis. For an example, see Cron job to purge the pg_cron history (p. 1609).</p>

Managing PostgreSQL partitions with the pg_partman extension

PostgreSQL table partitioning provides a framework for high-performance handling of data input and reporting. Use partitioning for databases that require very fast input of large amounts of data. Partitioning also provides for faster queries of large tables. Partitioning helps maintain data without impacting the database instance because it requires less I/O resources.

By using partitioning, you can split data into custom-sized chunks for processing. For example, you can partition time-series data for ranges such as hourly, daily, weekly, monthly, quarterly, yearly, custom, or any combination of these. For a time-series data example, if you partition the table by hour, each partition contains one hour of data. If you partition the time-series table by day, the partitions holds one day's worth of data, and so on. The partition key controls the size of a partition.

When you use an `INSERT` or `UPDATE` SQL command on a partitioned table, the database engine routes the data to the appropriate partition. PostgreSQL table partitions that store the data are child tables of the main table.

During database query reads, the PostgreSQL optimizer examines the `WHERE` clause of the query and, if possible, directs the database scan to only the relevant partitions.

Starting with version 10, PostgreSQL uses declarative partitioning to implement table partitioning. This is also known as native PostgreSQL partitioning. Before PostgreSQL version 10, you used triggers to implement partitions.

PostgreSQL table partitioning provides the following features:

- Creation of new partitions at any time.
- Variable partition ranges.
- Detachable and reattachable partitions using data definition language (DDL) statements.

For example, detachable partitions are useful for removing historical data from the main partition but keeping historical data for analysis.

- New partitions inherit the parent database table properties, including the following:
 - Indexes
 - Primary keys, which must include the partition key column
 - Foreign keys
 - Check constraints
 - References
- Creating indexes for the full table or each specific partition.

You can't alter the schema for an individual partition. However, you can alter the parent table (such as adding a new column), which propagates to partitions.

Topics

- [Overview of the PostgreSQL pg_partman extension \(p. 1615\)](#)
- [Enabling the pg_partman extension \(p. 1615\)](#)
- [Configuring partitions using the create_parent function \(p. 1616\)](#)
- [Configuring partition maintenance using the run_maintenance_proc function \(p. 1617\)](#)

Overview of the PostgreSQL pg_partman extension

You can use the PostgreSQL pg_partman extension to automate the creation and maintenance of table partitions. For more general information, see [PG Partition Manager](#) in the pg_partman documentation.

Note

The pg_partman extension is supported on RDS PostgreSQL engine versions 12.5 and higher.

Instead of having to manually create each partition, you configure pg_partman with the following settings:

- Table to be partitioned
- Partition type
- Partition key
- Partition granularity
- Partition precreation and management options

After you create a PostgreSQL partitioned table, you register it with pg_partman by calling the `create_parent` function. Doing this creates the necessary partitions based on the parameters you pass to the function.

The pg_partman extension also provides the `run_maintenance_proc` function, which you can call on a scheduled basis to automatically manage partitions. To ensure that the proper partitions are created as needed, schedule this function to run periodically (such as hourly). You can also ensure that partitions are automatically dropped.

Enabling the pg_partman extension

If you have multiple databases inside the same DB instance for which you want to manage partitions, enable the pg_partman extension separately for each database. To enable the pg_partman extension for a specific database, create the partition maintenance schema and then create the pg_partman extension as follows.

```
CREATE SCHEMA partman;
CREATE EXTENSION pg_partman WITH SCHEMA partman;
```

Note

To create the pg_partman extension, make sure that you have `rds_superuser` privileges.

If you receive an error such as the following, grant the `rds_superuser` privileges to the account or use your superuser account.

```
ERROR: permission denied to create extension "pg_partman"
HINT: Must be superuser to create this extension.
```

To grant `rds_superuser` privileges, connect with your superuser account and run the following command.

```
GRANT rds_superuser TO user-or-role;
```

For the examples that show using the pg_partman extension, we use the following sample database table and partition. This database uses a partitioned table based on a timestamp. A schema `data_mart` contains a table named `events` with a column named `created_at`. The following settings are included in the `events` table:

- Primary keys `event_id` and `created_at`, which must have the column used to guide the partition.
- A check constraint `ck_valid_operation` to enforce values for an operation table column.
- Two foreign keys, where one (`fk_orga_membership`) points to the external table `organization` and the other (`fk_parent_event_id`) is a self-referenced foreign key.
- Two indexes, where one (`idx_org_id`) is for the foreign key and the other (`idx_event_type`) is for the event type.

The follow DDL statements create these objects, which are automatically included on each partition.

```

CREATE SCHEMA data_mart;
CREATE TABLE data_mart.organization (
    org_id BIGSERIAL,
    org_name TEXT,
    CONSTRAINT pk_organization PRIMARY KEY (org_id)
);

CREATE TABLE data_mart.events(
    event_id      BIGSERIAL,
    operation     CHAR(1),
    value         FLOAT(24),
    parent_event_id BIGINT,
    event_type    VARCHAR(25),
    org_id        BIGSERIAL,
    created_at    timestamp,
    CONSTRAINT pk_data_mart_event PRIMARY KEY (event_id, created_at),
    CONSTRAINT ck_valid_operation CHECK (operation = 'C' OR operation = 'D'),
    CONSTRAINT fk_orga_membership
        FOREIGN KEY(org_id)
        REFERENCES data_mart.organization (org_id),
    CONSTRAINT fk_parent_event_id
        FOREIGN KEY(parent_event_id, created_at)
        REFERENCES data_mart.events (event_id, created_at)
) PARTITION BY RANGE (created_at);

CREATE INDEX idx_org_id      ON data_mart.events(org_id);
CREATE INDEX idx_event_type ON data_mart.events(event_type);

```

Configuring partitions using the `create_parent` function

After you enable the pg_partman extension, you use the `create_parent` function to configure partitions inside the partition maintenance schema. The following example uses the `events` table example created in [Enabling the pg_partman extension \(p. 1615\)](#). Call the `create_parent` function as follows.

```

SELECT partman.create_parent(
    p_parent_table => 'data_mart.events',
    p_control => 'created_at',
    p_type => 'native',
    p_interval=> 'daily',
    p_premake => 30);

```

The parameters are as follows:

- `p_parent_table` – The parent partitioned table. This table must already exist and be fully qualified including the schema.
- `p_control` – The column on which the partitioning is to be based. The data type must be integer or time-based.
- `p_type` – The type is either native or partman. You typically use the native type for its performance improvements and flexibility. The partman type relies on inheritance.

- `p_interval` – The time interval or integer range for each partition. Example values include daily, hourly, and so on.
- `p_premake` – The number of partitions to create in advance to support new inserts.

For a complete description of the `create_parent` function, see [Creation Functions](#) in the pg_partman documentation.

Configuring partition maintenance using the `run_maintenance_proc` function

You can run partition maintenance operations to automatically create new partitions, detach partitions, or remove old partitions. Partition maintenance relies on the `run_maintenance_proc` function of pg_partman and the pg_cron extension, which initiates an internal scheduler. The pg_cron scheduler automatically executes SQL statements, functions, and procedures defined in your databases.

The following example uses the events table example created in [Enabling the pg_partman extension \(p. 1615\)](#) to set partition maintenance operations to run automatically.

```
-- Prerequisite: add pg_cron to the shared_preload_libraries parameter in the DB instance's parameter group.

CREATE EXTENSION pg_cron;

UPDATE partman.part_config
SET infinite_time_partitions = true,
    retention = '3 months',
    retention_keep_table=true
WHERE parent_table = 'data_mart.events';
SELECT cron.schedule('@hourly', $$CALL partman.run_maintenance_proc()$$);
```

Following, you can find a step-by-step explanation of the preceding example:

1. Modify the parameter group associated with your DB instance and add pg_cron to the `shared_preload_libraries` parameter value. This change requires a DB instance restart for it to take effect. For more information, see [Modifying parameters in a DB parameter group \(p. 232\)](#).
2. Run the command `CREATE EXTENSION pg_cron;` using an account that has the `rds_superuser` permissions. Doing this enables the pg_cron extension. For more information, see [Scheduling maintenance with the PostgreSQL pg_cron extension \(p. 1607\)](#).
3. Run the command `UPDATE partman.part_config` to adjust the pg_partman settings for the `data_mart.events` table.
4. Run the command `SET ...` to configure the `data_mart.events` table, with these clauses:
 - a. `infinite_time_partitions = true`, – Configures the table to be able to automatically create new partitions without any limit.
 - b. `retention = '3 months'`, – Configures the table to have a maximum retention of three months.
 - c. `retention_keep_table=true` – Configures the table so that when the retention period is due, the table isn't deleted automatically. Instead, partitions that are older than the retention period are only detached from the parent table.
5. Run the command `SELECT cron.schedule ...` to make a pg_cron function call. This call defines how often the scheduler runs the pg_partman maintenance procedure, `partman.run_maintenance_proc`. For this example, the procedure runs every hour.

For a complete description of the `run_maintenance_proc` function, see [Maintenance Functions](#) in the pg_partman documentation.

Invoking an AWS Lambda function from an RDS for PostgreSQL DB instance

You can invoke AWS Lambda functions from an RDS for PostgreSQL DB instance. To do this, use the `aws_lambda` PostgreSQL extension provided with RDS for PostgreSQL.

AWS Lambda is a compute service that you can use to run code. For example, you can use Lambda functions to process event notifications from a DB instance. For more information about Lambda, see [What is AWS Lambda?](#) in the *AWS Lambda Developer Guide*.

Note

Invoking an AWS Lambda function is supported in the following RDS for PostgreSQL versions:

- 12.6 and later minor versions
- 13.2 and later minor versions

Topics

- [Overview of using a Lambda function \(p. 1618\)](#)
- [Specifying the Lambda function to use \(p. 1619\)](#)
- [Giving RDS access to Lambda \(p. 1619\)](#)
- [Invoking Lambda functions \(p. 1621\)](#)
- [Function reference \(p. 1623\)](#)

Overview of using a Lambda function

You can invoke a Lambda function from an RDS for PostgreSQL database with the following procedure.

To invoke a Lambda function from an RDS for PostgreSQL database

1. Install the required PostgreSQL extensions. These include the `aws_lambda` and `aws_commons` extensions. To do so, start `psql` and run the following commands.

```
CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;
```

The `aws_lambda` extension provides the [aws_lambda.invoke \(p. 1623\)](#) function that you use to invoke functions in Lambda. The `aws_commons` extension is included to provide additional helper functions.

2. Identify the name or Amazon Resource Name (ARN) for the Lambda function to use. For details about this process, see [Specifying the Lambda function to use \(p. 1619\)](#).
3. Provide permission to access the Lambda function.

To invoke a Lambda function, give the RDS for PostgreSQL DB instance permission to access the Lambda invoke API operation. Doing this includes the following steps:

1. Create an AWS Identity and Access Management (IAM) policy that provides access to a Lambda function that you want to invoke.
2. Create an IAM role.
3. Attach the IAM policy that you created to the role that you created.
4. Add this IAM role to your DB instance.

For details about this process, see [Giving RDS access to Lambda \(p. 1619\)](#).

4. Use the `aws_lambda.invoke` function to run the Lambda function. For details about this process, see [Invoking Lambda functions \(p. 1621\)](#).

Specifying the Lambda function to use

To identify the Lambda function to use, specify the following information:

- **Function name** – The name of the Lambda function, ARN, version, or alias. For a listing of possible formats, see [Lambda function name formats](#).
- **AWS Region** – (Optional) The AWS Region where the Lambda function is located. If you don't specify a Region value and it's not specified in the function ARN, RDS uses the same Region as the DB instance.

For a listing of AWS Region names and associated values, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

To hold the Lambda function name information, you can use the `aws_commons.create_lambda_function_arn (p. 1625)` function. This function creates an `aws_commons._lambda_function_arn_1` composite structure to store the name information, as shown following.

```
psql=> SELECT aws_commons.create_lambda_function_arn(
    'my-function',
    'us-west-2'
) AS aws_lambda_arn_1 \gset

psql=> SELECT aws_commons.create_lambda_function_arn(
    '123456789012:function:my-function',
    'us-west-2'
) AS lambda_partial_arn_1 \gset

psql=> SELECT aws_commons.create_lambda_function_arn(
    'arn:aws:lambda:us-west-2:123456789012:function:my-function'
) AS lambda_arn_1 \gset
```

You can later provide any of these values as a parameter in calls to the `aws_lambda.invoke (p. 1623)` function. For examples, see [Invoking Lambda functions \(p. 1621\)](#).

Giving RDS access to Lambda

To use a Lambda function, give your PostgreSQL DB instance permission to access Lambda. To do this, use the following procedure.

To give a PostgreSQL DB instance access to Lambda

1. Create an IAM policy.

This policy provides the permissions that allow your PostgreSQL DB instance to invoke Lambda functions.

As part of creating this policy, take the following steps:

- a. Include in the policy the required action `lambda:InvokeFunction` to allow Lambda invocation from your RDS for PostgreSQL DB instance.
- b. Include the Amazon Resource Name (ARN) that identifies the Lambda function. The ARN format for accessing Lambda is: `arn:aws:lambda:::function:example_function/*`

For more information on creating an IAM policy for RDS for PostgreSQL, see [Creating and using an IAM policy for IAM database access](#). See also [IAM Tutorial: Create and attach your first customer managed policy](#) in the *IAM User Guide*.

The following AWS CLI command creates an IAM policy named `rds-lambda-policy` with these options. It grants access to a function named `example_function`.

```
aws iam create-policy --policy-name rds-lambda-policy --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccessToExampleFunction",
            "Effect": "Allow",
            "Action": "lambda:InvokeFunction",
            "Resource": "arn:aws:lambda:<region>:<123456789012>:function:example_function"
        }
    ]
}'
```

After you create the policy, note the ARN of the policy. You need the ARN for a subsequent step when you attach the policy to an IAM role.

2. Create an IAM role.

You do this so that RDS for PostgreSQL can assume this IAM role on your behalf to access your Lambda function. For more information, see [Creating a role to delegate permissions to an IAM user](#) in the *IAM User Guide*.

The following example shows using the AWS CLI command to create a role named `rds-lambda-role`.

```
aws iam create-role --role-name rds-lambda-role --assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}'
```

3. Attach the IAM policy that you created to the IAM role that you created.

The following AWS CLI command attaches the policy created earlier to the role named `rds-lambda-role`. Replace `your-policy-arn` with the policy ARN that you noted in an earlier step.

```
aws iam attach-role-policy --policy-arn your-policy-arn --role-name rds-lambda-role
```

4. Add the IAM role to the DB instance. You do so by using the AWS CLI, as described following.

Use the following CLI command to add the IAM role to the RDS for PostgreSQL DB instance named `my-db-instance`. Replace `your-role-arn` with the role ARN that you noted in a previous step. Use `Lambda` for the value of the `--feature-name` option, as shown following.

```
aws rds add-role-to-db-instance \
```

```
--db-instance-identifier my-db-instance \
--feature-name Lambda \
--role-arn your-role-arn \
--region your-region
```

Invoking Lambda functions

Following, you can find some examples of calling the [aws_lambda.invoke \(p. 1623\)](#) function. Before you use the `aws_lambda.invoke` function, be sure to complete the following prerequisites:

- Install the required PostgreSQL extensions as described in [Overview of using a Lambda function \(p. 1618\)](#).
- Determine which Lambda function to invoke as described in [Specifying the Lambda function to use \(p. 1619\)](#).
- Make sure that the DB instance has invoke access to Lambda as described in [Giving RDS access to Lambda \(p. 1619\)](#).

You can invoke a Lambda function synchronously or asynchronously. You control this with the following values for the [aws_lambda.invoke \(p. 1623\)](#) function's `invocation_type` parameter:

- The `RequestResponse` type of invocation for a Lambda function is synchronous and returns a response payload in the result of the `aws_lambda.invoke` function. Use the `RequestResponse` invocation type when your workflow depends on receiving the Lambda function result immediately. Most of the following examples use synchronous invocation.

The `RequestResponse` type of invocation is the default.

- The `Event` type of invocation for a Lambda function is asynchronous and returns immediately without a returned payload. Use the `Event` type of invocation when you don't need to know the result of the Lambda function before your workflow moves on. For an example of asynchronous invocation, see [Asynchronous event invocation of Lambda functions \(p. 1622\)](#).

The following [aws_lambda.invoke \(p. 1623\)](#) examples use a `aws_lambda_arn_1` structure, which contains the identifying information for the Lambda function. To create the structure, use the [aws_commons.create_lambda_function_arn \(p. 1625\)](#) function. For an example of using the `aws_commons.create_lambda_function_arn` function, see [Specifying the Lambda function to use \(p. 1619\)](#).

Topics

- [Synchronous RequestResponse invocation of Lambda functions \(p. 1621\)](#)
- [Asynchronous event invocation of Lambda functions \(p. 1622\)](#)
- [Requesting a Lambda execution log in a function response \(p. 1622\)](#)
- [Including client context in a Lambda function \(p. 1622\)](#)
- [Invoking a specific version of a Lambda function \(p. 1622\)](#)
- [Lambda function error handling \(p. 1623\)](#)

Synchronous RequestResponse invocation of Lambda functions

Following is an example of a synchronous Lambda function invocation. The following two `aws_lambda.invoke` function call results are the same.

```
psql=> SELECT * FROM aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"'}::json);
```

```
psql=> SELECT * FROM aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}':json, 'RequestResponse');
```

The parameters are described as follows:

- `: 'aws_lambda_arn_1'` – This parameter is a structure that identifies the Lambda function to call. This example uses a variable to identify the previously created structure. You can instead create the structure by including the [aws_commons.create_lambda_function_arn \(p. 1625\)](#) function call inline within the [aws_lambda.invoke \(p. 1623\)](#) function call as follows.

```
psql=> SELECT * FROM aws_lambda.invoke(aws_commons.create_lambda_function_arn('my-function', 'us-west-2'), '{"body": "Hello from Postgres!"}':json );
```

- `'{"body": "Hello from PostgreSQL!"}':json` – The JSON payload to pass to the Lambda function.
- `'RequestResponse'` – The Lambda invocation type.

Asynchronous event invocation of Lambda functions

Following is an example of an asynchronous Lambda function invocation. The `Event` invocation type schedules the Lambda function invocation with the specified input payload and returns immediately. Use the `Event` invocation type in certain workflows that don't depend on the results of the Lambda function.

```
psql=> SELECT * FROM aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}':json, 'Event');
```

Requesting a Lambda execution log in a function response

You can request to include the last 4 KB of the execution log in the function response, as shown following.

```
psql=> SELECT *, select convert_from(decode(log_result, 'base64'), 'utf-8') as log FROM aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}':json, 'RequestResponse', 'Tail');
```

Set the [aws_lambda.invoke \(p. 1623\)](#) function's `log_type` parameter to `Tail` to include the execution log in the response. The default value for the `log_type` parameter is `None`.

The `log_result` that's returned is a `base64` encoded string. You can decode the contents using a combination of the `decode` and `convert_from` PostgreSQL functions.

Including client context in a Lambda function

You can pass in client context information that is separate from the payload, as shown following.

```
psql=> SELECT *, convert_from(decode(log_result, 'base64'), 'utf-8') as log FROM aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}':json, 'RequestResponse', 'Tail');
```

To include client context, use a JSON object for the [aws_lambda.invoke \(p. 1623\)](#) function's `context` parameter.

Invoking a specific version of a Lambda function

For an example of invoking a specific version of a Lambda function, see the following.

```
psql=> SELECT * FROM aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}':json, 'RequestResponse', 'None', NULL, 'custom_version');
```

To identify a Lambda function's version, use the [aws_lambda.invoke \(p. 1623\)](#) function's `qualifier` parameter. In this example, '`custom_version`' is an alias or version that identifies the version of the function to invoke.

You can instead supply a Lambda function qualifier with the function name information as follows.

```
psql=> SELECT * FROM aws_lambda.invoke(aws_commons.create_lambda_function_arn('my-function:custom_version', 'us-west-2'), '{"body": "Hello from Postgres!"}':json);
```

Lambda function error handling

If a Lambda function throws an exception during request processing, `aws_lambda.invoke` fails with a PostgreSQL error such as the following.

```
psql=> SELECT * FROM aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}':json);
ERROR: lambda invocation failed
DETAIL: "arn:aws:lambda:us-west-2:123456789012:function:my-function" returned error
"Unhandled", details: "<Error details string>".
```

Function reference

Following is the reference for the functions to use for invoking Lambda functions with RDS for PostgreSQL.

Functions

- [aws_lambda.invoke \(p. 1623\)](#)
- [aws_commons.create_lambda_function_arn \(p. 1625\)](#)

aws_lambda.invoke

Runs a Lambda function for an RDS for PostgreSQL DB instance.

For more details about invoking Lambda functions, see also [Invoke](#) in the *AWS Lambda Developer Guide*.

Syntax

JSON

```
aws_lambda.invoke(
    IN function_name TEXT,
    IN payload JSON,
    IN region TEXT DEFAULT NULL,
    IN invocation_type TEXT DEFAULT 'RequestResponse',
    IN log_type TEXT DEFAULT 'None',
    IN context JSON DEFAULT NULL,
    IN qualifier VARCHAR(128) DEFAULT NULL,
    OUT status_code INT,
    OUT payload JSON,
    OUT executed_version TEXT,
```

```
OUT log_result TEXT)
```

```
aws_lambda.invoke(
    IN function_name aws_commons._lambda_function_arn_1,
    IN payload JSON,
    IN invocation_type TEXT DEFAULT 'RequestResponse',
    IN log_type TEXT DEFAULT 'None',
    IN context JSON DEFAULT NULL,
    IN qualifier VARCHAR(128) DEFAULT NULL,
    OUT status_code INT,
    OUT payload JSON,
    OUT executed_version TEXT,
    OUT log_result TEXT)
```

JSONB

```
aws_lambda.invoke(
    IN function_name TEXT,
    IN payload JSONB,
    IN region TEXT DEFAULT NULL,
    IN invocation_type TEXT DEFAULT 'RequestResponse',
    IN log_type TEXT DEFAULT 'None',
    IN context JSONB DEFAULT NULL,
    IN qualifier VARCHAR(128) DEFAULT NULL,
    OUT status_code INT,
    OUT payload JSONB,
    OUT executed_version TEXT,
    OUT log_result TEXT)
```

```
aws_lambda.invoke(
    IN function_name aws_commons._lambda_function_arn_1,
    IN payload JSONB,
    IN invocation_type TEXT DEFAULT 'RequestResponse',
    IN log_type TEXT DEFAULT 'None',
    IN context JSONB DEFAULT NULL,
    IN qualifier VARCHAR(128) DEFAULT NULL,
    OUT status_code INT,
    OUT payload JSONB,
    OUT executed_version TEXT,
    OUT log_result TEXT
)
```

Input parameters

function_name

The identifying name of the Lambda function. The value can be the function name, an ARN, or a partial ARN. For a listing of possible formats, see [Lambda function name formats](#) in the *AWS Lambda Developer Guide*.

payload

The input for the Lambda function. The format can be JSON or JSONB. For more information, see [JSON Types](#) in the PostgreSQL documentation.

region

(Optional) The Lambda Region for the function. By default, RDS resolves the AWS Region from the full ARN in the *function_name* or it uses the RDS for PostgreSQL DB instance Region. If this Region value conflicts with the one provided in the *function_name* ARN, an error is raised.

invocation_type

The invocation type of the Lambda function. The value is case-sensitive. Possible values include the following:

- RequestResponse – The default. This type of invocation for a Lambda function is synchronous and returns a response payload in the result. Use the RequestResponse invocation type when your workflow depends on receiving the Lambda function result immediately.
- Event – This type of invocation for a Lambda function is asynchronous and returns immediately without a returned payload. Use the Event invocation type when you don't need results of the Lambda function before your workflow moves on.
- DryRun – This type of invocation tests access without running the Lambda function.

log_type

The type of Lambda log to return in the `log_result` output parameter. The value is case-sensitive. Possible values include the following:

- Tail – The returned `log_result` output parameter will include the last 4 KB of the execution log.
- None – No Lambda log information is returned.

context

Client context in JSON or JSONB format. Fields to use include `custom` and `env`.

qualifier

A qualifier that identifies a Lambda function's version to be invoked. If this value conflicts with one provided in the `function_name` ARN, an error is raised.

Output parameters

status_code

An HTTP status response code. For more information, see [Lambda Invoke response elements](#) in the [AWS Lambda Developer Guide](#).

payload

The information returned from the Lambda function that ran. The format is in JSON or JSONB.

executed_version

The version of the Lambda function that ran.

log_result

The execution log information returned if the `log_type` value is `Tail` when the Lambda function was invoked. The result contains the last 4 KB of the execution log encoded in Base64.

[aws_commons.create_lambda_function_arn](#)

Creates an `aws_commons._lambda_function_arn_1` structure to hold Lambda function name information. You can use the results of the `aws_commons.create_lambda_function_arn` function in the `function_name` parameter of the `aws_lambda.invoke` ([p. 1623](#)) function.

Syntax

```
aws_commons.create_lambda_function_arn(  
    function_name TEXT,  
    region TEXT DEFAULT NULL  
)
```

```
RETURNS aws_commons._lambda_function_arn_1
```

Input parameters

function_name

A required text string containing the Lambda function name. The value can be a function name, a partial ARN, or a full ARN.

region

An optional text string containing the AWS Region that the Lambda function is in. For a listing of Region names and associated values, see [Regions, Availability Zones, and Local Zones \(p. 49\)](#).

Security in Amazon RDS

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security of the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the [AWS compliance programs](#). To learn about the compliance programs that apply to Amazon RDS, see [AWS services in scope by compliance program](#).
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your organization's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Amazon RDS. The following topics show you how to configure Amazon RDS to meet your security and compliance objectives. You also learn how to use other AWS services that help you monitor and secure your Amazon RDS resources.

You can manage access to your Amazon RDS resources and your databases on a DB instance. The method you use to manage access depends on what type of task the user needs to perform with Amazon RDS:

- Run your DB instance in a virtual private cloud (VPC) based on the Amazon VPC service for the greatest possible network access control. For more information about creating a DB instance in a VPC, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).
- Use AWS Identity and Access Management (IAM) policies to assign permissions that determine who is allowed to manage Amazon RDS resources. For example, you can use IAM to determine who is allowed to create, describe, modify, and delete DB instances, tag resources, or modify security groups.
- Use security groups to control what IP addresses or Amazon EC2 instances can connect to your databases on a DB instance. When you first create a DB instance, its firewall prevents any database access except through rules specified by an associated security group.
- Use Secure Socket Layer (SSL) or Transport Layer Security (TLS) connections with DB instances running the MySQL, MariaDB, PostgreSQL, Oracle, or Microsoft SQL Server database engines. For more information on using SSL/TLS with a DB instance, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).
- Use Amazon RDS encryption to secure your DB instances and snapshots at rest. Amazon RDS encryption uses the industry standard AES-256 encryption algorithm to encrypt your data on the server that hosts your DB instance. For more information, see [Encrypting Amazon RDS resources \(p. 1630\)](#).
- Use network encryption and transparent data encryption with Oracle DB instances; for more information, see [Oracle native network encryption \(p. 1176\)](#) and [Oracle Transparent Data Encryption \(p. 1204\)](#).
- Use the security features of your DB engine to control who can log in to the databases on a DB instance. These features work just as if the database was on your local network.

Note

You only have to configure security for your use cases. You don't have to configure security access for processes that Amazon RDS manages. These include creating backups, replicating data between a primary DB instance and a read replica, and other processes.

For more information on managing access to Amazon RDS resources and your databases on a DB instance, see the following topics.

Topics

- [Database authentication with Amazon RDS \(p. 1628\)](#)
- [Data protection in Amazon RDS \(p. 1629\)](#)
- [Identity and access management in Amazon RDS \(p. 1644\)](#)
- [Logging and monitoring in Amazon RDS \(p. 1691\)](#)
- [Compliance validation for Amazon RDS \(p. 1693\)](#)
- [Resilience in Amazon RDS \(p. 1694\)](#)
- [Infrastructure security in Amazon RDS \(p. 1695\)](#)
- [Amazon RDS API and interface VPC endpoints \(AWS PrivateLink\) \(p. 1696\)](#)
- [Security best practices for Amazon RDS \(p. 1698\)](#)
- [Controlling access with security groups \(p. 1699\)](#)
- [Master user account privileges \(p. 1712\)](#)
- [Using service-linked roles for Amazon RDS \(p. 1714\)](#)
- [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#)

Database authentication with Amazon RDS

Amazon RDS supports several ways to authenticate database users.

Password, Kerberos, and IAM database authentication use different methods of authenticating to the database. Therefore, a specific user can log in to a database using only one authentication method.

For PostgreSQL, use only one of the following role settings for a user of a specific database:

- To use IAM database authentication, assign the `rds_iam` role to the user.
- To use Kerberos authentication, assign the `rds_ad` role to the user.
- To use password authentication, don't assign either the `rds_iam` or `rds_ad` roles to the user.

Don't assign both the `rds_iam` and `rds_ad` roles to a user of a PostgreSQL database either directly or indirectly by nested grant access. If the `rds_iam` role is added to the master user, IAM authentication takes precedence over password authentication so the master user has to log in as an IAM user.

Topics

- [Password authentication \(p. 1628\)](#)
- [IAM database authentication \(p. 1629\)](#)
- [Kerberos authentication \(p. 1629\)](#)

Password authentication

With *password authentication*, your DB instance performs all administration of user accounts. You create users with SQL statements such as `CREATE USER` and specify passwords in the `IDENTIFIED BY` clause.

All RDS DB engines support password authentication. For more information about password authentication, see the documentation for your DB engine.

With password authentication, your database controls and authenticates user accounts. If a DB engine has strong password management features, they can enhance security. Database authentication might be easier to administer using password authentication when you have small user communities. Because

clear text passwords are generated in this case, integrating with AWS Secrets Manager can enhance security.

For information about using Secrets Manager with Amazon RDS, see [Creating a basic secret](#) and [Rotating secrets for supported Amazon RDS databases](#) in the *AWS Secrets Manager User Guide*. For information about programmatically retrieving your secrets in your custom applications, see [Retrieving the secret value](#) in the *AWS Secrets Manager User Guide*.

IAM database authentication

You can authenticate to your DB instance using AWS Identity and Access Management (IAM) database authentication. IAM database authentication works with MySQL and PostgreSQL. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token.

For more information about IAM database authentication, including information about availability for specific DB engines, see [IAM database authentication for MySQL and PostgreSQL \(p. 1660\)](#).

Kerberos authentication

Amazon RDS supports external authentication of database users using Kerberos and Microsoft Active Directory. Kerberos is a network authentication protocol that uses tickets and symmetric-key cryptography to eliminate the need to transmit passwords over the network. Kerberos has been built into Active Directory and is designed to authenticate users to network resources, such as databases.

Amazon RDS support for Kerberos and Active Directory provides the benefits of single sign-on and centralized authentication of database users. You can keep your user credentials in Active Directory. Active Directory provides a centralized place for storing and managing credentials for multiple DB instances.

You can enable your database users to authenticate against DB instances in two ways. They can use credentials stored either in AWS Directory Service for Microsoft Active Directory or in your on-premises Active Directory.

Microsoft SQL Server, MySQL, and PostgreSQL DB instances support one- and two-way forest trust relationships. Oracle DB instances support one- and two-way external and forest trust relationships. For more information, see [When to create a trust relationship](#) in the *AWS Directory Service Administration Guide*.

For information about Kerberos authentication with a specific DB engine, see the following:

- [Using Windows Authentication with an Amazon RDS for SQL Server DB instance \(p. 711\)](#)
- [Using Kerberos authentication for MySQL \(p. 938\)](#)
- [Configuring Kerberos authentication for Amazon RDS for Oracle \(p. 1014\)](#)
- [Using Kerberos authentication with Amazon RDS for PostgreSQL \(p. 1520\)](#)

Note

Currently, Kerberos authentication isn't supported for MariaDB DB instances.

Data protection in Amazon RDS

The AWS [shared responsibility model](#) applies to data protection in Amazon Relational Database Service. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the

AWS services that you use. For more information about data privacy, see the [Data Privacy FAQ](#). For information about data protection in Europe, see the [AWS Shared Responsibility Model and GDPR](#) blog post on the [AWS Security Blog](#).

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-2](#).

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form fields such as a **Name** field. This includes when you work with Amazon RDS or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into Amazon RDS or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

Topics

- [Protecting data using encryption \(p. 1630\)](#)
- [Internetwork traffic privacy \(p. 1643\)](#)

Protecting data using encryption

You can enable encryption for database resources. You can also encrypt connections to DB instances.

Topics

- [Encrypting Amazon RDS resources \(p. 1630\)](#)
- [Customer master key \(CMK\) management \(p. 1633\)](#)
- [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#)
- [Rotating your SSL/TLS certificate \(p. 1636\)](#)

Encrypting Amazon RDS resources

Amazon RDS can encrypt your Amazon RDS DB instances. Data that is encrypted at rest includes the underlying storage for DB instances, its automated backups, read replicas, and snapshots.

Amazon RDS encrypted DB instances use the industry standard AES-256 encryption algorithm to encrypt your data on the server that hosts your Amazon RDS DB instances. After your data is encrypted, Amazon RDS handles authentication of access and decryption of your data transparently with a minimal impact on performance. You don't need to modify your database client applications to use encryption.

Note

For encrypted and unencrypted DB instances, data that is in transit between the source and the read replicas is encrypted, even when replicating across AWS Regions.

Topics

- [Overview of encrypting Amazon RDS resources \(p. 1631\)](#)
- [Enabling Amazon RDS encryption for a DB instance \(p. 1631\)](#)
- [Availability of Amazon RDS encryption \(p. 1632\)](#)
- [Limitations of Amazon RDS encrypted DB instances \(p. 1632\)](#)

Overview of encrypting Amazon RDS resources

Amazon RDS encrypted DB instances provide an additional layer of data protection by securing your data from unauthorized access to the underlying storage. You can use Amazon RDS encryption to increase data protection of your applications deployed in the cloud, and to fulfill compliance requirements for encryption at rest.

Amazon RDS also supports encrypting an Oracle or SQL Server DB instance with Transparent Data Encryption (TDE). TDE can be used with encryption at rest, although using TDE and encryption at rest simultaneously might slightly affect the performance of your database. You must manage different keys for each encryption method. For more information on TDE, see [Oracle Transparent Data Encryption \(p. 1204\)](#) or [Support for Transparent Data Encryption in SQL Server \(p. 754\)](#).

For an Amazon RDS encrypted DB instance, all logs, backups, and snapshots are encrypted. Amazon RDS uses an AWS KMS customer master key (CMK) to encrypt these resources. For more information about CMKs, see [Customer master keys \(CMKs\)](#) in the *AWS Key Management Service Developer Guide*. If you copy an encrypted snapshot, you can use a different CMK to encrypt the target snapshot than the one that was used to encrypt the source snapshot.

A read replica of an Amazon RDS encrypted instance must be encrypted using the same CMK as the primary DB instance when both are in the same AWS Region. If the primary DB instance and read replica are in different AWS Regions, you encrypt the read replica using the CMK for that AWS Region.

To manage the customer master keys (CMKs) used for encrypting and decrypting your Amazon RDS resources, you use the [AWS Key Management Service \(AWS KMS\)](#). AWS KMS combines secure, highly available hardware and software to provide a key management system scaled for the cloud. Using AWS KMS, you can create CMKs and define the policies that control how these CMKs can be used. AWS KMS supports CloudTrail, so you can audit CMK usage to verify that CMKs are being used appropriately. You can use your CMKs with Amazon RDS and supported AWS services such as Amazon S3, Amazon EBS, and Amazon Redshift. For a list of services that are integrated with AWS KMS, see [Supported services](#) in the *AWS Key Management Service Developer Guide*.

Enabling Amazon RDS encryption for a DB instance

To enable encryption for a new DB instance, choose **Enable encryption** on the Amazon RDS console. For information on creating a DB instance, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

If you use the `create-db-instance` AWS CLI command to create an encrypted DB instance, set the `--storage-encrypted` parameter. If you use the `CreateDBInstance` API operation, set the `StorageEncrypted` parameter to true.

When you create an encrypted DB instance, you can choose a customer managed CMK or the AWS managed CMK for Amazon RDS to encrypt your DB instance. If you don't specify the key identifier for a customer managed CMK, Amazon RDS uses the AWS managed CMK for your new DB instance. Amazon RDS creates an AWS managed CMK for Amazon RDS for your AWS account. Your AWS account has a different AWS managed CMK for Amazon RDS for each AWS Region.

Once you have created an encrypted DB instance, you can't change the CMK used by that DB instance. Therefore, be sure to determine your CMK requirements before you create your encrypted DB instance.

If you use the AWS CLI `create-db-instance` command to create an encrypted DB instance with a customer managed CMK, set the `--kms-key-id` parameter to any key identifier for the CMK. If you use

the Amazon RDS API `CreateDBInstance` operation, set the `KmsKeyId` parameter to any key identifier for the CMK. To use a customer managed CMK in a different AWS account, specify the key ARN or alias ARN.

Important

If Amazon RDS loses access to the CMK for a DB instance—for example, when RDS access to a CMK is revoked—then the encrypted DB instance goes into a terminal state. In this case, you can only restore the DB instance from a backup. We strongly recommend that you always enable backups for encrypted DB instances to guard against the loss of encrypted data in your databases.

Availability of Amazon RDS encryption

Amazon RDS encryption is currently available for all database engines and storage types.

Amazon RDS encryption is available for most DB instance classes. The following table lists DB instance classes that *do not support* Amazon RDS encryption:

Instance type	Instance class
General Purpose (M1)	db.m1.small
	db.m1.medium
	db.m1.large
	db.m1.xlarge
Memory Optimized (M2)	db.m2.xlarge
	db.m2.2xlarge
	db.m2.4xlarge
Burst Capable (T2)	db.t2.micro

Note

Encryption at rest is not available for DB instances running SQL Server Express Edition.

Limitations of Amazon RDS encrypted DB instances

The following limitations exist for Amazon RDS encrypted DB instances:

- You can only enable encryption for an Amazon RDS DB instance when you create it, not after the DB instance is created.

However, because you can encrypt a copy of an unencrypted snapshot, you can effectively add encryption to an unencrypted DB instance. That is, you can create a snapshot of your DB instance, and then create an encrypted copy of that snapshot. You can then restore a DB instance from the encrypted snapshot, and thus you have an encrypted copy of your original DB instance. For more information, see [Copying a snapshot \(p. 352\)](#).

- You can't disable encryption on an encrypted DB instance.
- You can't create an encrypted snapshot of an unencrypted DB instance.
- A snapshot of an encrypted DB instance must be encrypted using the same CMK as the DB instance.
- You can't have an encrypted read replica of an unencrypted DB instance or an unencrypted read replica of an encrypted DB instance.

- Encrypted read replicas must be encrypted with the same CMK as the source DB instance when both are in the same AWS Region.
- You can't restore an unencrypted backup or snapshot to an encrypted DB instance.
- To copy an encrypted snapshot from one AWS Region to another, you must specify the CMK in the destination AWS Region. This is because CMKs are specific to the AWS Region that they are created in.

The source snapshot remains encrypted throughout the copy process. Amazon RDS uses envelope encryption to protect data during the copy process. For more information about envelope encryption, see [Envelope encryption in the AWS Key Management Service Developer Guide](#).

- You can't unencrypt an encrypted DB instance. However, you can export data from an encrypted DB instance and import the data into an unencrypted DB instance.

Customer master key (CMK) management

Amazon RDS automatically integrates with AWS Key Management Service (AWS KMS) for key management. Amazon RDS uses envelope encryption. For more information about envelope encryption, see [Envelope encryption in the AWS Key Management Service Developer Guide](#).

A *customer master key (CMK)* is a logical representation of a master key. The CMK includes metadata, such as the key ID, creation date, description, and key state. The CMK also contains the key material used to encrypt and decrypt data. For more information about customer managed CMKs, see [Customer managed CMKs in the AWS Key Management Service Developer Guide](#).

You can manage CMKs used for Amazon RDS encrypted DB instances using the [AWS Key Management Service \(AWS KMS\)](#) in the [AWS KMS console](#), the AWS CLI, or the AWS KMS API. If you want full control over a CMK, then you must create a customer managed CMK.

AWS managed CMKs are CMKs in your account that are created, managed, and used on your behalf by an AWS service that is integrated with AWS KMS. You can't delete, edit, or rotate AWS managed CMKs. For more information about AWS managed CMKs, see [AWS managed CMKs in the AWS Key Management Service Developer Guide](#).

You can't share a snapshot that has been encrypted using the AWS managed CMK of the AWS account that shared the snapshot.

You can view audit logs of every action taken with an AWS managed or customer managed CMK by using [AWS CloudTrail](#).

Important

When RDS encounters a DB instance encrypted by a CMK that RDS doesn't have access to, RDS puts the DB instance into a terminal state. In this state, the DB instance is no longer available and the current state of the database can't be recovered. To restore the DB instance, you must re-enable access to the CMK for RDS, and then restore the DB instance from a backup.

Authorizing use of the CMK

When RDS uses a CMK in cryptographic operations, it acts on behalf of the user who is creating or changing the RDS resource.

To use the customer managed CMK for an RDS resource on your behalf, a user must have permissions to call the following operations on the CMK:

- kms:GenerateDataKey
- kms:Decrypt

You can specify these required permissions in a key policy, or in an IAM policy if the key policy allows it.

You can make the IAM policy stricter in various ways. For example, to allow the CMK to be used only for requests that originate in RDS , you can use the [kms:ViaService condition key](#) with the `rds.<region>.amazonaws.com` value.

You can also use the keys or values in the [encryption context](#) as a condition for using the CMK for cryptographic operations.

Using SSL/TLS to encrypt a connection to a DB instance

You can use Secure Socket Layer (SSL) or Transport Layer Security (TLS) from your application to encrypt a connection to a DB instance running MySQL, MariaDB, SQL Server, Oracle, or PostgreSQL. Each DB engine has its own process for implementing SSL/TLS. To learn how to implement SSL/TLS for your DB instance, use the link following that corresponds to your DB engine:

- [Using SSL with a MariaDB DB instance \(p. 584\)](#)
- [Using SSL with a Microsoft SQL Server DB instance \(p. 704\)](#)
- [Using SSL with a MySQL DB instance \(p. 835\)](#)
- [Encrypting client connections with SSL \(p. 1010\)](#)
- [Using SSL with a PostgreSQL DB instance \(p. 1513\)](#)

Important

For information about rotating your certificate, see [Rotating your SSL/TLS certificate \(p. 1636\)](#).

Note

All certificates are only available for download using SSL/TLS connections.

To get a root certificate that works for all AWS Regions, excluding opt-in AWS Regions, download it from <https://s3.amazonaws.com/rds-downloads/rds-ca-2019-root.pem>.

This root certificate is a trusted root entity and should work in most cases but might fail if your application doesn't accept certificate chains. If your application doesn't accept certificate chains, download the AWS Region-specific certificate from the list of intermediate certificates found later in this section.

To get a certificate bundle that contains both the intermediate and root certificates, download from <https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem>.

If your application is on Microsoft Windows and requires a PKCS7 file, you can download the PKCS7 certificate bundle. This bundle contains both the intermediate and root certificates at <https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.p7b>.

Note

Amazon RDS Proxy uses certificates from the AWS Certificate Manager (ACM). If you are using RDS Proxy, you don't need to download Amazon RDS certificates or update applications that use RDS Proxy connections. For more information about using TLS/SSL with RDS Proxy, see [Using TLS/SSL with RDS Proxy \(p. 169\)](#).

Root certificates for opt-in AWS Regions

If you are using an opt-in AWS Region, you can download the root certificate from the following table.

Opt-in AWS Region	Root certificate
Africa (Cape Town)	rds-ca-af-south-1-2019-root.pem
Asia Pacific (Hong Kong)	rds-ca-ap-east-1-2019-root.pem
Europe (Milan)	rds-ca-eu-south-1-2019-root.pem

Opt-in AWS Region	Root certificate
Middle East (Bahrain)	rds-ca-me-south-1-2019-root.pem

Intermediate certificates

You might need to use an intermediate certificate to connect to your AWS Region. For example, you must use an intermediate certificate to connect to the AWS GovCloud (US-West) Region using SSL/TLS. If you need an intermediate certificate for a particular AWS Region, download the certificate from the following table.

AWS Region	Intermediate certificate
Asia Pacific (Mumbai)	rds-ca-2019-ap-south-1.pem
Asia Pacific (Tokyo)	rds-ca-2019-ap-northeast-1.pem
Asia Pacific (Seoul)	rds-ca-2019-ap-northeast-2.pem
Asia Pacific (Osaka)	rds-ca-2019-ap-northeast-3.pem
Asia Pacific (Singapore)	rds-ca-2019-ap-southeast-1.pem
Asia Pacific (Sydney)	rds-ca-2019-ap-southeast-2.pem
Canada (Central)	rds-ca-2019-ca-central-1.pem
Europe (Frankfurt)	rds-ca-2019-eu-central-1.pem
Europe (Ireland)	rds-ca-2019-eu-west-1.pem
Europe (London)	rds-ca-2019-eu-west-2.pem
Europe (Paris)	rds-ca-2019-eu-west-3.pem
Europe (Stockholm)	rds-ca-2019-eu-north-1.pem
South America (São Paulo)	rds-ca-2019-sa-east-1.pem
US East (N. Virginia)	rds-ca-2019-us-east-1.pem
US East (Ohio)	rds-ca-2019-us-east-2.pem
US West (N. California)	rds-ca-2019-us-west-1.pem
US West (Oregon)	rds-ca-2019-us-west-2.pem

AWS GovCloud (US) certificates

You can download the root certificate for an AWS GovCloud (US) Region from the following list:

[AWS GovCloud \(US-East\) \(Root CA-2017\)](#)

[AWS GovCloud \(US-West\) \(Root CA-2017\)](#)

You can download the intermediate certificate for an AWS GovCloud (US) Region from the following list:

[AWS GovCloud \(US-East\) \(CA-2017\)](#)

[AWS GovCloud \(US-West\) \(CA-2017\)](#)

AWS GovCloud (US-West) (CA-2012)

To get a certificate bundle that contains both the intermediate and root certificates for the AWS GovCloud (US) Regions, download from <https://s3.us-gov-west-1.amazonaws.com/rds-downloads/rds-combined-ca-us-gov-bundle.pem>.

Rotating your SSL/TLS certificate

As of March 5, 2020, Amazon RDS CA-2015 certificates have expired. If you use or plan to use Secure Sockets Layer (SSL) or Transport Layer Security (TLS) with certificate verification to connect to your RDS DB instances, you require Amazon RDS CA-2019 certificates, which are enabled by default for new DB instances. If you currently do not use SSL/TLS with certificate verification, you might still have expired CA-2015 certificates and must update them to CA-2019 certificates if you plan to use SSL/TLS with certificate verification to connect to your RDS databases.

Follow these instructions to complete your updates. Before you update your DB instances to use the new CA certificate, make sure that you update your clients or applications connecting to your RDS databases.

Amazon RDS provides new CA certificates as an AWS security best practice. For information about the new certificates and the supported AWS Regions, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

Note

Amazon RDS Proxy uses certificates from the AWS Certificate Manager (ACM). If you are using RDS Proxy, when you rotate your SSL/TLS certificate, you don't need to update applications that use RDS Proxy connections. For more information about using TLS/SSL with RDS Proxy, see [Using TLS/SSL with RDS Proxy \(p. 169\)](#).

Note

If you are using a Go version 1.15 application with a DB instance that was created or updated to the rds-ca-2019 certificate prior to July 28, 2020, you must update the certificate again. Run the `modify-db-instance` command shown in the AWS CLI section using `rds-ca-2019` as the CA certificate identifier. In this case, it isn't possible to update the certificate using the AWS Management Console. If you created your DB instance or updated its certificate after July 28, 2020, no action is required. For more information, see [Go GitHub issue #39568](#).

Topics

- [Updating your CA certificate by modifying your DB instance \(p. 1636\)](#)
- [Updating your CA certificate by applying DB instance maintenance \(p. 1639\)](#)
- [Sample script for importing certificates into your trust store \(p. 1642\)](#)

Updating your CA certificate by modifying your DB instance

Complete the following steps to update your CA certificate.

To update your CA certificate by modifying your DB instance

1. Download the new SSL/TLS certificate as described in [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).
2. Update your applications to use the new SSL/TLS certificate.

The methods for updating applications for new SSL/TLS certificates depend on your specific applications. Work with your application developers to update the SSL/TLS certificates for your applications.

For information about checking for SSL/TLS connections and updating applications for each DB engine, see the following topics:

- [Updating applications to connect to MariaDB DB instances using new SSL/TLS certificates \(p. 594\)](#)

- [Updating applications to connect to Microsoft SQL Server DB instances using new SSL/TLS certificates \(p. 662\)](#)
- [Updating applications to connect to MySQL DB instances using new SSL/TLS certificates \(p. 848\)](#)
- [Updating applications to use new SSL/TLS certificates \(p. 1011\)](#)
- [Updating applications to connect to PostgreSQL DB instances using new SSL/TLS certificates \(p. 1516\)](#)

For a sample script that updates a trust store for a Linux operating system, see [Sample script for importing certificates into your trust store \(p. 1642\)](#).

Note

The certificate bundle contains certificates for both the old and new CA, so you can upgrade your application safely and maintain connectivity during the transition period. If you are using the AWS Database Migration Service to migrate a database to a DB instance, we recommend using the certificate bundle to ensure connectivity during the migration.

3. Modify the DB instance to change the CA from **rds-ca-2015** to **rds-ca-2019**.

Important

By default, this operation restarts your DB instance. If you don't want to restart your DB instance during this operation, you can use the `modify-db-instance` CLI command and specify the `--no-certificate-rotation-restart` option.

This option will not rotate the certificate until the next time the database restarts, either for planned or unplanned maintenance. This option is only recommended if you don't use SSL/TLS.

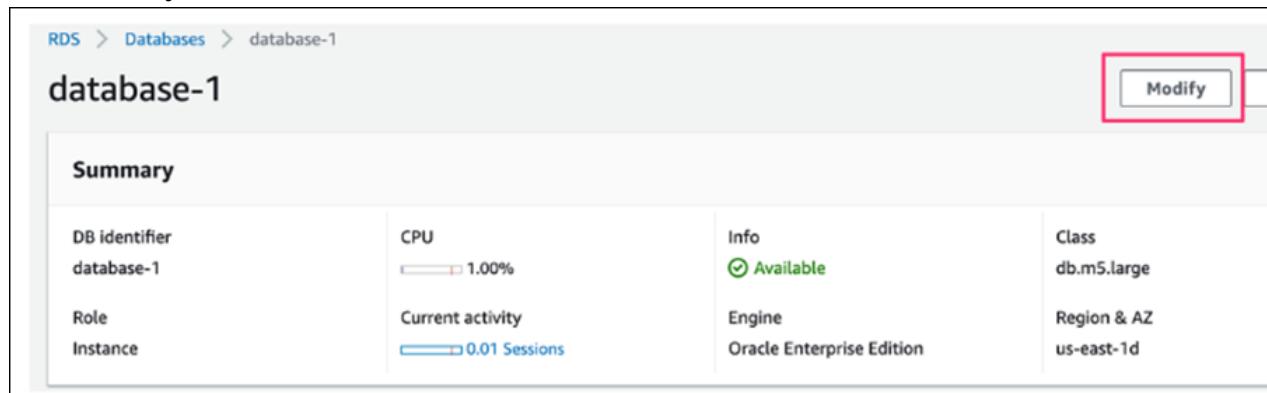
If you are experiencing connectivity issues after certificate expiry, use the `apply immediately` option by specifying **Apply immediately** in the console or by specifying the `--apply-immediately` option using the AWS CLI. By default, this operation is scheduled to run during your next maintenance window.

You can use the AWS Management Console or the AWS CLI to change the CA certificate from **rds-ca-2015** to **rds-ca-2019** for a DB instance.

Console

To change the CA from rds-ca-2015 to rds-ca-2019 for a DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
3. Choose **Modify**.



The **Modify DB Instance** page appears.

4. In the **Network & Security** section, choose **rds-ca-2019**.

Certificate authority
Certificate authority for this DB instance

rds-ca-2019

rds-ca-2015

rds-ca-2019

EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You can use more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

No

5. Choose **Continue** and check the summary of modifications.
6. To apply the changes immediately, choose **Apply immediately**.

Important

Choosing this option restarts your database immediately.

7. On the confirmation page, review your changes. If they are correct, choose **Modify DB Instance** to save your changes.

Important

When you schedule this operation, make sure that you have updated your client-side trust store beforehand.

Or choose **Back** to edit your changes or **Cancel** to cancel your changes.

AWS CLI

To use the AWS CLI to change the CA from **rds-ca-2015** to **rds-ca-2019** for a DB instance, call the [modify-db-instance](#) command. Specify the DB instance identifier and the `--ca-certificate-identifier` option.

Important

When you schedule this operation, make sure that you have updated your client-side trust store beforehand.

Example

The following code modifies `mydbinstance` by setting the CA certificate to `rds-ca-2019`. The changes are applied during the next maintenance window by using `--no-apply-immediately`. Use `--apply-immediately` to apply the changes immediately.

Important

By default, this operation reboots your DB instance. If you don't want to reboot your DB instance during this operation, you can use the `modify-db-instance` CLI command and specify the `--no-certificate-rotation-restart` option.

This option will not rotate the certificate until the next time the database restarts, either for planned or unplanned maintenance. This option is only recommended if you do not use SSL/TLS.

Use `--apply-immediately` to apply the update immediately. By default, this operation is scheduled to run during your next maintenance window.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
```

```
--db-instance-identifier mydbinstance \
--ca-certificate-identifier rds-ca-2019 \
--no-apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--ca-certificate-identifier rds-ca-2019 ^
--no-apply-immediately
```

Updating your CA certificate by applying DB instance maintenance

Complete the following steps to update your CA certificate by applying DB instance maintenance.

To update your CA certificate by applying DB instance maintenance

1. Download the new SSL/TLS certificate as described in [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).
2. Update your database applications to use the new SSL/TLS certificate.

The methods for updating applications for new SSL/TLS certificates depend on your specific applications. Work with your application developers to update the SSL/TLS certificates for your applications.

For information about checking for SSL/TLS connections and updating applications for each DB engine, see the following topics:

- [Updating applications to connect to MariaDB DB instances using new SSL/TLS certificates \(p. 594\)](#)
- [Updating applications to connect to Microsoft SQL Server DB instances using new SSL/TLS certificates \(p. 662\)](#)
- [Updating applications to connect to MySQL DB instances using new SSL/TLS certificates \(p. 848\)](#)
- [Updating applications to use new SSL/TLS certificates \(p. 1011\)](#)
- [Updating applications to connect to PostgreSQL DB instances using new SSL/TLS certificates \(p. 1516\)](#)

For a sample script that updates a trust store for a Linux operating system, see [Sample script for importing certificates into your trust store \(p. 1642\)](#).

Note

The certificate bundle contains certificates for both the old and new CA, so you can upgrade your application safely and maintain connectivity during the transition period.

3. Apply DB instance maintenance to change the CA from **rds-ca-2015** to **rds-ca-2019**.

Important

You can choose to apply the change immediately. By default, this operation is scheduled to run during your next maintenance window.

You can use the AWS Management Console to apply DB instance maintenance to change the CA certificate from **rds-ca-2015** to **rds-ca-2019** for multiple DB instances.

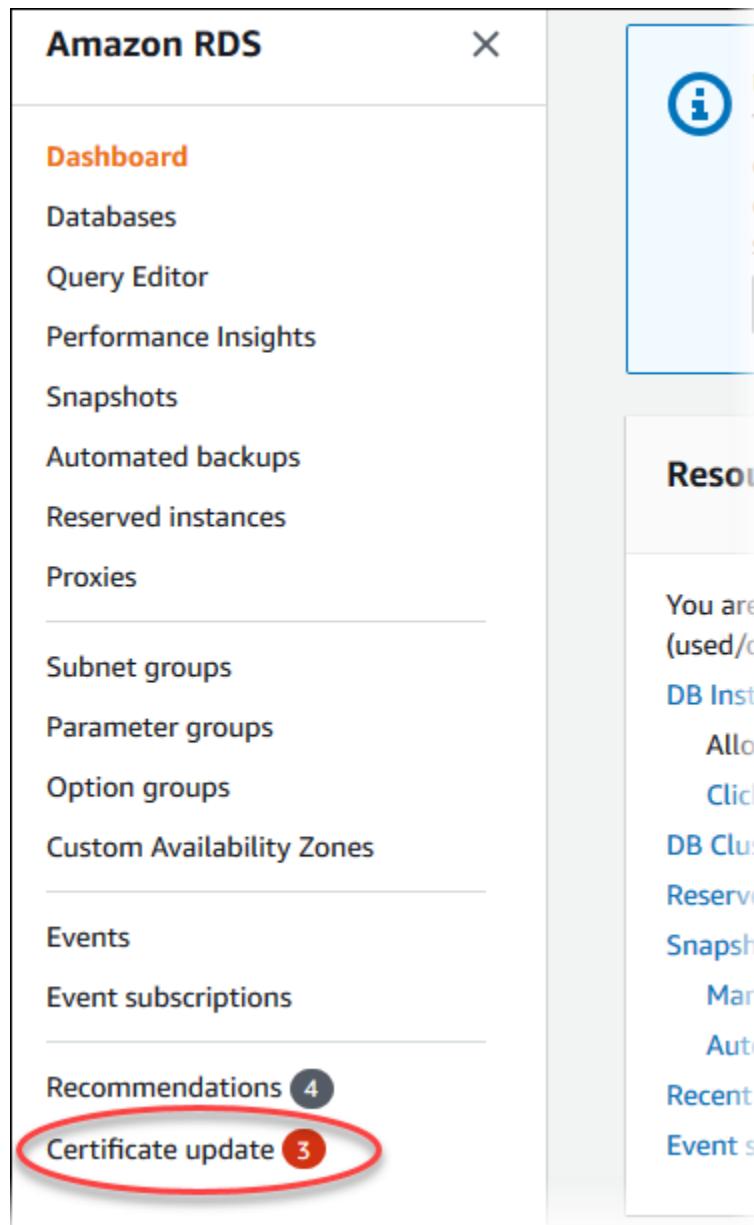
Updating your CA certificate by applying maintenance to multiple DB instances

Use the AWS Management Console to change the CA certificate for multiple DB instances.

To change the CA from rds-ca-2015 to rds-ca-2019 for multiple DB instances

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
 2. In the navigation pane, choose **Databases**.

In the navigation pane, there is a **Certificate update** option that shows the total number of affected DB instances.



Choose **Certificate update** in the navigation pane.

The **Update your Amazon RDS SSL/TLS certificates** page appears.

DB identifier	DB cluster identifier	Status	Apply date
mydbinstancecf	-	Requires Update	-
mydbinstancecf2	-	Requires Update	-
oracledb	-	Requires Update	-

Note

This page only shows the DB instances for the current AWS Region. If you have DB instances in more than one AWS Region, check this page in each AWS Region to see all DB instances with old SSL/TLS certificates.

3. Choose the DB instance you want to update.

You can schedule the certificate rotation for your next maintenance window by choosing **Update at the next maintenance window**. Apply the rotation immediately by choosing **Update now**.

Important

When your CA certificate is rotated, the operation restarts your DB instance.

If you experience connectivity issues after certificate expiry, use the **Update now** option.

4. If you choose **Update at the next maintenance window** or **Update now**, you are prompted to confirm the CA certificate rotation.

Important

Before scheduling the CA certificate rotation on your database, update any client applications that use SSL/TLS and the server certificate to connect. These updates are specific to your DB engine. To determine whether your applications use SSL/TLS and the server certificate to connect, see [Step 2: Update Your Database Applications to Use the New SSL/TLS Certificate \(p. 1639\)](#). After you have updated these client applications, you can confirm the CA certificate rotation.

Confirm rotation of CA certificate?

Before scheduling the CA certificate rotation, update client applications that connect to your database to use the new CA certificate. Not doing this will cause an interruption of connectivity between your applications and your database. [Get new CA certificates](#)

I understand that not doing so will break SSL/TLS connectivity to my database.

Cancel

Confirm

To continue, choose the check box, and then choose **Confirm**.

5. Repeat steps 3 and 4 for each DB instance that you want to update.

Sample script for importing certificates into your trust store

The following are sample shell scripts that import the certificate bundle into a trust store.

Topics

- [Sample script for importing certificates on Linux \(p. 1642\)](#)
- [Sample script for importing certificates on macOS \(p. 1642\)](#)

Sample script for importing certificates on Linux

The following is a sample shell script that imports the certificate bundle into a trust store on a Linux operating system.

```
mydir=tmp/certs
if [ ! -e "${mydir}" ]
then
mkdir -p "${mydir}"
fi

truststore=${mydir}/rds-truststore.jks
storepassword=changeit

curl -ss "https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem" > ${mydir}/
rds-combined-ca-bundle.pem
awk 'split_after == 1 {n++;split_after=0} /-----END CERTIFICATE-----/ {split_after=1}{print
> "rds-ca-" n ".pem"}' < ${mydir}/rds-combined-ca-bundle.pem

for CERT in rds-ca-*; do
alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /Subject:/; s/.*(CN=|CN = )//; print')
echo "Importing $alias"
keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -keystore
${truststore} -noprompt
rm ${CERT}
done

rm ${mydir}/rds-combined-ca-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep Alias | cut -d
" " -f3- | while read alias
do
    expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword} -alias
"${alias}" | grep Valid | perl -ne 'if(/until: (.*)\n/) { print "$1\n"; }'`|
    echo " Certificate ${alias} expires in '$expiry'"
done
```

Sample script for importing certificates on macOS

The following is a sample shell script that imports the certificate bundle into a trust store on macOS.

```
mydir=tmp/certs
if [ ! -e "${mydir}" ]
then
mkdir -p "${mydir}"
fi
```

```
truststore=${mydir}/rds-truststore.jks
storepassword=changeit

curl -sS "https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem" > ${mydir}/
rds-combined-ca-bundle.pem
split -p "-----BEGIN CERTIFICATE-----" ${mydir}/rds-combined-ca-bundle.pem rds-ca-
for CERT in rds-ca-*; do
    alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /Subject:/; s/.*(CN=|CN = )//; print')
    echo "Importing $alias"
    keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -keystore
${truststore} -noprompt
    rm ${CERT}
done

rm ${mydir}/rds-combined-ca-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep Alias | cut -d
" " -f3- | while read alias
do
    expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword} -alias
"${alias}" | grep Valid | perl -ne 'if(/until: (.*)\n/) { print "$1\n"; }'`^
    echo " Certificate ${alias} expires in '$expiry'"
done
```

Internetwork traffic privacy

Connections are protected both between Amazon RDS and on-premises applications and between Amazon RDS and other AWS resources within the same AWS Region.

Traffic between service and on-premises clients and applications

You have two connectivity options between your private network and AWS:

- An AWS Site-to-Site VPN connection. For more information, see [What is AWS Site-to-Site VPN?](#)
- An AWS Direct Connect connection. For more information, see [What is AWS Direct Connect?](#)

You get access to Amazon RDS through the network by using AWS-published API operations. Clients must support Transport Layer Security (TLS) 1.0. We recommend TLS 1.2. Clients must also support cipher suites with Perfect Forward Secrecy (PFS), such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). Most modern systems such as Java 7 and later support these modes. Additionally, you must sign requests using an access key identifier and a secret access key that are associated with an IAM principal. Or you can use the [AWS security token service \(STS\)](#) to generate temporary security credentials to sign requests.

Identity and access management in Amazon RDS

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Amazon RDS resources. IAM is an AWS service that you can use with no additional charge.

Topics

- [Audience \(p. 1644\)](#)
- [Authenticating with identities \(p. 1644\)](#)
- [Managing access using policies \(p. 1646\)](#)
- [How Amazon RDS works with IAM \(p. 1648\)](#)
- [Amazon RDS identity-based policy examples \(p. 1650\)](#)
- [IAM database authentication for MySQL and PostgreSQL \(p. 1660\)](#)
- [Troubleshooting Amazon RDS identity and access \(p. 1689\)](#)

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work you do in Amazon RDS.

Service user – If you use the Amazon RDS service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Amazon RDS features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Amazon RDS, see [Troubleshooting Amazon RDS identity and access \(p. 1689\)](#).

Service administrator – If you're in charge of Amazon RDS resources at your company, you probably have full access to Amazon RDS. It's your job to determine which Amazon RDS features and resources your employees should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Amazon RDS, see [How Amazon RDS works with IAM \(p. 1648\)](#).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Amazon RDS. To view example Amazon RDS identity-based policies that you can use in IAM, see [Amazon RDS identity-based policy examples \(p. 1650\)](#).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see [The IAM console and sign-in page](#) in the *IAM User Guide*.

You must be *authenticated* (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication, or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the [AWS Management Console](#), use your password with your root user email or your IAM user name. You can access AWS programmatically using your root user or IAM user access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. Do this using *Signature Version 4*, a protocol

for authenticating inbound API requests. For more information about authenticating requests, see [Signature Version 4 signing process](#) in the *AWS General Reference*.

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see [Using multi-factor authentication \(MFA\) in AWS](#) in the *IAM User Guide*.

AWS account root user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the *AWS account root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the [best practice of using the root user only to create your first IAM user](#). Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see [Managing access keys for IAM users](#) in the *IAM User Guide*. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see [When to create an IAM user \(instead of a role\)](#) in the *IAM User Guide*.

You can authenticate to your DB instance using IAM database authentication.

IAM database authentication works with the following DB engines:

- Amazon RDS for MySQL
- Amazon RDS for PostgreSQL

For more information about authenticating to your DB instance using IAM, see [IAM database authentication for MySQL and PostgreSQL \(p. 1660\)](#).

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by [switching roles](#). You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see [Using IAM roles](#) in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an [identity provider](#). For more information about federated users, see [Federated users and roles](#) in the *IAM User Guide*.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.
- **AWS service access** – A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using an IAM role to grant permissions to applications running on Amazon EC2 instances](#) in the *IAM User Guide*.

To learn whether to use IAM roles, see [When to create an IAM role \(instead of a user\)](#) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when an entity (root user, IAM user, or IAM role) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see [Overview of JSON policies](#) in the *IAM User Guide*.

An IAM administrator can use policies to specify who has access to AWS resources, and what actions they can perform on those resources. Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, role, or group. These policies control what actions that identity can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see [Creating IAM policies](#) in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see [Choosing between managed policies and inline policies](#) in the *IAM User Guide*.

The following AWS managed policies, which you can attach to users in your account, are specific to Amazon RDS:

- **AmazonRDSReadOnlyAccess** – Grants read-only access to all Amazon RDS resources for the AWS account specified.
- **AmazonRDSFullAccess** – Grants full access to all Amazon RDS resources for the AWS account specified.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see [Permissions boundaries for IAM entities](#) in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see [How SCPs work](#) in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see [Session policies](#) in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see [Policy evaluation logic](#) in the *IAM User Guide*.

For more information about identity and access management for Amazon RDS, continue to the following pages:

- [How Amazon RDS works with IAM \(p. 1648\)](#)
- [Troubleshooting Amazon RDS identity and access \(p. 1689\)](#)

How Amazon RDS works with IAM

Before you use IAM to manage access to Amazon RDS, you should understand what IAM features are available to use with Amazon RDS. To get a high-level view of how Amazon RDS and other AWS services work with IAM, see [AWS services that work with IAM](#) in the *IAM User Guide*.

Topics

- [Amazon RDS identity-based policies \(p. 1648\)](#)
- [Amazon RDS resource-based policies \(p. 1650\)](#)
- [Authorization based on Amazon RDS tags \(p. 1650\)](#)
- [Amazon RDS IAM roles \(p. 1650\)](#)

Amazon RDS identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. Amazon RDS supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see [IAM JSON policy elements reference](#) in the *IAM User Guide*.

Actions

The `Action` element of an IAM identity-based policy describes the specific action or actions that will be allowed or denied by the policy. Policy actions usually have the same name as the associated AWS API operation. The action is used in a policy to grant permissions to perform the associated operation.

Policy actions in Amazon RDS use the following prefix before the action: `rds:`. For example, to grant someone permission to describe DB instances with the Amazon RDS `DescribeDBInstances` API operation, you include the `rds:DescribeDBInstances` action in their policy. Policy statements must include either an `Action` or `NotAction` element. Amazon RDS defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": [  
    "rds:action1",  
    "rds:action2"]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action:

```
"Action": "rds:Describe*"
```

To see a list of Amazon RDS actions, see [Actions Defined by Amazon RDS in the Service Authorization Reference](#).

Resources

The `Resource` element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. You specify a resource using an ARN or using the wildcard (*) to indicate that the statement applies to all resources.

The DB instance resource has the following ARN:

```
arn:${Partition}:rds:${Region}:${Account}:{ResourceType}/${Resource}
```

For more information about the format of ARNs, see [Amazon Resource Names \(ARNs\) and AWS service namespaces](#).

For example, to specify the dbtest DB instance in your statement, use the following ARN:

```
"Resource": "arn:aws:rds:us-west-2:123456789012:db:dbtest"
```

To specify all DB instances that belong to a specific account, use the wildcard (*):

```
"Resource": "arn:aws:rds:us-east-1:123456789012:db:/*"
```

Some RDS API operations, such as those for creating resources, cannot be performed on a specific resource. In those cases, you must use the wildcard (*).

```
"Resource": "*"
```

Many Amazon RDS API operations involve multiple resources. For example, `CreateDBInstance` creates a DB instance. You can specify that an IAM user must use a specific security group and parameter group when creating a DB instance. To specify multiple resources in a single statement, separate the ARNs with commas.

```
"Resource": [  
    "resource1",  
    "resource2"]
```

To see a list of Amazon RDS resource types and their ARNs, see [Resources Defined by Amazon RDS](#) in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see [Actions Defined by Amazon RDS](#).

Condition keys

The `Condition` element (or `Condition block`) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can build conditional expressions that use [condition operators](#), such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical AND operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical OR operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see [IAM policy elements: Variables and tags](#) in the *IAM User Guide*.

Amazon RDS defines its own set of condition keys and also supports using some global condition keys. To see all AWS global condition keys, see [AWS global condition context keys](#) in the *IAM User Guide*.

All RDS API operations support the `aws:RequestedRegion` condition key.

To see a list of Amazon RDS condition keys, see [Condition Keys for Amazon RDS](#) in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see [Actions Defined by Amazon RDS](#).

Examples

To view examples of Amazon RDS identity-based policies, see [Amazon RDS identity-based policy examples \(p. 1650\)](#).

Amazon RDS resource-based policies

Amazon RDS does not support resource-based policies.

Authorization based on Amazon RDS tags

You can attach tags to Amazon RDS resources or pass tags in a request to Amazon RDS. To control access based on tags, you provide tag information in the [condition element](#) of a policy using the `rds:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys. For more information about tagging Amazon RDS resources, see [Specifying conditions: Using custom tags \(p. 1657\)](#).

To view an example identity-based policy for limiting access to a resource based on the tags on that resource, see [Grant permission for actions on a resource with a specific tag with two different values \(p. 1655\)](#).

Amazon RDS IAM roles

An [IAM role](#) is an entity within your AWS account that has specific permissions.

Using temporary credentials with Amazon RDS

You can use temporary credentials to sign in with federation, assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as [AssumeRole](#) or [GetFederationToken](#).

Amazon RDS supports using temporary credentials.

Service-linked roles

[Service-linked roles](#) allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in the Roles list in the IAM Management Console and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

Amazon RDS supports service-linked roles. For details about creating or managing Amazon RDS service-linked roles, see [Using service-linked roles for Amazon RDS \(p. 1714\)](#).

Service roles

This feature allows a service to assume a [service role](#) on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in the Roles list in the IAM Management Console and are owned by your account. This means that an IAM administrator can change the permissions for this role. However, doing so might break the functionality of the service.

Amazon RDS supports service roles.

Amazon RDS identity-based policy examples

By default, IAM users and roles don't have permission to create or modify Amazon RDS resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM

administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see [Creating policies on the JSON tab](#) in the *IAM User Guide*.

Topics

- [Policy best practices \(p. 1651\)](#)
- [Using the Amazon RDS console \(p. 1651\)](#)
- [Allow users to view their own permissions \(p. 1652\)](#)
- [Allow a user to create DB instances in an AWS account \(p. 1652\)](#)
- [Permissions required to use the console \(p. 1654\)](#)
- [Allow a user to perform any describe action on any RDS resource \(p. 1654\)](#)
- [Allow a user to create a DB instance that uses the specified DB parameter and security groups \(p. 1654\)](#)
- [Grant permission for actions on a resource with a specific tag with two different values \(p. 1655\)](#)
- [Prevent a user from deleting a DB instance \(p. 1655\)](#)
- [Deny all access to a resource \(p. 1656\)](#)
- [Example policies: Using condition keys \(p. 1656\)](#)
- [Specifying conditions: Using custom tags \(p. 1657\)](#)

Policy best practices

Identity-based policies are very powerful. They determine whether someone can create, access, or delete Amazon RDS resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get Started Using AWS Managed Policies** – To start using Amazon RDS quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see [Get started using permissions with AWS managed policies](#) in the *IAM User Guide*.
- **Grant Least Privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see [Grant least privilege](#) in the *IAM User Guide*.
- **Enable MFA for Sensitive Operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see [Using multi-factor authentication \(MFA\) in AWS](#) in the *IAM User Guide*.
- **Use Policy Conditions for Extra Security** – To the extent that it's practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see [IAM JSON policy elements: Condition](#) in the *IAM User Guide*.

Using the Amazon RDS console

To access the Amazon RDS console, you must have a minimum set of permissions. These permissions must enable you to list and view details about the Amazon RDS resources in your AWS account. You can

create an identity-based policy that is more restrictive than the minimum required permissions. However, if you do, the console doesn't function as intended for entities (IAM users or roles) with that policy.

To ensure that those entities can still use the Amazon RDS console, also attach the following AWS managed policy to the entities. For more information, see [Adding permissions to a user](#) in the *IAM User Guide*.

AmazonRDSReadOnlyAccess

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that you're trying to perform.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ViewOwnUserInfo",  
            "Effect": "Allow",  
            "Action": [  
                "iam:GetUserPolicy",  
                "iam>ListGroupsForUser",  
                "iam>ListAttachedUserPolicies",  
                "iam>ListUserPolicies",  
                "iam:GetUser"  
            ],  
            "Resource": [  
                "arn:aws:iam::*:user/${aws:username}"  
            ]  
        },  
        {  
            "Sid": "NavigateInConsole",  
            "Effect": "Allow",  
            "Action": [  
                "iam:GetGroupPolicy",  
                "iam:GetPolicyVersion",  
                "iam:GetPolicy",  
                "iam>ListAttachedGroupPolicies",  
                "iam>ListGroupPolicies",  
                "iam>ListPolicyVersions",  
                "iam>ListPolicies",  
                "iam>ListUsers"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Allow a user to create DB instances in an AWS account

The following is an example policy that allows the user with the ID 123456789012 to create DB instances for your AWS account. The policy requires that the name of the new DB instance begin with test. The new DB instance must also use the MySQL database engine and the db.t2.micro DB

instance class. In addition, the new DB instance must use an option group and a DB parameter group that starts with default, and it must use the default subnet group.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDBInstanceOnly",
            "Effect": "Allow",
            "Action": [
                "rds:CreateDBInstance"
            ],
            "Resource": [
                "arn:aws:rds:*:123456789012:db:test*",
                "arn:aws:rds:*:123456789012:og:default*",
                "arn:aws:rds:*:123456789012:pg:default*",
                "arn:aws:rds:*:123456789012:subgrp:default"
            ],
            "Condition": {
                "StringEquals": {
                    "rds:DatabaseEngine": "mysql",
                    "rds:DatabaseClass": "db.t2.micro"
                }
            }
        }
    ]
}
```

The policy includes a single statement that specifies the following permissions for the IAM user:

- The policy allows the IAM user to create a DB instance using the [CreateDBInstance](#) API operation (this also applies to the [create-db-instance](#) AWS CLI command and the AWS Management Console).
- The Resource element specifies that the user can perform actions on or with resources. You specify resources using an Amazon Resources Name (ARN). This ARN includes the name of the service that the resource belongs to (rds), the AWS Region (* indicates any region in this example), the user account number (123456789012 is the user ID in this example), and the type of resource. For more information about creating ARNs, see [Working with Amazon Resource Names \(ARNs\) in Amazon RDS \(p. 309\)](#).

The Resource element in the example specifies the following policy constraints on resources for the user:

- The DB instance identifier for the new DB instance must begin with test (for example, testCustomerData1, test-region2-data).
- The option group for the new DB instance must begin with default.
- The DB parameter group for the new DB instance must begin with default.
- The subnet group for the new DB instance must be the default subnet group.
- The Condition element specifies that the DB engine must be MySQL and the DB instance class must be db.t2.micro. The Condition element specifies the conditions when a policy should take effect. You can add additional permissions or restrictions by using the Condition element. For more information about specifying conditions, see [Condition keys \(p. 1649\)](#). This example specifies the rds:DatabaseEngine and rds:DatabaseClass conditions. For information about the valid condition values for rds:DatabaseEngine, see the list under the Engine parameter in [CreateDBInstance](#). For information about the valid condition values for rds:DatabaseClass, see [Supported DB engines for DB instance classes \(p. 8\)](#).

The policy doesn't specify the Principal element because in an identity-based policy you don't specify the principal who gets the permission. When you attach policy to a user, the user is the implicit principal.

When you attach a permission policy to an IAM role, the principal identified in the role's trust policy gets the permissions.

To see a list of Amazon RDS actions, see [Actions Defined by Amazon RDS in the Service Authorization Reference](#).

Permissions required to use the console

For a user to work with the console, that user must have a minimum set of permissions. These permissions allow the user to describe the Amazon RDS resources for their AWS account and to provide other related information, including Amazon EC2 security and network information.

If you create an IAM policy that is more restrictive than the minimum required permissions, the console doesn't function as intended for users with that IAM policy. To ensure that those users can still use the console, also attach the `AmazonRDSReadOnlyAccess` managed policy to the user, as described in [Managing access using policies \(p. 1646\)](#).

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the Amazon RDS API.

The following policy grants full access to all Amazon RDS resources for the root AWS account:

```
AmazonRDSFullAccess
```

Allow a user to perform any describe action on any RDS resource

The following permissions policy grants permissions to a user to run all of the actions that begin with `Describe`. These actions show information about an RDS resource, such as a DB instance. The wildcard character (*) in the `Resource` element indicates that the actions are allowed for all Amazon RDS resources owned by the account.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRDSDescribe",
            "Effect": "Allow",
            "Action": "rds:Describe*",
            "Resource": "*"
        }
    ]
}
```

Allow a user to create a DB instance that uses the specified DB parameter and security groups

The following permissions policy grants permissions to allow a user to only create a DB instance that must use the `mysql-production` DB parameter group and the `db-production` DB security group.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
{
    "Sid": "AllowMySQLProductionCreate",
    "Effect": "Allow",
    "Action": "rds:CreateDBInstance",
    "Resource": [
        "arn:aws:rds:us-west-2:123456789012:pg:mysql-production",
        "arn:aws:rds:us-west-2:123456789012:secgrp:db-production"
    ]
}
]
```

Grant permission for actions on a resource with a specific tag with two different values

You can use conditions in your identity-based policy to control access to Amazon RDS resources based on tags. The following policy allows permission to perform the `ModifyDBInstance` and `CreateDBSnapshot` APIs on DB instances with either the `stage` tag set to `development` or `test`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDevTestCreate",
            "Effect": "Allow",
            "Action": [
                "rds:ModifyDBInstance",
                "rds:CreateDBSnapshot"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "rds:db-tag/stage": [
                        "development",
                        "test"
                    ]
                }
            }
        }
    ]
}
```

Prevent a user from deleting a DB instance

The following permissions policy grants permissions to prevent a user from deleting a specific DB instance. For example, you might want to deny the ability to delete your production DB instances to any user that is not an administrator.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyDelete1",
            "Effect": "Deny",
            "Action": "rds:DeleteDBInstance",
            "Resource": "arn:aws:rds:us-west-2:123456789012:db:my-mysql-instance"
        }
    ]
}
```

Deny all access to a resource

You can explicitly deny access to a resource. Deny policies take precedence over allow policies. The following policy explicitly denies a user the ability to manage a resource:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": "rds:*",  
            "Resource": "arn:aws:rds:us-east-1:123456789012:db:mydb"  
        }  
    ]  
}
```

Example policies: Using condition keys

Following are examples of how you can use condition keys in Amazon RDS IAM permissions policies.

Example 1: Grant permission to create a DB instance that uses a specific DB engine and isn't MultiAZ

The following policy uses an RDS condition key and allows a user to create only DB instances that use the MySQL database engine and don't use MultiAZ. The Condition element indicates the requirement that the database engine is MySQL.

```
{  
  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowMySQLCreate",  
            "Effect": "Allow",  
            "Action": "rds:CreateDBInstance",  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "rds:DatabaseEngine": "mysql"  
                },  
                "Bool": {  
                    "rds:MultiAz": false  
                }  
            }  
        }  
    ]  
}
```

Example 2: Explicitly deny permission to create DB instances for certain DB instance classes and create DB instances that use Provisioned IOPS

The following policy explicitly denies permission to create DB instances that use the DB instance classes `r3.8xlarge` and `m4.10xlarge`, which are the largest and most expensive DB instance classes. This policy also prevents users from creating DB instances that use Provisioned IOPS, which incurs an additional cost.

Explicitly denying permission supersedes any other permissions granted. This ensures that identities to not accidentally get permission that you never want to grant.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyLargeCreate",
            "Effect": "Deny",
            "Action": "rds>CreateDBInstance",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "rds:DatabaseClass": [
                        "db.r3.8xlarge",
                        "db.m4.10xlarge"
                    ]
                }
            }
        },
        {
            "Sid": "DenyPIOPSCreate",
            "Effect": "Deny",
            "Action": "rds>CreateDBInstance",
            "Resource": "*",
            "Condition": {
                "NumericNotEquals": {
                    "rds:Piops": "0"
                }
            }
        }
    ]
}
```

Example 3: Limit the set of tag keys and values that can be used to tag a resource

The following policy uses an RDS condition key and allows the addition of a tag with the key `stage` to be added to a resource with the values `test`, `qa`, and `production`.

```
{
    {
        "Version" : "2012-10-17",
        "Statement" : [
            {
                "Effect" : "Allow",
                "Action" : [ "rds:AddTagsToResource", "rds:RemoveTagsFromResource" ],
                "Resource" : "*",
                "Condition" : { "streq" : { "rds:req-tag/stage" : [ "test", "qa", "production" ] } }
            }
        ]
    }
}
```

Specifying conditions: Using custom tags

Amazon RDS supports specifying conditions in an IAM policy using custom tags.

For example, suppose that you add a tag named `environment` to your DB instances with values such as `beta`, `staging`, `production`, and so on. If you do, you can create a policy that restricts certain users to DB instances based on the `environment` tag value.

Note

Custom tag identifiers are case-sensitive.

The following table lists the RDS tag identifiers that you can use in a Condition element.

RDS tag identifier	Applies to
db-tag	DB instances, including read replicas
snapshot-tag	DB snapshots
ri-tag	Reserved DB instances
secgrp-tag	DB security groups
og-tag	DB option groups
pg-tag	DB parameter groups
subgrp-tag	DB subnet groups
es-tag	Event subscriptions
cluster-tag	DB clusters
cluster-pg-tag	DB cluster parameter groups
cluster-snapshot-tag	DB cluster snapshots

The syntax for a custom tag condition is as follows:

```
"Condition": {"StringEquals": {"rds:rds-tag-identifier/tag-name": ["value"]}} }
```

For example, the following Condition element applies to DB instances with a tag named environment and a tag value of production.

```
"Condition": {"StringEquals": {"rds:db-tag/environment": ["production"]}} }
```

For information about creating tags, see [Tagging Amazon RDS resources \(p. 299\)](#).

Important

If you manage access to your RDS resources using tagging, we recommend that you secure access to the tags for your RDS resources. You can manage access to tags by creating policies for the AddTagsToResource and RemoveTagsFromResource actions. For example, the following policy denies users the ability to add or remove tags for all resources. You can then create policies to allow specific users to add or remove tags.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyTagUpdates",
            "Effect": "Deny",
            "Action": [
                "rds:AddTagsToResource",
                "rds:RemoveTagsFromResource"
            ],
            "Resource": "*"
        }
    ]
}
```

}

To see a list of Amazon RDS actions, see [Actions Defined by Amazon RDS in the Service Authorization Reference](#).

Example policies: Using custom tags

Following are examples of how you can use custom tags in Amazon RDS IAM permissions policies. For more information about adding tags to an Amazon RDS resource, see [Working with Amazon Resource Names \(ARNs\) in Amazon RDS \(p. 309\)](#).

Note

All examples use the us-west-2 region and contain fictitious account IDs.

Example 1: Grant permission for actions on a resource with a specific tag with two different values

The following policy allows permission to perform the `ModifyDBInstance` and `CreateDBSnapshot` APIs on DB instances with either the `stage` tag set to `development` or `test`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowDevTestCreate",  
            "Effect": "Allow",  
            "Action": [  
                "rds:ModifyDBInstance",  
                "rds:CreateDBSnapshot"  
            ],  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "rds:db-tag/stage": [  
                        "development",  
                        "test"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

Example 2: Explicitly deny permission to create a DB instance that uses specified DB parameter groups

The following policy explicitly denies permission to create a DB instance that uses DB parameter groups with specific tag values. You might apply this policy if you require that a specific customer-created DB parameter group always be used when creating DB instances. Policies that use `Deny` are most often used to restrict access that was granted by a broader policy.

Explicitly denying permission supersedes any other permissions granted. This ensures that identities to not accidentally get permission that you never want to grant.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "DenyProductionCreate",  
            "Effect": "Deny",  
            "Action": "rds:CreateDBInstance",  
            "Resource": "*"  
        }  
    ]  
}
```

```
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "rds:pg-tag/usage": "prod"
            }
        }
    ]
}
```

Example 3: Grant permission for actions on a DB instance with an instance name that is prefixed with a user name

The following policy allows permission to call any API (except to `AddTagsToResource` or `RemoveTagsFromResource`) on a DB instance that has a DB instance name that is prefixed with the user's name and that has a tag called `stage` equal to `devo` or that has no tag called `stage`.

The Resource line in the policy identifies a resource by its Amazon Resource Name (ARN). For more information about using ARNs with Amazon RDS resources, see [Working with Amazon Resource Names \(ARNs\) in Amazon RDS \(p. 309\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowFullDevAccessNoTags",
            "Effect": "Allow",
            "NotAction": [
                "rds:AddTagsToResource",
                "rds:RemoveTagsFromResource"
            ],
            "Resource": "arn:aws:rds:*:123456789012:db:${aws:username}*",
            "Condition": {
                "StringEqualsIfExists": {
                    "rds:db-tag/stage": "devo"
                }
            }
        ]
    ]
}
```

IAM database authentication for MySQL and PostgreSQL

You can authenticate to your DB instance using AWS Identity and Access Management (IAM) database authentication. IAM database authentication works with MySQL and PostgreSQL. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token.

An *authentication token* is a unique string of characters that Amazon RDS generates on request. Authentication tokens are generated using AWS Signature Version 4. Each token has a lifetime of 15 minutes. You don't need to store user credentials in the database, because authentication is managed externally using IAM. You can also still use standard database authentication.

IAM database authentication provides the following benefits:

- Network traffic to and from the database is encrypted using Secure Socket Layer (SSL) or Transport Layer Security (TLS). For more information about using SSL/TLS with Amazon RDS, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

- You can use IAM to centrally manage access to your database resources, instead of managing access individually on each DB instance.
- For applications running on Amazon EC2, you can use profile credentials specific to your EC2 instance to access your database instead of a password, for greater security.

Topics

- [Availability for IAM database authentication \(p. 1661\)](#)
- [Limitations for IAM database authentication \(p. 1661\)](#)
- [MySQL recommendations for IAM database authentication \(p. 1661\)](#)
- [Enabling and disabling IAM database authentication \(p. 1662\)](#)
- [Creating and using an IAM policy for IAM database access \(p. 1664\)](#)
- [Creating a database account using IAM authentication \(p. 1667\)](#)
- [Connecting to your DB instance using IAM authentication \(p. 1668\)](#)

Availability for IAM database authentication

IAM database authentication is available for the following database engines:

- MySQL 8.0, minor version 8.0.16 or higher
- MySQL 5.7, minor version 5.7.16 or higher
- MySQL 5.6, minor version 5.6.34 or higher
- PostgreSQL 13, all minor versions
- PostgreSQL 12, all minor versions
- PostgreSQL 11, all minor versions
- PostgreSQL 10, minor version 10.6 or higher
- PostgreSQL 9.6, minor version 9.6.11 or higher
- PostgreSQL 9.5, minor version 9.5.15 or higher

Limitations for IAM database authentication

When using IAM database authentication, the following limitations apply:

- The maximum number of connections per second for your DB instance might be limited depending on its DB instance class and your workload.
- Currently, IAM database authentication doesn't support all global condition context keys.

For more information about global condition context keys, see [AWS global condition context keys](#) in the *IAM User Guide*.

- Currently, IAM database authentication isn't supported for CNAMEs.
- For PostgreSQL, if the IAM role (`rds_iam`) is added to the master user, IAM authentication takes precedence over Password authentication so the master user has to log in as an IAM user.

MySQL recommendations for IAM database authentication

We recommend the following when using the MySQL DB engine:

- Use IAM database authentication as a mechanism for temporary, personal access to databases.

- Use IAM database authentication only for workloads that can be easily retried.
- Use IAM database authentication when your application requires fewer than 200 new IAM database authentication connections per second.

The database engines that work with Amazon RDS don't impose any limits on authentication attempts per second. However, when you use IAM database authentication, your application must generate an authentication token. Your application then uses that token to connect to the DB instance. If you exceed the limit of maximum new connections per second, then the extra overhead of IAM database authentication can cause connection throttling. The extra overhead can cause even existing connections to drop. For information about the maximum total connections for MySQL, see [Maximum MySQL and MariaDB connections \(p. 1752\)](#).

Enabling and disabling IAM database authentication

By default, IAM database authentication is disabled on DB instances. You can enable or disable IAM database authentication using the AWS Management Console, AWS CLI, or the API.

You can enable IAM database authentication when you perform one of the following actions:

- To create a new DB instance with IAM database authentication enabled, see [Creating an Amazon RDS DB instance \(p. 141\)](#).
- To modify a DB instance to enable IAM database authentication, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
- To restore a DB instance from a snapshot with IAM database authentication enabled, see [Restoring from a DB snapshot \(p. 349\)](#).
- To restore a DB instance to a point in time with IAM database authentication enabled, see [Restoring a DB instance to a specified time \(p. 389\)](#).

IAM authentication for PostgreSQL DB instances requires that the SSL value be 1. You can't enable IAM authentication for a PostgreSQL DB instance if the SSL value is 0. You can't change the SSL value to 0 if IAM authentication is enabled for a PostgreSQL DB instance.

Console

Each creation or modification workflow has a **Database authentication** section, where you can enable or disable IAM database authentication. In that section, choose **Password and IAM database authentication** to enable IAM database authentication.

To enable or disable IAM database authentication for an existing DB instance

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the DB instance that you want to modify.

Note

Make sure that the DB instance is compatible with IAM authentication. Check the compatibility requirements in [Availability for IAM database authentication \(p. 1661\)](#).

4. Choose **Modify**.
5. In the **Database authentication** section, choose **Password and IAM database authentication** to enable IAM database authentication.
6. Choose **Continue**.
7. To apply the changes immediately, choose **Immediately** in the **Scheduling of modifications** section.
8. Choose **Modify DB instance**.

AWS CLI

To create a new DB instance with IAM authentication by using the AWS CLI, use the [create-db-instance](#) command. Specify the --enable-iam-database-authentication option, as shown in the following example.

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m3.medium \
    --engine MySQL \
    --allocated-storage 20 \
    --master-username masterawsuser \
    --master-user-password masteruserpassword \
    --enable-iam-database-authentication
```

To update an existing DB instance to have or not have IAM authentication, use the AWS CLI command [modify-db-instance](#). Specify either the --enable-iam-database-authentication or --no-enable-iam-database-authentication option, as appropriate.

Note

Make sure that the DB instance is compatible with IAM authentication. Check the compatibility requirements in [Availability for IAM database authentication \(p. 1661\)](#).

By default, Amazon RDS performs the modification during the next maintenance window. If you want to override this and enable IAM DB authentication as soon as possible, use the --apply-immediately parameter.

The following example shows how to immediately enable IAM authentication for an existing DB instance.

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --apply-immediately \
    --enable-iam-database-authentication
```

If you are restoring a DB instance, use one of the following AWS CLI commands:

- [restore-db-instance-to-point-in-time](#)
- [restore-db-instance-from-db-snapshot](#)

The IAM database authentication setting defaults to that of the source snapshot. To change this setting, set the --enable-iam-database-authentication or --no-enable-iam-database-authentication option, as appropriate.

RDS API

To create a new DB instance with IAM authentication by using the API, use the API operation [CreateDBInstance](#). Set the EnableIAMDatabaseAuthentication parameter to true.

To update an existing DB instance to have IAM authentication, use the API operation [ModifyDBInstance](#). Set the EnableIAMDatabaseAuthentication parameter to true to enable IAM authentication, or false to disable it.

Note

Make sure that the DB instance is compatible with IAM authentication. Check the compatibility requirements in [Availability for IAM database authentication \(p. 1661\)](#).

If you are restoring a DB instance, use one of the following API operations:

- [RestoreDBInstanceFromDBSnapshot](#)

- [RestoreDBInstanceToPointInTime](#)

The IAM database authentication setting defaults to that of the source snapshot. To change this setting, set the `EnableIAMDatabaseAuthentication` parameter to `true` to enable IAM authentication, or `false` to disable it.

Creating and using an IAM policy for IAM database access

To allow an IAM user or role to connect to your DB instance, you must create an IAM policy. After that, you attach the policy to an IAM user or role.

Note

To learn more about IAM policies, see [Identity and access management in Amazon RDS \(p. 1644\)](#).

The following example policy allows an IAM user to connect to a DB instance using IAM database authentication.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "rds-db:connect"  
            ],  
            "Resource": [  
                "arn:aws:rds-db:us-east-2:1234567890:dbuser:db-ABCDEFGHIJKLM01234/db_user"  
            ]  
        }  
    ]  
}
```

Important

An IAM administrator user can access DB instances without explicit permissions in an IAM policy. The example in [Create an IAM user \(p. 67\)](#) creates an IAM administrator user. If you want to restrict administrator access to DB instances, you can create an IAM role with the appropriate, lesser privileged permissions and assign it to the administrator.

Note

Don't confuse the `rds-db:` prefix with other RDS API operation prefixes that begin with `rds:`. You use the `rds-db:` prefix and the `rds-db:connect` action only for IAM database authentication. They aren't valid in any other context.

Currently, the IAM console displays an error for policies with the `rds-db:connect` action. You can ignore this error.

The example policy includes a single statement with the following elements:

- **Effect** – Specify `Allow` to grant access to the DB instance. If you don't explicitly allow access, then access is denied by default.
- **Action** – Specify `rds-db:connect` to allow connections to the DB instance.
- **Resource** – Specify an Amazon Resource Name (ARN) that describes one database account in one DB instance. The ARN format is as follows.

```
arn:aws:rds-db:region:account-id:dbuser:DbiResourceId/db-user-name
```

In this format, replace the following:

- **region** is the AWS Region for the DB instance. In the example policy, the AWS Region is `us-east-2`.
- **account-id** is the AWS account number for the DB instance. In the example policy, the account number is `1234567890`.
- **DbiResourceId** is the identifier for the DB instance. This identifier is unique to an AWS Region and never changes. In the example policy, the identifier is `db-ABCDEFGHIJKL01234`.

To find a DB instance resource ID in the AWS Management Console for Amazon RDS, choose the DB instance to see its details. Then choose the **Configuration** tab. The **Resource ID** is shown in the **Configuration** section.

Alternatively, you can use the AWS CLI command to list the identifiers and resource IDs for all of your DB instance in the current AWS Region, as shown following.

```
aws rds describe-db-instances --query "DBInstances[*].  
[DBInstanceIdentifier,DbiResourceId]"
```

Note

If you are connecting to a database through RDS Proxy, specify the proxy resource ID, such as `prx-ABCDEFGHIJKL01234`. For information about using IAM database authentication with RDS Proxy, see [Connecting to a proxy using IAM authentication \(p. 183\)](#).

- **db-user-name** is the name of the database account to associate with IAM authentication. In the example policy, the database account is `db_user`.

You can construct other ARNs to support various access patterns. The following policy allows access to two different database accounts in a DB instance.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "rds-db:connect"  
            ],  
            "Resource": [  
                "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKL01234/jane_doe",  
                "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKL01234/mary_roe"  
            ]  
        }  
    ]  
}
```

The following policy uses the `"*"` character to match all DB instances and database accounts for a particular AWS account and AWS Region.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {
```

```
{  
    "Effect": "Allow",  
    "Action": [  
        "rds-db:connect"  
    ],  
    "Resource": [  
        "arn:aws:rds-db:us-east-2:1234567890:dbuser:/*/*"  
    ]  
}  
}  
]
```

The following policy matches all of the DB instances for a particular AWS account and AWS Region. However, the policy only grants access to DB instances that have a `jane_doe` database account.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "rds-db:connect"  
            ],  
            "Resource": [  
                "arn:aws:rds-db:us-east-2:123456789012:dbuser:/*/jane_doe"  
            ]  
        }  
    ]  
}
```

The IAM user or role has access to only those databases that the database user does. For example, suppose that your DB instance has a database named `dev`, and another database named `test`. If the database user `jane_doe` has access only to `dev`, any IAM users or roles that access that DB instance with the `jane_doe` user also have access only to `dev`. This access restriction is also true for other database objects, such as tables, views, and so on.

An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions. For examples of policies, see [Amazon RDS identity-based policy examples \(p. 1650\)](#).

Attaching an IAM policy to an IAM user or role

After you create an IAM policy to allow database authentication, you need to attach the policy to an IAM user or role. For a tutorial on this topic, see [Create and attach your first customer managed policy](#) in the *IAM User Guide*.

As you work through the tutorial, you can use one of the policy examples shown in this section as a starting point and tailor it to your needs. At the end of the tutorial, you have an IAM user with an attached policy that can make use of the `rds-db:connect` action.

Note

You can map multiple IAM users or roles to the same database user account. For example, suppose that your IAM policy specified the following resource ARN.

```
arn:aws:rds-db:us-east-2:123456789012:dbuser:db-12ABC34DEFG5HIJ6KLMNOP78QR/jane_doe
```

If you attach the policy to IAM users *Jane*, *Bob*, and *Diego*, then each of those users can connect to the specified DB instance using the `jane_doe` database account.

Creating a database account using IAM authentication

With IAM database authentication, you don't need to assign database passwords to the user accounts you create. If you remove an IAM user that is mapped to a database account, you should also remove the database account with the `DROP USER` statement.

Note

The user name used for IAM authentication must match the case of the user name in the database.

Topics

- [Using IAM authentication with MySQL \(p. 1667\)](#)
- [Using IAM authentication with PostgreSQL \(p. 1667\)](#)

Using IAM authentication with MySQL

With MySQL, authentication is handled by `AWSAuthenticationPlugin`—an AWS-provided plugin that works seamlessly with IAM to authenticate your IAM users. Connect to the DB instance and issue the `CREATE USER` statement, as shown in the following example.

```
CREATE USER jane_doe IDENTIFIED WITH AWSAuthenticationPlugin AS 'RDS';
```

The `IDENTIFIED WITH` clause allows MySQL to use the `AWSAuthenticationPlugin` to authenticate the database account (`jane_doe`). The `AS 'RDS'` clause refers to the authentication method. Make sure the specified database user name is the same as a resource in the IAM policy for IAM database access. For more information, see [Creating and using an IAM policy for IAM database access \(p. 1664\)](#).

Note

If you see the following message, it means that the AWS-provided plugin is not available for the current DB instance.

`ERROR 1524 (HY000): Plugin 'AWSAuthenticationPlugin' is not loaded`
To troubleshoot this error, verify that you are using a supported configuration and that you have enabled IAM database authentication on your DB instance. For more information, see [Availability for IAM database authentication \(p. 1661\)](#) and [Enabling and disabling IAM database authentication \(p. 1662\)](#).

After you create an account using `AWSAuthenticationPlugin`, you manage it in the same way as other database accounts. For example, you can modify account privileges with `GRANT` and `REVOKE` statements, or modify various account attributes with the `ALTER USER` statement.

Using IAM authentication with PostgreSQL

To use IAM authentication with PostgreSQL, connect to the DB instance, create database users, and then grant them the `rds_iam` role as shown in the following example.

```
CREATE USER db_userx;
GRANT rds_iam TO db_userx;
```

Make sure the specified database user name is the same as a resource in the IAM policy for IAM database access. For more information, see [Creating and using an IAM policy for IAM database access \(p. 1664\)](#).

Connecting to your DB instance using IAM authentication

With IAM database authentication, you use an authentication token when you connect to your DB instance. An *authentication token* is a string of characters that you use instead of a password. After you generate an authentication token, it's valid for 15 minutes before it expires. If you try to connect using an expired token, the connection request is denied.

Every authentication token must be accompanied by a valid signature, using AWS signature version 4. (For more information, see [Signature Version 4 signing process](#) in the *AWS General Reference*.) The AWS CLI and an AWS SDK, such as the AWS SDK for Java or AWS SDK for Python (Boto3), can automatically sign each token you create.

You can use an authentication token when you connect to Amazon RDS from another AWS service, such as AWS Lambda. By using a token, you can avoid placing a password in your code. Alternatively, you can use an AWS SDK to programmatically create and programmatically sign an authentication token.

After you have a signed IAM authentication token, you can connect to an Amazon RDS DB instance. Following, you can find out how to do this using either a command line tool or an AWS SDK, such as the AWS SDK for Java or AWS SDK for Python (Boto3).

For more information, see the following blog posts:

- [Use IAM authentication to connect with SQL Workbench/J to Aurora MySQL or Amazon RDS for MySQL](#)
- [Using IAM authentication to connect with pgAdmin Amazon Aurora PostgreSQL or Amazon RDS for PostgreSQL](#)

The following are prerequisites for connecting to your DB instance using IAM authentication:

- [Enabling and disabling IAM database authentication \(p. 1662\)](#)
- [Creating and using an IAM policy for IAM database access \(p. 1664\)](#)
- [Creating a database account using IAM authentication \(p. 1667\)](#)

Topics

- [Connecting to your DB instance using IAM authentication from the command line: AWS CLI and mysql client \(p. 1668\)](#)
- [Connecting to your DB instance using IAM authentication from the command line: AWS CLI and psql client \(p. 1670\)](#)
- [Connecting to your DB instance using IAM authentication and the AWS SDK for .NET \(p. 1672\)](#)
- [Connecting to your DB instance using IAM authentication and the AWS SDK for Go \(p. 1673\)](#)
- [Connecting to your DB instance using IAM authentication and the AWS SDK for Java \(p. 1679\)](#)
- [Connecting to your DB instance using IAM authentication and the AWS SDK for Python \(Boto3\) \(p. 1687\)](#)

Connecting to your DB instance using IAM authentication from the command line: AWS CLI and mysql client

You can connect from the command line to an Amazon RDS DB instance with the AWS CLI and `mysql` command line tool as described following.

The following are prerequisites for connecting to your DB instance using IAM authentication:

- [Enabling and disabling IAM database authentication \(p. 1662\)](#)

- [Creating and using an IAM policy for IAM database access \(p. 1664\)](#)
- [Creating a database account using IAM authentication \(p. 1667\)](#)

Note

For information about connecting to your database using SQL Workbench/J with IAM authentication, see the blog post [Use IAM authentication to connect with SQL Workbench/J to Aurora MySQL or Amazon RDS for MySQL](#).

Topics

- [Generating an IAM authentication token \(p. 1669\)](#)
- [Connecting to a DB instance \(p. 1669\)](#)

Generating an IAM authentication token

The following example shows how to get a signed authentication token using the AWS CLI.

```
aws rds generate-db-auth-token \
--hostname rdsmysql.123456789012.us-west-2.rds.amazonaws.com \
--port 3306 \
--region us-west-2 \
--username jane_doe
```

In the example, the parameters are as follows:

- **--hostname** – The host name of the DB instance that you want to access
- **--port** – The port number used for connecting to your DB instance
- **--region** – The AWS Region where the DB instance is running
- **--username** – The database account that you want to access

The first several characters of the token look like the following.

```
rdsmysql.123456789012.us-west-2.rds.amazonaws.com:3306/?Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

Connecting to a DB instance

The general format for connecting is shown following.

```
mysql --host=hostName --port=portNumber --ssl-ca=[full path]rds-combined-ca-bundle.pem --enable-cleartext-plugin --user=userName --password=authToken
```

The parameters are as follows:

- **--host** – The host name of the DB instance that you want to access
- **--port** – The port number used for connecting to your DB instance
- **--ssl-ca** – The SSL certificate file that contains the public key

For more information, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).

- **--enable-cleartext-plugin** – A value that specifies that `AWSAuthenticationPlugin` must be used for this connection

If you are using a MariaDB client, the **--enable-cleartext-plugin** option isn't required.

- **--user** – The database account that you want to access
- **--password** – A signed IAM authentication token

The authentication token consists of several hundred characters. It can be unwieldy on the command line. One way to work around this is to save the token to an environment variable, and then use that variable when you connect. The following example shows one way to perform this workaround.

```
RDSHOST="rdsmysql.123456789012.us-west-2.rds.amazonaws.com"  
TOKEN=$(aws rds generate-db-auth-token --hostname $RDSHOST --port 3306 --region us-west-2  
--username jane_doe)"  
  
mysql --host=$RDSHOST --port=3306 --ssl-ca=/sample_dir/rds-combined-ca-bundle.pem --enable-  
cleartext-plugin --user=jane_doe --password=$TOKEN
```

When you connect using `AWSAuthenticationPlugin`, the connection is secured using SSL. To verify this, type the following at the `mysql>` command prompt.

```
show status like 'Ssl%';
```

The following lines in the output show more details.

Variable_name	Value
...	...
Ssl_cipher	AES256-SHA
...	...
Ssl_version	TLSv1.1
...	...

Connecting to your DB instance using IAM authentication from the command line: AWS CLI and psql client

You can connect from the command line to an Amazon RDS for PostgreSQL DB instance with the AWS CLI and `psql` command line tool as described following.

The following are prerequisites for connecting to your DB instance using IAM authentication:

- [Enabling and disabling IAM database authentication \(p. 1662\)](#)
- [Creating and using an IAM policy for IAM database access \(p. 1664\)](#)
- [Creating a database account using IAM authentication \(p. 1667\)](#)

Note

For information about connecting to your database using pgAdmin with IAM authentication, see the blog post [Using IAM authentication to connect with pgAdmin Amazon Aurora PostgreSQL or Amazon RDS for PostgreSQL](#).

Topics

- [Generating an IAM authentication token \(p. 1671\)](#)
- [Connecting to an Amazon RDS PostgreSQL instance \(p. 1671\)](#)

Generating an IAM authentication token

The authentication token consists of several hundred characters so it can be unwieldy on the command line. One way to work around this is to save the token to an environment variable, and then use that variable when you connect. The following example shows how to use the AWS CLI to get a signed authentication token using the generated-db-auth-token command, and store it in a PGPASSWORD environment variable.

```
export RDHOST="rdspostgres.123456789012.us-west-2.rds.amazonaws.com"
export PGPASSWORD=$(aws rds generate-db-auth-token --hostname $RDHOST --port 5432 --
region us-west-2 --username jane_doe )"
```

In the example, the parameters to the generate-db-auth-token command are as follows:

- --hostname – The host name of the DB instance that you want to access
- --port – The port number used for connecting to your DB instance
- --region – The AWS Region where the DB instance is running
- --username – The database account that you want to access

The first several characters of the generated token look like the following.

```
rdspostgres.123456789012.us-west-2.rds.amazonaws.com:5432/?
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

Connecting to an Amazon RDS PostgreSQL instance

The general format for using psql to connect is shown following.

```
psql "host=hostName port=portNumber sslmode=verify-full sslrootcert=certificateFile
      dbname=DBName user=userName password=authToken"
```

The parameters are as follows:

- **host** – The host name of the DB instance that you want to access
- **port** – The port number used for connecting to your DB instance
- **sslmode** – The SSL mode to use

When you use **sslmode=verify-full**, the SSL connection verifies the DB instance endpoint against the endpoint in the SSL certificate.

- **sslrootcert** – The SSL certificate file that contains the public key

For more information, see [Using SSL with a PostgreSQL DB instance](#).

- **dbname** – The database that you want to access
- **user** – The database account that you want to access
- **password** – A signed IAM authentication token

The following example shows using psql to connect. In the example psql uses the environment variable PGPASSWORD that was set when the token was generated in the previous section.

```
psql "host=$RDSHOST port=5432 sslmode=verify-full sslrootcert=/sample_dir/rds-combined-ca-bundle.pem dbname=DBName user=jane_doe password=$PGPASSWORD"
```

Connecting to your DB instance using IAM authentication and the AWS SDK for .NET

You can connect to an RDS for MySQL or PostgreSQL for DB instance with the AWS SDK for .NET as described following.

The following are prerequisites for connecting to your DB instance using IAM authentication:

- [Enabling and disabling IAM database authentication \(p. 1662\)](#)
- [Creating and using an IAM policy for IAM database access \(p. 1664\)](#)
- [Creating a database account using IAM authentication \(p. 1667\)](#)

The following code example shows how to generate an authentication token, and then use it to connect to a DB instance.

To run this code example, you need the [AWS SDK for .NET](#), found on the AWS site. The AWSSDK.CORE and the AWSSDK.RDS packages are required. To connect to a DB instance, use the .NET database connector for the DB engine, such as MySqlConnector for MySQL or Npgsql for PostgreSQL.

Modify the values of the following variables as needed:

- **server** – The endpoint of the DB instance that you want to access
- **port** – The port number used for connecting to your DB instance
- **user** – The database account that you want to access.

This code connects to a MySQL DB instance.

```
using System;
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;
using Amazon;

namespace ubuntu
{
    class Program
    {
        static void Main(string[] args)
        {
            var pwd =
Amazon.RDS.Util.RDSSAuthTokenGenerator.GenerateAuthToken(RegionEndpoint.USEast1,
"mysqladb.123456789012.us-east-1.rds.amazonaws.com", 3306, "jane_doe");
            // for debug only Console.WriteLine("{0}\n", pwd); //this verifies the token is
generated

            MySqlConnection conn = new MySqlConnection("server=mysqladb.123456789012.us-
east-1.rds.amazonaws.com;user=jane_doe;database=mydB;port=3306;password={pwd};SslMode=Required;SslCa=..rds-ca-2019-root.pem");
            conn.Open();

            // Define a query
            MySqlCommand sampleCommand = new MySqlCommand("SHOW DATABASES;", conn);

            // Execute a query
            MySqlDataReader mysqlDataRdr = sampleCommand.ExecuteReader();
```

```
// Read all rows and output the first column in each row
while (mysqlDataRdr.Read())
    Console.WriteLine(mysqlDataRdr[0]);

mysqlDataRdr.Close();
// Close connection
conn.Close();
}
}
```

This code connects to a PostgreSQL DB instance.

```
using System;
using Npgsql;
using Amazon.RDS.Util;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            var pwd =
RDSAAuthGenerator.GenerateAuthToken("postgresql.123456789012.us-
east-1.rds.amazonaws.com", 5432, "jane_doe");
// for debug only Console.WriteLine("{0}\n", pwd); //this verifies the token is generated

            NpgsqlConnection conn = new
NpgsqlConnection($"Server=postgresql.123456789012.us-east-1.rds.amazonaws.com;User
Id=jane_doe;Password={pwd};Database=mydb;SSL Mode=Require;Trust Server
Certificate=true;");
            conn.Open();

            // Define a query
            NpgsqlCommand cmd = new NpgsqlCommand("select count(*) FROM pg_user",
conn);

            // Execute a query
            NpgsqlDataReader dr = cmd.ExecuteReader();

            // Read all rows and output the first column in each row
            while (dr.Read())
                Console.WriteLine("{0}\n", dr[0]);

            // Close connection
            conn.Close();
        }
    }
}
```

Connecting to your DB instance using IAM authentication and the AWS SDK for Go

You can connect to an RDS for MySQL or RDS for PostgreSQL DB instance with the AWS SDK for Go as described following.

The following are prerequisites for connecting to your DB instance using IAM authentication:

- Enabling and disabling IAM database authentication (p. 1662)
 - Creating and using an IAM policy for IAM database access (p. 1664)

- [Creating a database account using IAM authentication \(p. 1667\)](#)

To run these code examples, you need the [AWS SDK for Go](#), found on the AWS site.

Modify the values of the following variables as needed:

- `dbName` – The database that you want to access
- `dbUser` – The database account that you want to access
- `dbHost` – The endpoint of the DB instance that you want to access
- `dbPort` – The port number used for connecting to your DB instance
- `region` – The AWS Region where the DB instance is running

In addition, make sure the imported libraries in the sample code exist on your system.

Important

The examples in this section use the following code to provide credentials that access a database from a local environment:

```
creds := credentials.NewEnvCredentials()
```

If you are accessing a database from an AWS service, such as Amazon EC2 or Amazon ECS, you can replace the code with the following code:

```
sess := session.Must(session.NewSession())
```

```
creds := sess.Config.Credentials
```

If you make this change, make sure you add the following import:

```
"github.com/aws/aws-sdk-go/aws/session"
```

Topics

- [Connecting using IAM authentication and the AWS SDK for Go V2 \(p. 1674\)](#)
- [Connecting using IAM authentication and the AWS SDK for Go V1. \(p. 1677\)](#)

Connecting using IAM authentication and the AWS SDK for Go V2

You can connect to a DB instance using IAM authentication and the AWS SDK for Go V2.

Topics

- [Generating an IAM authentication token \(p. 1674\)](#)
- [Connecting to a DB instance \(p. 1675\)](#)

Generating an IAM authentication token

The auth package provides utilities for generating authentication tokens for connecting to Amazon RDS MySQL and PostgreSQL database instances. Using the `BuildAuthToken` method, you generate a database authorization token by providing the database endpoint, AWS Region, username, and a `aws.CredentialProvider` implementation that returns IAM credentials with permission connect to the database using IAM database authentication.

The following example shows how to use `BuildAuthToken` to create an authentication token for connecting to a MySQL DB instance.

```
package main

import "context"
import "github.com/aws/aws-sdk-go-v2/config"
import "github.com/aws/aws-sdk-go-v2/feature/rds/auth"

func main() {
```

```

cfg, err := config.LoadDefaultConfig(context.TODO())
if err != nil {
    panic("configuration error: " + err.Error())
}

authenticationToken, err := auth.BuildAuthToken(
    context.TODO(),
    "mydb.123456789012.us-east-1.rds.amazonaws.com:3306", // Database Endpoint (With Port)
    "us-east-1", // AWS Region
    "jane_doe", // Database Account
    cfg.Credentials,
)
if err != nil {
    panic("failed to create authentication token: " + err.Error())
}
}

```

The following example shows how to use `BuildAuthToken` to create an authentication token for connecting to a PostgreSQL DB instance.

```

package main

import "context"
import "github.com/aws/aws-sdk-go-v2/config"
import "github.com/aws/aws-sdk-go-v2/feature/rds/auth"

func main() {

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
        context.TODO(),
        "mydb.123456789012.us-east-1.rds.amazonaws.com:5432", // Database Endpoint (With Port)
        "us-east-1", // AWS Region
        "jane_doe", // Database Account
        cfg.Credentials,
    )
    if err != nil {
        panic("failed to create authentication token: " + err.Error())
    }
}

```

Connecting to a DB instance

The following code example shows how to generate an authentication token, and then use it to connect to a DB instance.

This code connects to a MySQL DB instance.

```

package main

import "context"
import "github.com/aws/aws-sdk-go-v2/config"
import "github.com/aws/aws-sdk-go-v2/feature/rds/auth"

func main() {

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {

```

```

        panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
        context.TODO(),
        "mydb.123456789012.us-east-1.rds.amazonaws.com:3306", // Database Endpoint (With Port)
        "us-east-1", // AWS Region
        "jane_doe", // Database Account
        cfg.Credentials,
    )
    if err != nil {
        panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authToken, dbEndpoint, dbName,
    )

    db, err := sql.Open("mysql", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}

```

This code connects to a PostgreSQL DB instance.

```

package main

import "context"
import "github.com/aws/aws-sdk-go-v2/config"
import "github.com/aws/aws-sdk-go-v2/feature/rds/auth"

func main() {

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
        context.TODO(),
        "mydb.123456789012.us-east-1.rds.amazonaws.com:5432", // Database Endpoint (With Port)
        "us-east-1", // AWS Region
        "jane_doe", // Database Account
        cfg.Credentials,
    )
    if err != nil {
        panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s",
        dbHost, dbPort, dbUser, authToken, dbName,
    )

    db, err := sql.Open("postgres", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
}

```

```
    if err != nil {
        panic(err)
    }
}
```

Connecting using IAM authentication and the AWS SDK for Go V1.

You can connect to a DB instance using IAM authentication and the AWS SDK for Go V1

Topics

- [Generating an IAM authentication token \(p. 1677\)](#)
- [Connecting to a DB instance \(p. 1678\)](#)

Generating an IAM authentication token

You can use the `rdsutils` package to generate tokens used to connect to a DB instance. Call the `BuildAuthToken` function to generate a token. Provide the DB instance endpoint, AWS region, username, and IAM credentials to generate the token for connecting to a DB instance with IAM credentials.

The following example shows how to use `BuildAuthToken` to create an authentication token for connecting to a MySQL DB instance.

```
package main

import (
    "database/sql"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go/aws/credentials"
    "github.com/aws/aws-sdk-go/service/rds/rdsutils"
)

func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "mydb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 3306
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        log.Fatalf("failed to build auth token %v", err)
    }
}
```

The following example shows how to use `BuildAuthToken` to create an authentication token for connecting to a PostgreSQL DB instance.

```
package main

import (
    "database/sql"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go/aws/credentials"
    "github.com/aws/aws-sdk-go/service/rds/rdsutils"
```

```

)
func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "mydb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 5432
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        log.Fatalf("failed to build auth token %v", err)
    }
}

```

Connecting to a DB instance

The following code example shows how to generate an authentication token, and then use it to connect to a DB instance.

This code connects to a MySQL DB instance.

```

package main

import (
    "database/sql"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go/aws/credentials"
    "github.com/aws/aws-sdk-go/service/rds/rdsutils"
    _ "github.com/go-sql-driver/mysql"
)

func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "mydb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 3306
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        panic(err)
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authToken, dbEndpoint, dbName,
    )

    db, err := sql.Open("mysql", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}

```

This code connects to a PostgreSQL DB instance.

```
package main

import (
    "database/sql"
    "fmt"

    "github.com/aws/aws-sdk-go/aws/credentials"
    "github.com/aws/aws-sdk-go/service/rds/rdsutils"
    _ "github.com/lib/pq"
)

func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "mydb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 5432
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        panic(err)
    }

    dsn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s",
        dbHost, dbPort, dbUser, authToken, dbName,
    )

    db, err := sql.Open("postgres", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Connecting to your DB instance using IAM authentication and the AWS SDK for Java

You can connect to an RDS for MySQL or RDS for PostgreSQL DB instance with the AWS SDK for Java as described following.

The following are prerequisites for connecting to your DB instance using IAM authentication:

- [Enabling and disabling IAM database authentication \(p. 1662\)](#)
- [Creating and using an IAM policy for IAM database access \(p. 1664\)](#)
- [Creating a database account using IAM authentication \(p. 1667\)](#)
- [Set up the AWS SDK for Java](#)

Topics

- [Generating an IAM authentication token \(p. 1680\)](#)
- [Manually constructing an IAM authentication token \(p. 1680\)](#)
- [Connecting to a DB instance \(p. 1683\)](#)

Generating an IAM authentication token

If you are writing programs using the AWS SDK for Java, you can get a signed authentication token using the `RdsIamAuthTokenGenerator` class. Using this class requires that you provide AWS credentials. To do this, you create an instance of the `DefaultAWSCredentialsProviderChain` class. `DefaultAWSCredentialsProviderChain` uses the first AWS access key and secret key that it finds in the [default credential provider chain](#). For more information about AWS access keys, see [Managing access keys for IAM users](#).

After you create an instance of `RdsIamAuthTokenGenerator`, you can call the `getAuthToken` method to obtain a signed token. Provide the AWS Region, host name, port number, and user name. The following code example illustrates how to do this.

```
package com.amazonaws.codesamples;

import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.services.rds.auth.GetIamAuthTokenRequest;
import com.amazonaws.services.rds.auth.RdsIamAuthTokenGenerator;

public class GenerateRDSAuthToken {

    public static void main(String[] args) {

        String region = "us-west-2";
        String hostname = "rdsmysql.123456789012.us-west-2.rds.amazonaws.com";
        String port = "3306";
        String username = "jane_doe";

        System.out.println(generateAuthToken(region, hostname, port, username));
    }

    static String generateAuthToken(String region, String hostName, String port, String
username) {

        RdsIamAuthTokenGenerator generator = RdsIamAuthTokenGenerator.builder()
            .credentials(new DefaultAWSCredentialsProviderChain())
            .region(region)
            .build();

        String authToken = generator.getAuthToken(
            GetIamAuthTokenRequest.builder()
                .hostname(hostName)
                .port(Integer.parseInt(port))
                .userName(username)
                .build());
    }

    return authToken;
}
}
```

Manually constructing an IAM authentication token

In Java, the easiest way to generate an authentication token is to use `RdsIamAuthTokenGenerator`. This class creates an authentication token for you, and then signs it using AWS signature version 4. For more information, see [Signature version 4 signing process](#) in the *AWS General Reference*.

However, you can also construct and sign an authentication token manually, as shown in the following code example.

```
package com.amazonaws.codesamples;

import com.amazonaws.SdkClientException;
```

```

import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.auth.SigningAlgorithm;
import com.amazonaws.util.BinaryUtils;
import org.apache.commons.lang3.StringUtils;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.Charset;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.SortedMap;
import java.util.TreeMap;

import static com.amazonaws.auth.internal.SignerConstants.AWS4_TERMINATOR;
import static com.amazonaws.util.StringUtils.UTF8;

public class CreateRDSAuthTokenManually {
    public static String httpMethod = "GET";
    public static String action = "connect";
    public static String canonicalURIParameter = "/";
    public static SortedMap<String, String> canonicalQueryParameters = new TreeMap();
    public static String payload = StringUtils.EMPTY;
    public static String signedHeader = "host";
    public static String algorithm = "AWS4-HMAC-SHA256";
    public static String serviceName = "rds-db";
    public static String requestWithoutSignature;

    public static void main(String[] args) throws Exception {

        String region = "us-west-2";
        String instanceName = "rdsmysql.123456789012.us-west-2.rds.amazonaws.com";
        String port = "3306";
        String username = "jane_doe";

        Date now = new Date();
        String date = new SimpleDateFormat("yyyyMMdd").format(now);
        String dateTimeStamp = new SimpleDateFormat("yyyyMMdd'T'HHmmss'Z'").format(now);
        DefaultAWSCredentialsProviderChain creds = new
DefaultAWSCredentialsProviderChain();
        String awsAccessKey = creds.getCredentials().getAWSAccessKeyId();
        String awsSecretKey = creds.getCredentials().getAWSSecretKey();
        String expiryMinutes = "900";

        System.out.println("Step 1: Create a canonical request:");
        String canonicalString = createCanonicalString(username, awsAccessKey, date,
dateTimeStamp, region, expiryMinutes, instanceName, port);
        System.out.println(canonicalString);
        System.out.println();

        System.out.println("Step 2: Create a string to sign:");
        String stringToSign = createStringToSign(dateTimeStamp, canonicalString,
awsAccessKey, date, region);
        System.out.println(stringToSign);
        System.out.println();

        System.out.println("Step 3: Calculate the signature:");
        String signature = BinaryUtils.toHexString(calculateSignature(stringToSign,
new SigningKey(awsSecretKey, date, region, serviceName)));
        System.out.println(signature);
        System.out.println();

        System.out.println("Step 4: Add the signing info to the request");
        System.out.println(appendSignature(signature));
        System.out.println();
    }
}

```

```

}

//Step 1: Create a canonical request date should be in format YYYYMMDD and date
//Time should be in format YYYYMMDDTHHMMSSZ
public static String createCanonicalString(String user, String accessKey, String date,
String date, String region, String expiryPeriod, String hostName, String port) throws
Exception {
    canonicalQueryParameters.put("Action", action);
    canonicalQueryParameters.put("DBUser", user);
    canonicalQueryParameters.put("X-Amz-Algorithm", "AWS4-HMAC-SHA256");
    canonicalQueryParameters.put("X-Amz-Credential", accessKey + "%2F" + date + "%2F" +
region + "%2F" + serviceName + "%2Faws4_request");
    canonicalQueryParameters.put("X-Amz-Date", date);
    canonicalQueryParameters.put("X-Amz-Expires", expiryPeriod);
    canonicalQueryParameters.put("X-Amz-SignedHeaders", signedHeader);
    String canonicalQueryString = "";
    while(!canonicalQueryParameters.isEmpty()) {
        String currentQueryParameter = canonicalQueryParameters.firstKey();
        String currentQueryParameterValue =
canonicalQueryParameters.remove(currentQueryParameter);
        canonicalQueryString = canonicalQueryString + currentQueryParameter + "=" +
currentQueryParameterValue;
        if (!currentQueryParameter.equals("X-Amz-SignedHeaders")) {
            canonicalQueryString += "&";
        }
    }
    String canonicalHeaders = "host:" + hostName + ":" + port + '\n';
    requestWithoutSignature = hostName + ":" + port + "/" + canonicalQueryString;

    String hashedPayload = BinaryUtils.toHexString(hash(payload));
    return httpMethod + '\n' + canonicalURIParameter + '\n' + canonicalQueryString +
'\n' + canonicalHeaders + '\n' + signedHeader + '\n' + hashedPayload;
}

//Step 2: Create a string to sign using sig v4
public static String createStringToSign(String date, String canonicalRequest,
String accessKey, String date, String region) throws Exception {
    String credentialScope = date + "/" + region + "/" + serviceName + "/aws4_request";
    return algorithm + '\n' + date + '\n' + credentialScope + '\n' +
BinaryUtils.toHexString(hash(canonicalRequest));

}

//Step 3: Calculate signature
/**
 * Step 3 of the AWS Signature version 4 calculation. It involves deriving
 * the signing key and computing the signature. Refer to
 * http://docs.aws.amazon
 * .com/general/latest/gr/sigv4-calculate-signature.html
 */
public static byte[] calculateSignature(String stringToSign,
                                         byte[] signingKey) {
    return sign(stringToSign.getBytes(Charset.forName("UTF-8")), signingKey,
               SigningAlgorithm.HmacSHA256);
}

public static byte[] sign(byte[] data, byte[] key,
                        SigningAlgorithm algorithm) throws SdkClientException {
    try {
        Mac mac = algorithm.getMac();
        mac.init(new SecretKeySpec(key, algorithm.toString()));
        return mac.doFinal(data);
    } catch (Exception e) {
        throw new SdkClientException(
            "Unable to calculate a request signature: "
        );
    }
}

```

```

                + e.getMessage(), e);
            }

        public static byte[] newSigningKey(String secretKey,
                                         String dateStamp, String regionName, String serviceName)
    {
        byte[] kSecret = ("AWS4" + secretKey).getBytes(Charset.forName("UTF-8"));
        byte[] kDate = sign(dateStamp, kSecret, SigningAlgorithm.HmacSHA256);
        byte[] kRegion = sign(regionName, kDate, SigningAlgorithm.HmacSHA256);
        byte[] kService = sign(serviceName, kRegion,
                               SigningAlgorithm.HmacSHA256);
        return sign(AWS4_TERMINATOR, kService, SigningAlgorithm.HmacSHA256);
    }

    public static byte[] sign(String stringData, byte[] key,
                             SigningAlgorithm algorithm) throws SdkClientException {
        try {
            byte[] data = stringData.getBytes(UTF8);
            return sign(data, key, algorithm);
        } catch (Exception e) {
            throw new SdkClientException(
                "Unable to calculate a request signature: "
                + e.getMessage(), e);
        }
    }

    //Step 4: append the signature
    public static String appendSignature(String signature) {
        return requestWithoutSignature + "&X-Amz-Signature=" + signature;
    }

    public static byte[] hash(String s) throws Exception {
        try {
            MessageDigest md = MessageDigest.getInstance("SHA-256");
            md.update(s.getBytes(UTF8));
            return md.digest();
        } catch (Exception e) {
            throw new SdkClientException(
                "Unable to compute hash while signing request: "
                + e.getMessage(), e);
        }
    }
}

```

Connecting to a DB instance

The following code example shows how to generate an authentication token, and then use it to connect to an instance running MySQL.

To run this code example, you need the [AWS SDK for Java](#), found on the AWS site. In addition, you need the following:

- MySQL Connector/J. This code example was tested with `mysql-connector-java-5.1.33-bin.jar`.
- An intermediate certificate for Amazon RDS that is specific to an AWS Region. (For more information, see [Using SSL/TLS to encrypt a connection to a DB instance \(p. 1634\)](#).) At runtime, the class loader looks for the certificate in the same directory as this Java code example, so that the class loader can find it.
- Modify the values of the following variables as needed:
 - `RDS_INSTANCE_HOSTNAME` – The host name of the DB instance that you want to access.
 - `RDS_INSTANCE_PORT` – The port number used for connecting to your PostgreSQL DB instance.

- REGION_NAME – The AWS Region where the DB instance is running.
- DB_USER – The database account that you want to access.
- SSL_CERTIFICATE – An SSL certificate for Amazon RDS that is specific to an AWS Region.

To download a certificate for your AWS Region, see [Intermediate certificates \(p. 1635\)](#). Place the SSL certificate in the same directory as this Java program file, so that the class loader can find the certificate at runtime.

This code example obtains AWS credentials from the [default credential provider chain](#).

```
package com.amazonaws.samples;

import com.amazonaws.services.rds.auth.RdsIamAuthTokenGenerator;
import com.amazonaws.services.rds.auth.GetIamAuthTokenRequest;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.auth.AWSStaticCredentialsProvider;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Properties;

import java.net.URL;

public class IAMDatabaseAuthenticationTester {
    //AWS Credentials of the IAM user with policy enabling IAM Database Authenticated
    access to the db by the db user.
    private static final DefaultAWSCredentialsProviderChain creds = new
DefaultAWSCredentialsProviderChain();
    private static final String AWS_ACCESS_KEY =
creds.getCredentials().getAWSAccessKeyId();
    private static final String AWS_SECRET_KEY = creds.getCredentials().getAWSSecretKey();

    //Configuration parameters for the generation of the IAM Database Authentication token
    private static final String RDS_INSTANCE_HOSTNAME = "rdsmysql.123456789012.us-
west-2.rds.amazonaws.com";
    private static final int RDS_INSTANCE_PORT = 3306;
    private static final String REGION_NAME = "us-west-2";
    private static final String DB_USER = "jane_doe";
    private static final String JDBC_URL = "jdbc:mysql://" + RDS_INSTANCE_HOSTNAME + ":" +
RDS_INSTANCE_PORT;

    private static final String SSL_CERTIFICATE = "rds-ca-2019-us-west-2.pem";

    private static final String KEY_STORE_TYPE = "JKS";
    private static final String KEY_STORE_PROVIDER = "SUN";
    private static final String KEY_STORE_FILE_PREFIX = "sys-connect-via-ssl-test-cacerts";
    private static final String KEY_STORE_FILE_SUFFIX = ".jks";
    private static final String DEFAULT_KEY_STORE_PASSWORD = "changeit";

    public static void main(String[] args) throws Exception {
        //get the connection
        Connection connection = getDBConnectionUsingIam();
```

```

        //verify the connection is successful
        Statement stmt= connection.createStatement();
        ResultSet rs=stmt.executeQuery("SELECT 'Success!' FROM DUAL;");
        while (rs.next()) {
            String id = rs.getString(1);
            System.out.println(id); //Should print "Success!"
        }

        //close the connection
        stmt.close();
        connection.close();

        clearSslProperties();

    }

    /**
     * This method returns a connection to the db instance authenticated using IAM Database
     Authentication
     * @return
     * @throws Exception
     */
    private static Connection getDBConnectionUsingIam() throws Exception {
        setSslProperties();
        return DriverManager.getConnection(JDBC_URL, set MySqlConnectionProperties());
    }

    /**
     * This method sets the mysql connection properties which includes the IAM Database
     Authentication token
     * as the password. It also specifies that SSL verification is required.
     * @return
     */
    private static Properties set MySqlConnectionProperties() {
        Properties mysqlConnectionProperties = new Properties();
        mysqlConnectionProperties.setProperty("verifyServerCertificate","true");
        mysqlConnectionProperties.setProperty("useSSL", "true");
        mysqlConnectionProperties.setProperty("user",DB_USER);
        mysqlConnectionProperties.setProperty("password",generateAuthToken());
        return mysqlConnectionProperties;
    }

    /**
     * This method generates the IAM Auth Token.
     * An example IAM Auth Token would look like follows:
     * btusii123.cmz7kenwo2ye.rds.cn-north-1.amazonaws.com.cn:3306/?Action=connect&DBUser=iamtestuser&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20171003T010726Z&X-Amz-SignedHeaders=host&X-Amz-Expires=899&X-Amz-Credential=AKIAFPXHGVDI5RNFO4AQ%2F20171003%2Fcna-north-1%2Frds-db%2Faws4_request&X-Amz-Signature=f9f45ef96c1f770cdad11a53e33ffa4c3730bc03fdee820cfdf1322eed15483b
     * @return
     */
    private static String generateAuthToken() {
        BasicAWSCredentials awsCredentials = new BasicAWSCredentials(AWS_ACCESS_KEY,
        AWS_SECRET_KEY);

        RdsIamAuthTokenGenerator generator = RdsIamAuthTokenGenerator.builder()
            .credentials(new
        AWSStaticCredentialsProvider(awsCredentials)).region(REGION_NAME).build();
        return generator.getAuthToken(GetIamAuthTokenRequest.builder()

        .hostname(RDS_INSTANCE_HOSTNAME).port(RDS_INSTANCE_PORT).userName(DB_USER).build());
    }

    /**

```

```

        * This method sets the SSL properties which specify the key store file, its type and
password:
        * @throws Exception
        */
private static void setSslProperties() throws Exception {
    System.setProperty("javax.net.ssl.trustStore", createKeyStoreFile());
    System.setProperty("javax.net.ssl.trustStoreType", KEY_STORE_TYPE);
    System.setProperty("javax.net.ssl.trustStorePassword", DEFAULT_KEY_STORE_PASSWORD);
}

/**
 * This method returns the path of the Key Store File needed for the SSL verification
during the IAM Database Authentication to
 * the db instance.
 * @return
 * @throws Exception
 */
private static String createKeyStoreFile() throws Exception {
    return createKeyStoreFile(createCertificate()).getPath();
}

/**
 * This method generates the SSL certificate
 * @return
 * @throws Exception
 */
private static X509Certificate createCertificate() throws Exception {
    CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
    URL url = new File(SSL_CERTIFICATE).toURI().toURL();
    if (url == null) {
        throw new Exception();
    }
    try (InputStream certInputStream = url.openStream()) {
        return (X509Certificate) certFactory.generateCertificate(certInputStream);
    }
}

/**
 * This method creates the Key Store File
 * @param rootX509Certificate - the SSL certificate to be stored in the KeyStore
 * @return
 * @throws Exception
 */
private static File createKeyStoreFile(X509Certificate rootX509Certificate) throws
Exception {
    File keyStoreFile = File.createTempFile(KEY_STORE_FILE_PREFIX,
KEY_STORE_FILE_SUFFIX);
    try (FileOutputStream fos = new FileOutputStream(keyStoreFile.getPath())) {
        KeyStore ks = KeyStore.getInstance(KEY_STORE_TYPE, KEY_STORE_PROVIDER);
        ks.load(null);
        ks.setCertificateEntry("rootCaCertificate", rootX509Certificate);
        ks.store(fos, DEFAULT_KEY_STORE_PASSWORD.toCharArray());
    }
    return keyStoreFile;
}

/**
 * This method clears the SSL properties.
 * @throws Exception
 */
private static void clearSslProperties() throws Exception {
    System.clearProperty("javax.net.ssl.trustStore");
    System.clearProperty("javax.net.ssl.trustStoreType");
    System.clearProperty("javax.net.ssl.trustStorePassword");
}

```

```
}
```

Connecting to your DB instance using IAM authentication and the AWS SDK for Python (Boto3)

You can connect to an RDS for MySQL or RDS for PostgreSQL DB instance with the AWS SDK for Python (Boto3) as described following.

The following are prerequisites for connecting to your DB instance using IAM authentication:

- [Enabling and disabling IAM database authentication \(p. 1662\)](#)
- [Creating and using an IAM policy for IAM database access \(p. 1664\)](#)
- [Creating a database account using IAM authentication \(p. 1667\)](#)

In addition, make sure the imported libraries in the sample code exist on your system.

The code examples use profiles for shared credentials. For information about the specifying credentials, see [Credentials](#) in the AWS SDK for Python (Boto3) documentation.

Topics

- [Generating an IAM authentication token \(p. 1687\)](#)
- [Connecting to a DB instance \(p. 1688\)](#)

Generating an IAM authentication token

You can call the `generate_db_auth_token` method to obtain a signed token. Provide the DB instance endpoint, port, user name, AWS Region, and DB engine to generate the token for connecting to a DB instance with IAM credentials.

This code generates an IAM authentication token for a MySQL DB instance.

```
import sys
import boto3
import os

ENDPOINT="mysqladb.123456789012.us-east-1.rds.amazonaws.com"
PORT="3306"
USR="jane_doe"
REGION="us-east-1"
os.environ['LIBMYSQL_ENABLE_CLEARTEXT_PLUGIN'] = '1'

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='RDSCreds')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USR,
Region=REGION)
```

This code generates an IAM authentication token for a PostgreSQL DB instance.

```
import sys
import boto3
import os

ENDPOINT="postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
```

```
PORT="5432"
USR="jane_doe"
REGION="us-east-1"

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='RDSCreds')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USR,
Region=REGION)
```

Connecting to a DB instance

The following code example shows how to generate an authentication token, and then use it to connect to a DB instance.

To run this code example, you need the [AWS SDK for Python \(Boto3\)](#), found on the AWS site.

Modify the values of the following variables as needed:

- **ENDPOINT** – The endpoint of the DB instance that you want to access
- **PORT** – The port number used for connecting to your DB instance
- **USER** – The database account that you want to access.
- **REGION** – The AWS Region where the DB instance is running
- **DBNAME** – The database that you want to access

This code connects to a MySQL DB instance.

```
import mysql.connector
import sys
import boto3
import os

ENDPOINT="mysqladb.123456789012.us-east-1.rds.amazonaws.com"
PORT="3306"
USR="jane_doe"
REGION="us-east-1"
DBNAME="mydb"
os.environ['LIBMYSQL_ENABLE_CLEARTEXT_PLUGIN'] = '1'

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='default')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USR,
Region=REGION)

try:
    conn = mysql.connector.connect(host=ENDPOINT, user=USR, passwd=token, port=PORT,
database=DBNAME)
    cur = conn.cursor()
    cur.execute("""SELECT now()""")
    query_results = cur.fetchall()
    print(query_results)
except Exception as e:
    print("Database connection failed due to {}".format(e))
```

This code connects to a PostgreSQL DB instance.

```
import psycopg2
import sys
import boto3
import os

ENDPOINT="postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
PORT="5432"
USR="jane_doe"
REGION="us-east-1"
DBNAME="mydb"

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='RDSCreds')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USR,
Region=REGION)

try:
    conn = psycopg2.connect(host=ENDPOINT, port=PORT, database=DBNAME, user=USR,
password=token)
    cur = conn.cursor()
    cur.execute("""SELECT now()""")
    query_results = cur.fetchall()
    print(query_results)
except Exception as e:
    print("Database connection failed due to {}".format(e))
```

Troubleshooting Amazon RDS identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Amazon RDS and IAM.

Topics

- [I'm not authorized to perform an action in Amazon RDS \(p. 1689\)](#)
- [I'm not authorized to perform iam:PassRole \(p. 1690\)](#)
- [I want to view my access keys \(p. 1690\)](#)
- [I'm an administrator and want to allow others to access Amazon RDS \(p. 1690\)](#)
- [I want to allow people outside of my AWS account to access my Amazon RDS resources \(p. 1690\)](#)

I'm not authorized to perform an action in Amazon RDS

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a `widget` but does not have `rds:GetWidget` permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
rds:GetWidget on resource: my-example-widget
```

In this case, Mateo asks his administrator to update his policies to allow him to access the `my-example-widget` resource using the `rds:GetWidget` action.

I'm not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to Amazon RDS.

Some AWS services allow you to pass an existing role to that service, instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Amazon RDS. However, the action requires the service to have permissions granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary asks her administrator to update her policies to allow her to perform the `iam:PassRole` action.

I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

Important

Do not provide your access keys to a third party, even to help [find your canonical user ID](#). By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see [Managing access keys](#) in the *IAM User Guide*.

I'm an administrator and want to allow others to access Amazon RDS

To enable others to access Amazon RDS, you must create an IAM entity (user or role) for the person or application that needs access. They use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in Amazon RDS.

To get started right away, see [Creating your first IAM delegated user and group](#) in the *IAM User Guide*.

I want to allow people outside of my AWS account to access my Amazon RDS resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Amazon RDS supports these features, see [How Amazon RDS works with IAM \(p. 1648\)](#).
- To learn how to provide access to your resources across AWS accounts that you own, see [Providing access to an IAM user in another AWS account that you own](#) in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see [Providing access to AWS accounts owned by third parties](#) in the *IAM User Guide*.
- To learn how to provide access through identity federation, see [Providing access to externally authenticated users \(identity federation\)](#) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.

Logging and monitoring in Amazon RDS

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon RDS and your AWS solutions. You should collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your Amazon RDS resources and responding to potential incidents:

Amazon CloudWatch Alarms

Using Amazon CloudWatch alarms, you watch a single metric over a time period that you specify. If the metric exceeds a given threshold, a notification is sent to an Amazon SNS topic or AWS Auto Scaling policy. CloudWatch alarms do not invoke actions because they are in a particular state. Rather the state must have changed and been maintained for a specified number of periods.

AWS CloudTrail Logs

CloudTrail provides a record of actions taken by a user, role, or an AWS service in Amazon RDS. CloudTrail captures all API calls for Amazon RDS as events, including calls from the console and from code calls to Amazon RDS API operations. Using the information collected by CloudTrail, you can determine the request that was made to Amazon RDS, the IP address from which the request was made, who made the request, when it was made, and additional details. For more information, see [Working with AWS CloudTrail and Amazon RDS \(p. 557\)](#).

Enhanced Monitoring

Amazon RDS provides metrics in real time for the operating system (OS) that your DB instance runs on. You can view the metrics for your DB instance using the console, or consume the Enhanced Monitoring JSON output from Amazon CloudWatch Logs in a monitoring system of your choice. For more information, see [Using Enhanced Monitoring \(p. 471\)](#).

Amazon RDS Performance Insights

Performance Insights expands on existing Amazon RDS monitoring features to illustrate your database's performance and help you analyze any issues that affect it. With the Performance Insights dashboard, you can visualize the database load and filter the load by waits, SQL statements, hosts, or users. For more information, see [Using Performance Insights on Amazon RDS \(p. 412\)](#).

Database Logs

You can view, download, and watch database logs using the AWS Management Console, AWS CLI, or RDS API. For more information, see [Accessing Amazon RDS database log files \(p. 504\)](#).

Amazon RDS Recommendations

Amazon RDS provides automated recommendations for database resources. These recommendations provide best practice guidance by analyzing DB instance configuration, usage, and performance data. For more information, see [Using Amazon RDS recommendations \(p. 407\)](#).

Amazon RDS Event Notification

Amazon RDS uses the Amazon Simple Notification Service (Amazon SNS) to provide notification when an Amazon RDS event occurs. These notifications can be in any notification form supported by Amazon SNS for an AWS Region, such as an email, a text message, or a call to an HTTP endpoint. For more information, see [Using Amazon RDS event notification \(p. 487\)](#).

AWS Trusted Advisor

Trusted Advisor draws upon best practices learned from serving hundreds of thousands of AWS customers. Trusted Advisor inspects your AWS environment and then makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps. All AWS customers have access to five Trusted Advisor checks. Customers with a Business or Enterprise support plan can view all Trusted Advisor checks.

Trusted Advisor has the following Amazon RDS-related checks:

- Amazon RDS Idle DB Instances
- Amazon RDS Security Group Access Risk
- Amazon RDS Backups
- Amazon RDS Multi-AZ

For more information on these checks, see [Trusted Advisor best practices \(checks\)](#).

For more information about monitoring Amazon RDS, see [Monitoring an Amazon RDS DB instance \(p. 399\)](#).

Compliance validation for Amazon RDS

Third-party auditors assess the security and compliance of Amazon RDS as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see [AWS services in scope by compliance program](#). For general information, see [AWS compliance programs](#).

You can download third-party audit reports using AWS Artifact. For more information, see [Downloading reports in AWS Artifact](#).

Your compliance responsibility when using Amazon RDS is determined by the sensitivity of your data, your organization's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- [Security and compliance quick start guides](#) – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- [Architecting for HIPAA security and compliance whitepaper](#) – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- [AWS compliance resources](#) – This collection of workbooks and guides that might apply to your industry and location.
- [AWS Config](#) – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- [AWS Security Hub](#) – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in Amazon RDS

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see [AWS global infrastructure](#).

In addition to the AWS global infrastructure, Amazon RDS offers features to help support your data resiliency and backup needs.

Backup and restore

Amazon RDS creates and saves automated backups of your DB instance. Amazon RDS creates a storage volume snapshot of your DB instance, backing up the entire DB instance and not just individual databases.

Amazon RDS creates automated backups of your DB instance during the backup window of your DB instance. Amazon RDS saves the automated backups of your DB instance according to the backup retention period that you specify. If necessary, you can recover your database to any point in time during the backup retention period. You can also back up your DB instance manually, by manually creating a DB snapshot.

You can create a DB instance by restoring from this DB snapshot as a disaster recovery solution if the source DB instance fails.

For more information, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#).

Replication

Amazon RDS uses the MariaDB, MySQL, Oracle, and PostgreSQL DB engines' built-in replication functionality to create a special type of DB instance called a read replica from a source DB instance. Updates made to the source DB instance are asynchronously copied to the read replica. You can reduce the load on your source DB instance by routing read queries from your applications to the read replica. Using read replicas, you can elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads. You can promote a read replica to a standalone instance as a disaster recovery solution if the source DB instance fails. For some DB engines, Amazon RDS also supports other replication options.

For more information, see [Working with read replicas \(p. 278\)](#).

Failover

Amazon RDS provides high availability and failover support for DB instances using Multi-AZ deployments. Amazon RDS uses several different technologies to provide failover support. Multi-AZ deployments for Oracle, PostgreSQL, MySQL, and MariaDB DB instances use Amazon's failover technology. SQL Server DB instances use SQL Server Database Mirroring (DBM).

For more information, see [High availability \(Multi-AZ\) for Amazon RDS \(p. 53\)](#).

Infrastructure security in Amazon RDS

As a managed service, Amazon RDS is protected by the AWS global network security procedures that are described in the [Amazon Web Services: Overview of security processes](#) whitepaper.

You use AWS published API calls to access Amazon RDS through the network. Clients must support Transport Layer Security (TLS) 1.0. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the [AWS Security Token Service \(AWS STS\)](#) to generate temporary security credentials to sign requests.

In addition, Amazon RDS offers features to help support infrastructure security.

Security groups

Security groups control the access that traffic has in and out of a DB instance. By default, network access is turned off to a DB instance. You can specify rules in a security group that allow access from an IP address range, port, or security group. After ingress rules are configured, the same rules apply to all DB instances that are associated with that security group.

For more information, see [Controlling access with security groups \(p. 1699\)](#).

Public accessibility

When you launch a DB instance inside a virtual private cloud (VPC) based on the Amazon VPC service, you can turn on or off public accessibility for that instance. To designate whether the DB instance that you create has a DNS name that resolves to a public IP address, you use the *Public accessibility* parameter. By using this parameter, you can designate whether there is public access to the DB instance. You can modify a DB instance to turn on or off public accessibility by modifying the *Public accessibility* parameter.

For more information, see [Hiding a DB instance in a VPC from the internet \(p. 1729\)](#).

Note

If your DB instance is in a VPC but isn't publicly accessible, you can also use an AWS Site-to-Site VPN connection or an AWS Direct Connect connection to access it from a private network. For more information, see [Internetwork traffic privacy \(p. 1643\)](#).

Amazon RDS API and interface VPC endpoints (AWS PrivateLink)

You can establish a private connection between your VPC and Amazon RDS API endpoints by creating an *interface VPC endpoint*. Interface endpoints are powered by [AWS PrivateLink](#).

AWS PrivateLink enables you to privately access Amazon RDS API operations without an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with Amazon RDS API endpoints to launch, modify, or terminate DB instances. Your instances also don't need public IP addresses to use any of the available RDS API operations. Traffic between your VPC and Amazon RDS doesn't leave the Amazon network.

Each interface endpoint is represented by one or more elastic network interfaces in your subnets. For more information on elastic network interfaces, see [Elastic network interfaces](#) in the *Amazon EC2 User Guide*.

For more information about VPC endpoints, see [Interface VPC endpoints \(AWS PrivateLink\)](#) in the *Amazon VPC User Guide*. For more information about RDS API operations, see [Amazon RDS API Reference](#).

Considerations for VPC endpoints

Before you set up an interface VPC endpoint for Amazon RDS API endpoints, ensure that you review [Interface endpoint properties and limitations](#) in the *Amazon VPC User Guide*.

All RDS API operations relevant to managing Amazon RDS resources are available from your VPC using AWS PrivateLink.

VPC endpoint policies are supported for RDS API endpoints. By default, full access to RDS API operations is allowed through the endpoint. For more information, see [Controlling access to services with VPC endpoints](#) in the *Amazon VPC User Guide*.

Availability

Amazon RDS API currently supports VPC endpoints in the following AWS Regions:

- US East (Ohio)
- US East (N. Virginia)
- US West (N. California)
- US West (Oregon)
- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Mumbai)
- Asia Pacific (Seoul)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Canada (Central)
- Europe (Frankfurt)

- Europe (Ireland)
- Europe (London)
- Europe (Paris)
- Europe (Stockholm)
- Europe (Milan)
- Middle East (Bahrain)
- South America (São Paulo)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Creating an interface VPC endpoint for Amazon RDS API

You can create a VPC endpoint for the Amazon RDS API using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see [Creating an interface endpoint](#) in the *Amazon VPC User Guide*.

Create a VPC endpoint for Amazon RDS API using the service name `com.amazonaws.region.rds`.

Excluding AWS Regions in China, if you enable private DNS for the endpoint, you can make API requests to Amazon RDS with the VPC endpoint using its default DNS name for the AWS Region, for example `rds.us-east-1.amazonaws.com`. For the China (Beijing) and China (Ningxia) AWS Regions, you can make API requests with the VPC endpoint using `rds-api.cn-north-1.amazonaws.com.cn` and `rds-api.cn-northwest-1.amazonaws.com.cn`, respectively.

For more information, see [Accessing a service through an interface endpoint](#) in the *Amazon VPC User Guide*.

Creating a VPC endpoint policy for Amazon RDS API

You can attach an endpoint policy to your VPC endpoint that controls access to Amazon RDS API. The policy specifies the following information:

- The principal that can perform actions.
- The actions that can be performed.
- The resources on which actions can be performed.

For more information, see [Controlling access to services with VPC endpoints](#) in the *Amazon VPC User Guide*.

Example: VPC endpoint policy for Amazon RDS API actions

The following is an example of an endpoint policy for Amazon RDS API. When attached to an endpoint, this policy grants access to the listed Amazon RDS API actions for all principals on all resources.

```
{  
    "Statement": [  
        {
```

```
        "Principal": "*",
        "Effect": "Allow",
        "Action": [
            "rds:CreateDBInstance",
            "rds:ModifyDBInstance",
            "rds>CreateDBSnapshot"
        ],
        "Resource": "*"
    }
}
```

Example: VPC endpoint policy that denies all access from a specified AWS account

The following VPC endpoint policy denies AWS account 123456789012 all access to resources using the endpoint. The policy allows all actions from other accounts.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

Security best practices for Amazon RDS

Use AWS Identity and Access Management (IAM) accounts to control access to Amazon RDS API operations, especially operations that create, modify, or delete Amazon RDS resources. Such resources include DB instances, security groups, and parameter groups. Also use IAM to control actions that perform common administrative actions such as backing up and restoring DB instances.

- Create an individual IAM user for each person who manages Amazon RDS resources, including yourself. Don't use AWS root credentials to manage Amazon RDS resources.
- Grant each user the minimum set of permissions required to perform his or her duties.
- Use IAM groups to effectively manage permissions for multiple users.
- Rotate your IAM credentials regularly.
- Configure AWS Secrets Manager to automatically rotate the secrets for Amazon RDS. For more information, see [Rotating your AWS Secrets Manager secrets](#) in the *AWS Secrets Manager User Guide*. You can also retrieve the credential from AWS Secrets Manager programmatically. For more information, see [Retrieving the secret value](#) in the *AWS Secrets Manager User Guide*.

For more information about IAM, see [AWS Identity and Access Management](#). For information on IAM best practices, see [IAM best practices](#).

Use the AWS Management Console, the AWS CLI, or the RDS API to change the password for your master user. If you use another tool, such as a SQL client, to change the master user password, it might result in privileges being revoked for the user unintentionally.

Controlling access with security groups

Security groups control the access that traffic has in and out of a DB instance. Three types of security groups are used with Amazon RDS: VPC security groups, DB security groups, and EC2-Classic security groups. In simple terms, these work as follows:

- A VPC security group controls access to DB instances and EC2 instances inside a VPC.
- A DB security group controls access to EC2-Classic DB instances that are not in a VPC.
- An EC2-Classic security group controls access to an EC2 instance. For more information about EC2-Classic security groups, see [EC2-Classic](#) in the Amazon EC2 documentation.

By default, network access is disabled for a DB instance. You can specify rules in a security group that allow access from an IP address range, port, or security group. Once ingress rules are configured, the same rules apply to all DB instances that are associated with that security group. You can specify up to 20 rules in a security group.

VPC security groups

Each VPC security group rule enables a specific source to access a DB instance in a VPC that is associated with that VPC security group. The source can be a range of addresses (for example, 203.0.113.0/24), or another VPC security group. By specifying a VPC security group as the source, you allow incoming traffic from all instances (typically application servers) that use the source VPC security group. VPC security groups can have rules that govern both inbound and outbound traffic, though the outbound traffic rules typically do not apply to DB instances. Outbound traffic rules only apply if the DB instance acts as a client. For example, outbound traffic rules apply to an Oracle DB instance with outbound database links. You must use the [Amazon EC2 API](#) or the **Security Group** option on the VPC Console to create VPC security groups.

When you create rules for your VPC security group that allow access to the instances in your VPC, you must specify a port for each range of addresses that the rule allows access for. For example, if you want to enable SSH access to instances in the VPC, then you create a rule allowing access to TCP port 22 for the specified range of addresses.

You can configure multiple VPC security groups that allow access to different ports for different instances in your VPC. For example, you can create a VPC security group that allows access to TCP port 80 for web servers in your VPC. You can then create another VPC security group that allows access to TCP port 3306 for RDS for MySQL DB instances in your VPC.

For more information on VPC security groups, see [Security groups](#) in the *Amazon Virtual Private Cloud User Guide*.

Note

If your DB instance is in a VPC but isn't publicly accessible, you can also use an AWS Site-to-Site VPN connection or an AWS Direct Connect connection to access it from a private network. For more information, see [Internetwork traffic privacy \(p. 1643\)](#).

DB security groups

DB security groups are used with DB instances that are not in a VPC and on the EC2-Classic platform. Each DB security group rule enables a specific source to access a DB instance that is associated with that

DB security group. The source can be a range of addresses (for example, 203.0.113.0/24), or an EC2-Classic security group. When you specify an EC2-Classic security group as the source, you allow incoming traffic from all EC2 instances that use that EC2-Classic security group. DB security group rules apply to inbound traffic only; outbound traffic is not currently permitted for DB instances.

You don't need to specify a destination port number when you create DB security group rules. The port number defined for the DB instance is used as the destination port number for all rules defined for the DB security group. DB security groups can be created using the Amazon RDS API operations or the Amazon RDS page of the AWS Management Console.

For more information about working with DB security groups, see [Working with DB security groups \(EC2-Classic platform\) \(p. 1704\)](#).

DB security groups vs. VPC security groups

The following table shows the key differences between DB security groups and VPC security groups.

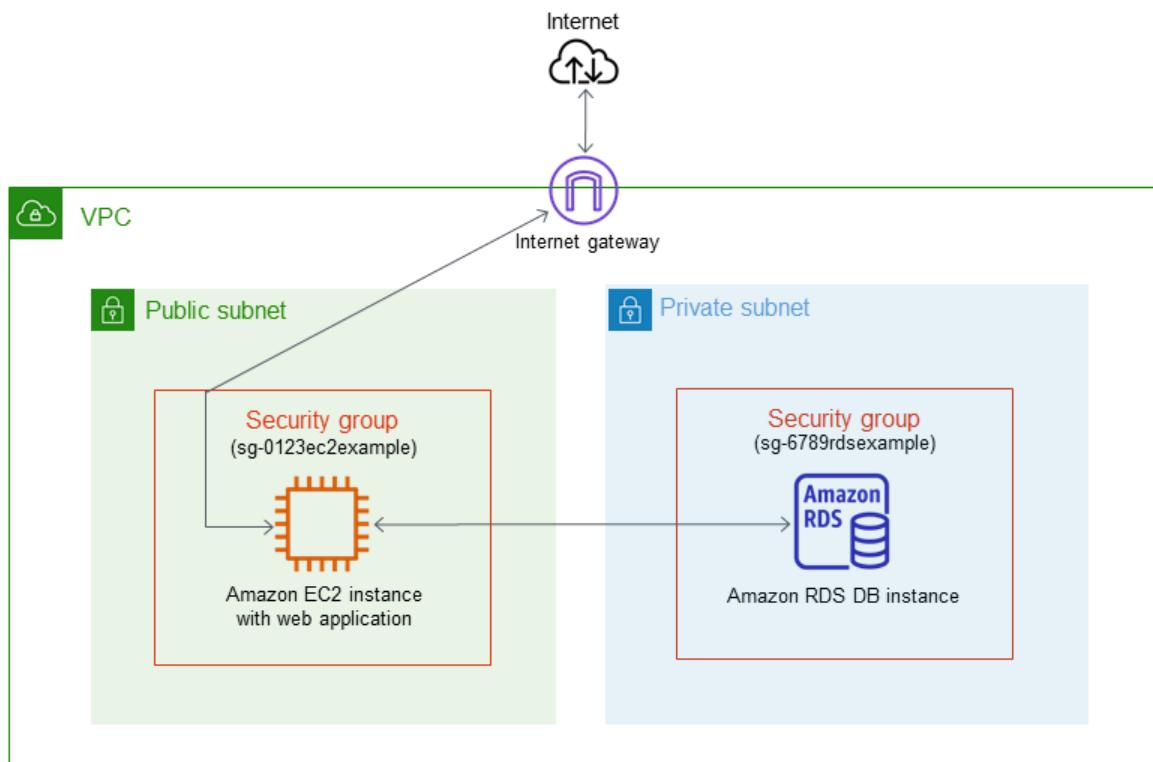
DB security group	VPC security group
Controls access to DB instances outside a VPC.	Controls access to DB instances in VPC.
Uses Amazon RDS API operations or the Amazon RDS page of the AWS Management Console to create and manage group and rules.	Uses Amazon EC2 API operations or the Amazon VPC page of the AWS Management Console to create and manage group and rules.
When you add a rule to a group, you don't need to specify port number or protocol.	When you add a rule to a group, specify the protocol as TCP. In addition, specify the same port number that you used to create the DB instances (or options) that you plan to add as members to the group.
Groups allow access from EC2-Classic security groups in your AWS account or other accounts.	Groups allow access from other VPC security groups in your VPC only.

Security group scenario

A common use of a DB instance in a VPC is to share data with an application server running in an Amazon EC2 instance in the same VPC, which is accessed by a client application outside the VPC. For this scenario, you use the RDS and VPC pages on the AWS Management Console or the RDS and EC2 API operations to create the necessary instances and security groups:

1. Create a VPC security group (for example, `sg-0123ec2example`) and define inbound rules that use the IP addresses of the client application as the source. This security group allows your client application to connect to EC2 instances in a VPC that uses this security group.
2. Create an EC2 instance for the application and add the EC2 instance to the VPC security group (`sg-0123ec2example`) that you created in the previous step.
3. Create a second VPC security group (for example, `sg-6789rdsexample`) and create a new rule by specifying the VPC security group that you created in step 1 (`sg-0123ec2example`) as the source.
4. Create a new DB instance and add the DB instance to the VPC security group (`sg-6789rdsexample`) that you created in the previous step. When you create the DB instance, use the same port number as the one specified for the VPC security group (`sg-6789rdsexample`) rule that you created in step 3.

The following diagram shows this scenario.



For more information about using a VPC, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).

Creating a VPC security group

You can create a VPC security group for a DB instance by using the VPC console. For information about creating a security group, see [Provide access to your DB instance in your VPC by creating a security group \(p. 70\)](#) and [Security groups](#) in the *Amazon Virtual Private Cloud User Guide*.

Associating a security group with a DB instance

You can associate a security group with a DB instance by using **Modify** on the RDS console, the `ModifyDBInstance` Amazon RDS API, or the `modify-db-instance` AWS CLI command.

For information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#). For security group considerations when you restore a DB instance from a DB snapshot, see [Security group considerations \(p. 349\)](#).

Deleting DB VPC security groups

DB VPC security groups are an RDS mechanism to synchronize security information with a VPC security group. However, this synchronization is no longer required, because RDS has been updated to use VPC security group information directly.

Note

DB VPC security groups are deprecated, and they are different from DB security groups, VPC security groups, and EC2-Classic security groups.

We strongly recommend that you delete any DB VPC security groups that you currently use. If you don't delete your DB VPC security groups, you might encounter unintended behaviors with your DB instances,

which can be as severe as losing access to a DB instance. The unintended behaviors might result from taking an action such as an update to a DB instance, a parameter group, or similar. Such updates cause RDS to resynchronize the DB VPC security group with the VPC security group. This resynchronization can result in your security information being overwritten with incorrect and outdated security information. This in turn can have a severe impact on your ability to access to your DB instances.

How can I determine if I have a DB VPC security group?

Because DB VPC security groups have been deprecated, they don't appear in the RDS console. However, you can call the [describe-db-security-groups](#) AWS CLI command or the [DescribeDBSecurityGroups](#) API operation to determine if you have any DB VPC security groups.

In this case, you can call the [describe-db-security-groups](#) AWS CLI command with JSON specified as the output format. If you do, you can identify DB VPC security groups by the VPC identifier on the second line of the output for the security group as shown in the following example.

```
{  
    "DBSecurityGroups": [  
        {  
            "VpcId": "vpc-abcd1234",  
            "DBSecurityGroupDescription": "default:vpc-abcd1234",  
            "IPRanges": [  
                {  
                    "Status": "authorized",  
                    "CIDRIP": "xxx.xxx.xxx.xxx/n"  
                },  
                {  
                    "Status": "authorized",  
                    "CIDRIP": "xxx.xxx.xxx.xxx/n "  
                }  
            ],  
            "OwnerId": "123456789012",  
            "EC2SecurityGroups": [],  
            "DBSecurityGroupName": "default:vpc-abcd1234"  
        }  
    ]  
}
```

If you run the [DescribeDBSecurityGroups](#) API operation, then you can identify DB VPC security groups using the <VpcId> response element as shown in the following example.

```
<DBSecurityGroup>  
    <EC2SecurityGroups/>  
    <DBSecurityGroupDescription>default:vpc-abcd1234</DBSecurityGroupDescription>  
    <IPRanges>  
        <IPRange>  
            <CIDRIP>xxx.xxx.xxx.xxx/n</CIDRIP>  
            <Status>authorized</Status>  
        </IPRange>  
        <IPRange>  
            <CIDRIP>xxx.xxx.xxx.xxx/n</CIDRIP>  
            <Status>authorized</Status>  
        </IPRange>  
    </IPRanges>  
    <VpcId>vpc-abcd1234</VpcId>  
    <OwnerId>123456789012</OwnerId>  
    <DBSecurityGroupName>default:vpc-abcd1234</DBSecurityGroupName>  
</DBSecurityGroup>
```

How do I delete a DB VPC security group?

Because DB VPC security groups don't appear in the RDS console, you must call the [delete-db-security-group](#) AWS CLI command or the [DeleteDBSecurityGroup](#) API operation to delete a DB VPC security group.

After you delete a DB VPC security group, your DB instances in your VPC continue to be secured by the VPC security group for that VPC. The DB VPC security group that was deleted was merely a copy of the VPC security group information.

Review your AWS CloudFormation templates

Older versions of AWS CloudFormation templates can contain instructions to create a DB VPC security group. Because DB VPC security groups are not yet fully deprecated, they can still be created. Make sure that any AWS CloudFormation templates that you use to provision a DB instance with security settings don't also create a DB VPC security group. Don't use AWS CloudFormation templates that create an RDS `DBSecurityGroup` with an `EC2VpcId` as shown in the following example.

```
{  
  "DbSecurityByEC2SecurityGroup" : {  
    "Type" : "AWS::RDS::DBSecurityGroup",  
    "Properties" : {  
      "GroupDescription" : "Ingress for security group",  
      "EC2VpcId" : "MyVPC",  
      "DBSecurityGroupIngress" : [ {  
        "EC2SecurityGroupId" : "sg-b0ff1111",  
        "EC2SecurityGroupOwnerId" : "111122223333"  
      }, {  
        "EC2SecurityGroupId" : "sg-ffd722222",  
        "EC2SecurityGroupOwnerId" : "111122223333"  
      } ]  
    }  
  }  
}
```

Instead, add security information for your DB instances in a VPC using VPC security groups, as shown in the following example.

```
{  
  "DBInstance" : {  
    "Type": "AWS::RDS::DBInstance",  
    "Properties": {  
      "DBName" : { "Ref" : "DBName" },  
      "Engine" : "MySQL",  
      "MultiAZ" : { "Ref": "MultiAZDatabase" },  
      "MasterUsername" : { "Ref" : "<master_username>" },  
      "DBInstanceClass" : { "Ref" : "DBClass" },  
      "AllocatedStorage" : { "Ref" : "DBAllocatedStorage" },  
      "MasterUserPassword": { "Ref" : "<master_password>" },  
      "VPCSecurityGroups" : [ { "Fn::GetAtt": [ "VPCSecurityGroup", "GroupId" ] } ]  
    }  
  }  
}
```

Working with DB security groups (EC2-Classic platform)

By default, network access is turned off to a DB instance. You can specify rules in a *security group* that allows access from an IP address range, port, or security group. Once ingress rules are configured, the same rules apply to all DB instances that are associated with that security group. You can specify up to 20 rules in a security group.

Amazon RDS supports two different kinds of security groups. The one you use depends on which Amazon RDS platform you are on:

- **VPC security groups** – for the EC2-VPC platform.
- **DB security groups** – for the EC2-Classic platform.

You are most likely on the EC2-VPC platform (and must use VPC security groups) if any of the following are true:

- If you are a new Amazon RDS customer.
- If you have never created a DB instance before.
- If you are creating a DB instance in an AWS Region you have not used before.

Otherwise, if you are on the EC2-Classic platform, you use DB security groups to manage access to your Amazon RDS DB instances. For more information about the differences between DB security groups and VPC security groups, see [Controlling access with security groups \(p. 1699\)](#).

Note

To determine which platform you are on, see [Determining whether you are using the EC2-VPC or EC2-Classic platform \(p. 1718\)](#).

If you are on the EC2-VPC platform, you must use VPC security groups instead of DB security groups. For more information about using a VPC, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).

Topics

- [Creating a DB security group \(p. 1704\)](#)
- [Listing available DB security groups \(p. 1706\)](#)
- [Viewing a DB security group \(p. 1706\)](#)
- [Associating a DB security group with a DB instance \(p. 1707\)](#)
- [Authorizing network access to a DB security group from an IP range \(p. 1708\)](#)
- [Authorizing network access to a DB instance from an Amazon EC2 instance \(p. 1709\)](#)
- [Revoking network access to a DB instance from an IP range \(p. 1711\)](#)

Creating a DB security group

To create a DB security group, you need to provide a name and a description.

Console

To create a DB security group

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. From the navigation pane, choose **Security Groups**.

Note

If you are on the EC2-VPC platform, the **Security Groups** option does not appear in the navigation pane. In this case, you must use VPC security groups instead of DB security groups. For more information about using a VPC, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).

3. Choose **Create DB Security Group**.
4. Type the name and description of the new DB security group in the **Name** and **Description** text boxes. The security group name can't contain spaces and can't start with a number.
5. Choose **Yes, Create**.

The DB security group is created.

A newly created DB security group doesn't provide access to a DB instance by default. You must specify a range of IP addresses or an EC2-Classic security group that can have access to the DB instance. To specify IP addresses or an EC2-Classic security group for a DB security group, see [Authorizing network access to a DB security group from an IP range \(p. 1708\)](#).

AWS CLI

To create a DB security group, use the AWS CLI command `create-db-security-group`.

Example

For Linux, macOS, or Unix:

```
aws rds create-db-security-group \
--db-security-group-name mydbsecuritygroup \
--db-security-group-description "My new security group"
```

For Windows:

```
aws rds create-db-security-group ^
--db-security-group-name mydbsecuritygroup ^
--db-security-group-description "My new security group"
```

A newly created DB security group doesn't provide access to a DB instance by default. You must specify a range of IP addresses or an EC2-Classic security group that can have access to the DB instance. To specify IP addresses or an EC2-Classic security group for a DB security group, see [Authorizing network access to a DB security group from an IP range \(p. 1708\)](#).

API

To create a DB security group, call the Amazon RDS function `CreateDBSecurityGroup` with the following parameters:

- `DBSecurityGroupName` = *mydbsecuritygroup*
- `Description` = *"My new security group"*

Example

```
https://rds.amazonaws.com/
?Action=CreateDBSecurityGroup
&DBSecurityGroupName=mydbsecuritygroup
&Description=My%20new%20db%20security%20group
&Version=2012-01-15
&SignatureVersion=2
```

```
&SignatureMethod=HmacSHA256
&Timestamp=2012-01-20T22%3A06%3A23.624Z
&AWSAccessKeyId=<Access Key ID>
&Signature=<Signature>
```

A newly created DB security group doesn't provide access to a DB instance by default. You must specify a range of IP addresses or an EC2-Classic security group that can have access to the DB instance. To specify IP addresses or an EC2-Classic security group for a DB security group, see [Authorizing network access to a DB security group from an IP range \(p. 1708\)](#).

Listing available DB security groups

You can list which DB security groups have been created for your AWS account.

Console

To list all available DB security groups for an AWS account

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. From the navigation pane, choose **Security Groups**.

The available DB security groups appear in the **DB Security Groups** list.

Note

If you are on the EC2-VPC platform, the **Security Groups** option does not appear in the navigation pane. In this case, you must use VPC security groups instead of DB security groups. For more information about using a VPC, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).

AWS CLI

To list all available DB security groups for an AWS account, Use the AWS CLI command `describe-db-security-groups` with no parameters.

Example

```
aws rds describe-db-security-groups
```

API

To list all available DB security groups for an AWS account, call `DescribeDBSecurityGroups` with no parameters.

Example

```
https://rds.amazonaws.com/
?Action=DescribeDBSecurityGroups
&MaxRecords=100
&Version=2009-10-16
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&AWSAccessKeyId=<Access Key ID>
&Signature=<Signature>
```

Viewing a DB security group

You can view detailed information about your DB security group to see what IP ranges have been authorized.

Console

To view properties of a specific DB security group

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. From the navigation pane, choose **Security Groups**.

Note

If you are on the EC2-VPC platform, the **Security Groups** option does not appear in the navigation pane. In this case, you must use VPC security groups instead of DB security groups. For more information about using a VPC, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).

3. Select the details icon for the DB security group you want to view. The detailed information for the DB security group is displayed.

AWS CLI

To view the properties of a specific DB security group use the AWS CLI `describe-db-security-groups`. Specify the DB security group you want to view.

Example

For Linux, macOS, or Unix:

```
aws rds describe-db-security-groups \
--db-security-group-name mydbsecuritygroup
```

For Windows:

```
aws rds describe-db-security-groups ^
--db-security-group-name mydbsecuritygroup
```

API

To view properties of a specific DB security group, call `DescribeDBSecurityGroups` with the following parameters:

- `DBSecurityGroupName=mydbsecuritygroup`

Example

```
https://rds.amazonaws.com/
?Action=DescribeDBSecurityGroups
&DBSecurityGroupName=mydbsecuritygroup
&Version=2009-10-16
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2009-10-16T22%3A23%3A07.107Z
&AWSAccessKeyId=<Access Key ID>
&Signature=<Signature>
```

Associating a DB security group with a DB instance

You can associate a DB security group with a DB instance using the RDS console's **Modify** option, the `ModifyDBInstance` Amazon RDS API, or the AWS CLI `modify-db-instance` command.

For information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Authorizing network access to a DB security group from an IP range

By default, network access is turned off to a DB instance. If you want to access a DB instance that is not in a VPC, you must set access rules for a DB security group to allow access from specific EC2-Classic security groups or CIDR IP ranges. You then must associate that DB instance with that DB security group. This process is called *ingress*. Once ingress is configured for a DB security group, the same ingress rules apply to all DB instances associated with that DB security group.

Warning

Talk with your network administrator if you are intending to access a DB instance behind a firewall to determine the IP addresses you should use.

In following example, you configure a DB security group with an ingress rule for a CIDR IP range.

Console

To configure a DB security group with an ingress rule for a CIDR IP range

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. From the navigation pane, choose **Security Groups**.

Note

If you are on the EC2-VPC platform, the **Security Groups** option does not appear in the navigation pane. In this case, you must use VPC security groups instead of DB security groups. For more information about using a VPC, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).

3. Select the details icon for the DB security group you want to authorize.
4. In the details page for your security group, select *CIDR/IP* from the **Connection Type** drop-down list, type the CIDR range for the ingress rule you want to add to this DB security group into the **CIDR** text box, and choose **Authorize**.

Tip

The AWS Management Console displays a CIDR IP based on your connection below the CIDR text field. If you are not accessing the DB instance from behind a firewall, you can use this CIDR IP.

5. The status of the ingress rule is **authorizing** until the new ingress rule has been applied to all DB instances that are associated with the DB security group that you modified. After the ingress rule has been successfully applied, the status changes to **authorized**.

AWS CLI

To configure a DB security group with an ingress rule for a CIDR IP range, use the AWS CLI command `aws rds authorize-db-security-group-ingress`.

Example

For Linux, macOS, or Unix:

```
aws rds authorize-db-security-group-ingress \
--db-security-group-name mydbsecuritygroup \
--cidr 192.168.1.10/27
```

For Windows:

```
aws rds authorize-db-security-group-ingress ^
--db-security-group-name mydbsecuritygroup ^
--cidrIp 192.168.1.10/27
```

The command should produce output similar to the following.

```
SECGROUP mydbsecuritygroup My new DBSecurityGroup
IP-RANGE 192.168.1.10/27 authorizing
```

API

To configure a DB security group with an ingress rule for a CIDR IP range, call the Amazon RDS API [AuthorizeDBSecurityGroupIngress](#) with the following parameters:

- DBSecurityGroupName = *mydbsecuritygroup*
- CIDRIP = *192.168.1.10/27*

Example

```
https://rds.amazonaws.com/
?Action=AuthorizeDBSecurityGroupIngress
&CIDRIP=192.168.1.10%2F27
&DBSecurityGroupName=mydbsecuritygroup
&Version=2009-10-16
&Action=AuthorizeDBSecurityGroupIngress
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2009-10-22T17%3A10%3A50.274Z
&AWSAccessKeyId=<Access Key ID>
&Signature=<Signature>
```

Authorizing network access to a DB instance from an Amazon EC2 instance

If you want to access your DB instance from an Amazon EC2 instance, you must first determine if your EC2 instance and DB instance are in a VPC. If you are using a default VPC, you can assign the same EC2 or VPC security group that you used for your EC2 instance when you create or modify the DB instance that the EC2 instance accesses.

If your DB instance and EC2 instance are not in a VPC, you must configure the DB instance's security group with an ingress rule that allows traffic from the Amazon EC2 instance. You do this by adding the EC2-Classic security group for the EC2 instance to the DB security group for the DB instance. In this example, you add an ingress rule to a DB security group for an EC2-Classic security group.

Important

- Adding an ingress rule to a DB security group for an EC2-Classic security group only grants access to your DB instances from Amazon EC2 instances associated with that EC2-Classic security group.
- You can't authorize an EC2-Classic security group that is in a different AWS Region than your DB instance. You can authorize an IP range, or specify an EC2-Classic security group in the same AWS Region that refers to IP address in another AWS Region. If you specify an IP range, we recommend that you use the private IP address of your Amazon EC2 instance, which provides a more direct network route from your Amazon EC2 instance to your Amazon RDS DB instance, and doesn't incur network charges for data sent outside of the Amazon network.

Console

To add an EC2-Classic security group to a DB security group

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. From the navigation pane, choose **Security Groups**.

Note

If you are on the EC2-VPC platform, the **Security Groups** option does not appear in the navigation pane. In this case, you must use VPC security groups instead of DB security groups. For more information about using a VPC, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).

3. Select the details icon for the DB security group you want to grant access.
4. In the details page for your security group, choose **EC2 Security Group** for **Connection Type**, and then select the EC2-Classic security group you want to use. Then choose **Authorize**.
5. The status of the ingress rule is **authorizing** until the new ingress rule has been applied to all DB instances that are associated with the DB security group that you modified. After the ingress rule has been successfully applied, the status changes to **authorized**.

AWS CLI

To grant access to an EC2-Classic security group, use the AWS CLI command `authorize-db-security-group-ingress`.

Example

For Linux, macOS, or Unix:

```
aws rds authorize-db-security-group-ingress \
  --db-security-group-name default \
  --ec2-security-group-name myec2group \
  --ec2-security-group-owner-id 987654321021
```

For Windows:

```
aws rds authorize-db-security-group-ingress ^
  --db-security-group-name default ^
  --ec2-security-group-name myec2group ^
  --ec2-security-group-owner-id 987654321021
```

The command should produce output similar to the following:

SECGROUP	Name	Description
SECGROUP	default	default
EC2-SECGROUP	myec2group	987654321021 authorizing

API

To authorize network access to an EC2-Classic security group, call that Amazon RDS API function, https://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_AuthorizeDBSecurityGroupIngress.html with the following parameters:

- `EC2SecurityGroupName = myec2group`
- `EC2SecurityGroupOwnerId = 987654321021`

Example

```
https://rds.amazonaws.com/  
?Action=AuthorizeDBSecurityGroupIngress  
&EC2SecurityGroupOwnerId=987654321021  
&EC2SecurityGroupName=myec2group  
&Version=2009-10-16  
&SignatureVersion=2  
&SignatureMethod=HmacSHA256  
&Timestamp=2009-10-22T17%3A10%3A50.274Z  
&AWSAccessKeyId=<Access Key ID>  
&Signature=<Signature>
```

Revoking network access to a DB instance from an IP range

You can easily revoke network access from a CIDR IP range to DB instances belonging to a DB security group by revoking the associated CIDR IP ingress rule.

In this example, you revoke an ingress rule for a CIDR IP range on a DB security group.

Console

To revoke an ingress rule for a CIDR IP range on a DB security group

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. From the navigation pane, choose **Security Groups**.

Note

If you are on the EC2-VPC platform, the **Security Groups** option does not appear in the navigation pane. In this case, you must use VPC security groups instead of DB security groups. For more information about using a VPC, see [Amazon Virtual Private Cloud VPCs and Amazon RDS \(p. 1718\)](#).

3. Select the details icon for the DB security group that has the ingress rule you want to revoke.
4. In the details page for your security group, choose **Remove** next to the ingress rule you want to revoke.
5. The status of the ingress rule is **revoking** until the ingress rule has been removed from all DB instances that are associated with the DB security group that you modified. After the ingress rule has been successfully removed, the ingress rule is removed from the DB security group.

AWS CLI

To revoke an ingress rule for a CIDR IP range on a DB security group, use the AWS CLI command `revoke-db-security-group-ingress`.

Example

For Linux, macOS, or Unix:

```
aws rds revoke-db-security-group-ingress \  
--db-security-group-name mydbsecuritygroup \  
--cidrip 192.168.1.1/27
```

For Windows:

```
aws rds revoke-db-security-group-ingress ^  
--db-security-group-name mydbsecuritygroup ^  
--cidrip 192.168.1.1/27
```

The command should produce output similar to the following.

```
SECGROUP mydbsecuritygroup My new DBSecurityGroup
IP-RANGE 192.168.1.1/27 revoking
```

API

To revoke an ingress rule for a CIDR IP range on a DB security group, call the Amazon RDS API operation https://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_RevokeDBSecurityGroupIngress.html with the following parameters:

- DBSecurityGroupName = *mydbsecuritygroup*
- CIDRIP = *192.168.1.10/27*

Example

```
https://rds.amazonaws.com/
?Action=RevokeDBSecurityGroupIngress
&DBSecurityGroupName=mydbsecuritygroup
&CIDRIP=192.168.1.10%2F27
&Version=2009-10-16
&SignatureVersion=2&SignatureMethod=HmacSHA256
&Timestamp=2009-10-22T22%3A32%3A12.515Z
&AWSAccessKeyId=<Access Key ID>
&Signature=<Signature>
```

Master user account privileges

When you create a new DB instance, the default master user that you use gets certain privileges for that DB instance. The following table shows the privileges and database roles the master user gets for each of the database engines.

Important

We strongly recommend that you do not use the master user directly in your applications. Instead, adhere to the best practice of using a database user created with the minimal privileges required for your application.

Note

If you accidentally delete the permissions for the master user, you can restore them by modifying the DB instance and setting a new master user password. For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Database engine	System privilege	Database role
MySQL and MariaDB	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES, INDEX, ALTER, SHOW DATABASES, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER ON *.* WITH GRANT OPTION, REPLICATION SLAVE (only for RDS for MySQL versions 5.6, 5.7 and 8.0, RDS for MariaDB)	—

Database engine	System privilege	Database role
PostgreSQL	<pre>CREATE ROLE, CREATE DB, PASSWORD VALID UNTIL INFINITY, CREATE EXTENSION, ALTER EXTENSION, DROP EXTENSION, CREATE TABLESPACE, ALTER < OBJECT> OWNER, CHECKPOINT, PG_CANCEL_BACKEND(), PG_TERMINATE_BACKEND(), SELECT PG_STAT_REPLICATION, EXECUTE PG_STAT_STATEMENTS_RESET(), OWN POSTGRES_FDW_HANDLER(), OWN POSTGRES_FDW_VALIDATOR(), OWN POSTGRES_FDW, EXECUTE PG_BUFFERCACHE_PAGES(), SELECT PG_BUFFERCACHE</pre>	RDS_SUPERUSER
Oracle	<pre>ALTER DATABASE LINK, ALTER PUBLIC DATABASE LINK, DROP ANY DIRECTORY, EXEMPT ACCESS POLICY, EXEMPT IDENTITY POLICY, GRANT ANY OBJECT PRIVILEGE, RESTRICTED SESSION, EXEMPT REDACTION POLICY</pre>	AQ_ADMINISTRATOR_ROLE, AQ_USER_ROLE, CONNECT, CTXAPP, DBA, EXECUTE_CATALOG_ROLE, RECOVERY_CATALOG_OWNER, RESOURCE, SELECT_CATALOG_ROLE
Microsoft SQL Server	ADMINISTER BULK OPERATIONS, ALTER ANY CONNECTION, ALTER ANY LINKED SERVER, ALTER ANY LOGIN, ALTER SERVER STATE, ALTER TRACE, CONNECT SQL, CREATE ANY DATABASE, VIEW ANY DATABASE, VIEW ANY DEFINITION, VIEW SERVER STATE, ALTER ANY SERVER ROLE, ALTER ANY USER, ALTER ON ROLE SQLAgentOperatorRole	DB_OWNER (database-level role), PROCESSADMIN (server-level role), SETUPADMIN (server-level role), SQLAgentUserRole (database-level role)

Using service-linked roles for Amazon RDS

Amazon RDS uses AWS Identity and Access Management (IAM) [service-linked roles](#). A service-linked role is a unique type of IAM role that is linked directly to Amazon RDS. Service-linked roles are predefined by Amazon RDS and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes using Amazon RDS easier because you don't have to manually add the necessary permissions. Amazon RDS defines the permissions of its service-linked roles, and unless defined otherwise, only Amazon RDS can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete the roles only after first deleting their related resources. This protects your Amazon RDS resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see [AWS services that work with IAM](#) and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for Amazon RDS

Amazon RDS uses the service-linked role named **AWSServiceRoleForRDS** – to allow Amazon RDS to call AWS services on behalf of your DB instances.

The **AWSServiceRoleForRDS** service-linked role trusts the following services to assume the role:

- `rds.amazonaws.com`

The role permissions policy allows Amazon RDS to complete the following actions on the specified resources:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AuthorizeSecurityGroupIngress",  
                "ec2>CreateNetworkInterface",  
                "ec2>CreateSecurityGroup",  
                "ec2>DeleteNetworkInterface",  
                "ec2>DeleteSecurityGroup",  
                "ec2:DescribeAvailabilityZones",  
                "ec2:DescribeInternetGateways",  
                "ec2:DescribeSecurityGroups",  
                "ec2:DescribeSubnets",  
                "ec2:DescribeVpcAttribute",  
                "ec2:DescribeVpcs",  
                "ec2:ModifyNetworkInterfaceAttribute",  
                "ec2:ModifyVpcEndpoint",  
                "ec2:RevokeSecurityGroupIngress",  
                "ec2>CreateVpcEndpoint",  
                "ec2:DescribeVpcEndpoints",  
                "ec2>DeleteVpcEndpoints",  
                "ec2:AssignPrivateIpAddresses",  
                "ec2:UnassignPrivateIpAddresses"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

```

},
{
    "Effect": "Allow",
    "Action": [
        "sns:Publish"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "logs>CreateLogGroup"
    ],
    "Resource": [
        "arn:aws:logs:***:log-group:/aws/rds/*",
        "arn:aws:logs:***:log-group:/aws/docdb/*",
        "arn:aws:logs:***:log-group:/aws/neptune/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "logs>CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
    "Resource": [
        "arn:aws:logs:***:log-group:/aws/rds/*:log-stream:*",
        "arn:aws:logs:***:log-group:/aws/docdb/*:log-stream:*",
        "arn:aws:logs:***:log-group:/aws/neptune/*:log-stream:*
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kinesis>CreateStream",
        "kinesis:PutRecord",
        "kinesis:PutRecords",
        "kinesis:DescribeStream",
        "kinesis:SplitShard",
        "kinesis:MergeShards",
        "kinesis>DeleteStream",
        "kinesis:UpdateShardCount"
    ],
    "Resource": [
        "arn:aws:kinesis:***:stream/aws-rds-das-*"
    ]
}
]
}

```

Note

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. If you encounter the following error message:

**Unable to create the resource. Verify that you have permission to create service linked role.
Otherwise wait and try again later.**

Make sure you have the following permissions enabled:

```
{
    "Action": "iam>CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam:***:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
```

```
"Condition": {  
    "StringLike": {  
        "iam:AWSServiceName": "rds.amazonaws.com"  
    }  
}
```

For more information, see [Service-linked role permissions in the IAM User Guide](#).

Creating a service-linked role for Amazon RDS

You don't need to manually create a service-linked role. When you create a DB instance, Amazon RDS creates the service-linked role for you.

Important

If you were using the Amazon RDS service before December 1, 2017, when it began supporting service-linked roles, then Amazon RDS created the AWSServiceRoleForRDS role in your account. To learn more, see [A new role appeared in my AWS account](#).

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you create a DB instance, Amazon RDS creates the service-linked role for you again.

Editing a service-linked role for Amazon RDS

Amazon RDS does not allow you to edit the AWSServiceRoleForRDS service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see [Editing a service-linked role](#) in the *IAM User Guide*.

Deleting a service-linked role for Amazon RDS

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must delete all of your DB instances before you can delete the service-linked role.

Cleaning up a service-linked role

Before you can use IAM to delete a service-linked role, you must first confirm that the role has no active sessions and remove any resources used by the role.

To check whether the service-linked role has an active session in the IAM console

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane of the IAM console, choose **Roles**. Then choose the name (not the check box) of the AWSServiceRoleForRDS role.
3. On the **Summary** page for the chosen role, choose the **Access Advisor** tab.
4. On the **Access Advisor** tab, review recent activity for the service-linked role.

Note

If you are unsure whether Amazon RDS is using the AWSServiceRoleForRDS role, you can try to delete the role. If the service is using the role, then the deletion fails and you can view the AWS Regions where the role is being used. If the role is being used, then you must wait

for the session to end before you can delete the role. You cannot revoke the session for a service-linked role.

If you want to remove the AWSServiceRoleForRDS role, you must first delete *all* of your DB instances .

Deleting all of your instances

Use one of these procedures to delete each of your instances.

To delete an instance (console)

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the instance that you want to delete.
4. For **Actions**, choose **Delete**.
5. If you are prompted for **Create final Snapshot?**, choose **Yes** or **No**.
6. If you chose **Yes** in the previous step, for **Final snapshot name** enter the name of your final snapshot.
7. Choose **Delete**.

To delete an instance (CLI)

See [delete-db-instance](#) in the *AWS CLI Command Reference*.

To delete an instance (API)

See [DeleteDBInstance](#) in the *Amazon RDS API Reference*.

You can use the IAM console, the IAM CLI, or the IAM API to delete the AWSServiceRoleForRDS service-linked role. For more information, see [Deleting a service-linked role](#) in the *IAM User Guide*.

Amazon Virtual Private Cloud VPCs and Amazon RDS

There are two Amazon Elastic Compute Cloud (EC2) platforms that host Amazon RDS DB instances, *EC2-VPC* and *EC2-Classic*. Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources, such as Amazon RDS DB instances, into a virtual private cloud (VPC).

When you use an Amazon VPC, you have control over your virtual networking environment: you can choose your own IP address range, create subnets, and configure routing and access control lists. The basic functionality of Amazon RDS is the same whether your DB instance is running in an Amazon VPC or not: Amazon RDS manages backups, software patching, automatic failure detection, and recovery. There is no additional cost to run your DB instance in an Amazon VPC.

Accounts that support only the *EC2-VPC* platform have a default VPC. All new DB instances are created in the default VPC unless you specify otherwise. If you are a new Amazon RDS customer, if you have never created a DB instance before, or if you are creating a DB instance in an AWS Region you have not used before, you are most likely on the *EC2-VPC* platform and have a default VPC.

Some legacy DB instances on the *EC2-Classic* platform are not in a VPC. The legacy *EC2-Classic* platform does not have a default VPC, but as is true for either platform, you can create your own VPC and specify that a DB instance be located in that VPC.

Topics

- [Determining whether you are using the EC2-VPC or EC2-Classic platform \(p. 1718\)](#)
- [Scenarios for accessing a DB instance in a VPC \(p. 1720\)](#)
- [Working with a DB instance in a VPC \(p. 1727\)](#)
- [Updating the VPC for a DB instance \(p. 1734\)](#)
- [Tutorial: Create an Amazon VPC for use with a DB instance \(p. 1737\)](#)

This documentation only discusses VPC functionality relevant to Amazon RDS DB instances. For more information about Amazon VPC, see [Amazon VPC Getting Started Guide](#) and [Amazon VPC User Guide](#). For information about using a network address translation (NAT) gateway, see [NAT gateways](#) in the [Amazon Virtual Private Cloud User Guide](#).

Determining whether you are using the EC2-VPC or EC2-Classic platform

Your AWS account and the AWS Region you choose determines which of the two RDS platforms your DB instance is created on: *EC2-VPC* or *EC2-Classic*. The type of platform determines if you have a default VPC, and which type of security group you use to provide access to your DB instance.

The legacy *EC2-Classic* platform is the original platform used by Amazon RDS. If you are on this platform and want to use a VPC, you must create the VPC using the Amazon VPC console or Amazon VPC API. Accounts that only support the *EC2-VPC* platform have a default VPC where all DB instances are created, and you must use either an EC2 or VPC security group to provide access to the DB instance.

Important

If you are a new Amazon RDS customer, if you have never created a DB instance before, or if you are creating a DB instance in an AWS Region you have not used before, in almost all cases you are on the *EC2-VPC* platform and have a default VPC.

You can tell which platform your AWS account in a given AWS Region is using by looking at the dashboard on the [RDS console](#) or [EC2](#) console. If you are a new Amazon RDS customer, if you have never

created a DB instance before, or if you are creating a DB instance in an AWS Region you have not used before, you might be redirected to the first-run console page and not see the home page following.

EC2-VPC platform in the console

If **Supported platforms** indicates VPC, as shown following in the RDS console, your AWS account in the current AWS Region uses the *EC2-VPC* platform, and uses a default VPC.

The screenshot shows the 'Resources' section of the Amazon RDS console. On the left, there's a sidebar with links like 'DB Instances (3/40)', 'DB Clusters (0/40)', 'Reserved instances (0/40)', 'Snapshots (24)', 'Recent events (12)', and 'Event subscriptions (1/20)'. On the right, there are several counts: 'Parameter groups (32)', 'Default (23)', 'Custom (9/100)'; 'Option groups (22)', 'Default (18)', 'Custom (4/20)'; and 'Subnet groups (4/50)'. Below these is a link 'Supported platforms VPC' which is circled in red. At the bottom, it says 'Default network vpc-2aed394c'.

Similarly, if **Supported platforms** indicates VPC, as shown following in the EC2 console, your AWS account in the current AWS Region uses the *EC2-VPC* platform, and uses a default VPC.

The screenshot shows the 'Account attributes' section of the EC2 console. It lists various settings: 'Supported platforms' (with a red circle around it), 'Default VPC' (with a red circle around it), 'Settings', 'EBS encryption', 'Zones', 'Default credit specification', and 'Console experiments'. Below 'Default VPC', the name 'vpc-2aed394c' is listed.

In both the RDS and EC2 console, the name of the default VPC is shown below the supported platform. To provide access to a DB instance created on the *EC2-VPC* platform, you must create a VPC security group. For information about creating a VPC security group, see [Tutorial: Create an Amazon VPC for use with a DB instance \(p. 1737\)](#).

EC2-Classic platform in the console

In both the RDS and EC2 console, if **Supported platforms** indicates EC2, VPC, your AWS account in the current AWS Region uses the *EC2-Classic* platform, and you do not have a default VPC. To provide

access to a DB instance created on the *EC2-Classic* platform, you must create a DB security group. For information about creating a DB security group, see [Creating a DB security group \(p. 1704\)](#).

Note

- You can create a VPC on the *EC2-Classic* platform, but one is not created for you by default as it is on accounts that support the *EC2-VPC* platform.
- If you want to move an existing DB instance into a VPC, you can use the AWS Management Console to do it easily. For more information, see [Moving a DB instance not in a VPC into a VPC \(p. 1735\)](#).

Scenarios for accessing a DB instance in a VPC

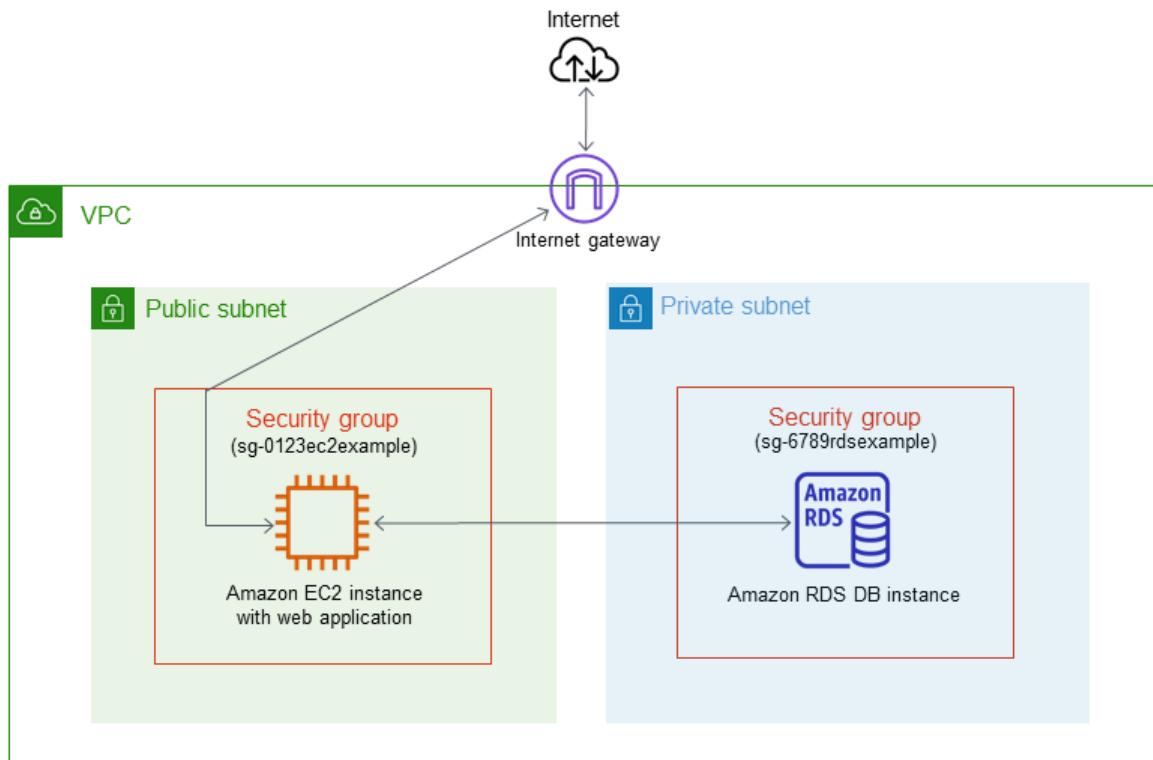
Amazon RDS supports the following scenarios for accessing a DB instance in a VPC:

- An EC2 instance in the same VPC (p. 1720)
- An EC2 instance in a different VPC (p. 1722)
- A client application through the internet (p. 1723)
- A private network (p. 1723)
- An EC2 instance not in a VPC (p. 1724)

A DB instance in a VPC accessed by an EC2 instance in the same VPC

A common use of a DB instance in a VPC is to share data with an application server that is running in an EC2 instance in the same VPC. This is the user scenario created if you use AWS Elastic Beanstalk to create an EC2 instance and a DB instance in the same VPC.

The following diagram shows this scenario.



The simplest way to manage access between EC2 instances and DB instances in the same VPC is to do the following:

- Create a VPC security group for your DB instances to be in. This security group can be used to restrict access to the DB instances. For example, you can create a custom rule for this security group that allows TCP access using the port you assigned to the DB instance when you created it and an IP address you use to access the DB instance for development or other purposes.
- Create a VPC security group for your EC2 instances (web servers and clients) to be in. This security group can, if needed, allow access to the EC2 instance from the internet by using the VPC's routing table. For example, you can set rules on this security group to allow TCP access to the EC2 instance over port 22.
- Create custom rules in the security group for your DB instances that allow connections from the security group you created for your EC2 instances. This would allow any member of the security group to access the DB instances.

For a tutorial that shows you how to create a VPC with both public and private subnets for this scenario, see [Tutorial: Create an Amazon VPC for use with a DB instance \(p. 1737\)](#).

To create a rule in a VPC security group that allows connections from another security group, do the following:

1. Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc>.
2. In the navigation pane, choose **Security Groups**.
3. Choose or create a security group for which you want to allow access to members of another security group. In the scenario preceding, this is the security group that you use for your DB instances. Choose the **Inbound rules** tab, and then choose **Edit inbound rules**.
4. On the **Edit inbound rules** page, choose **Add rule**.

5. From **Type**, choose the entry that corresponds to the port you used when you created your DB instance, such as **MYSQL/Aurora**.
6. In the **Source** box, start typing the ID of the security group, which lists the matching security groups. Choose the security group with members that you want to have access to the resources protected by this security group. In the scenario preceding, this is the security group that you use for your EC2 instance.
7. If required, repeat the steps for the TCP protocol by creating a rule with **All TCP** as the **Type** and your security group in the **Source** box. If you intend to use the UDP protocol, create a rule with **All UDP** as the **Type** and your security group in the **Source** box.
8. Choose **Save rules** when you are done.

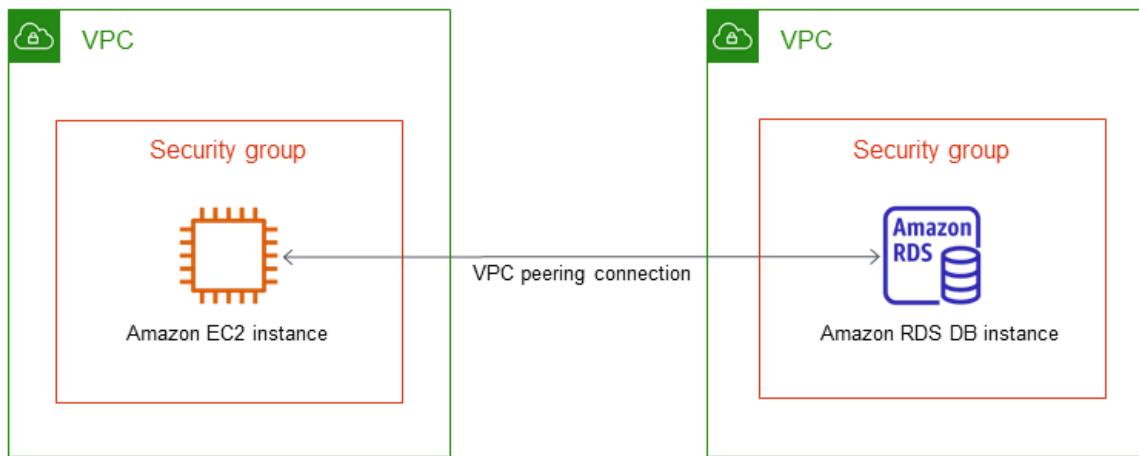
The following screen shows an inbound rule with a security group for its source.

Details	Inbound rules	Outbound rules	Tags
Inbound rules			
Type	Protocol	Port range	Source
MYSQL/Aurora	TCP	3306	sg-00bd2328e37926844 (tutorial-securitygroup)

A DB instance in a VPC accessed by an EC2 instance in a different VPC

When your DB instance is in a different VPC from the EC2 instance you are using to access it, you can use VPC peering to access the DB instance.

The following diagram shows this scenario.

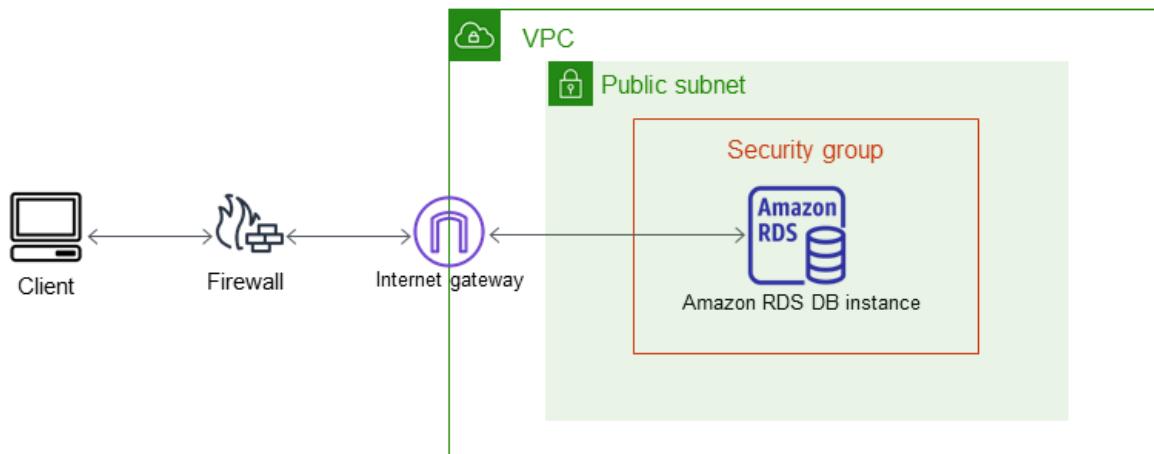


A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IP addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, with a VPC in another AWS account, or with a VPC in a different AWS Region. To learn more about VPC peering, see [VPC peering](#) in the *Amazon Virtual Private Cloud User Guide*.

A DB instance in a VPC accessed by a client application through the internet

To access a DB instance in a VPC from a client application through the internet, you configure a VPC with a single public subnet, and an internet gateway to enable communication over the internet.

The following diagram shows this scenario.



We recommend the following configuration:

- A VPC of size /16 (for example CIDR: 10.0.0.0/16). This size provides 65,536 private IP addresses.
- A subnet of size /24 (for example CIDR: 10.0.0.0/24). This size provides 256 private IP addresses.
- An Amazon RDS DB instance that is associated with the VPC and the subnet. Amazon RDS assigns an IP address within the subnet to your DB instance.
- An internet gateway which connects the VPC to the internet and to other AWS products.
- A security group associated with the DB instance. The security group's inbound rules allow your client application to access to your DB instance.

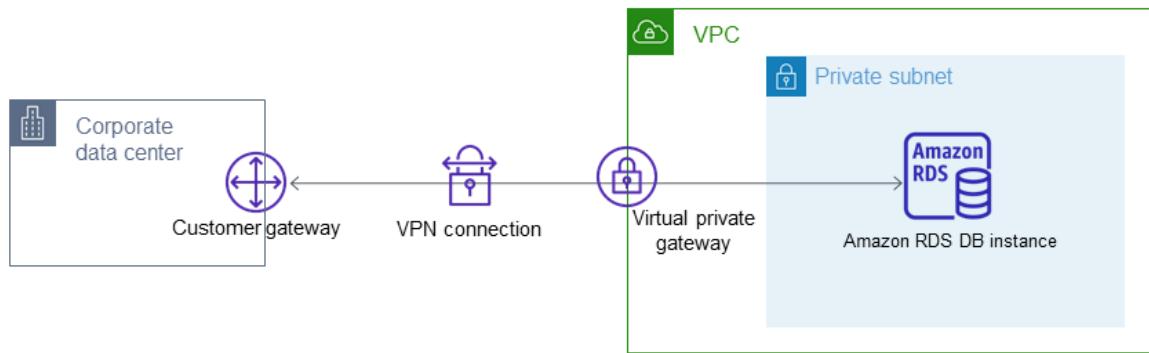
For information about creating a DB instance in a VPC, see [Creating a DB instance in a VPC \(p. 1730\)](#).

A DB instance in a VPC accessed by a private network

If your DB instance isn't publicly accessible, you have the following options for accessing it from a private network:

- An AWS Site-to-Site VPN connection. For more information, see [What is AWS Site-to-Site VPN?](#)
- An AWS Direct Connect connection. For more information, see [What is AWS Direct Connect?](#)

The following diagram shows a scenario with an AWS Site-to-Site VPN connection.



For more information, see [Internetwork traffic privacy \(p. 1643\)](#).

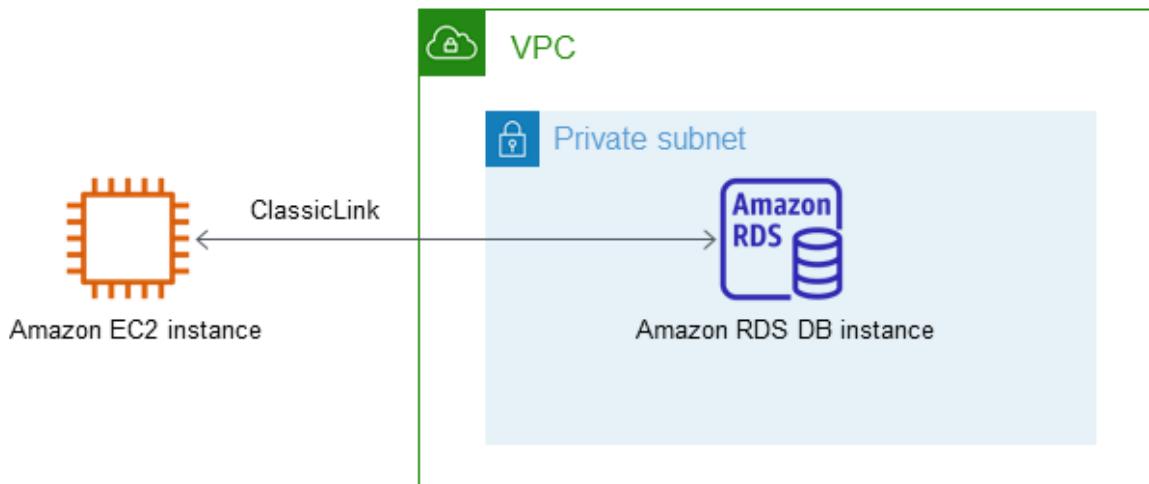
A DB instance in a VPC accessed by an EC2 instance not in a VPC

You can communicate between an Amazon RDS DB instance that is in a VPC and an EC2 instance that is not in an Amazon VPC by using *ClassicLink*. When you use ClassicLink, an application on the EC2 instance can connect to the DB instance by using the endpoint for the DB instance. ClassicLink is available at no charge.

Important

If your EC2 instance was created after 2013, it is probably in a VPC.

The following diagram shows this scenario.



Using ClassicLink, you can connect an EC2 instance to a logically isolated database where you define the IP address range and control the access control lists (ACLs) to manage network traffic. You don't have to use public IP addresses or tunneling to communicate with the DB instance in the VPC. This arrangement provides you with higher throughput and lower latency connectivity for inter-instance communications.

To enable ClassicLink between a DB instance in a VPC and an EC2 instance not in a VPC

1. Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc>.
2. In the navigation pane, choose **Your VPCs**.
3. Choose the VPC used by the DB instance.
4. In **Actions**, choose **Enable ClassicLink**. In the confirmation dialog box, choose **Yes, Enable**.

5. On the EC2 console, choose the EC2 instance you want to connect to the DB instance in the VPC.
6. In **Actions**, choose **ClassicLink**, and then choose **Link to VPC**.
7. On the **Link to VPC** page, choose the security group you want to use, and then choose **Link to VPC**.

Note

The ClassicLink features are only visible in the consoles for accounts and regions that support EC2-Classic. For more information, see [ClassicLink](#) in the *Amazon EC2 User Guide for Linux Instances*.

Scenarios for accessing a DB instance not in a VPC

Amazon RDS supports the following scenarios for accessing a DB instance that is not in a VPC:

- [An EC2 instance in a VPC \(p. 1720\)](#)
- [A client application through the internet \(p. 1726\)](#)
- [An EC2 instance not in a VPC \(p. 1726\)](#)

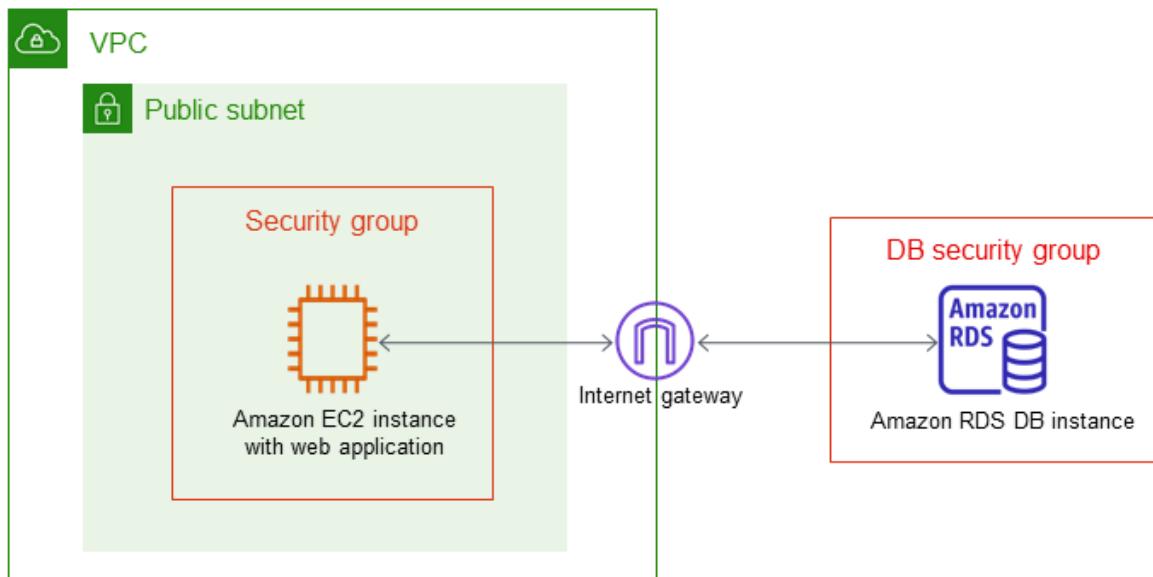
Important

If your DB instance was created after 2013, it is probably in a VPC. For information about accessing a DB instance in a VPC, see [Scenarios for accessing a DB instance in a VPC \(p. 1720\)](#).

A DB instance not in a VPC accessed by an EC2 instance in a VPC

In the case where you have an EC2 instance in a VPC and an RDS DB instance not in a VPC, you can connect them over the public internet.

The following diagram shows this scenario.



Note

ClassicLink, as described in [A DB instance in a VPC accessed by an EC2 instance not in a VPC \(p. 1724\)](#), is not available for this scenario.

To connect your DB instance and your EC2 instance over the public internet, do the following:

- Ensure that the EC2 instance is in a public subnet in the VPC.

- Ensure that the RDS DB instance was marked as publicly accessible.
- A note about network ACLs here. A network ACL is like a firewall for your entire subnet. Therefore, all instances in that subnet are subject to network ACL rules. By default, network ACLs allow all traffic and you generally don't need to worry about them, unless you particularly want to add rules as an extra layer of security. A security group, on the other hand, is associated with individual instances, and you do need to worry about security group rules.
- Add the necessary ingress rules to the DB security group for the RDS DB instance.

An ingress rule specifies a network port and a CIDR/IP range. For example, you can add an ingress rule that allows port 3306 to connect to a MySQL RDS DB instance, and a CIDR/IP range of 203.0.113.25/32. For more information, see [Authorizing network access to a DB security group from an IP range \(p. 1708\)](#).

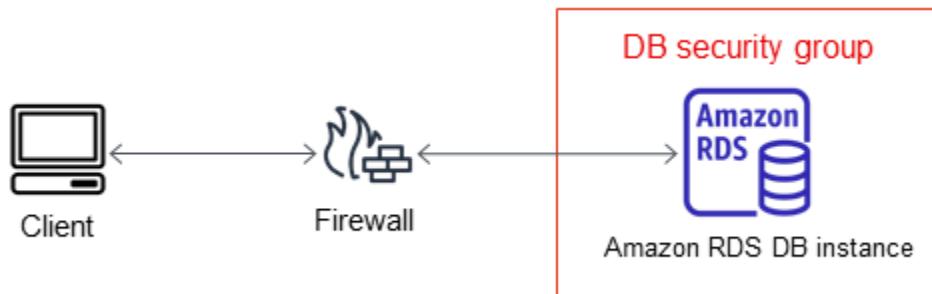
Note

If you are interested in moving an existing DB instance into a VPC, you can use the AWS Management Console to do it easily. For more information, see [Moving a DB instance not in a VPC into a VPC \(p. 1735\)](#).

A DB instance not in a VPC accessed by a client application through the internet

New Amazon RDS customers can only create a DB instance in a VPC. However, you might need to connect to an existing Amazon RDS DB instance that is not in a VPC from a client application through the internet.

The following diagram shows this scenario.



In this scenario, you must ensure that the DB security group for the RDS DB instance includes the necessary ingress rules for your client application to connect. An ingress rule specifies a network port and a CIDR/IP range. For example, you can add an ingress rule that allows port 3306 to connect to a MySQL RDS DB instance, and a CIDR/IP range of 203.0.113.25/32. For more information, see [Authorizing network access to a DB security group from an IP range \(p. 1708\)](#).

Warning

If you intend to access a DB instance behind a firewall, talk with your network administrator to determine the IP addresses you should use.

Note

If you are interested in moving an existing DB instance into a VPC, you can use the AWS Management Console to do it easily. For more information, see [Moving a DB instance not in a VPC into a VPC \(p. 1735\)](#).

A DB instance not in a VPC accessed by an EC2 instance not in a VPC

When neither your DB instance nor an application on an EC2 instance are in a VPC, you can access the DB instance by using its endpoint and port.

The following diagram shows this scenario.



You must create a security group for the DB instance that permits access from the port you specified when creating the DB instance. For example, you could use a connection string similar to this connection string used with *sqlplus* to access an Oracle DB instance:

```
PROMPT>sqlplus 'mydbusr@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<endpoint>)(PORT=<port number>))(CONNECT_DATA=(SID=<database name>)))'
```

For more information, see the following documentation.

Database engine	Relevant documentation
MariaDB	Connecting to a DB instance running the MariaDB database engine (p. 588)
Microsoft SQL Server	Connecting to a DB instance running the Microsoft SQL Server database engine (p. 656)
MySQL	Connecting to a DB instance running the MySQL database engine (p. 840)
Oracle	Connecting to your Oracle DB instance (p. 1001)
PostgreSQL	Connecting to a DB instance running the PostgreSQL database engine (p. 1508)

Note

If you are interested in moving an existing DB instance into a VPC, you can use the AWS Management Console to do it easily. For more information, see [Moving a DB instance not in a VPC into a VPC \(p. 1735\)](#).

Working with a DB instance in a VPC

Unless you are working with a legacy DB instance, your DB instance is in a virtual private cloud (VPC). A VPC is a virtual network that is logically isolated from other virtual networks in the AWS Cloud. Amazon VPC lets you launch AWS resources, such as an Amazon RDS DB instance or Amazon EC2 instance, into a VPC. The VPC can either be a default VPC that comes with your account or one that you create. All VPCs are associated with your AWS account.

Your default VPC has three subnets you can use to isolate resources inside the VPC. The default VPC also has an internet gateway that can be used to provide access to resources inside the VPC from outside the VPC.

For a list of scenarios involving Amazon RDS DB instances in a VPC and outside of a VPC, see [Scenarios for accessing a DB instance in a VPC \(p. 1720\)](#).

For a tutorial that shows you how to create a VPC that you can use with a common Amazon RDS scenario, see [Tutorial: Create an Amazon VPC for use with a DB instance \(p. 1737\)](#).

To learn how to work with DB instances inside a VPC, see the following:

Topics

- [Working with a DB instance in a VPC \(p. 1728\)](#)
- [Working with DB subnet groups \(p. 1729\)](#)
- [Hiding a DB instance in a VPC from the internet \(p. 1729\)](#)
- [Creating a DB instance in a VPC \(p. 1730\)](#)

Working with a DB instance in a VPC

Here are some tips on working with a DB instance in a VPC:

- Your VPC must have at least two subnets. These subnets must be in two different Availability Zones in the AWS Region where you want to deploy your DB instance. A subnet is a segment of a VPC's IP address range that you can specify and that lets you group instances based on your security and operational needs.

Note

The DB subnet group for a Local Zone can have only one subnet.

- If you want your DB instance in the VPC to be publicly accessible, you must enable the VPC attributes *DNS hostnames* and *DNS resolution*.
- Your VPC must have a DB subnet group that you create (for more information, see the next section). You create a DB subnet group by specifying the subnets you created. Amazon RDS chooses a subnet and an IP address within that subnet to associate with your DB instance. The DB instance uses the Availability Zone that contains the subnet.
- Your VPC must have a VPC security group that allows access to the DB instance.
- The CIDR blocks in each of your subnets must be large enough to accommodate spare IP addresses for Amazon RDS to use during maintenance activities, including failover and compute scaling.
- A VPC can have an *instance tenancy* attribute of either *default* or *dedicated*. All default VPCs have the instance tenancy attribute set to default, and a default VPC can support any DB instance class.

If you choose to have your DB instance in a dedicated VPC where the instance tenancy attribute is set to dedicated, the DB instance class of your DB instance must be one of the approved Amazon EC2 dedicated instance types. For example, the m3.medium EC2 dedicated instance corresponds to the db.m3.medium DB instance class. For information about instance tenancy in a VPC, see [Dedicated instances in the Amazon Elastic Compute Cloud User Guide](#).

For more information about the instance types that can be in a dedicated instance, see [Amazon EC2 dedicated instances](#) on the EC2 pricing page.

Note

When you set the instance tenancy attribute to dedicated for an Amazon RDS DB instance, it doesn't guarantee that the DB instance will run on a dedicated host.

- When an option group is assigned to a DB instance, it is linked to the supported platform the DB instance is on, either VPC or EC2-Classic (non-VPC). Furthermore, if a DB instance is in a VPC, the option group associated with the DB instance is linked to that VPC. This linkage means that you cannot use the option group assigned to a DB instance if you attempt to restore the DB instance into a different VPC or onto a different platform.
- If you restore a DB instance into a different VPC or onto a different platform, you must either assign the default option group to the DB instance, assign an option group that is linked to that VPC or platform, or create a new option group and assign it to the DB instance. With persistent or permanent options, such as Oracle TDE, you must create a new option group that includes the persistent or permanent option when restoring a DB instance into a different VPC.

Working with DB subnet groups

Subnets are segments of a VPC's IP address range that you designate to group your resources based on security and operational needs. A DB subnet group is a collection of subnets (typically private) that you create in a VPC and that you then designate for your DB instances. A DB subnet group allows you to specify a particular VPC when creating DB instances using the CLI or API; if you use the console, you can just choose the VPC and subnets you want to use.

Each DB subnet group should have subnets in at least two Availability Zones in a given AWS Region. When creating a DB instance in a VPC, you must choose a DB subnet group. From the DB subnet group, Amazon RDS chooses a subnet and an IP address within that subnet to associate with your DB instance. The DB instance uses the Availability Zone that contains the subnet. If the primary DB instance of a Multi-AZ deployment fails, Amazon RDS can promote the corresponding standby and subsequently create a new standby using an IP address of the subnet in one of the other Availability Zones.

The subnets in a DB subnet group are either public or private. They can't be a mix of both public and private subnets. The subnets are public or private, depending on the configuration that you set for their network access control lists (network ACLs) and routing tables.

Note

The DB subnet group for a Local Zone can have only one subnet.

When Amazon RDS creates a DB instance in a VPC, it assigns a network interface to your DB instance by using an IP address from your DB subnet group. However, we strongly recommend that you use the DNS name to connect to your DB instance because the underlying IP address changes during failover.

Note

For each DB instance that you run in a VPC, make sure to reserve at least one address in each subnet in the DB subnet group for use by Amazon RDS for recovery actions.

Hiding a DB instance in a VPC from the internet

One common Amazon RDS scenario is to have a VPC in which you have an EC2 instance with a public-facing web application and a DB instance with a database that is not publicly accessible. For example, you can create a VPC that has a public subnet and a private subnet. Amazon EC2 instances that function as web servers can be deployed in the public subnet, and the DB instances are deployed in the private subnet. In such a deployment, only the web servers have access to the DB instances. For an illustration of this scenario, see [A DB instance in a VPC accessed by an EC2 instance in the same VPC \(p. 1720\)](#).

When you launch a DB instance inside a VPC, the DB instance has a private IP address for traffic inside the VPC. This private IP address isn't publicly accessible. You can use the *Public access* option to designate whether the DB instance also has a public IP address in addition to the private IP address. If the DB instance is designated as publicly accessible, its DNS endpoint resolves to the private IP address from within the DB instance's VPC, and to the public IP address from outside of the DB instance's VPC. Access to the DB instance is ultimately controlled by the security group it uses, and that public access is not permitted if the security group assigned to the DB instance doesn't permit it.

You can modify a DB instance to turn on or off public accessibility by modifying the *Public access* option. For more information, see the modifying section for your DB engine.

The following illustration shows the **Public access** option in the **Additional connectivity configuration** section. To set the option, open the **Additional connectivity configuration** section in the **Connectivity** section.

Connectivity

Subnet group
default ▾

Security group
List of DB security groups to associate with this DB instance.
Choose security groups ▾
default X

Certificate authority
rds-ca-2019 ▾

▼ Additional connectivity configuration

Public access

Publicly accessible
EC2 instances and devices outside the VPC can connect to the instance. You define the security groups for supported devices and instances.

Not publicly accessible
No IP address is assigned to the DB instance. EC2 instances and devices outside the VPC can't connect.

⚠ Setting your DB instance accessibility to private might result in the loss of connectivity to your database. [Learn more](#)

Database port
Specify the TCP/IP port that the DB instance will use for application connections. The application connection string must specify the port number. The DB security group and your firewall must allow connections to the port. [Learn more](#) ▾
3306 ▾

For information about modifying a DB instance to set the **Public access** option, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Creating a DB instance in a VPC

The following procedures help you create a DB instance in a VPC. If your account has a default VPC, you can begin with step 3 because the VPC and DB subnet group have already been created for you. If your AWS account doesn't have a default VPC, or if you want to create an additional VPC, you can create a new VPC.

If you don't know if you have a default VPC, see [Determining whether you are using the EC2-VPC or EC2-Classic platform \(p. 1718\)](#).

Note

If you want your DB instance in the VPC to be publicly accessible, you must update the DNS information for the VPC by enabling the VPC attributes *DNS hostnames* and *DNS resolution*. For

information about updating the DNS information for a VPC instance, see [Updating DNS support for your VPC](#).

Follow these steps to create a DB instance in a VPC:

- [Step 1: Create a VPC \(p. 1731\)](#)
- [Step 2: Add subnets to the VPC \(p. 1731\)](#)
- [Step 3: Create a DB subnet group \(p. 1731\)](#)
- [Step 4: Create a VPC security group \(p. 1734\)](#)
- [Step 5: Create a DB instance in the VPC \(p. 1734\)](#)

Step 1: Create a VPC

If your AWS account does not have a default VPC or if you want to create an additional VPC, follow the instructions for creating a new VPC. See [Create a VPC with private and public subnets \(p. 1737\)](#), or see [Step 1: Create a VPC](#) in the Amazon VPC documentation.

Step 2: Add subnets to the VPC

Once you have created a VPC, you need to create subnets in at least two Availability Zones. You use these subnets when you create a DB subnet group. If you have a default VPC, a subnet is automatically created for you in each Availability Zone in the AWS Region.

For instructions on how to create subnets in a VPC, see [Create a VPC with private and public subnets \(p. 1737\)](#).

Step 3: Create a DB subnet group

A DB subnet group is a collection of subnets (typically private) that you create for a VPC and that you then designate for your DB instances. A DB subnet group allows you to specify a particular VPC when you create DB instances using the CLI or API. If you use the console, you can just choose the VPC and subnets you want to use. Each DB subnet group must have at least one subnet in at least two Availability Zones in the AWS Region.

As a best practice, each DB subnet group should have at least one subnet for every Availability Zone in the AWS Region. For Multi-AZ deployments, defining a subnet for all Availability Zones in an AWS Region enables Amazon RDS to create a new standby replica in another Availability Zone if necessary. You can follow this best practice even for Single-AZ deployments, because you might convert them to Multi-AZ deployments in the future.

For a DB instance to be publicly accessible, the subnets in the DB subnet group must have an internet gateway. For more information about internet gateways for subnets, see [Internet gateways](#) in the Amazon VPC documentation.

Note

The DB subnet group for a Local Zone can have only one subnet.

When you create a DB instance in a VPC, make sure to choose a DB subnet group. Amazon RDS chooses a subnet and an IP address within that subnet to associate with your DB instance. Amazon RDS creates and associates an Elastic Network Interface to your DB instance with that IP address. The DB instance uses the Availability Zone that contains the subnet. For Multi-AZ deployments, defining a subnet for two or more Availability Zones in an AWS Region allows Amazon RDS to create a new standby in another Availability Zone should the need arise. You need to do this even for Single-AZ deployments, just in case you want to convert them to Multi-AZ deployments at some point.

In this step, you create a DB subnet group and add the subnets that you created for your VPC.

To create a DB subnet group

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Subnet groups**.
3. Choose **Create DB Subnet Group**.
4. For **Name**, type the name of your DB subnet group.
5. For **Description**, type a description for your DB subnet group.
6. For **VPC**, choose the VPC that you created.
7. In the **Add subnets** section, choose the Availability Zones that include the subnets from **Availability Zones**, and then choose the subnets from **Subnets**.

Create DB Subnet Group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

Name

You won't be able to modify the name after your subnet group has been created.

mydbsubnetgroup

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

My DB Subnet Group

VPC

Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

tutorial-vpc (vpc-068fe388385afc014)

Add subnets

Availability Zones

Choose the Availability Zones that include the subnets you want to add.

Choose an availability zone

us-east-1a X us-east-1c X

Subnets

Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Select subnets

subnet-079bd4b8953aee1dd (10.0.0.0/24) X

subnet-057e85b72c46fdd9a (10.0.1.0/24) X

Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1a	subnet-079bd4b8953aee1dd	10.0.0.0/24
us-east-1c	subnet-057e85b72c46fdd9a	10.0.1.0/24

Cancel

Create

Note

If you have enabled a Local Zone, you can choose an Availability Zone group on the [Create DB subnet group](#) page. In this case, choose the **Availability Zone group, Availability Zones, and Subnets**.

8. Choose **Create**.

Your new DB subnet group appears in the DB subnet groups list on the RDS console. You can choose the DB subnet group to see details, including all of the subnets associated with the group, in the details pane at the bottom of the window.

Step 4: Create a VPC security group

Before you create your DB instance, you must create a VPC security group to associate with your DB instance. For instructions on how to create a security group for your DB instance, see [Create a VPC security group for a private DB instance \(p. 1740\)](#), or see [Security groups for your VPC](#) in the Amazon VPC documentation.

Step 5: Create a DB instance in the VPC

In this step, you create a DB instance and use the VPC name, the DB subnet group, and the VPC security group you created in the previous steps.

Note

If you want your DB instance in the VPC to be publicly accessible, you must enable the VPC attributes *DNS hostnames* and *DNS resolution*. For information on updating the DNS information for a VPC instance, see [Updating DNS support for your VPC](#).

For details on how to create a DB instance, see [Creating an Amazon RDS DB instance \(p. 141\)](#).

When prompted in the **Network & Security** section, enter the VPC name, the DB subnet group, and the VPC security group you created in the previous steps.

Updating the VPC for a DB instance

You can use the AWS Management Console to move your DB instance to a different VPC.

For information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#). In the **Network & Security** section of the modify page, shown following, enter the new subnet group for **Subnet group**. The new subnet group must be a subnet group in a new VPC.

The screenshot shows the 'Network & Security' section of the AWS Management Console. It includes fields for 'Subnet group' (containing 'mydbsubnetgroup') and 'Security group' (containing 'mydbsecuritygroup').

Network & Security	
Subnet group	Use this field to move the DB instance to a new subnet group in another VPC. Learn more .
mydbsubnetgroup	
Security group	List of DB security groups to associate with this DB instance.
mydbsecuritygroup	

Moving a DB instance not in a VPC into a VPC

Some legacy DB instances on the EC2-Classic platform are not in a VPC. If your DB instance is not in a VPC, you can use the AWS Management Console to easily move your DB instance into a VPC. Before you can move a DB instance not in a VPC, into a VPC, you must create the VPC.

Follow these steps to create a VPC for your DB instance.

- [Step 1: Create a VPC \(p. 1731\)](#)
- [Step 2: Add subnets to the VPC \(p. 1731\)](#)
- [Step 3: Create a DB subnet group \(p. 1731\)](#)
- [Step 4: Create a VPC security group \(p. 1734\)](#)

Each DB subnet group must include at least the Availability Zones in which the DB instance is located.

After you create the VPC, follow these steps to move your DB instance into the VPC.

- [Updating the VPC for a DB instance \(p. 1734\)](#)

We highly recommend that you create a backup of your DB instance immediately before the migration. Doing so ensures that you can restore the data if the migration fails. For more information, see [Backing up and restoring an Amazon RDS DB instance \(p. 327\)](#).

The following are some limitations to moving your DB instance into the VPC.

- **Previous generation DB instance classes** – Previous generation DB instance classes might not be supported on the VPC platform. When moving a DB instance to a VPC, choose a db.m3 or db.r3 DB instance class. After you move the DB instance to a VPC, you can scale the DB instance to use a later DB instance class. For a full list of VPC supported instance classes, see [Amazon RDS instance types](#).
- **Multi-AZ** – Moving a Multi-AZ DB instance not in a VPC into a VPC is not currently supported. To move your DB instance to a VPC, first modify the DB instance so that it is a single-AZ deployment. Change the **Multi-AZ deployment** setting to **No**. After you move the DB instance to a VPC, modify it again to make it a Multi-AZ deployment. For more information, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).
- **Read replicas** – Moving a DB instance with read replicas not in a VPC into a VPC is not currently supported. To move your DB instance to a VPC, first delete all of its read replicas. After you move the DB instance to a VPC, recreate the read replicas. For more information, see [Working with read replicas \(p. 278\)](#).
- **Option groups** – If you move your DB instance to a VPC, and the DB instance is using a custom option group, change the option group that is associated with your DB instance. Option groups are platform-specific, and moving to a VPC is a change in platform. To use a custom option group in this case, assign the default VPC option group to the DB instance, assign an option group that is used by other DB instances in the VPC you are moving to, or create a new option group and assign it to the DB instance. For more information, see [Working with option groups \(p. 212\)](#).

Alternatives for moving a DB instance not in a VPC into a VPC with minimal downtime

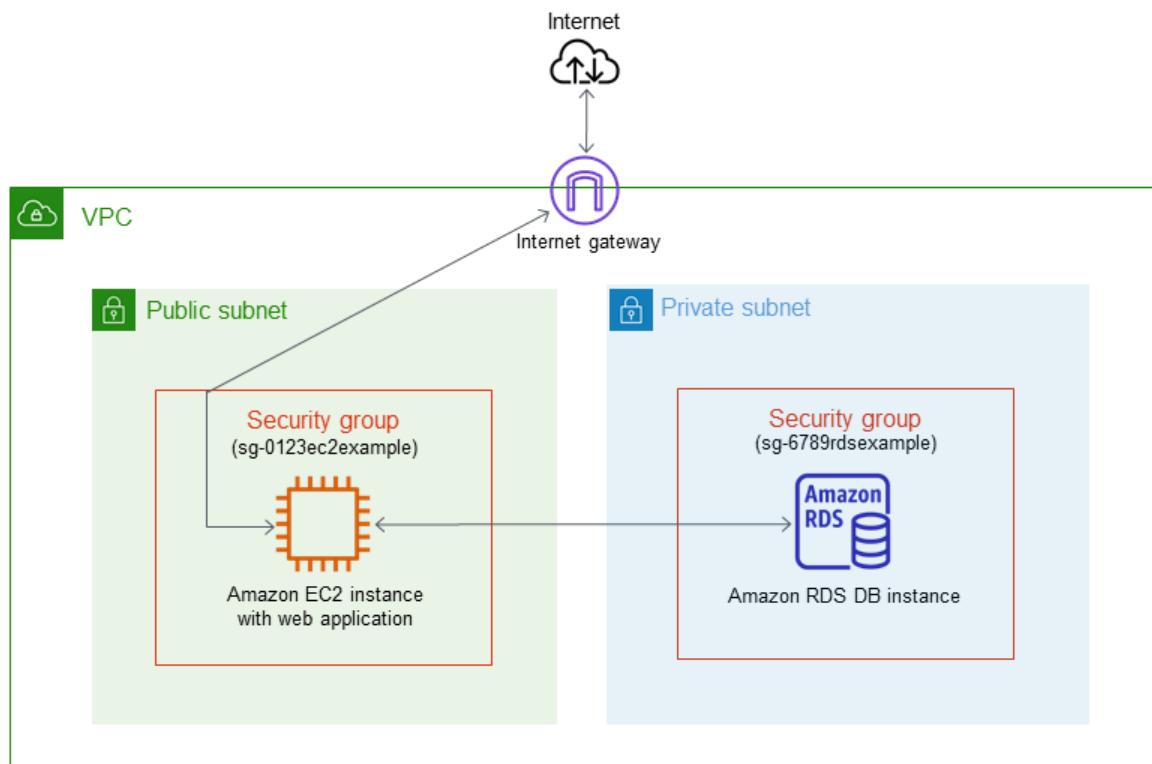
Using the following alternatives, you can move a DB instance not in a VPC into a VPC with minimal downtime. These alternatives cause minimum disruption to the source DB instance and allow it to serve user traffic during the migration. However, the time required to migrate to a VPC will vary based on the database size and the live workload characteristics.

- **AWS Database Migration Service (AWS DMS)** – AWS DMS enables the live migration of data while keeping the source DB instance fully operational, but it replicates only a limited set of DDL statements. AWS DMS doesn't propagate items such as indexes, users, privileges, stored procedures, and other database changes not directly related to table data. In addition, AWS DMS doesn't automatically use RDS snapshots for the initial DB instance creation, which can increase migration time. For more information, see [AWS Database Migration Service](#).
- **DB snapshot restore or point-in-time recovery** – You can move a DB instance to a VPC by restoring a snapshot of the DB instance or by restoring a DB instance to a point in time. For more information, see [Restoring from a DB snapshot \(p. 349\)](#) and [Restoring a DB instance to a specified time \(p. 389\)](#).

Tutorial: Create an Amazon VPC for use with a DB instance

A common scenario includes a DB instance in an Amazon VPC, that shares data with a web server that is running in the same VPC. In this tutorial you create the VPC for this scenario.

The following diagram shows this scenario. For information about other scenarios, see [Scenarios for accessing a DB instance in a VPC \(p. 1720\)](#).



Because your DB instance only needs to be available to your web server, and not to the public Internet, you create a VPC with both public and private subnets. The web server is hosted in the public subnet, so that it can reach the public Internet. The DB instance is hosted in a private subnet. The web server is able to connect to the DB instance because it is hosted within the same VPC, but the DB instance is not available to the public Internet, providing greater security.

This tutorial describes configuring a VPC for Amazon RDS DB instances. For more information about Amazon VPC, see [Amazon VPC Getting Started Guide](#) and [Amazon VPC User Guide](#).

Note

For a tutorial that shows you how to create a web server for this VPC scenario, see [Tutorial: Create a web server and an Amazon RDS DB instance \(p. 108\)](#).

Create a VPC with private and public subnets

Use the following procedure to create a VPC with both public and private subnets.

To create a VPC and subnets

1. If you don't have an Elastic IP address to associate with a network address translation (NAT) gateway, allocate one now. A NAT gateway is required for this tutorial. If you have an available Elastic IP address, move on to the next step.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. In the top-right corner of the AWS Management Console, choose the Region to allocate your Elastic IP address in. The Region of your Elastic IP address should be the same as the Region where you want to create your VPC. This example uses the US West (Oregon) Region.
 - c. In the navigation pane, choose **Elastic IPs**.
 - d. Choose **Allocate Elastic IP address**.
 - e. If the console shows the **Network Border Group** field, keep the default value for it.
 - f. For **Public IPv4 address pool**, choose **Amazon's pool of IPv4 addresses**.
 - g. Choose **Allocate**.

Note the allocation ID of the new Elastic IP address because you'll need this information when you create your VPC.

For more information about Elastic IP addresses, see [Elastic IP addresses](#) in the *Amazon EC2 User Guide*. For more information about NAT gateways, see [NAT gateways](#) in the *Amazon VPC User Guide*.

2. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
3. In the top-right corner of the AWS Management Console, choose the Region to create your VPC in. This example uses the US West (Oregon) Region.
4. In the upper-left corner, choose **VPC Dashboard**. To begin creating a VPC, choose **Launch VPC Wizard**.
5. On the **Step 1: Select a VPC Configuration** page, choose **VPC with Public and Private Subnets**, and then choose **Select**.
6. On the **Step 2: VPC with Public and Private Subnets** page, set these values:
 - **IPv4 CIDR block:** 10.0.0.0/16
 - **IPv6 CIDR block:** No IPv6 CIDR Block
 - **VPC name:** tutorial-vpc
 - **Public subnet's IPv4 CIDR:** 10.0.0.0/24
 - **Availability Zone:** us-west-2a
 - **Public subnet name:** Tutorial public
 - **Private subnet's IPv4 CIDR:** 10.0.1.0/24
 - **Availability Zone:** us-west-2a
 - **Private subnet name:** Tutorial private 1
 - **Elastic IP Allocation ID:** An Elastic IP address to associate with the NAT gateway
 - **Service endpoints:** Skip this field.
 - **Enable DNS hostnames:** Yes
 - **Hardware tenancy:** Default
7. Choose **Create VPC**.

Create additional subnets

You must have either two private subnets or two public subnets available to create a DB subnet group for a DB instance to use in a VPC. Because the DB instance for this tutorial is private, add a second private subnet to the VPC.

To create an additional subnet

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
 2. To add the second private subnet to your VPC, choose **VPC Dashboard**, choose **Subnets**, and then choose **Create subnet**.
 3. On the **Create subnet** page, set these values:
 - **VPC ID:** Choose the VPC that you created in the previous step, for example: vpc-*identifier* (tutorial-vpc)
 - **Subnet name:** Tutorial private 2
 - **Availability Zone:** us-west-2b
- Note**
Choose an Availability Zone that is different from the one that you chose for the first private subnet.
4. Choose **Create subnet**. Next, choose **Close** on the confirmation page.
 5. To ensure that the second private subnet that you created uses the same route table as the first private subnet, complete the following steps:
 - a. Choose **VPC Dashboard**, choose **Subnets**, and then choose the first private subnet that you created for the VPC, Tutorial private 1.
 - b. Below the list of subnets, choose the **Route table** tab, and note the value for **Route Table**—for example: rtb-98b613fd.
 - c. In the list of subnets, deselect the first private subnet.
 - d. In the list of subnets, choose the second private subnet Tutorial private 2, and choose the **Route table** tab.
 - e. If the current route table is not the same as the route table for the first private subnet, choose **Edit route table association**. For **Route table ID**, choose the route table that you noted earlier—for example: rtb-98b613fd. Next, to save your selection, choose **Save**.

Create a VPC security group for a public web server

Next you create a security group for public access. To connect to public instances in your VPC, you add inbound rules to your VPC security group that allow traffic to connect from the internet.

To create a VPC security group

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. Choose **VPC Dashboard**, choose **Security Groups**, and then choose **Create security group**.
3. On the **Create security group** page, set these values:
 - **Security group name:** tutorial-securitygroup
 - **Description:** Tutorial Security Group
 - **VPC:** Choose the VPC that you created earlier, for example: vpc-*identifier* (tutorial-vpc)
4. Add inbound rules to the security group.
 - a. Determine the IP address to use to connect to instances in your VPC. To determine your public IP address, in a different browser window or tab, you can use the service at <https://checkip.amazonaws.com>. An example of an IP address is 203.0.113.25/32.

If you are connecting through an Internet service provider (ISP) or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

Warning

If you use 0.0.0.0/0, you enable all IP addresses to access your public instances.

This approach is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you'll authorize only a specific IP address or range of addresses to access your instances.

- b. In the **Inbound rules** section, choose **Add rule**.
- c. Set the following values for your new inbound rule to allow Secure Shell (SSH) access to your EC2 instance. If you do this, you can connect to your EC2 instance to install the web server and other utilities, and to upload content for your web server.
 - **Type:** SSH
 - **Source:** The IP address or range from Step a, for example: 203.0.113.25/32.
- d. Choose **Add rule**.
- e. Set the following values for your new inbound rule to allow HTTP access to your web server.
 - **Type:** HTTP
 - **Source:** 0.0.0.0/0.

5. To create the security group, choose **Create security group**.

Note the security group ID because you need it later in this tutorial.

Create a VPC security group for a private DB instance

To keep your DB instance private, create a second security group for private access. To connect to private instances in your VPC, you add inbound rules to your VPC security group that allow traffic from your web server only.

To create a VPC security group

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. Choose **VPC Dashboard**, choose **Security Groups**, and then choose **Create security group**.
3. On the **Create security group** page, set these values:
 - **Security group name:** tutorial-db-securitygroup
 - **Description:** Tutorial DB Instance Security Group
 - **VPC:** Choose the VPC that you created earlier, for example: vpc-*identifier* (tutorial-vpc)
4. Add inbound rules to the security group.
 - a. In the **Inbound rules** section, choose **Add rule**.
 - b. Set the following values for your new inbound rule to allow MySQL traffic on port 3306 from your EC2 instance. If you do this, you can connect from your web server to your DB instance to store and retrieve data from your web application to your database.
 - **Type:** MySQL/Aurora
 - **Source:** The identifier of the tutorial-securitygroup security group that you created previously in this tutorial, for example: sg-9edd5cfb.
5. To create the security group, choose **Create security group**.

Create a DB subnet group

A DB subnet group is a collection of subnets that you create in a VPC and that you then designate for your DB instances. A DB subnet group allows you to specify a particular VPC when creating DB instances.

To create a DB subnet group

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.

Note

Make sure you connect to the Amazon RDS console, not to the Amazon VPC console.

2. In the navigation pane, choose **Subnet groups**.
3. Choose **Create DB Subnet Group**.
4. On the **Create DB subnet group** page, set these values in **Subnet group details**:

- **Name:** tutorial-db-subnet-group
- **Description:** Tutorial DB Subnet Group
- **VPC:** tutorial-vpc (`vpc-identifier`)

5. In the **Add subnets** section, choose the **Availability Zones** and **Subnets**.

For this tutorial, choose `us-west-2a` and `us-west-2b` for the **Availability Zones**. Next, for **Subnets**, choose the subnets for IPv4 CIDR block `10.0.0.0/24`, `10.0.1.0/24`, and `10.0.2.0/24`.

Note

If you have enabled a Local Zone, you can choose an Availability Zone group on the **Create DB subnet group** page. In this case, choose the **Availability Zone group**, **Availability Zones**, and **Subnets**.

6. Choose **Create**.

Your new DB subnet group appears in the DB subnet groups list on the RDS console. You can click the DB subnet group to see details, including all of the subnets associated with the group, in the details pane at the bottom of the window.

Note

If you created this VPC to complete [Tutorial: Create a web server and an Amazon RDS DB instance \(p. 108\)](#), create the DB instance by following the instructions in [Create a DB instance \(p. 109\)](#).

Quotas and constraints for Amazon RDS

Following, you can find a description of the resource quotas and naming constraints for Amazon RDS.

Topics

- [Quotas in Amazon RDS \(p. 1742\)](#)
- [Naming constraints in Amazon RDS \(p. 1743\)](#)
- [Maximum number of database connections \(p. 1744\)](#)
- [File size limits in Amazon RDS \(p. 1745\)](#)

Quotas in Amazon RDS

Each AWS account has quotas, for each AWS Region, on the number of Amazon RDS resources that can be created. After a quota for a resource has been reached, additional calls to create that resource fail with an exception.

The following table lists the resources and their quotas per AWS Region.

Resource	Default quota
Authorizations per DB security group	20
Burst balance for instances <1 tebibyte (TiB)	3000 IOPS
Concurrent DB snapshot export tasks	5
Cross-Region DB snapshot copy requests	5
DB instances	40
DB security groups	25
DB subnet groups	50
Event subscriptions	20
IAM roles per DB instance	5
Manual DB snapshots	100
Option groups	20
Parameter groups	50
Proxies	20
Read replicas per primary	5
Reserved DB instances	40

Resource	Default quota
Rules per security group	20
Rules per virtual private cloud (VPC) security group	50 inbound, 50 outbound
Subnets per subnet group	20
Tags per resource	50
Total storage for all DB instances	100 terabytes (TB)
VPC security groups	5

Note

By default, you can have up to a total of 40 DB instances. RDS DB instances, Aurora DB instances, Amazon Neptune instances, and Amazon DocumentDB instances apply to this quota. The following limitations apply to the Amazon RDS DB instances:

- 10 for each SQL Server edition (Enterprise, Standard, Web, and Express) under the "license-included" model
- 10 for Oracle under the "license-included" model
- 40 for MySQL, MariaDB, or PostgreSQL
- 40 for Oracle under the "bring-your-own-license" (BYOL) licensing model

If your application requires more DB instances, you can request additional DB instances by opening the [Service Quotas console](#). In the navigation pane, choose **AWS services**. Choose **Amazon Relational Database Service (Amazon RDS)**, choose a quota, and follow the directions to request a quota increase. For more information, see [Requesting a quota increase](#) in the [Service Quotas User Guide](#).

Backups managed by AWS Backup are considered manual DB snapshots, but don't count toward the manual snapshot quota. For information about AWS Backup, see the [AWS Backup Developer Guide](#).

Naming constraints in Amazon RDS

The following table describes naming constraints in Amazon RDS.

Resource or item	Constraints
DB instance identifier	<p>Identifiers have these naming constraints:</p> <ul style="list-style-type: none"> • Must contain 1–63 alphanumeric characters or hyphens. • First character must be a letter. • Can't end with a hyphen or contain two consecutive hyphens. • Must be unique for all DB instances per AWS account, per AWS Region.
Database name	<p>Database name constraints differ for each database engine . For more information, see the available settings when creating each DB instance.</p>

Resource or item	Constraints
	Note This approach doesn't apply to SQL Server. For SQL Server, you create your databases after you create your DB instance.
Master user name	Master user name constraints differ for each database engine. For more information, see the available settings when creating each DB instance.
Master password	The password for the database master user can include any printable ASCII character except /, ", @, or a space. Master password length constraints differ for each database engine. For more information, see the available settings when creating each DB instance.
DB parameter group name	These names have these constraints: <ul style="list-style-type: none"> • Must contain 1–255 alphanumeric characters. • First character must be a letter. • Hyphens are allowed, but the name cannot end with a hyphen or contain two consecutive hyphens.
DB subnet group name	These names have these constraints: <ul style="list-style-type: none"> • Must contain 1–255 characters. • Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Maximum number of database connections

The maximum number of simultaneous database connections varies by the DB engine type and the memory allocation for the DB instance class. The maximum number of connections is set in the parameter group associated with the DB instance, except for Microsoft SQL Server, where it is set in the server properties for the DB instance in SQL Server Management Studio (SSMS).

`DBInstanceClassMemory` is in bytes. You can find the value of `DBInstanceClassMemory` in gibibytes (GiB) in the table of [Hardware specifications for DB instance classes \(p. 33\)](#).

Note

For Oracle, you set the maximum number of user processes and user and system sessions.

Maximum database connections

DB engine	Parameter	Allowed values	Default value	Description
MariaDB and MySQL	<code>max_connections</code>	1–100000	$\{DBInstanceClassMemory/125\}$ or $\{DBInstanceClassMemory/200\}$	Number of simultaneous client connections allowed
Oracle	<code>processes</code>	80–20000	$\text{LEAST}(\{DBInstanceClassMemory/200\}, 20000)$	User processes
	<code>sessions</code>	100–65535	–	User and system sessions

DB engine	Parameter	Allowed values	Default value	Description
PostgreSQL	max_connections	6–8388607	LEAST({DBInstanceClassMemory / 5000), 1365}	Maximum number of concurrent connections
SQL Server	Maximum number of concurrent connections	0–32767	0 (unlimited)	Maximum number of concurrent connections

The following example shows how to calculate `max_connections` for a MariaDB or MySQL DB instance using the db.m5.xlarge instance class. `DBInstanceClassMemory` is 16 GiB, or 17,179,869,184 bytes. That divided by 12,582,880 = 1365 connections maximum.

For MariaDB and MySQL DB instances, setting the `max_connections` parameter to a large value might cause a DB instance to be placed in the **incompatible-parameters** status. For more information, see [Diagnosing and resolving incompatible parameters status for a memory limit \(p. 1752\)](#).

Note

You might see fewer than the maximum number of DB connections. This is to avoid potential out-of-memory issues.

File size limits in Amazon RDS

File size limits apply to certain Amazon RDS DB instances. For more information, see the following engine-specific limits:

- [MariaDB file size limits in Amazon RDS \(p. 581\)](#)
- [MySQL file size limits in Amazon RDS \(p. 950\)](#)
- [Oracle file size limits in Amazon RDS \(p. 999\)](#)

Troubleshooting for Amazon RDS

Use the following sections to help troubleshoot problems you have with DB instances in Amazon RDS and Aurora.

Topics

- [Can't connect to Amazon RDS DB instance \(p. 1746\)](#)
- [Amazon RDS security issues \(p. 1748\)](#)
- [Resetting the DB instance owner password \(p. 1748\)](#)
- [Amazon RDS DB instance outage or reboot \(p. 1749\)](#)
- [Amazon RDS DB parameter changes not taking effect \(p. 1749\)](#)
- [Amazon RDS DB instance running out of storage \(p. 1750\)](#)
- [Amazon RDS insufficient DB instance capacity \(p. 1751\)](#)
- [MySQL and MariaDB issues \(p. 1751\)](#)
- [Can't set backup retention period to 0 \(p. 1758\)](#)

For information about debugging problems using the Amazon RDS API, see [Troubleshooting applications on Amazon RDS \(p. 1760\)](#).

Can't connect to Amazon RDS DB instance

When you can't connect to a DB instance, the following are common causes:

- **Inbound rules** – The access rules enforced by your local firewall and the IP addresses authorized to access your DB instance might not match. The problem is most likely the inbound rules in your security group.

By default, DB instances don't allow access. Access is granted through a security group associated with the VPC that allows traffic into and out of the DB instance. If necessary, add inbound and outbound rules for your particular situation to the security group. You can specify an IP address, a range of IP addresses, or another VPC security group.

Note

When adding a new inbound rule, you can choose **My IP** for **Source** to allow access to the DB instance from the IP address detected in your browser.

For more information about setting up security groups, see [Provide access to your DB instance in your VPC by creating a security group \(p. 70\)](#).

Note

Client connections from IP addresses within the range 169.254.0.0/16 aren't permitted. This is the Automatic Private IP Addressing Range (APIPA), which is used for local-link addressing.

- **Public accessibility** – To connect to your DB instance from outside of the VPC, such as by using a client application, the instance must have a public IP address assigned to it.

To make the instance publicly accessible, modify it and choose **Yes** under **Public accessibility**. For more information, see [Hiding a DB instance in a VPC from the internet \(p. 1729\)](#).

- **Port** – The port that you specified when you created the DB instance can't be used to send or receive communications due to your local firewall restrictions. To determine if your network allows the

specified port to be used for inbound and outbound communication, check with your network administrator.

- **Availability** – For a newly created DB instance, the DB instance has a status of *creating* until the DB instance is ready to use. When the state changes to *available*, you can connect to the DB instance. Depending on the size of your DB instance, it can take up to 20 minutes before an instance is available.
- **Internet gateway** – For a DB instance to be publicly accessible, the subnets in its DB subnet group must have an internet gateway.

To configure an internet gateway for a subnet

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the name of the DB instance.
3. In the **Connectivity & security** tab, write down the values of the VPC ID under **VPC** and the subnet ID under **Subnets**.
4. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
5. In the navigation pane, choose **Internet Gateways**. Verify that there is an internet gateway attached to your VPC. Otherwise, choose **Create Internet Gateway** to create an internet gateway. Select the internet gateway, and then choose **Attach to VPC** and follow the directions to attach it to your VPC.
6. In the navigation pane, choose **Subnets**, and then select your subnet.
7. On the **Route Table** tab, verify that there is a route with 0.0.0.0/0 as the destination and the internet gateway for your VPC as the target.
 - a. Choose the ID of the route table (rtb-xxxxxxx) to navigate to the route table.
 - b. On the **Routes** tab, choose **Edit routes**. Choose **Add route**, use 0.0.0.0/0 as the destination and the internet gateway as the target.
 - c. Choose **Save routes**.

For more information, see [Working with a DB instance in a VPC \(p. 1727\)](#).

For engine-specific connection issues, see the following topics:

- [Troubleshooting connections to your SQL Server DB instance \(p. 661\)](#)
- [Troubleshooting connections to your Oracle DB instance \(p. 1006\)](#)
- [Troubleshooting connections to your PostgreSQL instance \(p. 1511\)](#)
- [Maximum MySQL and MariaDB connections \(p. 1752\)](#)

Testing a connection to a DB instance

You can test your connection to a DB instance using common Linux or Microsoft Windows tools.

From a Linux or Unix terminal, you can test the connection by entering the following (replace **DB-instance-endpoint** with the endpoint and **port** with the port of your DB instance).

```
nc -zv DB-instance-endpoint port
```

For example, the following shows a sample command and the return value.

```
nc -zv postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 8299
```

```
Connection to postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 8299 port [tcp/vvr-data] succeeded!
```

Windows users can use Telnet to test the connection to a DB instance. Telnet actions aren't supported other than for testing the connection. If a connection is successful, the action returns no message. If a connection isn't successful, you receive an error message such as the following.

```
C:\>telnet sg-postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 819
Connecting To sg-postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com...Could not open
connection to the host, on port 819: Connect failed
```

If Telnet actions return success, your security group is properly configured.

Note

Amazon RDS doesn't accept internet control message protocol (ICMP) traffic, including ping.

Troubleshooting connection authentication

If you can connect to your DB instance but you get authentication errors, you might want to reset the master user password for the DB instance. You can do this by modifying the RDS instance.

For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Amazon RDS security issues

To avoid security issues, never use your master AWS user name and password for a user account. Best practice is to use your master AWS account to create AWS Identity and Access Management (IAM) users and assign those to DB user accounts. You can also use your master account to create other user accounts, if necessary.

For more information on creating IAM users, see [Create an IAM user \(p. 67\)](#).

Error message "failed to retrieve account attributes, certain console functions may be impaired."

You can get this error for several reasons. It might be because your account is missing permissions, or your account hasn't been properly set up. If your account is new, you might not have waited for the account to be ready. If this is an existing account, you might lack permissions in your access policies to perform certain actions such as creating a DB instance. To fix the issue, your IAM administrator needs to provide the necessary roles to your account. For more information, see [the IAM documentation](#).

Resetting the DB instance owner password

If you get locked out of your DB instance, you can log in as the master user. Then you can reset the credentials for other administrative users or roles. If you can't log in as the master user, the AWS account owner can reset the master user password. For details of which administrative accounts or roles you might need to reset, see [Master user account privileges \(p. 1712\)](#).

You can change the DB instance password by using the Amazon RDS console, the AWS CLI command [modify-db-instance](#), or by using the [ModifyDBInstance](#) API operation. For more information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Amazon RDS DB instance outage or reboot

A DB instance outage can occur when a DB instance is rebooted. It can also occur when the DB instance is put into a state that prevents access to it, and when the database is restarted. A reboot can occur when you either manually reboot your DB instance or change a DB instance setting that requires a reboot before it can take effect.

A DB instance reboot occurs when you change a setting that requires a reboot, or when you manually cause a reboot. A reboot can occur immediately if you change a setting and request that the change take effect immediately or it can occur during the DB instance's maintenance window.

A DB instance reboot occurs immediately when one of the following occurs:

- You change the backup retention period for a DB instance from 0 to a nonzero value or from a nonzero value to 0 and set **Apply Immediately** to **true**.
- You change the DB instance class, and **Apply Immediately** is set to **true**.
- You change the storage type from **Magnetic (Standard)** to **General Purpose (SSD)** or **Provisioned IOPS (SSD)**, or from **Provisioned IOPS (SSD)** or **General Purpose (SSD)** to **Magnetic (Standard)**.

A DB instance reboot occurs during the maintenance window when one of the following occurs:

- You change the backup retention period for a DB instance from 0 to a nonzero value or from a nonzero value to 0, and **Apply Immediately** is set to **false**.
- You change the DB instance class, and **Apply Immediately** is set to **false**.

When you change a static parameter in a DB parameter group, the change doesn't take effect until the DB instance associated with the parameter group is rebooted. The change requires a manual reboot. The DB instance isn't automatically rebooted during the maintenance window.

To see a table that shows DB instance actions and the effect that setting the **Apply Immediately** value has, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

Amazon RDS DB parameter changes not taking effect

In some cases, you might change a parameter in a DB parameter group but don't see the changes take effect. If so, you likely need to reboot the DB instance associated with the DB parameter group. When you change a dynamic parameter, the change takes effect immediately. When you change a static parameter, the change doesn't take effect until you reboot the DB instance associated with the parameter group.

You can reboot a DB instance using the RDS console or explicitly calling the [RebootDBInstance](#) API operation (without failover, if the DB instance is in a Multi-AZ deployment). The requirement to reboot the associated DB instance after a static parameter change helps mitigate the risk of a parameter misconfiguration affecting an API call. An example of this might be calling [ModifyDBInstance](#) to change the DB instance class. For more information, see [Modifying parameters in a DB parameter group \(p. 232\)](#).

Amazon RDS DB instance running out of storage

If your DB instance runs out of storage space, it might no longer be available. We highly recommend that you constantly monitor the `FreeStorageSpace` metric published in CloudWatch to make sure that your DB instance has enough free storage space.

If your database instance runs out of storage, its status changes to `storage-full`. For example, a call to the `DescribeDBInstances` API operation for a DB instance that has used up its storage outputs the following.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance

DBINSTANCE mydbinstance 2009-12-22T23:06:11.915Z db.m5.large mysql8.0 50 sa
storage-full mydbinstance.clla4j4jgyp.us-east-1.rds.amazonaws.com 3306
us-east-1b 3
SECGROUP default active
PARAMGRP default.mysql8.0 in-sync
```

To recover from this scenario, add more storage space to your instance using the `ModifyDBInstance` API operation or the following AWS CLI command.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--allocated-storage 60 \
--apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--allocated-storage 60 ^
--apply-immediately
```

```
DBINSTANCE mydbinstance 2009-12-22T23:06:11.915Z db.m5.large mysql8.0 50 sa
storage-full mydbinstance.clla4j4jgyp.us-east-1.rds.amazonaws.com 3306
us-east-1b 3 60
SECGROUP default active
PARAMGRP default.mysql8.0 in-sync
```

Now, when you describe your DB instance, you see that your DB instance has `modifying` status, which indicates the storage is being scaled.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE mydbinstance 2009-12-22T23:06:11.915Z db.m5.large mysql8.0 50 sa
modifying mydbinstance.clla4j4jgyp.us-east-1.rds.amazonaws.com
3306 us-east-1b 3 60
SECGROUP default active
PARAMGRP default.mysql8.0 in-sync
```

After storage scaling is complete, your DB instance status changes to `available`.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE mydbinstance 2009-12-22T23:06:11.915Z db.m5.large mysql8.0 60 sa
available mydbinstance.clla4j4jgyp.us-east-1.rds.amazonaws.com 3306
us-east-1b 3
SECGROUP default active
PARAMGRP default.mysql8.0 in-sync
```

You can receive notifications when your storage space is exhausted using the `DescribeEvents` operation. For example, in this scenario, if you make a `DescribeEvents` call after these operations you see the following output.

```
aws rds describe-events --source-type db-instance --source-identifier mydbinstance
```

```
2009-12-22T23:44:14.374Z mydbinstance Allocated storage has been exhausted db-instance
2009-12-23T00:14:02.737Z mydbinstance Applying modification to allocated storage db-
instance
2009-12-23T00:31:54.764Z mydbinstance Finished applying modification to allocated storage
```

Amazon RDS insufficient DB instance capacity

The `InsufficientDBInstanceCapacity` error can be returned when you try to create or modify a DB instance, or when you try to restore a DB instance from a DB snapshot. When this error is returned, the following are common causes:

- The specific DB instance class isn't available in the requested Availability Zone. You can try one of the following to solve the problem:
 - Retry the request with a different DB instance class.
 - Retry the request with a different Availability Zone.
 - Retry the request without specifying an explicit Availability Zone.

For information about troubleshooting instance capacity issues for Amazon EC2, see [Insufficient instance capacity](#) in the *Amazon Elastic Compute Cloud User Guide*.

- The DB instance is on the EC2-Classic platform and therefore isn't in a VPC. Some DB instance classes require a VPC. For example, if you're on the EC2-Classic platform and try to increase capacity by switching to a DB instance class that requires a VPC, this error results. For information about Amazon EC2 instance types that are only available in a VPC, see [Instance types available in EC2-Classic](#) in the *Amazon Elastic Compute Cloud User Guide*. To correct the problem, you can move the DB instance into a VPC. For more information, see [Moving a DB instance not in a VPC into a VPC \(p. 1735\)](#).

For information about modifying a DB instance, see [Modifying an Amazon RDS DB instance \(p. 250\)](#).

MySQL and MariaDB issues

You can diagnose and correct issues with MySQL and MariaDB DB instances.

Topics

- [Maximum MySQL and MariaDB connections \(p. 1752\)](#)
- [Diagnosing and resolving incompatible parameters status for a memory limit \(p. 1752\)](#)
- [Diagnosing and resolving lag between read replicas \(p. 1753\)](#)
- [Diagnosing and resolving a MySQL or MariaDB read replication failure \(p. 1754\)](#)
- [Creating triggers with binary logging enabled requires SUPER privilege \(p. 1755\)](#)

- [Diagnosing and resolving point-in-time restore failures \(p. 1757\)](#)
- [Replication stopped error \(p. 1757\)](#)
- [Read replica create fails or replication breaks with fatal error 1236 \(p. 1758\)](#)

Maximum MySQL and MariaDB connections

The maximum number of connections allowed to an RDS for MySQL or RDS for MariaDB DB instance is based on the amount of memory available for its DB instance class. A DB instance class with more memory available results in a larger number of connections available. For more information on DB instance classes, see [DB instance classes \(p. 7\)](#).

The connection limit for a DB instance is set by default to the maximum for the DB instance class. You can limit the number of concurrent connections to any value up to the maximum number of connections allowed. Use the `max_connections` parameter in the parameter group for the DB instance. For more information, see [Maximum number of database connections \(p. 1744\)](#) and [Working with DB parameter groups \(p. 228\)](#).

You can retrieve the maximum number of connections allowed for a MySQL or MariaDB DB instance by running the following query.

```
SELECT @@max_connections;
```

You can retrieve the number of active connections to a MySQL or MariaDB DB instance by running the following query.

```
SHOW STATUS WHERE `variable_name` = 'Threads_connected';
```

Diagnosing and resolving incompatible parameters status for a memory limit

A MariaDB or MySQL DB instance can be placed in **incompatible-parameters** status for a memory limit when both of the following conditions are met:

- The DB instance is either restarted at least three time in one hour or at least five times in one day, or an attempt to restart the DB instance fails.
- The potential memory usage of the DB instance exceeds 1.2 times the memory allocated to its DB instance class.

When a DB instance is restarted for the third time in one hour or for the fifth time in one day, Amazon RDS for MySQL performs a check for memory usage. The check makes the a calculation of the potential memory usage of the DB instance. The value returned by the calculation is the sum of the following values:

- **Value 1** – The sum of the following parameters:
 - `innodb_additional_mem_pool_size`
 - `innodb_buffer_pool_size`
 - `innodb_log_buffer_size`
 - `key_buffer_size`
 - `query_cache_size` (MySQL version 5.6 and 5.7 only)
 - `tmp_table_size`
- **Value 2** – The `max_connections` parameter multiplied by the sum of the following parameters:

- binlog_cache_size
 - join_buffer_size
 - read_buffer_size
 - read_rnd_buffer_size
 - sort_buffer_size
 - thread_stack
- **Value 3 – If the performance_schema parameter is enabled, then multiply the max_connections parameter by 257700.**

If the performance_schema parameter is disabled, then this value is zero.

So, the value returned by the calculation is the following:

Value 1 + Value 2 + Value 3

When this value exceeds 1.2 times the memory allocated to the DB instance class used by the DB instance, the DB instance is placed in **incompatible-parameters** status. For information about the memory allocated to DB instance classes, see [Hardware specifications for DB instance classes \(p. 33\)](#).

The calculation multiplies the value of the max_connections parameter by the sum of several parameters. If the max_connections parameter is set to a large value, it might cause the check to return an inordinately high value for the potential memory usage of the DB instance. In this case, consider lowering the value of the max_connections parameter.

To resolve the problem, complete the following steps:

1. Adjust the memory parameters in the DB parameter group associated with the DB instance so that the potential memory usage is lower than 1.2 times the memory allocated to its DB instance class.

For information about setting parameters, see [Modifying parameters in a DB parameter group \(p. 232\)](#).

2. Restart the DB instance.

For information about setting parameters, see [Starting an Amazon RDS DB instance that was previously stopped \(p. 249\)](#).

Diagnosing and resolving lag between read replicas

After you create a MySQL or MariaDB read replica and the replica is available, Amazon RDS first replicates the changes made to the source DB instance from the time the read replica create operation started. During this phase, the replication lag time for the read replica is greater than 0. You can monitor this lag time in Amazon CloudWatch by viewing the Amazon RDS ReplicaLag metric.

The ReplicaLag metric reports the value of the Seconds_Behind_Master field of the MySQL or MariaDB SHOW SLAVE STATUS command. For more information, see [SHOW SLAVE STATUS](#). When the ReplicaLag metric reaches 0, the replica has caught up to the source DB instance. If the ReplicaLag metric returns -1, replication might not be active. To troubleshoot a replication error, see [Diagnosing and resolving a MySQL or MariaDB read replication failure \(p. 1754\)](#). A ReplicaLag value of -1 can also mean that the Seconds_Behind_Master value can't be determined or is NULL.

The ReplicaLag metric returns -1 during a network outage or when a patch is applied during the maintenance window. In this case, wait for network connectivity to be restored or for the maintenance window to end before you check the ReplicaLag metric again.

The MySQL and MariaDB read replication technology is asynchronous. Thus, you can expect occasional increases for the BinLogDiskUsage metric on the source DB instance and for the ReplicaLag metric

on the read replica. For example, consider a situation where a high volume of write operations to the source DB instance occur in parallel. At the same time, write operations to the read replica are serialized using a single I/O thread. Such a situation can lead to a lag between the source instance and read replica.

For more information about read replicas and MySQL, see [Replication implementation details](#) in the MySQL documentation. For more information about read replicas and MariaDB, see [Replication overview](#) in the MariaDB documentation.

You can reduce the lag between updates to a source DB instance and the subsequent updates to the read replica by doing the following:

- Set the DB instance class of the read replica to have a storage size comparable to that of the source DB instance.
- Make sure that parameter settings in the DB parameter groups used by the source DB instance and the read replica are compatible. For more information and an example, see the discussion of the `max_allowed_packet` parameter in the next section.
- Disable the query cache. For tables that are modified often, using the query cache can increase replica lag because the cache is locked and refreshed often. If this is the case, you might see less replica lag if you disable the query cache. You can disable the query cache by setting the `query_cache_type` parameter to 0 in the DB parameter group for the DB instance. For more information on the query cache, see [Query cache configuration](#).
- Warm the buffer pool on the read replica for InnoDB for MySQL, InnoDB for MariaDB 10.2 or higher, or XtraDB for MariaDB 10.1 or lower. For example, suppose that you have a small set of tables that are being updated often and you're using the InnoDB or XtraDB table schema. In this case, dump those tables on the read replica. Doing this causes the database engine to scan through the rows of those tables from the disk and then cache them in the buffer pool. This approach can reduce replica lag. The following shows an example.

For Linux, macOS, or Unix:

```
PROMPT> mysqldump \
-h <endpoint> \
--port=<port> \
-u=<username> \
-p <password> \
database_name table1 table2 > /dev/null
```

For Windows:

```
PROMPT> mysqldump ^
-h <endpoint> ^
--port=<port> ^
-u=<username> ^
-p <password> ^
database_name table1 table2 > /dev/null
```

Diagnosing and resolving a MySQL or MariaDB read replication failure

Amazon RDS monitors the replication status of your read replicas and updates the **Replication State** field of the read replica instance to **Error** if replication stops for any reason. You can review the details of the associated error thrown by the MySQL or MariaDB engines by viewing the **Replication Error** field. Events that indicate the status of the read replica are also generated, including [RDS-EVENT-0045 \(p. 492\)](#), [RDS-EVENT-0046 \(p. 492\)](#), and [RDS-EVENT-0047 \(p. 491\)](#). For more information about events and subscribing to events, see [Using Amazon RDS event notification \(p. 487\)](#). If a MySQL

error message is returned, check the error in the [MySQL error message documentation](#). If a MariaDB error message is returned, check the error in the [MariaDB error message documentation](#).

Common situations that can cause replication errors include the following:

- The value for the `max_allowed_packet` parameter for a read replica is less than the `max_allowed_packet` parameter for the source DB instance.

The `max_allowed_packet` parameter is a custom parameter that you can set in a DB parameter group. The `max_allowed_packet` parameter is used to specify the maximum size of data manipulation language (DML) that can be run on the database. If the `max_allowed_packet` value for the source DB instance is larger than the `max_allowed_packet` value for the read replica, the replication process can throw an error and stop replication. The most common error is `packet bigger than 'max_allowed_packet' bytes`. You can fix the error by having the source and read replica use DB parameter groups with the same `max_allowed_packet` parameter values.

- Writing to tables on a read replica. If you're creating indexes on a read replica, you need to have the `read_only` parameter set to `0` to create the indexes. If you're writing to tables on the read replica, it can break replication.
- Using a nontransactional storage engine such as MyISAM. Read replicas require a transactional storage engine. Replication is only supported for the following storage engines: InnoDB for MySQL, InnoDB for MariaDB 10.2 or higher, or XtraDB for MariaDB 10.1 or lower.

You can convert a MyISAM table to InnoDB with the following command:

```
alter table <schema>.<table_name> engine=innodb;
```

- Using unsafe nondeterministic queries such as `SYSDATE()`. For more information, see [Determination of safe and unsafe statements in binary logging](#) in the MySQL documentation.

The following steps can help resolve your replication error:

- If you encounter a logical error and you can safely skip the error, follow the steps described in [Skipping the current replication error \(p. 933\)](#). Your MySQL or MariaDB DB instance must be running a version that includes the `mysql_rds_skip_repl_error` procedure. For more information, see [mysql.rds_skip_repl_error \(p. 968\)](#).
- If you encounter a binary log (binlog) position issue, you can change the replica replay position with the `mysql_rds_next_master_log` command. Your MySQL or MariaDB DB instance must be running a version that supports the `mysql_rds_next_master_log` command to change the replica replay position. For version information, see [mysql.rds_next_master_log \(p. 969\)](#).
- If you encounter a temporary performance issue due to high DML load, you can set the `innodb_flush_log_at_trx_commit` parameter to `2` in the DB parameter group on the read replica. Doing this can help the read replica catch up, though it temporarily reduces atomicity, consistency, isolation, and durability (ACID).
- You can delete the read replica and create an instance using the same DB instance identifier. If you do this, the endpoint remains the same as that of your old read replica.

If a replication error is fixed, the **Replication State** changes to **replicating**. For more information, see [Troubleshooting a MySQL read replica problem \(p. 908\)](#).

Creating triggers with binary logging enabled requires SUPER privilege

When trying to create triggers in an RDS for MySQL or RDS for MariaDB DB instance, you might receive the following error.

"You do not have the SUPER privilege and binary logging is enabled"

To use triggers when binary logging is enabled requires the SUPER privilege, which is restricted for RDS for MySQL and RDS for MariaDB DB instances. You can create triggers when binary logging is enabled without the SUPER privilege by setting the `log_bin_trust_function_creators` parameter to true. To set the `log_bin_trust_function_creators` to true, create a new DB parameter group or modify an existing DB parameter group.

To create a new DB parameter group that allows you to create triggers in your RDS for MySQL or RDS for MariaDB DB instance with binary logging enabled, use the following CLI commands. To modify an existing parameter group, start with step 2.

To create a new parameter group to allow triggers with binary logging enabled using the CLI

1. Create a new parameter group.

For Linux, macOS, or Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name allow-triggers \
    --db-parameter-group-family mysql8.0 \
    --description "parameter group allowing triggers"
```

For Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name allow-triggers ^
    --db-parameter-group-family mysql8.0 ^
    --description "parameter group allowing triggers"
```

2. Modify the DB parameter group to allow triggers.

For Linux, macOS, or Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name allow-triggers \
    --parameters "ParameterName=log_bin_trust_function_creators, ParameterValue=true,
    ApplyMethod=pending-reboot"
```

For Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name allow-triggers ^
    --parameters "ParameterName=log_bin_trust_function_creators, ParameterValue=true,
    ApplyMethod=pending-reboot"
```

3. Modify your DB instance to use the new DB parameter group.

For Linux, macOS, or Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --db-parameter-group-name allow-triggers \
    --apply-immediately
```

For Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--db-parameter-group-name allow-triggers ^
--apply-immediately
```

4. For the changes to take effect, manually reboot the DB instance.

```
aws rds reboot-db-instance --db-instance-identifier mydbinstance
```

Diagnosing and resolving point-in-time restore failures

Restoring a DB Instance That Includes Temporary Tables

When attempting a point-in-time restore (PITR) of your MySQL or MariaDB DB instance, you might encounter the following error.

```
Database instance could not be restored because there has been incompatible database activity for restore functionality. Common examples of incompatible activity include using temporary tables, in-memory tables, or using MyISAM tables. In this case, use of Temporary table was detected.
```

PITR relies on both backup snapshots and binary logs (binlogs) from MySQL or MariaDB to restore your DB instance to a particular time. Temporary table information can be unreliable in binlogs and can cause a PITR failure. If you use temporary tables in your MySQL or MariaDB DB instance, you can minimize the possibility of a PITR failure by performing more frequent backups. A PITR failure is most probable in the time between a temporary table's creation and the next backup snapshot.

Restoring a DB Instance That Includes In-Memory Tables

You might encounter a problem when restoring a database that has in-memory tables. In-memory tables are purged during a restart. As a result, your in-memory tables might be empty after a reboot. We recommend that when you use in-memory tables, you architect your solution to handle empty tables in the event of a restart. If you're using in-memory tables with replicated DB instances, you might need to recreate the read replicas after a restart. This might be necessary if a read replica reboots and can't restore data from an empty in-memory table.

For more information about backups and PITR, see [Working with backups \(p. 328\)](#) and [Restoring a DB instance to a specified time \(p. 389\)](#).

Replication stopped error

When you call the `mysql.rds_skip_repl_error` command, you might receive the following error message: `Slave is down or disabled`.

This error message appears because replication is stopped and can't be restarted.

If you need to skip a large number of errors, the replication lag can increase beyond the default retention period for binary log files. In this case, you might encounter a fatal error due to binary log files being purged before they have been replayed on the replica. This purge causes replication to stop, and you can no longer call the `mysql.rds_skip_repl_error` command to skip replication errors.

You can mitigate this issue by increasing the number of hours that binary log files are retained on your replication source. After you have increased the binlog retention time, you can restart replication and call the `mysql.rds_skip_repl_error` command as needed.

To set the binlog retention time, use the [mysql.rds_set_configuration \(p. 972\)](#) procedure. Specify a configuration parameter of 'binlog retention hours' along with the number of hours to retain binlog files on the DB cluster, up to 720 (30 days). The following example sets the retention period for binlog files to 48 hours.

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

Read replica create fails or replication breaks with fatal error 1236

After changing default parameter values for a MySQL or MariaDB DB instance, you might encounter one of the following problems:

- You can't create a read replica for the DB instance.
- Replication fails with `fatal error 1236`.

Some default parameter values for MySQL and MariaDB DB instances help to make sure that the database is ACID compliant and read replicas are crash-safe. They do this by making sure that each commit is fully synchronized by writing the transaction to the binary log before it's committed. Changing these parameters from their default values to improve performance can cause replication to fail when a transaction hasn't been written to the binary log.

To resolve this issue, set the following parameter values:

- `sync-binlog = 1`
- `innodb_support_xa = 1`
- `innodb_flush_log_at_trx_commit = 1`

Can't set backup retention period to 0

There are several reasons why you might need to set the backup retention period to 0. For example, you can disable automatic backups immediately by setting the retention period to 0.

In some cases, you might set the value to 0 and receive a message saying that the retention period must be between 1 and 35. In these cases, check to make sure that you haven't set up a read replica for the instance. Read replicas require backups for managing read replica logs, and therefore you can't set a retention period of 0.

Amazon RDS application programming interface (API) reference

In addition to the AWS Management Console, and the AWS Command Line Interface (AWS CLI), Amazon Relational Database Service (Amazon RDS) also provides an application programming interface (API). You can use the API to automate tasks for managing your DB instances and other objects in Amazon RDS.

- For an alphabetical list of API operations, see [Actions](#).
- For an alphabetical list of data types, see [Data types](#).
- For a list of common query parameters, see [Common parameters](#).
- For descriptions of the error codes, see [Common errors](#).

For more information about the AWS CLI, see [AWS Command Line Interface reference for Amazon RDS](#).

Topics

- [Using the Query API \(p. 1759\)](#)
- [Troubleshooting applications on Amazon RDS \(p. 1760\)](#)

Using the Query API

The following sections briefly discuss the parameters and request authentication used with the Query API.

For general information about how the Query API works, see [Query requests](#) in the *Amazon EC2 API Reference*.

Query parameters

HTTP Query-based requests are HTTP requests that use the HTTP verb GET or POST and a Query parameter named `Action`.

Each Query request must include some common parameters to handle authentication and selection of an action.

Some operations take lists of parameters. These lists are specified using the `param.n` notation. Values of `n` are integers starting from 1.

For information about Amazon RDS regions and endpoints, go to [Amazon Relational Database Service \(RDS\)](#) in the Regions and Endpoints section of the *Amazon Web Services General Reference*.

Query request authentication

You can only send Query requests over HTTPS, and you must include a signature in every Query request. You must use either AWS signature version 4 or signature version 2. For more information, see [Signature Version 4 signing process](#) and [Signature version 2 signing process](#).

Troubleshooting applications on Amazon RDS

Amazon RDS provides specific and descriptive errors to help you troubleshoot problems while interacting with the Amazon RDS API.

Topics

- [Retrieving errors \(p. 1760\)](#)
- [Troubleshooting tips \(p. 1760\)](#)

For information about troubleshooting for Amazon RDS DB instances, see [Troubleshooting for Amazon RDS \(p. 1746\)](#).

Retrieving errors

Typically, you want your application to check whether a request generated an error before you spend any time processing results. The easiest way to find out if an error occurred is to look for an `Error` node in the response from the Amazon RDS API.

XPath syntax provides a simple way to search for the presence of an `Error` node, as well as an easy way to retrieve the error code and message. The following code snippet uses Perl and the `XML::XPath` module to determine if an error occurred during a request. If an error occurred, the code prints the first error code and message in the response.

```
use XML::XPath;
my $xp = XML::XPath->new(xml =>$response);
if ( $xp->find("//Error") )
{print "There was an error processing your request:\n", " Error code: ",
$xp->findvalue("//Error[1]/Code"), "\n", " ",
$xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Troubleshooting tips

We recommend the following processes to diagnose and resolve problems with the Amazon RDS API.

- Verify that Amazon RDS is operating normally in the AWS Region you are targeting by visiting <http://status.aws.amazon.com>.
- Check the structure of your request

Each Amazon RDS operation has a reference page in the *Amazon RDS API Reference*. Double-check that you are using parameters correctly. In order to give you ideas regarding what might be wrong, look at the sample requests or user scenarios to see if those examples are doing similar operations.

- Check the forum

Amazon RDS has a development community forum where you can search for solutions to problems others have experienced along the way. To view the forum, go to

<https://forums.aws.amazon.com/>

Document history

Current API version: 2014-10-31

The following table describes important changes in each release of the *Amazon RDS User Guide* after May 2018. For notification about updates to this documentation, you can subscribe to an RSS feed.

update-history-change	update-history-description	update-history-date
Amazon RDS on AWS Outposts supports Amazon CloudWatch monitoring (p. 1761)	RDS on Outposts now supports Amazon CloudWatch monitoring. For more information, see Amazon RDS on AWS Outposts support for Amazon RDS features .	April 21, 2021
RDS for PostgreSQL supports AWS Lambda functions (p. 1761)	You can now invoke AWS Lambda functions for your RDS for PostgreSQL DB instances. For more information, see Invoking an AWS Lambda function from an RDS for PostgreSQL DB instance .	April 13, 2021
Amazon RDS for PostgreSQL versions 13.2, 12.6, 11.11, 10.16, 9.6.21, and 9.5.25 (p. 1761)	Amazon RDS for PostgreSQL now supports versions 13.2, 12.6, 11.11, 10.16, 9.6.21, and 9.5.25. For more information, see Supported PostgreSQL database versions .	April 13, 2021
RDS for SQL Server supports extended events (p. 1761)	You can use SQL Server extended events to capture debugging and troubleshooting information. For more information, see Using extended events with Amazon RDS for Microsoft SQL Server .	April 8, 2021
Support for MySQL 8.0.23, 5.7.33, and 5.6.51 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 8.0.23, 5.7.33, and 5.6.51. For more information, see MySQL on Amazon RDS versions .	March 31, 2021
Automatic rollback on failed Amazon RDS for MySQL upgrade (p. 1761)	If a DB instance upgrade from MySQL version 5.7 to MySQL version 8.0 fails, Amazon RDS rolls back the changes performed for the upgrade automatically. After the rollback, the MySQL DB instance is running MySQL version 5.7. For more information, see Rollback	March 18, 2021

	<p>after failure to upgrade from MySQL 5.7 to 8.0.</p>	
Amazon RDS supports cross-Region read replicas in opt-in Regions (p. 1761)	You can now replicate DB instances to opt-in Regions. For more information, see Creating a read replica in a different AWS Region .	March 18, 2021
Amazon RDS for Oracle includes the January 2021 PSU for release 12.1.0.2 (p. 1761)	Amazon RDS has released a database engine update for Oracle Database 12c Release 1 (12.1.0.2). Also, Amazon RDS has released database version 19.0.0.0.ru-2021-01.rur-2021-01.r2, which we recommend as an alternative to 19.0.0.0.ru-2021-01.rur-2021-01.r1. For more information, see Oracle database engine release notes .	March 18, 2021
Amazon RDS plans to deprecate Oracle Database 18c (p. 1761)	Oracle Database 18c (18.0.0.0) is on a deprecation path. Oracle Corporation will no longer provide patches for Oracle Database 18c after the end-of-support date. On July 1, 2021, Amazon RDS plans to begin automatically upgrading Oracle Database 18c instances to Oracle Database 19c. Before the automatic upgrades begin, we highly recommend that you manually upgrade your existing Oracle Database 18c instances to Oracle Database 19c. For more information, see Preparing for the automatic upgrade of Oracle Database 18c .	March 11, 2021
Amazon RDS has ended support for Oracle Database 11g (p. 1761)	You can only create DB instances for Oracle Database 12c Release 1 (12.1.0.2) and later. If you have Oracle Database 11g snapshots, upgrade them to a later release. For more information, see Upgrading an Oracle DB snapshot .	March 11, 2021
Amazon RDS supports continuous backups of DB instances in AWS Backup (p. 1761)	You can now create automated backups in AWS Backup and restore DB instances from these backups to a specified time. For more information, see Using AWS Backup to manage automated backups .	March 10, 2021

Amazon RDS for Oracle supports Oracle Management Agent (OMA) version 13.4 (p. 1761)	You can use Oracle Management Agent (OMA) version 13.4 with Oracle Enterprise Manager (OEM) Cloud Control 13c Release 4 Update 9. Amazon RDS for Oracle installs OMA, which then communicates with your Oracle Management Service (OMS) to provide monitoring information. If you run OMS 13.4, you can manage databases by installing OMA 13.4. For more information, see Oracle Management Agent for Enterprise Manager Cloud Control .	March 10, 2021
RDS Proxy endpoint enhancements (p. 1761)	You can create additional endpoints associated with each RDS proxy. Creating an endpoint in a different VPC enables cross-VPC access for the proxy. Proxies for Aurora MySQL clusters can also have read-only endpoints. These reader endpoints connect to reader DB instances in the clusters and can improve read scalability and availability for query-intensive applications. For more information about RDS Proxy, see "Managing Connections with Amazon RDS Proxy" in the Amazon RDS User Guide or the Aurora user guide .	March 8, 2021
Amazon RDS extends supports for cross-Region automated backups (p. 1761)	You can now configure Amazon RDS database instances running PostgreSQL to replicate DB snapshots and transaction logs to a different AWS Region. For more information, see Replicating automated backups to another AWS Region .	March 8, 2021
Replication filters for Amazon RDS for MariaDB and MySQL supported in the China (Beijing) Region and China (Ningxia) Region (p. 1761)	Replication filtering is now supported in the China (Beijing) Region and China (Ningxia) Region. For more information, see Configuring replication filters with MariaDB and Configuring replication filters with MySQL .	March 5, 2021

Amazon RDS supports cross-Region DB snapshot copy in opt-in Regions (p. 1761)	You can now copy DB snapshots to and from opt-in AWS Regions. For more information, see Copying snapshots across AWS Regions .	March 4, 2021
Amazon RDS for PostgreSQL version 13.1 (p. 1761)	Amazon RDS for PostgreSQL now supports version 13.1. For more information, see Supported PostgreSQL database versions .	February 24, 2021
Amazon RDS for SQL Server supports Always On Availability Groups for Standard Edition (p. 1761)	When you create a DB instance using the Multi-AZ configuration on SQL Server 2019 for the Standard Edition database engine, RDS automatically uses Availability Groups. For more information, see Multi-AZ deployments for Microsoft SQL Server .	February 23, 2021
Amazon RDS for Oracle introduces advisor-related procedures (p. 1761)	The <code>rdsadmin_util</code> package includes the procedures <code>advisor_task_set_parameter</code> , <code>advisor_task_drop</code> , and <code>dbms_stats_init</code> . You can use these procedures to modify, stop, and re-enable advisor tasks such as <code>AUTO_STATS_ADVISOR_TASK</code> . For more information, see Setting parameters for advisor tasks .	February 23, 2021
Amazon RDS for Oracle includes January 2021 RU, RUR, and PSU updates (p. 1761)	Amazon RDS has released database engine updates for Oracle Database versions 12.2, 18c, and 19c. For more information, see Oracle database engine release notes .	February 22, 2021
Amazon RDS provides failover reasons for Multi-AZ DB instances (p. 1761)	You can now see more detailed explanations when a Multi-AZ DB instance fails over to a standby replica. For more information, see Failover process for Amazon RDS .	February 18, 2021
Amazon RDS extends support for exporting snapshots to Amazon S3 (p. 1761)	You can now export DB snapshot data to Amazon S3 in China. For more information, see Exporting DB snapshot data to Amazon S3 .	February 17, 2021

Replication filters for Amazon RDS for MariaDB and MySQL (p. 1761)	You can configure replication filters for MySQL and MariaDB instances. Replication filters specify which databases and tables are replicated in a read replica. You can create lists of databases and tables to include or exclude for each read replica. For more information, see Configuring replication filters with MariaDB and Configuring replication filters with MySQL .	February 12, 2021
RDS for Oracle supports APEX 20.2v1 (p. 1761)	You can use APEX 20.2.v1 with all supported versions of Oracle Database. For more information, see Oracle Application Express .	February 2, 2021
Amazon RDS for SQL Server supports local instance storage for the tempdb database (p. 1761)	You can now launch Amazon RDS for SQL Server on Amazon EC2 db.r5d and db.m5d instance types with the tempdb database configured to use an instance store. By placing tempdb data files and log files locally, you can achieve lower read and write latencies when compared to standard storage based on Amazon EBS. For more information, see Instance store support for the tempdb database on Amazon RDS for SQL Server .	January 27, 2021
Amazon RDS for PostgreSQL versions 12.5, 11.10, 10.15, 9.6.20, and 9.5.24 (p. 1761)	Amazon RDS for PostgreSQL now supports versions 12.5, 11.10, 10.15, 9.6.20, and 9.5.24. For more information, see Supported PostgreSQL database versions .	January 12, 2021
Amazon RDS for PostgreSQL supports pg_partman and pg_cron (p. 1761)	Amazon RDS for PostgreSQL now supports the pg_partman and pg_cron extensions. For more information on the pg_partman extension, see Managing PostgreSQL partitions with the pg_partman extension . For more information on the pg_cron extension, see Scheduling maintenance with the PostgreSQL pg_cron extension .	January 12, 2021

Amazon RDS supports publishing the Oracle Management Agent log to Amazon CloudWatch Logs (p. 1761)	The Oracle Management Agent log consists of emctl.log, emdctlj.log, gcagent.log, gcagent_errors.log, emagent.nohup, and secure.log. Amazon RDS publishes each of these logs as a separate CloudWatch log stream. For more information, see Publishing Oracle logs to Amazon CloudWatch Logs .	December 28, 2020
Amazon RDS on AWS Outposts supports additional database versions (p. 1761)	RDS on Outposts now supports additional MySQL and PostgreSQL versions. For more information, see Amazon RDS on AWS Outposts support for Amazon RDS features .	December 23, 2020
Amazon RDS on AWS Outposts supports CoIPs (p. 1761)	RDS on Outposts now supports customer-owned IP addresses (CoIPs). CoIPs provide local or external connectivity to resources in your Outpost subnets through your on-premises network. For more information, see Customer-owned IP addresses for RDS on Outposts .	December 22, 2020
Amazon RDS for Oracle includes the October 2020 PSU for Oracle Database 11g (p. 1761)	Amazon RDS for Oracle has released database engine updates for Oracle Database 11g Enterprise Edition. For more information, see Oracle database engine release notes .	December 21, 2020
Amazon RDS for Oracle plans upgrade of 11g BYOL instances to 19c (p. 1761)	On January 4, 2021, we plan to begin automatically upgrading all editions of Oracle Database 11g instances on the Bring Your Own License (BYOL) model to Oracle Database 19c. All Oracle Database 11g instances, including reserved instances, will move to the latest available Release Update (RU). For more information, see Preparing for the automatic upgrade of Oracle Database 11g BYOL .	December 11, 2020

Amazon RDS supports replicating automated backups to another AWS Region (p. 1761)	You can now configure your Amazon RDS database instances to replicate snapshots and transaction logs to a destination AWS Region of your choice. For more information, see Replicating automated backups to another AWS Region .	December 4, 2020
Amazon RDS for Oracle and Microsoft SQL Server support a new DB instance class (p. 1761)	You can now use the db.r5b instance class to create Amazon RDS DB instances running Oracle or SQL Server. For more information, see Supported DB engines for DB instance classes .	December 4, 2020
Support for MariaDB 10.2.32 (p. 1761)	You can now create Amazon RDS DB instances running MariaDB version 10.2.32. For more information, see MariaDB on Amazon RDS versions .	November 25, 2020
Amazon RDS for SQL Server supports the Microsoft Business Intelligence Suite on SQL Server 2019 (p. 1761)	You can now run SQL Server Analysis Services, SQL Server Integration Services, and SQL Server Reporting Services on DB instances using the latest major version. For more information, see Options for the Microsoft SQL Server database engine .	November 24, 2020
Amazon RDS for PostgreSQL version 13 in the database preview environment (p. 1761)	Amazon RDS for PostgreSQL now supports PostgreSQL version 13 in the database preview environment. For more information, see PostgreSQL 13 versions .	November 24, 2020
Amazon RDS for Oracle includes October 2020 RU, RUR, and PSU updates (p. 1761)	Amazon RDS for Oracle has released database engine updates for all versions except 11.2. Also, Amazon RDS for Oracle provides support for setting and unsetting system diagnostic events using the <code>rdsadmin.rdsadmin_util</code> package, and troubleshooting a GoldenGate error using the procedure <code>rdsadmin_util.truncate_apply\$_cdr_info</code> . For more information, see Oracle database engine release notes .	November 24, 2020

Amazon RDS Performance Insights introduces new dimensions (p. 1761)	You can group database load according to the dimension groups for database (PostgreSQL, MySQL, and MariaDB), application (PostgreSQL), and session type (PostgreSQL). Amazon RDS also supports the dimensions db.name (PostgreSQL, MySQL, and MariaDB), db.application.name (PostgreSQL), and db.session_type.name (PostgreSQL). For more information, see Top load table .	November 24, 2020
Amazon RDS for MariaDB supports a new major version (p. 1761)	You can now create Amazon RDS DB instances running MariaDB version 10.5. For more information, see MariaDB on Amazon RDS versions .	November 23, 2020
Support for MySQL 5.6.49 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 5.6.49. For more information, see MySQL on Amazon RDS versions .	November 20, 2020
Support for MySQL 5.5.62 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 5.5.62. For more information, see MySQL on Amazon RDS versions .	November 20, 2020
Performance Insights supports analyzing statistics for running PostgreSQL queries (p. 1761)	You can now analyze statistics for running queries with Performance Insights for PostgreSQL DB instances. For more information, see Statistics for PostgreSQL .	November 18, 2020
Amazon RDS extends support for storage autoscaling (p. 1761)	You can now enable storage autoscaling when creating a read replica, restoring a DB instance to a specified time, or restoring a MySQL DB instance from an Amazon S3 backup. For more information, see Managing capacity automatically with Amazon RDS storage autoscaling .	November 18, 2020

Amazon RDS for SQL Server supports Database Mail (p. 1761)	With Database Mail you can send email messages from your Amazon RDS for SQL Server database instance. After specifying the email recipients, you can add files or query results to the message you send. For more information, see Using Database Mail on Amazon RDS for SQL Server .	November 4, 2020
Support for MySQL 8.0.21 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 8.0.21. For more information, see MySQL on Amazon RDS versions .	October 22, 2020
Amazon RDS extends support for exporting snapshots to Amazon S3 (p. 1761)	You can now export DB snapshot data to Amazon S3 in all commercial AWS Regions. For more information, see Exporting DB snapshot data to Amazon S3 .	October 22, 2020
Amazon RDS for PostgreSQL supports read replica upgrades (p. 1761)	With Amazon RDS for PostgreSQL, when you do a major version upgrade of the primary DB instance, read replicas are also automatically upgraded. For more information, see Upgrading the PostgreSQL DB engine .	October 15, 2020
Amazon RDS for MariaDB, MySQL and PostgreSQL support the Graviton2 DB instance classes (p. 1761)	You can now use the Graviton2 DB instance classes db.m6g.x and db.r6g.x to create Amazon RDS DB instances running MariaDB, MySQL or PostgreSQL. For more information, see Supported DB Engines for All Available DB Instance Classes .	October 15, 2020
Amazon RDS for SQL Server supports upgrades to SQL Server 2019 (p. 1761)	You can upgrade your SQL Server DB instances to SQL Server 2019. For more information, see Upgrading the Microsoft SQL Server DB Engine .	October 6, 2020
Amazon RDS for Oracle supports specifying the national character set (p. 1761)	The national character set, also called the NCHAR character set, is used in the NCHAR, NVARCHAR2, and NCLOB data types. When you create a database, you can specify either AL16UTF16 (default) or UTF8 as the NCHAR character set. For more information, see Oracle character sets supported in Amazon RDS .	October 2, 2020

Support for MySQL 5.7.31 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 5.7.31. For more information, see MySQL on Amazon RDS versions .	October 1, 2020
Amazon RDS for PostgreSQL supports exporting data to Amazon S3 (p. 1761)	You can query data from a PostgreSQL DB instance and export it directly into files stored in an Amazon S3bucket. For more information, see Exporting data from an RDS for PostgreSQL DB instance to Amazon S3 .	September 24, 2020
Amazon RDS for PostgreSQL 12.4, 11.9, 10.14, 9.6.19, and 9.5.23 (p. 1761)	Amazon RDS for PostgreSQL supports versions 12.4, 11.9, 10.14, 9.6.19, and 9.5.23. For more information, see Supported PostgreSQL database versions .	September 24, 2020
Amazon RDS for MySQL 8.0 supports Percona XtraBackup (p. 1761)	You can now use Percona XtraBackup to restore a backup into an Amazon RDS for MySQL 8.0 DB instance. For more information, see Restoring a backup into a MySQL DB instance .	September 17, 2020
Amazon RDS for SQL Server supports native backup and restore on DB instances with read replicas (p. 1761)	You can restore a SQL Server native backup onto a DB instance that has read replicas configured. For more information, see Importing and exporting SQL Server databases .	September 16, 2020
Amazon RDS for SQL Server supports additional time zones (p. 1761)	You can match your DB instance time zone with your chosen time zone. For more information, see Local time zone for Microsoft SQL Server DB instances .	September 11, 2020
Amazon RDS for PostgreSQL version 13 beta 3 in the database preview environment (p. 1761)	Amazon RDS for PostgreSQL now supports PostgreSQL Version 13 Beta 3 in the Database Preview Environment. For more information, see PostgreSQL 13 versions .	September 9, 2020
RDS for Oracle includes July 2020 RU, RUR, and PSU updates (p. 1761)	Amazon RDS for Oracle has released database engine updates for July 2020. For more information, see Oracle database engine release notes .	August 28, 2020

Amazon RDS for SQL Server supports trace flag 692 (p. 1761)	You can now use trace flag 692 as a startup parameter using DB parameter groups. Enabling this trace flag disables fast inserts while bulk loading data into heap or clustered indexes. For more information, see Disabling fast inserts during bulk loading .	August 27, 2020
Amazon RDS for SQL Server supports Microsoft SQL Server 2019 (p. 1761)	You can now create RDS DB instances that use SQL Server 2019. For more information, see Microsoft SQL Server versions on Amazon RDS .	August 26, 2020
RDS for Oracle supports mounted replica database (p. 1761)	When creating or modifying an Oracle replica, you can place it in mounted mode. Because the replica database doesn't accept user connections, it can't serve a read-only workload. The mounted replica deletes archived redo log files after it applies them. The primary use for mounted replicas is cross-Region disaster recovery. For more information, see Overview of Oracle replicas .	August 13, 2020
RDS for Oracle plans upgrade of 11g SE1 LI instances (p. 1761)	On November 1, 2020, we plan to begin automatically upgrading Oracle Database 11g SE1 License Included (LI) instances to Oracle Database 19c for Amazon RDS for Oracle. All 11g instances, including reserved instances, will move to the latest available Oracle Release Update (RU). For more information, see Preparing for the automatic upgrade of Oracle Database 11g SE1 .	July 31, 2020
Amazon RDS supports new Graviton2 DB instance classes in preview release for PostgreSQL and MySQL (p. 1761)	You can now create Amazon RDS DB instances running PostgreSQL or MySQL that use the db.m6g.x and db.r6g.x DB instance classes. For more information, see Supported DB engines for all available DB instance classes .	July 30, 2020
RDS for Oracle supports APEX 20.1v1 (p. 1761)	You can use APEX 20.1v1 with all supported versions of Oracle Database. For more information, see Oracle application Express .	July 28, 2020

Support for MySQL 8.0.20 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 8.0.20. For more information, see MySQL on Amazon RDS versions .	July 23, 2020
Amazon RDS for MariaDB and MySQL support new DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running MariaDB and MySQL that use the db.m5.16xlarge, db.m5.8xlarge, db.r5.16xlarge, and db.r5.8xlarge DB instance classes. For more information, see Supported DB engines for all available DB instance classes .	July 23, 2020
RDS for SQL Server supports disabling old versions of TLS and ciphers (p. 1761)	You can turn certain security protocols and ciphers on and off. For more information, see Configuring security protocols and ciphers .	July 21, 2020
RDS supports Oracle Spatial on SE2 (p. 1761)	You can use Oracle Spatial in Standard Edition 2 (SE2) for all versions of 12.2, 18c, and 19c. For more information, see Oracle Spatial .	July 9, 2020
Amazon RDS supports AWS PrivateLink (p. 1761)	Amazon RDS now supports creating Amazon VPC endpoints for Amazon RDS API calls to keep traffic between applications and Amazon RDS in the AWS network. For more information, see Amazon RDS and interface VPC endpoints (AWS PrivateLink) .	July 9, 2020
Amazon RDS for PostgreSQL versions 9.4.x are deprecated (p. 1761)	Amazon RDS for PostgreSQL no longer supports versions 9.4.x. For supported versions, see Supported PostgreSQL database versions .	July 8, 2020
Support for MariaDB 10.3.23 and 10.4.13 (p. 1761)	You can now create Amazon RDS DB instances running MariaDB version 10.3.23 and 10.4.13. For more information, see MariaDB on Amazon RDS versions .	July 6, 2020
Amazon RDS on AWS Outposts (p. 1761)	You can create Amazon RDS DB instances on AWS Outposts. For more information, see Working with Amazon RDS on AWS Outposts .	July 6, 2020

Amazon RDS for Oracle creates inventory files automatically (p. 1761)	To open service requests for BYOL customers, Oracle Support requests inventory files generated by Opatch. Amazon RDS for Oracle automatically creates inventory files every hour in the BDUMP directory. For more information, see Accessing Opatch files .	July 6, 2020
Support for MySQL 5.7.30 and 5.6.48 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 5.7.30 and 5.6.48. For more information, see MySQL on Amazon RDS versions .	June 25, 2020
Amazon RDS for PostgreSQL supports PostgreSQL 12.3 (p. 1761)	Amazon RDS for PostgreSQL supports version 12.3. For more information, see Supported PostgreSQL database versions .	June 17, 2020
Amazon RDS for Oracle supports ADRCI (p. 1761)	The Automatic Diagnostic Repository Command Interpreter (ADRCI) utility is an Oracle command-line tool that you use to manage diagnostic data. By using the functions in the Amazon RDS package <code>rdsadmin_adrci_util</code> , you can list and package problems and incidents, and also show trace files. For more information, see Common DBA diagnostic tasks for Oracle DB instances .	June 17, 2020
Support for MySQL 8.0.19 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 8.0.19. For more information, see MySQL on Amazon RDS versions .	June 2, 2020
MySQL 8.0 supports lower case table names (p. 1761)	You can now set the <code>lower_case_table_names</code> parameter to 1 for Amazon RDS DB instances running MySQL version 8.0.19 and higher 8.0 versions. For more information, see MySQL parameter exceptions for Amazon RDS DB instances .	June 2, 2020

Amazon RDS for Microsoft SQL Server supports SQL Server Integration Services (SSIS) (p. 1761)	SSIS is a platform for data integration and workflow applications. You can enable SSIS on existing or new DB instances. It's installed on the same DB instance as your database engine. For more information, see Support for SQL Server Integration Services in SQL Server .	May 19, 2020
Amazon RDS for Microsoft SQL Server supports SQL Server Reporting Services (SSRS) (p. 1761)	SSRS is a server-based application used for report generation and distribution. You can enable SSRS on existing or new DB instances. It's installed on the same DB instance as your database engine. For more information, see Support for SQL Server Reporting Services in SQL Server .	May 15, 2020
Amazon RDS for Microsoft SQL Server supports S3 integration on Multi-AZ instances (p. 1761)	You can now use Amazon S3 with SQL Server features such as bulk insert on Multi-AZ DB instances. For more information, see Integrating an Amazon RDS for SQL Server DB instance with Amazon S3 .	May 15, 2020
Amazon RDS for Oracle supports purging the recycle bin (p. 1761)	The <code>rdsadmin.rdsadmin_util.purge_dba_recyclebin</code> procedure purges the recycle bin. For more information, see Purging the recycle bin .	May 13, 2020
Amazon RDS for Oracle improves manageability of Automatic Workload Repository (AWR) (p. 1761)	The <code>rdsadmin.rdsadmin_diagnostic_util</code> procedures generate AWR reports and extract AWR data into dump files. For more information, see Generating performance reports with Automatic Workload Repository (AWR) .	May 13, 2020
Amazon RDS for Oracle April 2020 RU, RUR, and PSU (p. 1761)	Amazon RDS for Oracle has released database engine updates for April 2020. For more information, see Oracle database engine release notes .	May 13, 2020

Amazon RDS for Microsoft SQL Server supports Microsoft Distributed Transaction Coordinator (MSDTC) (p. 1761)	Amazon RDS for SQL Server supports distributed transactions between hosts. For more information, see Support for Microsoft Distributed Transaction Coordinator in SQL Server.	May 4, 2020
Amazon RDS for PostgreSQL versions 11.7, 10.12, 9.6.17, and 9.5.21 (p. 1761)	Amazon RDS for PostgreSQL supports versions 11.7, 10.12, 9.6.17, and 9.5.21. For more information, see Supported PostgreSQL database versions.	April 28, 2020
Amazon RDS for Microsoft SQL Server supports new versions (p. 1761)	You can now create Amazon RDS DB instances running SQL Server versions 2017 CU19 14.00.3281.6, 2016 SP2 CU11 13.00.5598.27, 2014 SP3 CU4 12.00.6329.1, and 2012 SP4 GDR 11.0.7493.4 for all editions. For more information, see Microsoft SQL Server versions on Amazon RDS.	April 28, 2020
Amazon RDS available in the Europe (Milan) Region (p. 1761)	Amazon RDS is now available in the Europe (Milan) Region. For more information, see Regions and Availability Zones.	April 28, 2020
Amazon RDS support for Local Zones (p. 1761)	You can now launch DB instances into a Local Zone subnet. For more information, see Regions, Availability Zones, and Local Zones.	April 23, 2020
Amazon RDS available in the Africa (Cape Town) Region (p. 1761)	Amazon RDS is now available in the Africa (Cape Town) Region. For more information, see Regions and Availability Zones.	April 22, 2020
Amazon RDS for Microsoft SQL Server supports SQL Server Analysis Services (SSAS) (p. 1761)	SSAS is an online analytical processing (OLAP) and data mining tool that is installed within SQL Server. You can enable SSAS on existing or new DB instances. It's installed on the same DB instance as your database engine. For more information, see Support for SQL Server Analysis Services in SQL Server.	April 17, 2020

Amazon RDS proxy for PostgreSQL (p. 1761)	Amazon RDS Proxy is now available for PostgreSQL. You can use RDS Proxy to reduce the overhead of connection management on your DB instance and also the chance of "too many connections" errors. The RDS Proxy is currently in public preview for PostgreSQL. For more information, see Managing connections with Amazon RDS proxy (preview) .	April 8, 2020
Amazon RDS for Oracle supports Oracle APEX version 19.2.v1 (p. 1761)	Amazon RDS for Oracle now supports Oracle Application Express (APEX) version 19.2.v1. For more information, see Oracle application Express .	April 8, 2020
Amazon RDS for MariaDB supports a new major version (p. 1761)	You can now create Amazon RDS DB instances running MariaDB version 10.4. For more information, see MariaDB on Amazon RDS versions .	April 6, 2020
Amazon RDS Performance Insights is available for Amazon RDS for MariaDB 10.4 (p. 1761)	Amazon RDS Performance Insights is now available for Amazon RDS for MariaDB version 10.4. For more information, see Using Amazon RDS performance insights .	April 6, 2020
Amazon RDS for PostgreSQL versions 9.3.x are deprecated (p. 1761)	Amazon RDS for PostgreSQL no longer supports versions 9.3.x. For supported versions, see Supported PostgreSQL database versions .	April 3, 2020
Amazon RDS for Microsoft SQL Server supports read replicas (p. 1761)	You can now create read replicas for SQL Server DB instances. For more information, see Working with read replicas .	April 3, 2020
Amazon RDS for Microsoft SQL Server supports multifile backups (p. 1761)	You can now back up databases to multiple files using SQL Server native backup and restore. For more information, see Backing up a database .	April 2, 2020
Amazon RDS for PostgreSQL supports PostgreSQL 12.2 (p. 1761)	Amazon RDS for PostgreSQL supports version 12.2. For more information, see Supported PostgreSQL database versions .	March 31, 2020

Amazon RDS for Oracle integration with AWS License Manager (p. 1761)	Amazon RDS for Oracle is now integrated with AWS License Manager. If you use the Bring Your Own License model, AWS License Manager integration makes it easier to monitor your Oracle license usage within your organization. For more information, see Integrating with AWS License Manager .	March 23, 2020
Support for 64 TiB on db.r5 instances in Amazon RDS for MariaDB and MySQL (p. 1761)	You can now create Amazon RDS DB instances for MariaDB and MySQL that use the db.r5 DB instance class with up to 64 TiB of storage. For more information, see Factors that affect storage performance .	March 18, 2020
Support for MySQL 8.0.17 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 8.0.17. For more information, see MySQL on Amazon RDS versions .	March 10, 2020
Amazon RDS Performance Insights is available for Amazon RDS for MySQL 8.0 (p. 1761)	Amazon RDS Performance Insights is now available for Amazon RDS for MySQL version 8.0.17 and higher 8.0 versions. For more information, see Using Amazon RDS performance insights .	March 10, 2020
Support for MySQL 5.6.46 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 5.6.46. For more information, see MySQL on Amazon RDS versions .	February 28, 2020
Amazon RDS Performance Insights is available for Amazon RDS for MariaDB 10.3 (p. 1761)	Amazon RDS Performance Insights is now available for Amazon RDS for MariaDB version 10.3.13 and higher 10.3 versions. For more information, see Using Amazon RDS performance insights .	February 26, 2020
Support for MySQL 5.7.28 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 5.7.28. For more information, see MySQL on Amazon RDS versions .	February 20, 2020
Support for MariaDB 10.3.20 (p. 1761)	You can now create Amazon RDS DB instances running MariaDB version 10.3.20. For more information, see MariaDB on Amazon RDS versions .	February 20, 2020

Amazon RDS for Oracle January 2020 RU, RUR, and PSU (p. 1761)	Amazon RDS for Oracle has released database engine updates for January 2020. For more information, see Oracle database engine release notes .	February 20, 2020
Amazon RDS for Microsoft SQL Server supports a new DB instance class (p. 1761)	You can now create Amazon RDS DB instances running SQL Server that use the db.z1d DB instance class. For more information, see DB instance class support for Microsoft SQL Server .	February 19, 2020
Support for cross-account, cross-VPC Active Directory domains in Amazon RDS for SQL Server (p. 1761)	Amazon RDS for Microsoft SQL Server now supports associating DB instances with Active Directory domains owned by different accounts and VPCs. For more information, see Using Windows authentication with a Microsoft SQL Server DB instance .	February 13, 2020
Oracle OLAP option (p. 1761)	Amazon RDS for Oracle now supports the On-line Analytical Processing (OLAP) option for Oracle DB instances. You can use Oracle OLAP to analyze large amounts of data by creating dimensional objects and cubes in accordance with the OLAP standard. For more information, see Oracle OLAP .	February 13, 2020
FIPS 140-2 support for Oracle (p. 1761)	Amazon RDS for Oracle supports the Federal Information Processing Standard Publication 140-2 (FIPS 140-2) for SSL/TLS connections. For more information, see FIPS support .	February 11, 2020
Amazon RDS for PostgreSQL supports new versions (p. 1761)	Amazon RDS for PostgreSQL supports versions 11.6, 10.11, 9.6.16, 9.5.20, and 9.4.25. For more information, see Supported PostgreSQL database versions .	February 11, 2020
Amazon RDS for PostgreSQL supports new DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running PostgreSQL that use the db.m5.16xlarge, db.m5.8xlarge, db.r5.16xlarge, and db.r5.8xlarge DB instance classes. For more information, see Supported DB engines for all available DB instance classes .	February 11, 2020

Performance Insights supports analyzing statistics of running MariaDB and MySQL queries (p. 1761)	You can now analyze statistics of running queries with Performance Insights for MariaDB and MySQL DB instances. For more information, see Analyzing statistics of running queries .	February 4, 2020
Support for exporting DB snapshot data to Amazon S3 for MariaDB, MySQL, and PostgreSQL (p. 1761)	Amazon RDS supports exporting DB snapshot data to Amazon S3 for MariaDB, MySQL, and PostgreSQL. For more information, see Exporting DB snapshot data to Amazon S3 .	January 23, 2020
Amazon RDS for MySQL supports Kerberos authentication (p. 1761)	You can now use Kerberos authentication to authenticate users when they connect to your Amazon RDS for MySQL DB instances. For more information, see Using Kerberos authentication for MySQL .	January 21, 2020
Amazon RDS for Oracle October 2019 RU and RUR for Oracle Database 19c (p. 1761)	Amazon RDS for Oracle has released database engine updates for October 2019 for Oracle Database 19c. For more information, see Oracle database engine release notes .	January 9, 2020
Amazon RDS Performance Insights supports viewing more SQL text for Amazon RDS for Microsoft SQL Server (p. 1761)	Amazon RDS Performance Insights now supports viewing more SQL text in the Performance Insights dashboard for Amazon RDS for Microsoft SQL Server DB instances. For more information, see Viewing more SQL text in the Performance Insights dashboard .	December 17, 2019
Amazon RDS proxy (p. 1761)	You can reduce the overhead of connection management on your cluster, and reduce the chance of "too many connections" errors, by using the Amazon RDS Proxy. You associate each proxy with an RDS DB instance or Aurora DB cluster. Then you use the proxy endpoint in the connection string for your application. The Amazon RDS Proxy is currently in a public preview state. It supports the RDS for MySQL database engine. For more information, see Managing connections with Amazon RDS proxy (preview) .	December 3, 2019

Amazon RDS on AWS Outposts (preview) (p. 1761)	With Amazon RDS on AWS Outposts, you can create AWS-managed relational databases in your on-premises data centers. RDS on Outposts enables you to run RDS databases on AWS Outposts. For more information, see Amazon RDS on AWS Outposts (preview) .	December 3, 2019
Amazon RDS for Oracle supports cross-region read replicas (p. 1761)	Amazon RDS for Oracle now supports cross-region read replicas with Active Data Guard. For more information, see Working with read replicas and Working with Oracle read replicas .	November 26, 2019
Performance Insights supports analyzing statistics of running Oracle queries (p. 1761)	You can now analyze statistics of running queries with Performance Insights for Oracle DB instances. For more information, see Analyzing statistics of running queries .	November 25, 2019
Amazon RDS for Microsoft SQL Server supports publishing logs to CloudWatch Logs (p. 1761)	You can configure your Amazon RDS for SQL Server DB instance to publish log events directly to Amazon CloudWatch Logs. For more information, see Publishing SQL Server logs to Amazon CloudWatch Logs .	November 25, 2019
Amazon RDS for Microsoft SQL Server supports new DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running SQL Server that use the db.x1e and db.x1 DB instance classes. For more information, see DB instance class support for Microsoft SQL Server .	November 25, 2019
Amazon RDS for Microsoft SQL Server supports differential and log restores (p. 1761)	You can restore differential backups and logs using SQL Server native backup and restore. For more information, see Using native backup and restore .	November 25, 2019
Multi-AZ supported on Amazon RDS for Microsoft SQL Server in new regions (p. 1761)	Multi-AZ on SQL Server is now available in China, Middle East (Bahrain), and Europe (Stockholm). For more information, see Multi-AZ deployments for Microsoft SQL Server .	November 22, 2019

Amazon RDS for Microsoft SQL Server now supports bulk insert and S3 integration (p. 1761)	You can transfer files between a SQL Server DB instance and an Amazon S3 bucket. Then you can use Amazon S3 with SQL Server features such as bulk insert. For more information, see Integrating an Amazon RDS for SQL Server DB instance with Amazon S3 .	November 21, 2019
Amazon RDS for Oracle October 2019 RU, RUR, and PSU (p. 1761)	Amazon RDS for Oracle has released database engine updates for October 2019. For more information, see Oracle database engine release notes .	November 19, 2019
Performance Insights counters for Amazon RDS for Microsoft SQL Server (p. 1761)	You can now add performance counters to your Performance Insights charts for Microsoft SQL Server DB instances. For more information, see Performance Insights counters for Amazon RDS for Microsoft SQL Server .	November 12, 2019
Amazon RDS for Microsoft SQL Server supports new DB instance class sizes (p. 1761)	You can now create Amazon RDS DB instances running SQL Server that use the 8xlarge and 16xlarge instance sizes for the db.m5 and db.r5 DB instance classes. Instance sizes ranging from small to 2xlarge are now available for the db.t3 instance class. For more information, see DB instance class support for Microsoft SQL Server .	November 11, 2019
Support for PostgreSQL snapshot upgrades (p. 1761)	If you have existing manual DB snapshots of your Amazon RDS PostgreSQL DB instances, you can now upgrade them to a later version of the PostgreSQL database engine. For more information, see Upgrading a PostgreSQL DB snapshot .	November 7, 2019
Amazon RDS for Oracle supports a new major version (p. 1761)	You can now create Amazon RDS DB instances running Oracle Database 19c (19.0). For more information, see Oracle Database 19c with Amazon RDS .	November 7, 2019
Amazon RDS for PostgreSQL version 12.0 in the database preview environment (p. 1761)	Amazon RDS for PostgreSQL now supports PostgreSQL Version 12.0 in the Database Preview Environment. For more information, see PostgreSQL version 12.0 in the database preview environment .	November 1, 2019

Amazon RDS for PostgreSQL supports Kerberos authentication (p. 1761)	You can now use Kerberos authentication to authenticate users when they connect to your Amazon RDS DB instance running PostgreSQL. For more information, see Using Kerberos authentication with Amazon RDS for PostgreSQL .	October 28, 2019
OEM Management Agent database tasks for Oracle DB instances (p. 1761)	Amazon RDS for Oracle DB instances now support procedures to invoke certain EMCTL commands on the Management Agent. For more information, see OEM agent database tasks .	October 24, 2019
Amazon RDS for PostgreSQL versions 11.5, 10.10, 9.6.15, 9.5.19, and 9.4.24 (p. 1761)	Amazon RDS for PostgreSQL supports versions 11.5, 10.10, 9.6.15, 9.5.19, and 9.4.24. For more information, see Supported PostgreSQL database versions .	October 8, 2019
Amazon RDS for PostgreSQL supports PostgreSQL transportable databases (p. 1761)	PostgreSQL Transportable Databases provide an extremely fast method of migrating an RDS PostgreSQL database between two DB instances. For more information, see Transporting PostgreSQL databases between DB instances .	October 8, 2019
Amazon RDS for Oracle supports Kerberos authentication (p. 1761)	You can now use Kerberos authentication to authenticate users when they connect to your Amazon RDS DB instance running Oracle. For more information, see Using Kerberos authentication with Amazon RDS for Oracle .	September 30, 2019
Amazon RDS for PostgreSQL version 12 beta 3 in the database preview environment (p. 1761)	Amazon RDS for PostgreSQL now supports PostgreSQL Version 12 Beta 3 in the Database Preview Environment. For more information, see PostgreSQL version 12 beta 3 on Amazon RDS in the database preview environment .	August 28, 2019
Support for MySQL 8.0.16 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 8.0.16. For more information, see MySQL on Amazon RDS versions .	August 19, 2019

Amazon RDS for Oracle supports a new major version (p. 1761)	You can now create Amazon RDS DB instances running Oracle Database 18c (18.0). For more information, see Oracle Database 18c with Amazon RDS .	August 15, 2019
Amazon RDS for Oracle July 2019 RU, RUR, and PSU (p. 1761)	Amazon RDS for Oracle has released database engine version 12.2.0.1.ru-2019-07.rur-2019-07.r1 to support the July 2019 Release Update (RU) and Release Update Revision (RUR). Amazon RDS for Oracle has also released database engine versions 12.1.0.2.v17 and 11.2.0.4.v21 to support the July 2019 Oracle Database Patch Set Update (PSU). For more information, see Oracle database engine release notes .	August 8, 2019
Management Agent for OEM 13c release 3 (p. 1761)	Amazon RDS for Oracle DB instances now support the Management Agent for Oracle Enterprise Manager (OEM) Cloud Control 13c Release 3. For more information, see Oracle Management Agent for Enterprise Manager cloud control .	August 7, 2019
Amazon RDS for PostgreSQL version 12 beta 2 in the database preview environment (p. 1761)	Amazon RDS for PostgreSQL now supports PostgreSQL Version 12 Beta 2 in the Database Preview Environment. For more information, see PostgreSQL version 12 beta 2 on Amazon RDS in the database preview environment .	August 6, 2019
Amazon RDS supports server collations for SQL Server (p. 1761)	Amazon RDS for SQL Server supports a selection of collations for new DB instances. For more information, see Collations and character sets for Microsoft SQL Server .	July 29, 2019
Amazon RDS for PostgreSQL versions 11.4, 10.9, 9.6.14, 9.5.18, and 9.4.23 (p. 1761)	Amazon RDS for PostgreSQL supports minor versions 11.4, 10.9, 9.6.14, 9.5.18, and 9.4.23. For more information, see Supported PostgreSQL database versions .	July 3, 2019

Amazon RDS for Oracle supports Oracle APEX version 19.1.v1 (p. 1761)	Amazon RDS for Oracle now supports Oracle Application Express (APEX) version 19.1.v1. For more information, see Oracle application Express .	June 28, 2019
Amazon RDS for PostgreSQL version 13 beta 1 in the database preview environment (p. 1761)	Amazon RDS for PostgreSQL now supports PostgreSQL Version 13 Beta 1 in the Database Preview Environment. For more information, see PostgreSQL 13 versions .	June 22, 2019
Amazon RDS storage autoscaling (p. 1761)	Storage autoscaling for Amazon RDS DB instances enables Amazon RDS to automatically expand the storage associated with a DB instance to reduce the chance of out-of-space conditions. For information about storage autoscaling, see Working with storage for Amazon RDS DB instances .	June 20, 2019
Amazon RDS for Oracle supports db.z1d DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running Oracle that use the db.z1d DB instance classes. For more information, see DB instance class .	June 13, 2019
Amazon RDS Performance Insights supports viewing more SQL text for Amazon RDS for Oracle (p. 1761)	Amazon RDS Performance Insights now supports viewing more SQL text in the Performance Insights dashboard for Amazon RDS for Oracle DB instances. For more information, see Viewing more SQL text in the Performance Insights dashboard .	June 10, 2019
Amazon RDS adds support native restores of SQL Server databases up to 16 TB (p. 1761)	You can now do native restores of up to 16 TB from SQL Server to Amazon RDS. For more information, see Amazon RDS for SQL Server: Limitations and recommendations .	June 4, 2019
Amazon RDS adds support for Microsoft SQL Server audit (p. 1761)	Using Amazon RDS for Microsoft SQL Server, you can audit server and database level events using SQL Server Audit, and view the results on your DB instance or send the audit log files directly to Amazon S3. For more information, see SQL Server Audit .	May 23, 2019

Improvements to Amazon RDS recommendations (p. 1761)	Amazon RDS has improved its automated recommendations for database resources. For example, Amazon RDS now provides recommendations for database parameters. For more information, see Using Amazon RDS recommendations .	May 22, 2019
Support for more databases per DB instance for Amazon RDS for SQL Server (p. 1761)	You can create up to 30 databases on each of your DB instances running Microsoft SQL Server. For more information, see Limits for Microsoft SQL Server DB instances .	May 21, 2019
Support for 64 TiB and 80k IOPS of storage for Amazon RDS for MariaDB, MySQL and PostgreSQL (p. 1761)	You can now create Amazon RDS DB instances for MariaDB, MySQL and PostgreSQL with up to 64 TiB of storage and up to 80,000 provisioned IOPS. For more information, see DB instance storage .	May 20, 2019
Amazon RDS for MySQL supports upgrade prechecks (p. 1761)	When you upgrade a DB instance from MySQL 5.7 to MySQL 8.0, Amazon RDS performs prechecks for incompatibilities. For more information, see Prechecks for upgrades from MySQL 5.7 to 8.0 .	May 17, 2019
Support for the MySQL password validation plugin (p. 1761)	You can now use the MySQL validate_password plugin for improved security of Amazon RDS for MySQL DB instances. For more information, see Using the Password Validation Plugin .	May 16, 2019
Amazon RDS for Oracle April 2019 RU, RUR, and PSU (p. 1761)	Amazon RDS for Oracle has released database engine version 12.2.0.1.ru-2019-04.rur-2019-04.r1 to support the April 2019 Release Update (RU) and Release Update Revision (RUR). Amazon RDS for Oracle has also released database engine versions 12.1.0.2.v16 and 11.2.0.4.v20 to support the April 2019 Oracle Database Patch Set Update (PSU). For more information, see Oracle database engine release notes .	May 16, 2019

Performance Insights counters for Amazon RDS for Oracle (p. 1761)	You can now add performance counters to your Performance Insights charts for Oracle DB instances. For more information, see Performance Insights counters for Amazon RDS for Oracle .	May 8, 2019
Amazon RDS for PostgreSQL versions 11.2, 10.7, 9.6.12, 9.5.16, and 9.4.21 (p. 1761)	Amazon RDS for PostgreSQL supports minor versions 11.2, 10.7, 9.6.12, 9.5.16, and 9.4.21. For more information, see Supported PostgreSQL database versions .	May 1, 2019
Support for per-second billing (p. 1761)	Amazon RDS is now billed in 1-second increments in all AWS Regions except AWS GovCloud (US) for on-demand instances. For more information, see DB instance billing for Amazon RDS .	April 25, 2019
Support for importing data from Amazon S3 for Amazon RDS for PostgreSQL (p. 1761)	You can now import data from Amazon S3 file into a table in an RDS PostgreSQL DB instance. For more information, see Importing Amazon S3 data into an RDS PostgreSQL DB instance .	April 24, 2019
Support for restoring 5.7 backups from Amazon S3 (p. 1761)	You can now create a backup of your MySQL version 5.7 database, store it on Amazon S3, and then restore the backup file onto a new Amazon RDS DB instance running MySQL. For more information, see Restoring a backup into a MySQL DB instance .	April 17, 2019
Support for multiple major version upgrades for Amazon RDS for PostgreSQL (p. 1761)	With Amazon RDS for PostgreSQL, you can now choose from multiple major versions when you upgrade the DB engine. This feature enables you to skip ahead to a newer major version when you upgrade select PostgreSQL engine versions. For more information, see Upgrading the PostgreSQL DB engine .	April 16, 2019
Support for 64 TiB of storage for Amazon RDS for Oracle (p. 1761)	You can now create Amazon RDS DB instances for Oracle with up to 64 TiB of storage and up to 80,000 provisioned IOPS. For more information, see DB instance storage .	April 4, 2019

Support for MySQL 8.0.15 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 8.0.15. For more information, see MySQL on Amazon RDS versions .	April 3, 2019
Support for MariaDB 10.3.13 (p. 1761)	You can now create Amazon RDS DB instances running MariaDB version 10.3.13. For more information, see MariaDB on Amazon RDS versions .	April 3, 2019
Microsoft SQL Server 2008 R2 deprecated on Amazon RDS (p. 1761)	Support for Microsoft SQL Server 2008 R2 is deprecated, coinciding with the Microsoft plan to terminate extended support for this version on July 9, 2019. Any existing Microsoft SQL Server 2008 R2 snapshots are to be automatically upgraded to the latest minor version of Microsoft SQL Server 2012 starting on June 1, 2019. For more information, see Microsoft SQL Server 2008 R2 support on Amazon RDS .	April 2, 2019
Always On availability groups supported in Microsoft SQL Server 2017 (p. 1761)	You can now use Always On Availability Groups in SQL Server 2017 Enterprise Edition 14.00.3049.1 or later. For more information, see Multi-AZ deployments for Microsoft SQL Server .	March 29, 2019
View volume metrics (p. 1761)	You can now view metrics for the Amazon Elastic Block Store (Amazon EBS) volumes, which are the physical devices used for database and log storage. For more information, see Viewing Enhanced Monitoring .	March 20, 2019
Support for MySQL 5.7.25 (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 5.7.25. For more information, see MySQL on Amazon RDS versions .	March 19, 2019
Amazon RDS for Oracle supports RMAN DBA tasks (p. 1761)	Amazon RDS for Oracle now supports Oracle Recovery Manager (RMAN) DBA tasks, including RMAN backups. For more information, see Common DBA Recovery Manager (RMAN) tasks for Oracle DB instances .	March 14, 2019

Amazon RDS for PostgreSQL supports version 11.1 (p. 1761)	You can now create Amazon RDS DB instances running PostgreSQL version 11.1. For more information, see PostgreSQL version 11.1 on Amazon RDS .	March 12, 2019
Multiple-file restore is available in Amazon RDS for SQL Server (p. 1761)	You can now restore from multiple files with Amazon RDS for SQL Server. For more information, see Restoring a database .	March 11, 2019
MariaDB 10.2.21 (p. 1761)	You can now create Amazon RDS DB instances running MariaDB version 10.2.21. For more information, see MariaDB on Amazon RDS versions .	March 11, 2019
Amazon RDS for Oracle supports read replicas (p. 1761)	Amazon RDS for Oracle now supports read replicas with Active Data Guard. For more information, see Working with read replicas and Working with Oracle read replicas .	March 11, 2019
Amazon RDS Performance Insights is available for Amazon RDS for MariaDB (p. 1761)	Amazon RDS Performance Insights is now available for Amazon RDS for MariaDB. For more information, see Using Amazon RDS Performance Insights .	March 11, 2019
MySQL 8.0.13 and 5.7.24 (p. 1761)	You can now create Amazon RDS DB instances running MySQL versions 8.0.13 and 5.7.24. For more information, see MySQL on Amazon RDS versions .	March 8, 2019
Amazon RDS Performance Insights is available for Amazon RDS for SQL Server (p. 1761)	Amazon RDS Performance Insights is now available for Amazon RDS for SQL Server. For more information, see Using Amazon RDS Performance Insights .	March 4, 2019
Amazon RDS for Oracle supports Amazon S3 integration (p. 1761)	You can now transfer files between an Amazon RDS for Oracle DB instance and an Amazon S3 bucket. For more information, see Integrating Amazon RDS for Oracle and Amazon S3 .	February 26, 2019

Amazon RDS for MySQL and Amazon RDS for MariaDB support db.t3 DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running MySQL or MariaDB that use the db.t3 DB instance classes. For more information, see DB instance class .	February 20, 2019
Amazon RDS for MySQL and Amazon RDS for MariaDB support db.r5 DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running MySQL or MariaDB that use the db.r5 DB instance classes. For more information, see DB instance class .	February 20, 2019
Performance Insights counters for RDS for MySQL and PostgreSQL (p. 1761)	You can now add performance counters to your Performance Insights charts for MySQL and PostgreSQL DB instances. For more information, see Performance Insights dashboard components .	February 19, 2019
Amazon RDS for PostgreSQL now supports adaptive autovacuum parameter tuning (p. 1761)	Adaptive autovacuum parameter tuning with Amazon RDS for PostgreSQL helps prevent transaction ID wraparound by adjusting autovacuum parameter values automatically. For more information, see Reducing the likelihood of transaction ID wraparound .	February 12, 2019
Amazon RDS for Oracle supports Oracle APEX versions 18.1.v1 and 18.2.v1 (p. 1761)	Amazon RDS for Oracle now supports Oracle Application Express (APEX) versions 18.1.v1 and 18.2.v1. For more information, see Oracle application Express .	February 11, 2019
Amazon RDS for Oracle January 2019 RU, RUR, and PSU (p. 1761)	Amazon RDS for Oracle has released database engine version 12.2.0.1.ru-2019-01.rur-2019-01.r1 to support the January 2019 Release Update (RU) and Release Update Revision (RUR). Amazon RDS for Oracle has also released database engine versions 12.1.0.2.v15 and 11.2.0.4.v19 to support the January 2019 Oracle Database Patch Set Update (PSU). For more information, see Oracle database engine release notes .	February 8, 2019

Amazon RDS Performance Insights supports viewing more SQL text for RDS for MySQL (p. 1761)	Amazon RDS Performance Insights now supports viewing more SQL text in the Performance Insights dashboard for MySQL DB instances. For more information, see Viewing more SQL text in the Performance Insights dashboard .	February 6, 2019
Amazon RDS for PostgreSQL supports db.t3 DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running PostgreSQL that use the db.t3 DB instance classes. For more information, see DB instance class .	January 25, 2019
Amazon RDS for Oracle supports db.t3 DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running Oracle that use the db.t3 DB instance classes. For more information, see DB instance class .	January 25, 2019
Amazon RDS Performance Insights supports viewing more SQL text for Amazon RDS PostgreSQL (p. 1761)	Amazon RDS Performance Insights now supports viewing more SQL text in the Performance Insights dashboard for Amazon RDS PostgreSQL DB instances. For more information, see Viewing more SQL text in the Performance Insights dashboard .	January 24, 2019
Amazon RDS for Oracle supports a new version of SQLT (p. 1761)	Amazon RDS for Oracle now supports SQLT version 12.2.180725. For more information, see Oracle SQLT .	January 22, 2019
Amazon RDS for PostgreSQL supports new minor versions (p. 1761)	Amazon RDS for PostgreSQL now supports the following new minor versions: 10.6, 9.6.11, 9.5.15, 9.4.20, and 9.3.25. For more information, see Amazon RDS for PostgreSQL versions and extensions .	December 19, 2018
Amazon RDS for PostgreSQL supports db.r5 DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running PostgreSQL that use the db.r5 DB instance classes. For more information, see DB instance class .	December 19, 2018

Amazon RDS for PostgreSQL now supports restricted password management (p. 1761)	Amazon RDS for PostgreSQL enables you to restrict who can manage user passwords and password expiration changes by using the parameter <code>rds.restrict_password_commands</code> and the role <code>rds_password</code> . For more information, see Restricting password management .	December 19, 2018
Amazon RDS for PostgreSQL supports uploading database logs to Amazon CloudWatch Logs (p. 1761)	Amazon RDS for PostgreSQL supports uploading database logs to CloudWatch Logs. For more information, see Publishing PostgreSQL logs to CloudWatch Logs .	December 10, 2018
Amazon RDS for Oracle supports db.r5 DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running Oracle that use the db.r5 DB instance classes. For more information, see DB instance class .	November 20, 2018
Retain backups when deleting a DB instance (p. 1761)	Amazon RDS supports retaining automated backups when you delete a DB instance. For more information, see Working with backups .	November 15, 2018
Amazon RDS for PostgreSQL supports db.m5 DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running PostgreSQL that use the db.m5 DB instance classes. For more information, see DB instance class .	November 15, 2018
Amazon RDS for Oracle supports a new major version (p. 1761)	You can now create Amazon RDS DB instances running Oracle version 12.2. For more information, see Oracle Database 12c Release 2 (12.2.0.1) with Amazon RDS .	November 13, 2018
Amazon RDS for Oracle October 2018 PSU (p. 1761)	Amazon RDS for Oracle has released database engine versions 12.1.0.2.v14 and 11.2.0.4.v18 to support the October 2018 Oracle Database Patch Set Update (PSU). For more information, see Oracle database engine release notes .	November 13, 2018
Amazon RDS for SQL Server supports Always On (p. 1761)	Amazon RDS for SQL Server supports Always On Availability Groups. For more information, see Multi-AZ deployments for Microsoft SQL Server .	November 8, 2018

Amazon RDS for PostgreSQL supports outbound network access using custom DNS servers (p. 1761)	You can now enable extended data types on Amazon RDS DB instances running Oracle. With extended data types, the maximum size is 32,767 bytes for the VARCHAR2, NVARCHAR2, and RAW data types. For more information, see Using extended data types .	November 8, 2018
Amazon RDS for MariaDB, MySQL, and PostgreSQL supports 32 TiB of storage (p. 1761)	You can now create Amazon RDS DB instances with up to 32 TiB of storage for MySQL, MariaDB, and PostgreSQL. For more information, see DB instance storage .	November 7, 2018
Amazon RDS for Oracle supports extended data types (p. 1761)	You can now enable extended data types on Amazon RDS DB instances running Oracle. With extended data types, the maximum size is 32,767 bytes for the VARCHAR2, NVARCHAR2, and RAW data types. For more information, see Using extended data types .	November 6, 2018
Amazon RDS for Oracle supports db.m5 DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running Oracle that use the db.m5 DB instance classes. For more information, see DB instance class .	November 2, 2018
Amazon RDS for Oracle migration from SE, SE1, or SE2 to EE (p. 1761)	You can now migrate from any Oracle Database Standard Edition (SE, SE1, or SE2) to Oracle Database Enterprise Edition (EE). For more information, see Migrating between Oracle editions .	October 31, 2018
Amazon RDS can now stop Multi-AZ instances (p. 1761)	Amazon RDS can now stop a DB instance that is part of a Multi-AZ deployment. Formerly, the stop instance feature had a limitation for multi-AZ instances. For more information, see Stopping an Amazon RDS DB instance temporarily .	October 29, 2018
Amazon RDS Performance Insights is available for Amazon RDS for Oracle (p. 1761)	Amazon RDS Performance Insights is now available for Amazon RDS for Oracle. For more information, see Using Amazon RDS Performance Insights .	October 29, 2018

Amazon RDS for PostgreSQL supports PostgreSQL version 11 in the database preview environment (p. 1761)	Amazon RDS for PostgreSQL now supports PostgreSQL version 11 in the Database Preview Environment. For more information, see PostgreSQL version 11 on Amazon RDS in the database preview environment .	October 25, 2018
MySQL supports a new major version (p. 1761)	You can now create Amazon RDS DB instances running MySQL version 8.0. For more information, see MySQL on Amazon RDS versions .	October 23, 2018
MariaDB supports a new major version (p. 1761)	You can now create Amazon RDS DB instances running MariaDB version 10.3. For more information, see MariaDB on Amazon RDS versions .	October 23, 2018
Amazon RDS for Oracle supports Oracle JVM (p. 1761)	Amazon RDS for Oracle now supports the Oracle Java Virtual Machine (JVM) option. For more information, see Oracle Java virtual machine .	October 16, 2018
Custom parameter group for restore and point in time recovery (p. 1761)	You can now specify a custom parameter group when you restore a snapshot or perform a point in time recovery operation. For more information, see Restoring from a DB snapshot and Restoring a DB instance to a specified time .	October 15, 2018
Amazon RDS for Oracle supports 32 TiB storage (p. 1761)	You can now create Oracle RDS DB instances with up to 32 TiB of storage. For more information, see DB instance storage .	October 15, 2018
Amazon RDS for MySQL supports GTIDs (p. 1761)	Amazon RDS for MySQL now supports global transaction identifiers (GTIDs), which are unique across all DB instances and in a replication configuration. For more information, see Using GTID-based replication for RDS for MySQL .	October 10, 2018
MySQL 5.7.23, 5.6.41, and 5.5.61 (p. 1761)	You can now create Amazon RDS DB instances running MySQL versions 5.7.23, 5.6.41, and 5.5.61. For more information, see MySQL on Amazon RDS versions .	October 8, 2018

Amazon RDS for PostgreSQL supports new minor versions (p. 1761)	Amazon RDS for PostgreSQL now supports the following new minor versions: 10.5, 9.6.10, 9.5.14, 9.4.19, and 9.3.24. For more information, see Amazon RDS for PostgreSQL versions and extensions .	October 4, 2018
Amazon RDS for Oracle supports a new version of SQLT (p. 1761)	Amazon RDS for Oracle now supports SQLT version 12.2.180331. For more information, see Oracle SQLT .	October 4, 2018
Amazon RDS for Oracle July 2018 PSU (p. 1761)	Amazon RDS for Oracle has released database engine versions 12.1.0.2.v13 and 11.2.0.4.v17 to support the July 2018 Oracle Database Patch Set Update (PSU). For more information, see Oracle database engine release notes .	October 3, 2018
Amazon RDS for PostgreSQL now supports IAM authentication (p. 1761)	Amazon RDS for PostgreSQL now supports IAM authentication. For more information see IAM database authentication for MySQL and PostgreSQL .	September 27, 2018
You can enable deletion protection for your Amazon RDS DB instances (p. 1761)	When you enable deletion protection for a DB instance, the database cannot be deleted by any user. For more information, see Deleting a DB instance .	September 26, 2018
Amazon RDS for MySQL and Amazon RDS for MariaDB support db.m5 DB instance classes (p. 1761)	You can now create Amazon RDS DB instances running MySQL or MariaDB that use the db.m5 DB instance classes. For more information, see DB instance class .	September 18, 2018
Amazon RDS now supports upgrades to SQL Server 2017 (p. 1761)	You can upgrade your existing DB instance to SQL Server 2017 from any version except SQL Server 2008. To upgrade from SQL Server 2008, first upgrade to one of the other versions first. For information, see Upgrading the Microsoft SQL Server DB engine .	September 11, 2018

Amazon RDS for PostgreSQL now supports PostgreSQL version 11 beta 3 in the database preview environment (p. 1761)	In this release, the Write-Ahead Log (WAL) segment size (<code>wal_segment_size</code>) is now set to 64MB. For more about PostgreSQL version 11 Beta 3, see PostgreSQL 11 beta 3 released . For information on the Database Preview Environment, see Working with the database preview environment .	September 7, 2018
Amazon Aurora User Guide (p. 1761)	The Amazon Aurora User Guide describes all Amazon Aurora concepts and provides instructions on using the various features with both the console and the command line interface. The Amazon RDS User Guide now covers non-Aurora database engines.	August 31, 2018
Amazon RDS Performance Insights is available for RDS for MySQL (p. 1761)	Amazon RDS Performance Insights is now available for RDS for MySQL. For more information, see Using Amazon RDS Performance Insights .	August 28, 2018
Aurora PostgreSQL-Compatible Edition now supports Aurora Auto Scaling (p. 1761)	Auto Scaling of Aurora replicas is now available for Aurora PostgreSQL-Compatible Edition. For more information, see Using Amazon Aurora auto scaling with Aurora replicas .	August 16, 2018
Aurora Serverless for Aurora MySQL (p. 1761)	Aurora Serverless is an on-demand, autoscaling configuration for Amazon Aurora. For more information, see Using Amazon Aurora Serverless .	August 9, 2018
MySQL 5.7.22 and 5.6.40 (p. 1761)	You can now create Amazon RDS DB instances running MySQL versions 5.7.22 and 5.6.40. For more information, see MySQL on Amazon RDS versions .	August 6, 2018
Aurora is now available in the China (Ningxia) region (p. 1761)	Aurora MySQL and Aurora PostgreSQL are now available in the China (Ningxia) region. For more information, see Availability for Amazon Aurora MySQL and Availability for Amazon Aurora PostgreSQL .	August 6, 2018

Amazon RDS for MySQL supports delayed replication (p. 1761)	Amazon RDS for MySQL now supports delayed replication as a strategy for disaster recovery. For more information, see Configuring delayed replication with MySQL .	August 6, 2018
Amazon RDS Performance Insights is available for Aurora MySQL (p. 1761)	Amazon RDS Performance Insights is now available for Aurora MySQL. For more information, see Using Amazon RDS Performance Insights .	August 6, 2018
Amazon RDS Performance Insights integration with Amazon CloudWatch (p. 1761)	Amazon RDS Performance Insights automatically publishes metrics to Amazon CloudWatch. For more information, see Performance Insights metrics published to CloudWatch .	August 6, 2018
Amazon RDS recommendations (p. 1761)	Amazon RDS now provides automated recommendations for database resources. For more information, see Using Amazon RDS recommendations .	July 25, 2018
Amazon RDS for PostgreSQL supports new minor versions (p. 1761)	Amazon RDS for PostgreSQL now supports the following new minor versions: 10.4, 9.6.9, 9.5.13, 9.4.18, and 9.3.23. For more information, see Amazon RDS for PostgreSQL versions and extensions .	July 25, 2018
Incremental snapshot copies across AWS Regions (p. 1761)	Amazon RDS supports incremental snapshot copies across AWS Regions for both unencrypted and encrypted instances. For more information, see Copying snapshots across AWS Regions .	July 24, 2018
Amazon RDS Performance Insights is available for Amazon RDS for PostgreSQL (p. 1761)	Amazon RDS Performance Insights is now available for Amazon RDS for PostgreSQL. For more information, see Using Amazon RDS Performance Insights .	July 18, 2018
Amazon RDS for Oracle supports Oracle APEX version 5.1.4.v1 (p. 1761)	Amazon RDS for Oracle now supports Oracle Application Express (APEX) version 5.1.4.v1. For more information, see Oracle application Express .	July 10, 2018

Amazon RDS for Oracle supports publishing logs to Amazon CloudWatch Logs (p. 1761)	Amazon RDS for Oracle now supports publishing alert, audit, trace, and listener log data to a log group in CloudWatch Logs. For more information, see Publishing Oracle logs to Amazon CloudWatch Logs .	July 9, 2018
MariaDB 10.2.15, 10.1.34, and 10.0.35 (p. 1761)	You can now create Amazon RDS DB instances running MariaDB versions 10.2.15, 10.1.34, and 10.0.35. For more information, see MariaDB on Amazon RDS versions .	July 5, 2018
Aurora PostgreSQL 1.2 is available and compatible with PostgreSQL 9.6.8 (p. 1761)	Aurora PostgreSQL 1.2 is now available and is compatible with PostgreSQL 9.6.8. For more information, see Version 1.2 .	June 27, 2018
Read replicas for Amazon RDS PostgreSQL support Multi-AZ deployments (p. 1761)	RDS read replicas in Amazon RDS PostgreSQL now support multiple Availability Zones. For more information, see Working with PostgreSQL read replicas .	June 25, 2018
Performance Insights available for Aurora PostgreSQL (p. 1761)	Performance Insights is generally available for Aurora PostgreSQL, with support for extended retention of performance data. For more information, see Using Amazon RDS performance insights .	June 21, 2018
Aurora PostgreSQL available in western US (northern California) region (p. 1761)	Aurora PostgreSQL is now available in the western United States (Northern California) region. For more information, see Availability for Amazon Aurora PostgreSQL .	June 11, 2018
Amazon RDS for Oracle now supports CPU configuration (p. 1761)	Amazon RDS for Oracle supports configuring the number of CPU cores and the number of threads for each core for the processor of a DB instance class. For more information, see Configuring the processor of the DB instance class .	June 5, 2018
Amazon RDS for Oracle April 2018 PSU (p. 1761)	Amazon RDS for Oracle has released database engine versions 12.1.0.2.v12 and 11.2.0.4.v16 to support the April 2018 Oracle Database Patch Set Update (PSU). For more information, see Oracle database engine release notes .	June 1, 2018

Earlier updates

The following table describes the important changes in each release of the *Amazon RDS User Guide* before June 2018.

Change	Description	Date changed
Amazon RDS for PostgreSQL now supports PostgreSQL Version 11 Beta 1 in the Database Preview Environment	<p>PostgreSQL version 11 Beta 1 contains several improvements that are described in PostgreSQL 11 beta 1 released!</p> <p>For information on the Database Preview Environment, see Working with the database preview environment (p. 1457).</p>	May 31, 2018
Amazon RDS for Oracle now supports TLS versions 1.0 and 1.2	Amazon RDS for Oracle supports Transport Layer Security (TLS) versions 1.0 and 1.2. For more information, see TLS versions for the Oracle SSL option (p. 1182) .	May 30, 2018
Aurora MySQL supports publishing logs to Amazon CloudWatch Logs	Aurora MySQL now supports publishing general, slow, audit, and error log data to a log group in CloudWatch Logs. For more information, see Publishing Aurora MySQL to CloudWatch Logs .	May 23, 2018
Database Preview Environment for Amazon RDS PostgreSQL	You can now launch a new instance of Amazon RDS PostgreSQL in a preview mode. For more information about the Database Preview Environment see, Working with the database preview environment (p. 1457) .	May 22, 2018
Amazon RDS for Oracle DB instances support new DB instance classes	Oracle DB instances now support the db.x1e and db.x1 DB instance classes. For more information, see DB instance classes (p. 7) and RDS for Oracle instance classes (p. 992) .	May 22, 2018
Amazon RDS PostgreSQL now supports <code>postgres_fdw</code> on a read replica.	You can now use <code>postgres_fdw</code> to connect to a remote server from a read replica. For more information see, Accessing external data with the postgres_fdw extension (p. 1592) .	May 17, 2018
Amazon RDS for Oracle now supports setting <code>sqlnet.ora</code> parameters	You can now set <code>sqlnet.ora</code> parameters with Amazon RDS for Oracle. For more information, see Modifying connection properties using sqlnet.ora parameters (p. 1007) .	May 10, 2018
Aurora PostgreSQL available in Asia Pacific (Seoul) region.	Aurora PostgreSQL is now available in the Asia Pacific (Seoul) region. For more information, see Availability for Amazon Aurora PostgreSQL .	May 9, 2018
Aurora MySQL supports backtracking	Aurora MySQL now supports "rewinding" a DB cluster to a specific time, without restoring data from a backup. For more information, see Backtracking an Aurora DB cluster .	May 9, 2018
Aurora MySQL supports encrypted	Aurora MySQL now supports encrypted migration and replication from an external MySQL database. For more	April 25, 2018

Change	Description	Date changed
migration and replication from external MySQL	information, see Migrating data from an external MySQL database to an Amazon Aurora MySQL DB cluster and Replication between Aurora and MySQL or between Aurora and another Aurora DB cluster .	
Aurora PostgreSQL-Compatible Edition support for the Copy-on-Write protocol.	You can now clone databases in an Aurora PostgreSQL database cluster. For more information see, Cloning databases in an Aurora DB cluster .	April 10, 2018
MariaDB 10.2.12, 10.1.31, and 10.0.34	You can now create Amazon RDS DB instances running MariaDB versions 10.2.12, 10.1.31, and 10.0.34. For more information, see MariaDB on Amazon RDS versions (p. 576) .	March 21, 2018
Aurora PostgreSQL Support for new regions	Aurora PostgreSQL is now available in the EU (London) and Asia Pacific (Singapore) regions. For more information, see Availability for Amazon Aurora PostgreSQL .	March 13, 2018
MySQL 5.7.21, 5.6.39, and 5.5.59	You can now create Amazon RDS DB instances running MySQL versions 5.7.21, 5.6.39, and 5.5.59. For more information, see MySQL on Amazon RDS versions (p. 828) .	March 9, 2018
Amazon RDS for Oracle now supports Oracle REST Data Services	Amazon RDS for Oracle supports Oracle REST Data Services as part of the APEX option. For more information, see Oracle Application Express (APEX) (p. 1140) .	March 9, 2018
Amazon Aurora MySQL-Compatible Edition available in new AWS Region	Aurora MySQL is now available in the Asia Pacific (Singapore) region. For the complete list of AWS Regions for Aurora MySQL, see Availability for Amazon Aurora MySQL .	March 6, 2018
Support for PostgreSQL 10.1	Amazon RDS now supports version 10.1 of PostgreSQL. For more information, see PostgreSQL version 10.1 on Amazon RDS (p. 1470)	February 27, 2018
Oracle January 2018 PSU	Amazon RDS for Oracle has released database engine versions 12.1.0.2.v11 and 11.2.0.4.v15 to support the January 2018 Oracle Database Patch Set Update (PSU). For more information, see Oracle database engine release notes (p. 1245) .	February 22, 2018
Amazon RDS DB instances running Microsoft SQL Server support change data capture (CDC)	DB instances running Amazon RDS for Microsoft SQL Server now support change data capture (CDC). For more information, see Change data capture support for Microsoft SQL Server DB instances (p. 642) .	February 6, 2018
Aurora MySQL supports a new major version	You can now create Aurora MySQL DB clusters running MySQL version 5.7. For more information, see Amazon Aurora MySQL database engine updates 2018-02-06 .	February 6, 2018

Change	Description	Date changed
Support for PostgreSQL 9.6.6	Amazon RDS PostgreSQL now supports version 9.6.6. This release also includes support for the prefix and orafce extensions. For more information, see PostgreSQL version 9.6.6 on Amazon RDS (p. 1474) .	January 19, 2018
Publish MySQL and MariaDB logs to Amazon CloudWatch Logs	You can now publish MySQL and MariaDB log data to CloudWatch Logs. For more information, see Publishing MySQL logs to Amazon CloudWatch Logs (p. 521) and Publishing MariaDB logs to Amazon CloudWatch Logs (p. 509) .	January 17, 2018
Multi-AZ support for read replicas	You can now create a read replica as a Multi-AZ DB instance. Amazon RDS creates a standby of your replica in another Availability Zone for failover support for the replica. Creating your read replica as a Multi-AZ DB instance is independent of whether the source database is a Multi-AZ DB instance. For more information, see Working with read replicas (p. 278) .	January 11, 2018
Amazon RDS for MariaDB supports a new major version	You can now create Amazon RDS DB instances running MariaDB version 10.2. For more information, see MariaDB 10.2 support on Amazon RDS (p. 579) .	January 3, 2018
Amazon Aurora PostgreSQL-Compatible Edition available in new AWS Region	Aurora PostgreSQL is now available in the EU (Paris) region. For the complete list of AWS Regions for Aurora PostgreSQL, see Availability for Amazon Aurora PostgreSQL .	December 22, 2017
Aurora PostgreSQL supports new instance types	Aurora PostgreSQL now supports new instance types. For the complete list of instance types, see Choosing the DB instance class .	December 20, 2017
Oracle October 2017 PSU	Amazon RDS for Oracle has released database engine versions 12.1.0.2.v10 and 11.2.0.4.v14 to support the October 2017 Oracle Database Patch Set Update (PSU). For more information, see Oracle database engine release notes (p. 1245) .	December 19, 2017
Amazon Aurora MySQL-Compatible Edition available in new AWS Region	Aurora MySQL is now available in the EU (Paris) region. For the complete list of AWS Regions for Aurora MySQL, see Availability for Amazon Aurora MySQL .	December 18, 2017
Aurora MySQL supports hash joins	This feature can improve query performance when you need to join a large amount of data by using an equijoin. For more information, see Working with hash joins in Aurora MySQL .	December 11, 2017
Aurora MySQL supports native functions to invoke AWS Lambda functions	You can call the native functions <code>lambda_sync</code> and <code>lambda_async</code> when you use Aurora MySQL. For more information, see Invoking a Lambda function from an Amazon Aurora MySQL DB cluster .	December 11, 2017

Change	Description	Date changed
Added Aurora PostgreSQL HIPAA eligibility	Aurora PostgreSQL now supports building HIPAA compliant applications. For more information, see Working with Amazon Aurora PostgreSQL .	December 6, 2017
Additional AWS Regions available for Amazon Aurora with PostgreSQL compatibility	Amazon Aurora with PostgreSQL compatibility is now available in four new AWS Regions. For more information, see Availability for Amazon Aurora PostgreSQL .	November 22, 2017
Modify storage for Amazon RDS DB instances running Microsoft SQL Server	You can now modify the storage of your Amazon RDS DB instances running SQL Server. For more information, see Modifying an Amazon RDS DB instance (p. 250) .	November 21, 2017
Amazon RDS supports 16 TiB storage for Linux-based engines	You can now create MySQL, MariaDB, PostgreSQL, and Oracle RDS DB instances with up to 16 TiB of storage. For more information, see Amazon RDS DB instance storage (p. 40) .	November 21, 2017
Amazon RDS supports fast scale up of storage	You can now add storage to MySQL, MariaDB, PostgreSQL, and Oracle RDS DB instances in a few minutes. For more information, see Amazon RDS DB instance storage (p. 40) .	November 21, 2017
Amazon RDS supports MariaDB versions 10.1.26 and 10.0.32	You can now create Amazon RDS DB instances running MariaDB versions 10.1.26 and 10.0.32. For more information, see MariaDB on Amazon RDS versions (p. 576) .	November 20, 2017
Amazon RDS for Microsoft SQL Server now supports new DB instance classes	You can now create Amazon RDS DB instances running SQL Server that use the db.r4 and db.m4.16xlarge DB instance classes. For more information, see DB instance class support for Microsoft SQL Server (p. 634) .	November 20, 2017
Amazon RDS for MySQL and MariaDB now supports new DB instance classes	You can now create Amazon RDS DB instances running MySQL and MariaDB that use the db.r4, db.m4.16xlarge, db.t2.xlarge, and db.t2.2xlarge DB instance classes. For more information, see DB instance classes (p. 7) .	November 20, 2017
SQL Server 2017	You can now create Amazon RDS DB instances running Microsoft SQL Server 2017. You can also create DB instances running SQL Server 2016 SP1 CU5. For more information, see Microsoft SQL Server on Amazon RDS (p. 630) .	November 17, 2017
Restore MySQL backups from Amazon S3	You can now create a backup of your on-premises database, store it on Amazon S3, and then restore the backup file onto a new Amazon RDS DB instance running MySQL. For more information, see Restoring a backup into a MySQL DB instance (p. 871) .	November 17, 2017

Change	Description	Date changed
Auto Scaling with Aurora Replicas	Amazon Aurora MySQL now supports Aurora Auto Scaling. Aurora Auto Scaling dynamically adjusts the number of Aurora Replicas based on increases or decreases in connectivity or workload. For more information, see Using Amazon Aurora Auto Scaling with Aurora replicas .	November 17, 2017
Oracle default edition support	Amazon RDS for Oracle DB instances now supports setting the default edition for the DB instance. For more information, see Setting the default edition for a DB instance (p. 1057) .	November 3, 2017
Oracle DB instance file validation	Amazon RDS for Oracle DB instances now supports validating DB instance files with the Oracle Recovery Manager (RMAN) logical validation utility. For more information, see Validating DB instance files (p. 1072) .	November 3, 2017
Oracle July 2017 PSU	Amazon RDS for Oracle has released database engine versions 12.1.0.2.v9 and 11.2.0.4.v13 to support the July 2017 Oracle Database Patch Set Update (PSU). For more information, see Oracle database engine release notes (p. 1245) .	November 3, 2017
Management Agent for OEM 13c	Amazon RDS for Oracle DB instances now support the Management Agent for Oracle Enterprise Manager (OEM) Cloud Control 13c. For more information, see Oracle Management Agent for Enterprise Manager Cloud Control (p. 1154) .	November 1, 2017
PostgreSQL 9.6.5, 9.5.9, 9.4.14, and 9.3.19	You can now create Amazon RDS DB instances running PostgreSQL versions 9.6.5., 9.5.9, 9.4.14, and 9.3.19. For more information, see Supported PostgreSQL database versions (p. 1461) .	November 1, 2017
Storage reconfiguration for Microsoft SQL Server snapshots	You can now reconfigure the storage when you restore a snapshot to an Amazon RDS DB instance running Microsoft SQL Server. For more information, see Restoring from a DB snapshot (p. 349) .	October 26, 2017
Asynchronous key prefetch for Aurora MySQL-Compatible Edition	Asynchronous key prefetch (AKP) improves the performance of noncached index joins, by prefetching keys in memory ahead of when they are needed. For more information, see Working with asynchronous key prefetch in Amazon Aurora .	October 26, 2017
MySQL 5.7.19, 5.6.37, and 5.5.57	You can now create Amazon RDS DB instances running MySQL versions 5.7.19, 5.6.37, and 5.5.57. For more information, see MySQL on Amazon RDS versions (p. 828) .	October 25, 2017
General availability of Amazon Aurora with PostgreSQL compatibility	Amazon Aurora with PostgreSQL compatibility makes it simple and cost-effective to set up, operate, and scale your new and existing PostgreSQL deployments, thus freeing you to focus on your business and applications. For more information, see Working with Amazon Aurora PostgreSQL .	October 24, 2017

Change	Description	Date changed
Amazon RDS for Oracle DB instances support new DB instance classes	Amazon RDS for Oracle DB instances now support Memory Optimized Next Generation (db.r4) instance classes. Amazon RDS for Oracle DB instances also now support the following new current generation instance classes: db.m4.16xlarge, db.t2.xlarge, and db.t2.2xlarge. For more information, see DB instance classes (p. 7) and RDS for Oracle instance classes (p. 992) .	October 23, 2017
New feature	Your new and existing Reserved Instances can now cover multiple sizes in the same DB instance class. Size-flexible reserved instances are available for DB instances with the same AWS Region, database engine, and instance family, and across AZ configuration. Size-flexible reserved instances are available for the following database engines: Amazon Aurora, MariaDB, MySQL, Oracle (Bring Your Own License), PostgreSQL. For more information, see Size-flexible reserved DB instances (p. 59) .	October 11, 2017
New feature	You can now use the Oracle SQLT option to tune a SQL statement for optimal performance. For more information, see Oracle SQLT (p. 1193) .	September 22, 2017
New feature	If you have existing manual DB snapshots of your Amazon RDS for Oracle DB instances, you can now upgrade them to a later version of the Oracle database engine. For more information, see Upgrading an Oracle DB snapshot (p. 1217) .	September 20, 2017
New feature	You can now use Oracle Spatial to store, retrieve, update, and query spatial data in your Amazon RDS DB instances running Oracle. For more information, see Oracle Spatial (p. 1190) .	September 15, 2017
New feature	You can now use Oracle Locator to support internet and wireless service-based applications and partner-based GIS solutions with your Amazon RDS DB instances running Oracle. For more information, see Oracle Locator (p. 1170) .	September 15, 2017
New feature	You can now use Oracle Multimedia to store, manage, and retrieve images, audio, video, and other heterogeneous media data in your Amazon RDS DB instances running Oracle. For more information, see Oracle Multimedia (p. 1173) .	September 15, 2017
New feature	You can now export audit logs from your Amazon Aurora MySQL DB clusters to Amazon CloudWatch Logs. For more information, see Publishing Aurora MySQL logs to Amazon CloudWatch Logs .	September 14, 2017
New feature	Amazon RDS now supports multiple versions of Oracle Application Express (APEX) for your DB instances running Oracle. For more information, see Oracle Application Express (APEX) (p. 1140) .	September 13, 2017

Change	Description	Date changed
New feature	You can now use Amazon Aurora to migrate an unencrypted or encrypted DB snapshot or MySQL DB instance to an encrypted Aurora MySQL DB cluster. For more information, see Migrating an RDS for MySQL snapshot to Aurora and Migrating data from a MySQL DB instance to an Amazon Aurora MySQL DB cluster by using an Aurora read replica .	September 5, 2017
New feature	You can use Amazon RDS for Microsoft SQL Server databases to build HIPAA-compliant applications. For more information, see Compliance program support for Microsoft SQL Server DB instances (p. 636) .	August 31, 2017
New feature	You can now use Amazon RDS for MariaDB databases to build HIPAA-compliant applications. For more information, see MariaDB on Amazon RDS (p. 574) .	August 31, 2017
New feature	You can now create Amazon RDS DB instances running Microsoft SQL Server with allocated storage up to 16 TiB, and Provisioned IOPS to storage ranges of 1:1–50:1. For more information, see Amazon RDS DB instance storage (p. 40) .	August 22, 2017
New feature	You can now use Multi-AZ deployments for DB instances running Microsoft SQL Server in the EU (Frankfurt) region. For more information, see Multi-AZ deployments for Microsoft SQL Server (p. 698) .	August 3, 2017
New feature	You can now create Amazon RDS DB instances running MariaDB versions 10.1.23 and 10.0.31. For more information, see MariaDB on Amazon RDS versions (p. 576) .	July 17, 2017
New feature	Amazon RDS now supports Microsoft SQL Server Enterprise Edition with the License Included model in all AWS Regions. For more information, see Licensing Microsoft SQL Server on Amazon RDS (p. 655) .	July 13, 2017
New feature	Amazon RDS for Oracle now supports Linux kernel huge pages for increased database scalability. The use of huge pages results in smaller page tables and less CPU time spent on memory management, increasing the performance of large database instances. You can use huge pages with your Amazon RDS DB instances running all editions of Oracle versions 12.1.0.2 and 11.2.0.4. For more information, see Enabling HugePages for an Oracle DB instance (p. 1101) .	July 7, 2017
New feature	Updated to support encryption at rest (EAR) for db.t2.small and db.t2.medium DB instance classes for all non-Aurora DB engines. For more information, see Availability of Amazon RDS encryption (p. 1632) .	June 27, 2017
New feature	Updated to support Amazon Aurora in the Europe (Frankfurt) region. For more information, see Availability for Amazon Aurora MySQL .	June 16, 2017

Change	Description	Date changed
New feature	You can now specify an option group when you copy a DB snapshot across AWS regions. For more information, see Option group considerations (p. 356) .	June 12, 2017
New feature	You can now copy DB snapshots created from specialized DB instances across AWS regions. You can copy snapshots from DB instances that use Oracle TDE, Microsoft SQL Server TDE, and Microsoft SQL Server Multi-AZ with Mirroring. For more information, see Copying a DB snapshot (p. 357) .	June 12, 2017
New feature	Amazon Aurora now allows you to quickly and cost-effectively copy all of your databases in an Amazon Aurora DB cluster. For more information, see Cloning databases in an Aurora DB cluster .	June 12, 2017
New feature	Amazon RDS now supports Microsoft SQL Server 2016 SP1 CU2. For more information, see Microsoft SQL Server on Amazon RDS (p. 630) .	June 7, 2017
New feature	Amazon RDS for Oracle has released database engine versions 12.1.0.2.v8 and 11.2.0.4.v12 to support the April 2017 Oracle Database Patch Set Update (PSU). For more information, see Oracle database engine release notes (p. 1245) .	May 23, 2017
New Feature	Amazon RDS now supports PostgreSQL versions 9.6.2, 9.5.6, 9.4.11, and 9.3.16. For more information, see Supported PostgreSQL database versions (p. 1461)	May 3, 2017
Preview	Public preview of Amazon Aurora with PostgreSQL Compatibility. For more information, see Working with Amazon Aurora PostgreSQL .	April 19, 2017
New feature	Amazon Aurora now allows you to run an ALTER TABLE <i>tbl_name</i> ADD COLUMN <i>col_name column_definition</i> operation nearly instantaneously. The operation completes without requiring the table to be copied and without materially impacting other DML statements. For more information, see Altering tables in Amazon Aurora using fast DDL .	April 5, 2017
New feature	We have added a new monitoring command, SHOW VOLUME STATUS, to display the number of nodes and disks in a volume. For more information, see Displaying volume status for an Aurora DB cluster .	April 5, 2017
New feature	Amazon RDS for Oracle now includes the January 2017 Oracle Database Patch Set Update (PSU). This adds support for database engine versions 12.1.0.2.v7 and 11.2.0.4.v11. For more information, see Oracle database engine release notes (p. 1245) .	March 21, 2017
New feature	You can now use your own custom logic in your custom password verification functions for Oracle on Amazon RDS. For more information, see Creating custom functions to verify passwords (p. 1042) .	March 21, 2017

Change	Description	Date changed
New feature	You can now access your online and archived redo log files on your Oracle DB instances on Amazon RDS. For more information, see Accessing transaction logs (p. 1068) .	March 21, 2017
New feature	You can now copy both encrypted and unencrypted DB cluster snapshots between accounts in the same region. For more information, see Copying a DB cluster snapshot across accounts .	March 7, 2017
New feature	You can now share encrypted DB cluster snapshots between accounts in the same region. For more information, see Sharing a DB cluster snapshot .	March 7, 2017
New feature	You can now replicate encrypted Amazon Aurora MySQL DB clusters to create cross-region Aurora Replicas. For more information, see Replicating Aurora MySQL DB clusters across AWS Regions .	March 7, 2017
New feature	You can now require that all connections to your DB instance running Microsoft SQL Server use Secure Sockets Layer (SSL). For more information, see Using SSL with a Microsoft SQL Server DB instance (p. 704) .	February 27, 2017
New feature	You can now set your local time zone to one of 15 additional time zones. For more information, see Supported time zones (p. 647) .	February 27, 2017
New feature	You can now use the Amazon RDS procedure <code>msdb.dbo.rds_shrink_tempdbfile</code> to shrink the tempdb database on your DB instances running Microsoft SQL Server. For more information, see Shrinking the tempdb database (p. 810) .	February 17, 2017
New feature	You can now compress your backup file when you export your Enterprise and Standard Edition Microsoft SQL Server database from an Amazon RDS DB instance to Amazon S3. For more information, see Compressing backup files (p. 685) .	February 17, 2017
New feature	Amazon RDS now supports custom DNS servers to resolve DNS names used in outbound network access on your DB instances running Oracle. For more information, see Setting up a custom DNS server (p. 1045) .	January 26, 2017
New feature	Amazon RDS now supports creating an encrypted read replica in another region. For more information, see Creating a read replica in a different AWS Region (p. 290) and CreateDBInstanceReadReplica .	January 23, 2017
New feature	Amazon RDS now supports upgrading a MySQL DB snapshot from MySQL 5.1 to MySQL 5.5. For more information, see Upgrading a MySQL DB snapshot (p. 863) and ModifyDBSnapshot .	January 20, 2017

Change	Description	Date changed
New feature	Amazon RDS now supports copying an encrypted DB snapshot to another region for the MariaDB, MySQL, Oracle, PostgreSQL, and Microsoft SQL Server database engines. For more information, see Copying a DB snapshot (p. 357) and CopyDBSnapshot .	December 20, 2016
New feature	Amazon RDS now supports migrating an RDS for MySQL 5.6 DB snapshot to a new DB instance running MariaDB 10.1. For more information, see Migrating data from a MySQL DB snapshot to a MariaDB DB instance (p. 603) .	December 20, 2016
New feature	Amazon Aurora MySQL now supports spatial indexing. Spatial indexing improves query performance on large datasets for queries that use spatial data. For more information, see Amazon Aurora MySQL and spatial data .	December 14, 2016
New feature	Amazon RDS for Oracle now includes the October 2016 Oracle Database Patch Set Update (PSU). This adds support for Oracle database engine versions 12.1.0.2.v6 and 11.2.0.4.v10. For more information, see Oracle database engine release notes (p. 1245) .	December 12, 2016
New feature	Amazon RDS now supports outbound network access on your DB instances running Oracle. You can use <code>utl_http</code> , <code>utl_tcp</code> , and <code>utl_smtp</code> to connect from your DB instance to the network. For more information, see Configuring outbound network access on your Oracle DB instance (p. 1025) .	December 5, 2016
New feature	Amazon RDS has retired support for MySQL version 5.1. However, you can restore existing MySQL 5.1 snapshots to a MySQL 5.5 instance. For more information, see Supported storage engines for MySQL on Amazon RDS (p. 832) .	November 15, 2016
New feature	Amazon RDS now supports PostgreSQL version 9.6.1. For more information, see PostgreSQL version 9.6.1 on Amazon RDS (p. 1475) .	November 11, 2016
New feature	Amazon RDS now supports Microsoft SQL Server 2016 RTM CU2. For more information, see Microsoft SQL Server on Amazon RDS (p. 630) .	November 4, 2016
New feature	Amazon RDS now supports major version upgrades for DB instances running Oracle. You can now upgrade your Oracle DB instances from 11g to 12c. For more information, see Upgrading the Oracle DB engine (p. 1209) .	November 2, 2016
New feature	You can now create DB instances running Microsoft SQL Server 2014 Enterprise Edition. Amazon RDS now supports SQL Server 2014 SP2 for all editions and all regions. For more information, see Microsoft SQL Server on Amazon RDS (p. 630) .	October 25, 2016

Change	Description	Date changed
New feature	Amazon Aurora MySQL now integrates with other AWS services: You can load text or XML data into a table from an Amazon S3 bucket, or invoke an AWS Lambda function from database code. For more information, see Integrating Aurora MySQL with other AWS services .	October 18, 2016
New feature	You can now access the tempdb database on your Amazon RDS DB instances running Microsoft SQL Server. You can access the tempdb database by using Transact-SQL through Microsoft SQL Server Management Studio (SSMS), or any other standard SQL client application. For more information, see Accessing the tempdb database on Microsoft SQL Server DB instances on Amazon RDS (p. 810) .	September 29, 2016
New feature	You can now use the UTL_MAIL package with your Amazon RDS DB instances running Oracle. For more information, see Oracle UTL_MAIL (p. 1206) .	September 20, 2016
New feature	Amazon RDS for Oracle now includes the July 2016 Oracle Database Patch Set Update (PSU). This adds support for Oracle database engine versions 12.1.0.2.v5, 12.1.0.1.v6, and 11.2.0.4.v9. For more information, see Oracle database engine release notes (p. 1245) .	September 20, 2016
New features	You can now set the time zone of your new Microsoft SQL Server DB instances to a local time zone, to match the time zone of your applications. For more information, see Local time zone for Microsoft SQL Server DB instances (p. 646) .	September 19, 2016
New features	Added support for new PostgreSQL versions 9.5.4, 9.4.9, and 9.3.14. Also added support for PostgreSQL logical replication, PostgreSQL event triggers, and RAM disk for the PostgreSQL stats_temp_directory. For more information, see Supported PostgreSQL database versions (p. 1461) , Logical replication for PostgreSQL on Amazon RDS (p. 1502) , Event triggers for PostgreSQL on Amazon RDS (p. 1504) , and RAM disk for the stats_temp_directory (p. 1506) .	September 14, 2016
New feature	You can now use the Oracle Label Security option to control access to individual table rows in your Amazon RDS DB instances running Oracle Database 12c. With Oracle Label Security, you can enforce regulatory compliance with a policy-based administration model, and ensure that an access to sensitive data is restricted to only users with the appropriate clearance level. For more information, see Oracle Label Security (p. 1167) .	September 8, 2016

Change	Description	Date changed
New feature	You can now connect to an Amazon Aurora DB cluster using the reader endpoint, which load-balances connections across the Aurora Replicas that are available in the DB cluster. As clients request new connections to the reader endpoint, Aurora distributes the connection requests among the Aurora Replicas in the DB cluster. This functionality can help balance your read workload across multiple Aurora Replicas in your DB cluster. For more information, see Amazon Aurora endpoints .	September 8, 2016
New feature	You can now support the Oracle Enterprise Manager Cloud Control on your Amazon RDS DB instances running Oracle. You can enable the Management Agent on your DB instances, and share data with your Oracle Management Service (OMS). For more information, see Oracle Management Agent for Enterprise Manager Cloud Control (p. 1154) .	September 1, 2016
New feature	This release adds support to get an ARN for a resource. For more information, see Getting an existing ARN (p. 312) .	August 23, 2016
New feature	You can now assign up to 50 tags for each Amazon RDS resource, for managing your resources and tracking costs. For more information, see Tagging Amazon RDS resources (p. 299) .	August 19, 2016
New feature	Amazon RDS now supports the License Included model for Oracle Standard Edition Two. For more information, see Creating an Amazon RDS DB instance (p. 141) . You can now change the license model of your Amazon RDS DB instances running Microsoft SQL Server and Oracle. For more information, see Licensing Microsoft SQL Server on Amazon RDS (p. 655) and Oracle licensing options (p. 990) .	August 5, 2016
New feature	You can now use the AWS Management Console to easily move your DB instance to a different VPC, or to a different subnet group in the same VPC. For more information, see Updating the VPC for a DB instance (p. 1734) . If your DB instance is not in a VPC, you can now use the AWS Management Console to easily move your DB instance into a VPC. For more information, see Moving a DB instance not in a VPC into a VPC (p. 1735) .	August 4, 2016
New feature	Amazon RDS now supports native backup and restore for Microsoft SQL Server databases using full backup files (.bak files). You can now easily migrate SQL Server databases to Amazon RDS, and import and export databases in a single, easily-portable file, using Amazon S3 for storage, and AWS KMS for encryption. For more information, see Importing and exporting SQL Server databases (p. 671) .	July 27, 2016

Change	Description	Date changed
New feature	You can now copy the source files from a MySQL database to an Amazon Simple Storage Service (Amazon S3) bucket, and then restore an Amazon Aurora DB cluster from those files. This option can be considerably faster than migrating data using <code>mysqldump</code> . For more information, see Migrating data from an external MySQL database to an Aurora MySQL DB cluster .	July 20, 2016
New feature	You can now restore an unencrypted Amazon Aurora DB cluster snapshot to create an encrypted Amazon Aurora DB cluster by including an AWS Key Management Service (AWS KMS) encryption key during the restore operation. For more information, see Encrypting Amazon RDS resources .	June 30, 2016
New feature	Amazon RDS for Oracle now includes the April 2016 Oracle Database Patch Set Update (PSU). This PSU adds support for Oracle database engine versions 12.1.0.2.v4, 12.1.0.1.v5, and 11.2.0.4.v8. For more information, see Oracle database engine release notes (p. 1245) .	June 17, 2016
New feature	You can use the Oracle Repository Creation Utility (RCU) to create a repository on Amazon RDS for Oracle. For more information, see Using the Oracle Repository Creation Utility on Amazon RDS for Oracle (p. 1237) .	June 17, 2016
New feature	Adds support for PostgreSQL cross-region read replicas. For more information, see Creating a read replica in a different AWS Region (p. 290) .	June 16, 2016
New feature	You can now use the AWS Management Console to easily add Multi-AZ with Mirroring to a Microsoft SQL Server DB instance. For more information, see Adding Multi-AZ to a Microsoft SQL Server DB instance (p. 699) .	June 9, 2016
New feature	You can now use Multi-AZ Deployments Using SQL Server Mirroring in the following additional regions: Asia Pacific (Sydney), Asia Pacific (Tokyo), and South America (Sao Paulo). For more information, see Multi-AZ deployments for Microsoft SQL Server (p. 698) .	June 9, 2016
New feature	Updated to support MariaDB version 10.1. For more information, see MariaDB on Amazon RDS (p. 574) .	June 1, 2016
New feature	Updated to support Amazon Aurora cross-region DB clusters that are read replicas. For more information, see Replicating Aurora MySQL DB clusters across AWS Regions .	June 1, 2016
New feature	Enhanced Monitoring is now available for Oracle DB instances. For more information, see Using Enhanced Monitoring (p. 471) and Modifying an Amazon RDS DB instance (p. 250) .	May 27, 2016
New feature	Updated to support manual snapshot sharing for Amazon Aurora DB cluster snapshots. For more information, see Sharing a DB cluster snapshot .	May 18, 2016

Change	Description	Date changed
New feature	You can now use the MariaDB Audit Plugin to log database activity on MariaDB and MySQL database instances. For more information, see Options for MariaDB database engine (p. 616) and Options for MySQL DB instances (p. 925) .	April 27, 2016
New feature	In-place, major version upgrades are now available for upgrading from MySQL version 5.6 to version 5.7. For more information, see Upgrading the MySQL DB engine (p. 853) .	April 26, 2016
New feature	Enhanced Monitoring is now available for Microsoft SQL Server DB instances. For more information, see Using Enhanced Monitoring (p. 471) .	April 22, 2016
New feature	Added support for PostgreSQL versions 9.5.2, 9.4.7, and 9.3.12. For more information, see Supported PostgreSQL database versions (p. 1461) .	April 8, 2016
New feature	Updated to support Oracle database versions 11.2.0.4.v7, 12.1.0.1.v4, and 12.1.0.2.v3 with the January 2016 Oracle Patch Set Updates (PSU). For more information, see Oracle database engine release notes (p. 1245) .	April 1, 2016
New feature	Updated to provide an Amazon Aurora Clusters view in the Amazon RDS console. For more information, see Viewing an Aurora DB cluster .	April 1, 2016
New feature	Updated to support SQL Server Multi-AZ with mirroring in the Asia Pacific (Seoul) region. For more information, see Multi-AZ deployments for Microsoft SQL Server (p. 698) .	March 31, 2016
New feature	Updated to support Amazon Aurora Multi-AZ with mirroring in the Asia Pacific (Seoul) region. For more information, see Availability for Amazon Aurora MySQL .	March 31, 2016
New feature	PostgreSQL DB instances have the ability to require connections to use SSL. For more information, see Using SSL with a PostgreSQL DB instance (p. 1513) .	March 25, 2016
New feature	Enhanced Monitoring is now available for PostgreSQL DB instances. For more information, see Using Enhanced Monitoring (p. 471) .	March 25, 2016
New feature	Microsoft SQL Server DB instances can now use Windows Authentication for user authentication. For more information, see Using Windows Authentication with an Amazon RDS for SQL Server DB instance (p. 711) .	March 23, 2016
New feature	Enhanced Monitoring is now available in the Asia Pacific (Seoul) region. For more information, see Using Enhanced Monitoring (p. 471) .	March 16, 2016

Change	Description	Date changed
New feature	You can now customize the order in which Aurora Replicas are promoted to primary instance during a failover. For more information, see Fault tolerance for an Aurora DB cluster .	March 14, 2016
New feature	Updated to support encryption when migrating to an Aurora DB cluster. For more information, see Migrating data to an Aurora DB cluster .	March 2, 2016
New feature	Updated to support local time zone for Aurora DB clusters. For more information, see Local time zone for Aurora DB clusters .	March 1, 2016
New feature	Updated to add support for MySQL version 5.7 for current generation Amazon RDS DB instance classes.	February 22, 2016
New feature	Updated to support <i>db.r3</i> and <i>db.t2</i> DB instance classes in the AWS GovCloud (US-West) region.	February 11, 2016
New feature	Updated to support encrypting copies of DB snapshots and sharing encrypted DB snapshots. For more information, see Copying a snapshot (p. 352) and Sharing a DB snapshot (p. 365) .	February 11, 2016
New feature	Updated to support Amazon Aurora in the Asia Pacific (Sydney) region. For more information, see Availability for Amazon Aurora MySQL .	February 11, 2016
New feature	Updated to support SSL for Oracle DB Instances. For more information, see Encrypting client connections with SSL (p. 1010) .	February 9, 2016
New feature	Updated to support local time zone for MySQL and MariaDB DB instances. For more information, see Local time zone for MySQL DB instances (p. 838) and Local time zone for MariaDB DB instances (p. 587) .	December 21, 2015
New feature	Updated to support Enhanced Monitoring of OS metrics for MySQL and MariaDB instances and Aurora DB clusters. For more information, see Viewing DB instance metrics (p. 548) .	December 18, 2015
New feature	Updated to support Oracle Standard Edition Two with Bring-Your- Own-License licensing. Also added support for Oracle versions 11.2.0.4.v5, 12.1.0.1.v3, and 12.1.0.2.v2. For more information, see Oracle database engine release notes (p. 1245) .	December 14, 2015
New feature	Updated to support <i>db.t2</i> , <i>db.r3</i> , and <i>db.m4</i> DB instance classes for MySQL version 5.5. For more information, see DB instance classes (p. 7) .	December 4, 2015
New feature	Updated to support modifying the database port for an existing DB instance.	December 3, 2015

Change	Description	Date changed
New feature	Updated to support three new extensions for PostgreSQL versions 9.3.10 and 9.4.5 DB instances. For more information, see Supported PostgreSQL database versions (p. 1461) .	December 1, 2015
New feature	Updated to support PostgreSQL versions 9.3.10 and 9.4.5 DB instances. For more information, see Supported PostgreSQL database versions (p. 1461) .	November 27, 2015
New feature	Updated to support major version upgrades of the database engine for PostgreSQL instances. For more information, see Upgrading the PostgreSQL DB engine for Amazon RDS (p. 1533) .	November 19, 2015
New feature	Updated to support modifying the public accessibility of an existing DB instance. Updated to support db.m4 standard DB instance classes.	November 11, 2015
New feature	Updated to support manual DB snapshot sharing. For more information, see Sharing a DB snapshot (p. 365) .	October 28, 2015
New feature	Updated to support Microsoft SQL Server 2014 for the Web, Express, and Standard editions.	October 26, 2015
New feature	Updated to support the MySQL-based MariaDB database engine. For more information, see MariaDB on Amazon RDS (p. 574) .	October 7, 2015
New feature	Updated to support Amazon Aurora in the Asia Pacific (Tokyo) region. For more information, see Availability for Amazon Aurora MySQL .	October 7, 2015
New feature	Updated to support db.t2 burst-capable DB instance classes for all DB engines and the addition of the db.t2.large DB instance class. For more information, see DB instance classes (p. 7) .	September 25, 2015
New feature	Updated to support Oracle DB instances on R3 and T2 DB instance classes. For more information, see DB instance classes (p. 7) .	August 5, 2015
New feature	Updated to support PostgreSQL versions 9.4.4 and 9.3.9. For more information, see Supported PostgreSQL database versions (p. 1461) .	July 30, 2015
New feature	Microsoft SQL Server Enterprise Edition is now available with the License Included service model. For more information, see Licensing Microsoft SQL Server on Amazon RDS (p. 655) .	July 29, 2015
New feature	Amazon Aurora has officially released. The Amazon Aurora DB engine supports multiple DB instances in a DB cluster. For detailed information, see What is Amazon Aurora? .	July 27, 2015
New feature	Updated to support copying tags to DB snapshots.	July 20, 2015

Change	Description	Date changed
New feature	Updated to support Oracle Database 12c Release 1 (12.1.0.2), including the In-Memory option, Oracle Database 11g April PSU patches, and improved integration with AWS CloudHSM.	July 20, 2015
New feature	Updated to support increases in storage size for all DB engines and an increase in Provisioned IOPS for SQL Server.	June 18, 2015
New feature	Updated options for reserved DB instances.	June 15, 2015
New feature	Updated to support Oracle Database 12c.	April 2, 2015
New feature	Updated to support PostgreSQL versions 9.3.6 and 9.4.1.	March 18, 2015
New feature	Updated to support using Amazon CloudHSM with Oracle DB instances using TDE.	January 8, 2015
New feature	Updated to support encrypting data at rest and new API version 2014-10-31.	January 6, 2015
New feature	Updated to support version 11.2.0.4.v3 of Oracle Database 11g, which includes the PSU released in October 2014.	November 20, 2014
New feature	Updated to include the new Amazon DB engine: Aurora. The Amazon Aurora DB engine supports multiple DB instances in a DB cluster. Amazon Aurora is currently in preview release and is subject to change. For detailed information, see What is Amazon Aurora? .	November 12, 2014
New feature	Updated to support PostgreSQL read replicas.	November 10, 2014
New features	Updated to support version 11.2.0.4v2 of Oracle Database 11g.	October 16, 2014
New API and features	Updated to support the GP2 storage type and new API version 2014-09-01. Updated to support the ability to copy an existing option or parameter group to create a new option or parameter group.	October 7, 2014
New feature	Updated to support InnoDB Cache Warming for DB instances running MySQL version 5.6.19 and later.	September 3, 2014
New feature	Updated to support SSL certificate verification when connecting to MySQL version 5.6, SQL Server, and PostgreSQL database engines.	August 5, 2014
New feature	Updated to support the db.t2 burst-capable DB instance classes.	August 4, 2014
New feature	Updated to support the db.r3 memory-optimized DB instance classes for use with the MySQL (version 5.6), SQL Server, and PostgreSQL database engines.	May 28, 2014
New feature	Updated to support SQL Server Multi-AZ deployments using SQL Server Mirroring.	May 19, 2014

Change	Description	Date changed
New feature	Updated to support upgrades from MySQL version 5.5 to version 5.6.	April 23, 2014
New feature	Updated to support Oracle Database 11g (11.2.0.4).	April 23, 2014
New feature	Updated to support Oracle GoldenGate.	April 3, 2014
New feature	Updated to support the M3 DB instance classes.	February 20, 2014
New feature	Updated to support the Oracle Timezone option.	January 13, 2014
New feature	Updated to support replication between MySQL DB instances in different regions.	November 26, 2013
New feature	Updated to support the PostgreSQL DB engine.	November 14, 2013
New feature	Updated to support SQL Server transparent data encryption (TDE).	November 7, 2013
New API and new feature	Updated to support cross region DB snapshot copies; new API version, 2013-09-09.	October 31, 2013
New features	Updated to support Oracle Statspack.	September 26, 2013
New features	Updated to support using replication to import or export data between instances of MySQL running in Amazon RDS and instances of MySQL running on-premises or on Amazon EC2.	September 5, 2013
New features	Updated to support the db.cr1.8xlarge DB instance class for MySQL 5.6.	September 4, 2013
New feature	Updated to support replication of read replicas.	August 28, 2013
New feature	Updated to support parallel read replica creation.	July 22, 2013
New feature	Updated to support fine-grained permissions and tagging for all Amazon RDS resources.	July 8, 2013
New feature	Updated to support MySQL 5.6 for new instances, including support for the MySQL 5.6 memcached interface and binary log access.	July 1, 2013
New feature	Updated to support major version upgrades from MySQL 5.1 to MySQL 5.5.	June 20, 2013
New feature	Updated DB parameter groups to allow expressions for parameter values.	June 20, 2013
New API and new feature	Updated to support read replica status; new API version, 2013-05-15.	May 23, 2013
New features	Updated to support Oracle Advanced Security features for native network encryption and Oracle Transparent Data Encryption.	April 18, 2013
New features	Updated to support major version upgrades for SQL Server and additional functionality for Provisioned IOPS.	March 13, 2013

Change	Description	Date changed
New feature	Updated to support VPC By Default for RDS.	March 11, 2013
New API and feature	Updated to support log access; new API version 2013-02-12	March 4, 2013
New feature	Updated to support RDS event notification subscriptions.	February 4, 2013
New API and feature	Updated to support DB instance renaming and the migration of DB security group members in a VPC to a VPC security group.	January 14, 2013
New feature	Updated for AWS GovCloud (US-West) support.	December 17, 2012
New feature	Updated to support m1.medium and m1.xlarge DB Instance classes.	November 6, 2012
New feature	Updated to support read replica promotion.	October 11, 2012
New feature	Updated to support SSL in Microsoft SQL Server DB Instances.	October 10, 2012
New feature	Updated to support Oracle micro DB Instances.	September 27, 2012
New feature	Updated to support SQL Server 2012.	September 26, 2012
New API and feature	Updated to support provisioned IOPS. API version 2012-09-17.	September 25, 2012
New features	Updated for SQL Server support for DB Instances in VPC and Oracle support for Data Pump.	September 13, 2012
New feature	Updated for support for SQL Server Agent.	August 22, 2012
New feature	Updated for support for tagging of DB Instances.	August 21, 2012
New features	Updated for support for Oracle APEX and XML DB, Oracle time zones, and Oracle DB Instances in a VPC.	August 16, 2012
New features	Updated for support for SQL Server Database Engine Tuning Advisor and Oracle DB Instances in VPC.	July 18, 2012
New feature	Updated for support for option groups and first option, Oracle Enterprise Manager Database Control.	May 29, 2012
New feature	Updated for support for read replicas in Amazon Virtual Private Cloud.	May 17, 2012
New feature	Updated for Microsoft SQL Server support.	May 8, 2012
New features	Updated for support for forced failover, Multi-AZ deployment of Oracle DB Instances, and nondefault character sets for Oracle DB Instances.	May 2, 2012
New feature	Updated for Amazon Virtual Private Cloud (VPC) Support.	February 13, 2012
Updated content	Updated for new Reserved Instance types.	December 19, 2011
New feature	Updated for Oracle engine support.	May 23, 2011
Updated content	Console updates.	May 13, 2011

Change	Description	Date changed
Updated content	Edited content for shortened backup and maintenance windows.	February 28, 2011
New feature	Added support for MySQL 5.5.	January 31, 2011
New feature	Added support for read replicas.	October 4, 2010
New feature	Added support for AWS Identity and Access Management (IAM).	September 2, 2010
New feature	Added DB Engine Version Management.	August 16, 2010
New feature	Added Reserved DB Instances.	August 16, 2010
New Feature	Amazon RDS now supports SSL connections to your DB Instances.	June 28, 2010
New Guide	This is the first release of the Amazon RDS User Guide.	June 7, 2010

AWS glossary

For the latest AWS terminology, see the [AWS glossary](#) in the *AWS General Reference*.