



SRV310

Optimizing Relational Databases on AWS: Deep Dive on Amazon RDS

Jignesh Shah

Senior Product Manager, Amazon RDS

Yoav Eilat

Senior Product Marketing Manager, Amazon Aurora

What is Amazon RDS?

Managed relational database service with a choice of six popular engines

Amazon
Aurora

MySQL

PostgreSQL

MariaDB

Microsoft SQL Server

ORACLE



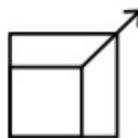
Easy to administer

No need to provision infrastructure, install, and maintain DB software



Available & durable

Automatic Multi-AZ data replication; automated backup, snapshots, and failover



Highly scalable

Scale DB compute and storage with a few clicks; minimal downtime for your application



Fast & secure

SSD storage and guaranteed provisioned I/O; data encryption at rest and in transit

Why use Amazon RDS?



Lower TCO because we manage critical administrative functions

- Automated hardware provisioning, database setup, patching, and backups
- Get more leverage from your teams
- Focus on the things that differentiate you



Built-in high availability and disaster recovery across multiple data centers

- Enabled with a single API call or click of a button in the console
- Even a small startup can leverage enterprise-level availability, durability, and scalability

Configuring your database instance in Amazon RDS

Which RDS engine should I use?

Commercial

ORACLE



Amazon EBS-based storage

Open source



PostgreSQL



Cloud native

**Amazon
Aurora**

MySQL-compatible
PostgreSQL-compatible

Aurora storage system

Which instance type should I choose?

T2 family

- Burstable instances
- 1 vCPU/1 GB RAM > 8 vCPU 32 GB RAM
- Moderate networking performance
- Good for smaller or variable workloads
- Monitor CPU credit metrics in Amazon CloudWatch
- T2.micro is eligible for free tier

M4 family

- General-purpose instances
- 2 vCPU/8 GiB RAM > 64 vCPU 256 GiB RAM
- High-performance networking
- Good for running CPU intensive workloads (e.g., WordPress)

R4 family

- Memory-optimized instances
- 2 vCPU/16 GiB RAM > 64 vCPU 488 GiB RAM
- High-performance networking
- Good for query-intensive workloads or high connection counts



Which storage type should I choose?

General purpose (GP2)

- SSD storage
- Maximum of 16 TB!
- Leverages Amazon EBS Elastic Volumes
- IOPS determined by volume size
- Minimum of 100 IOPS (below 33.33 GiB)
- Bursts to 3,000 IOPS (applicable below 1.3 TB)
- Baseline of 10,000 IOPS (at 3.3 TB and above)
- Affordable performance

Provisioned IOPS (IO1)

- SSD storage
- Maximum of 16 TB!
- Leverages Amazon EBS Elastic Volumes
- Maximum of 40K IOPS (32K for SQL Server)
- Delivers within 10% of the IOPS performance 99.9% of the time
- High performance and consistency

Magnetic

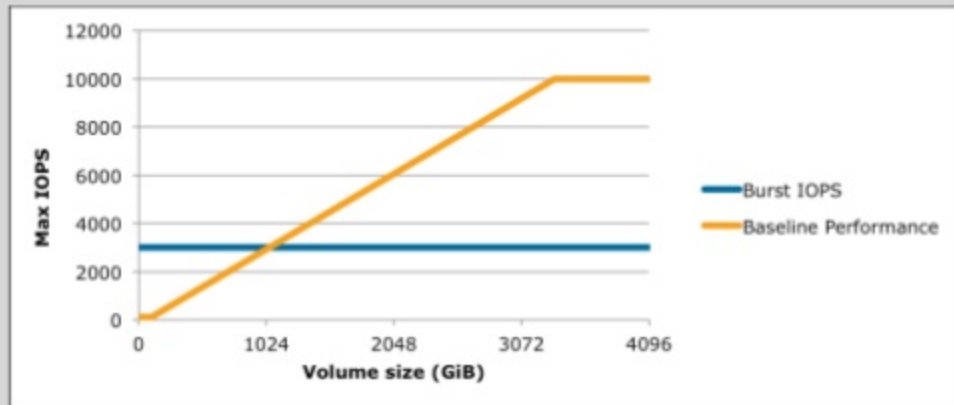
- Magnetic storage
- Maximum of 1 TB
- Supported for legacy databases



How do I decide between GP2 and IO1?

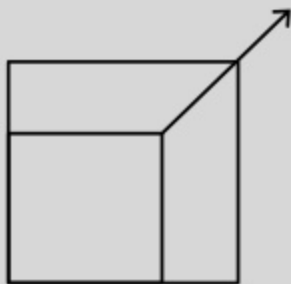
Why am I not seeing 40K IOPS?

- GP2 is a great choice, but be aware of burst credits on volumes < 1 TB
 - Hitting credit-depletion results in IOPS drop—latency and queue depth metrics will spike until credits are replenished
 - Monitor BurstBalance to see percent of burst-bucket I/O credits available
 - Monitor read/write IOPS to see if average IOPS is greater than the baseline
- Think of GP2 burst rate and PIOPS stated rate as maximum I/O rates



How do I scale my database instance?

Why am I not seeing 40K IOPS?



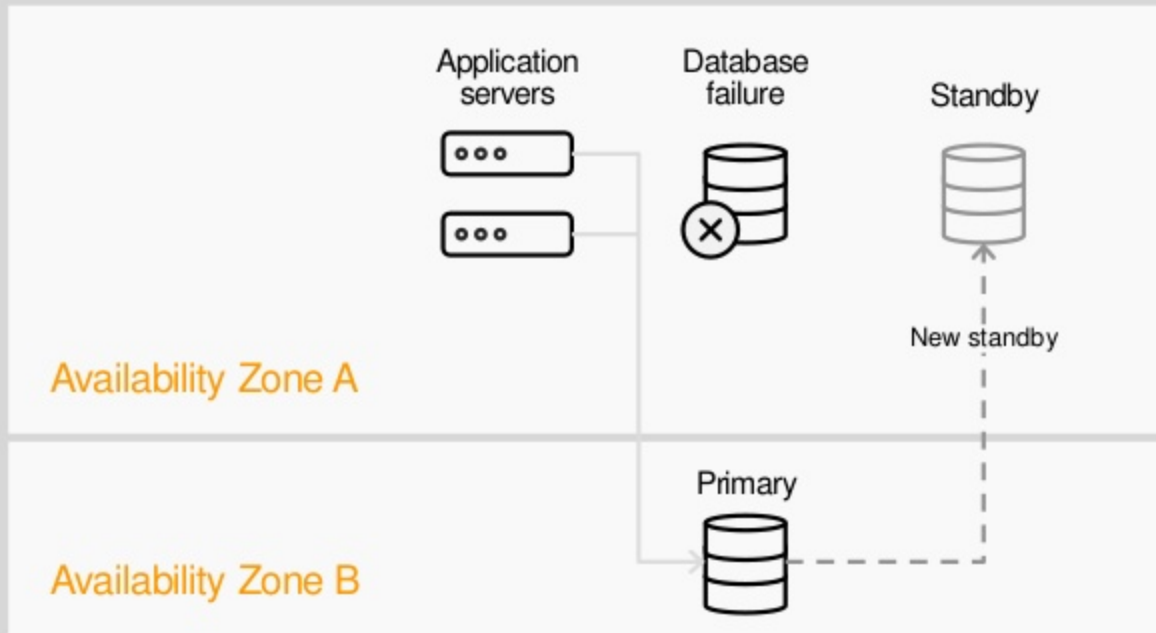
- Scale compute/memory vertically up or down
 - Handle higher load to grow over time
 - Lower usage to control costs
 - New host is attached to existing storage with minimal downtime
- Scale up Amazon EBS storage (now up to 16 TB!)
 - Amazon EBS engines now support Elastic Volumes for fast scaling (now including SQL Server)
 - No downtime for storage scaling
 - Initial scaling operation may take longer, because storage is reconfigured on older instances
 - Can reprovision IOPS on the fly

Managing high availability, read replicas, and backups in Amazon RDS

How do I ensure database high availability?

Multi-AZ provides enterprise-grade fault-tolerance across multiple data centers

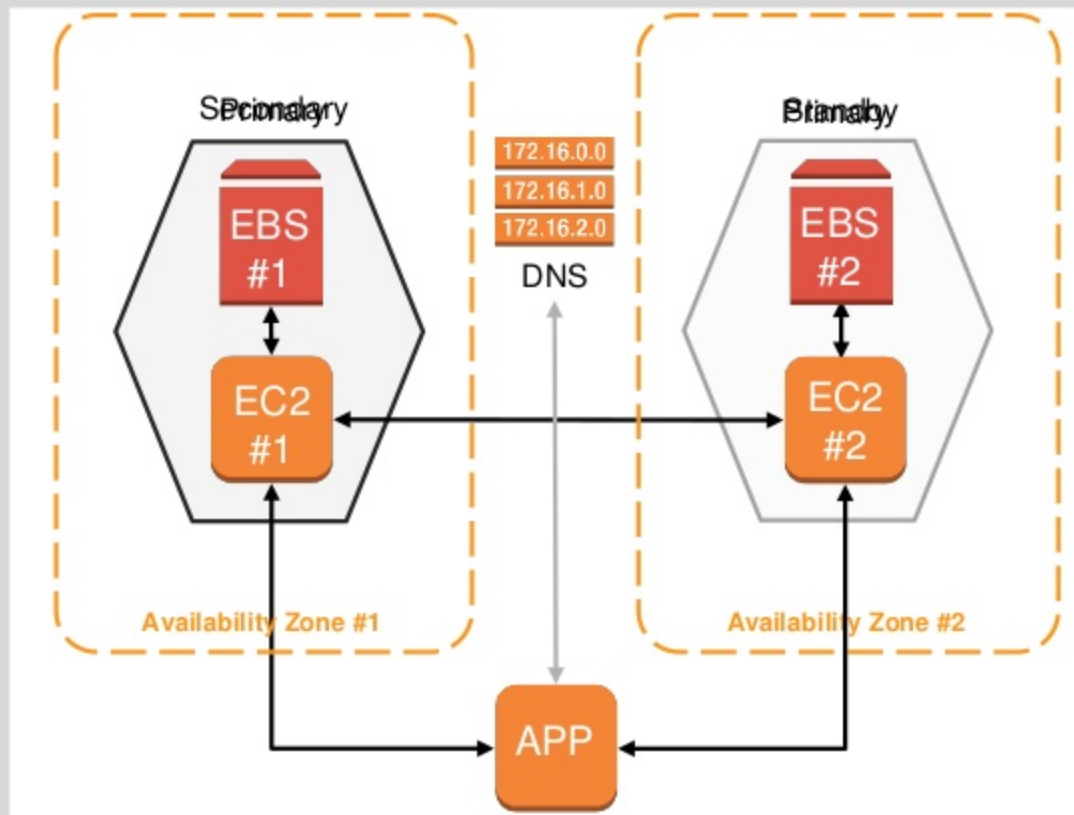
- Automatic failover
- Synchronous replication
- Enabled with one click



What happens during a Multi-AZ failover?

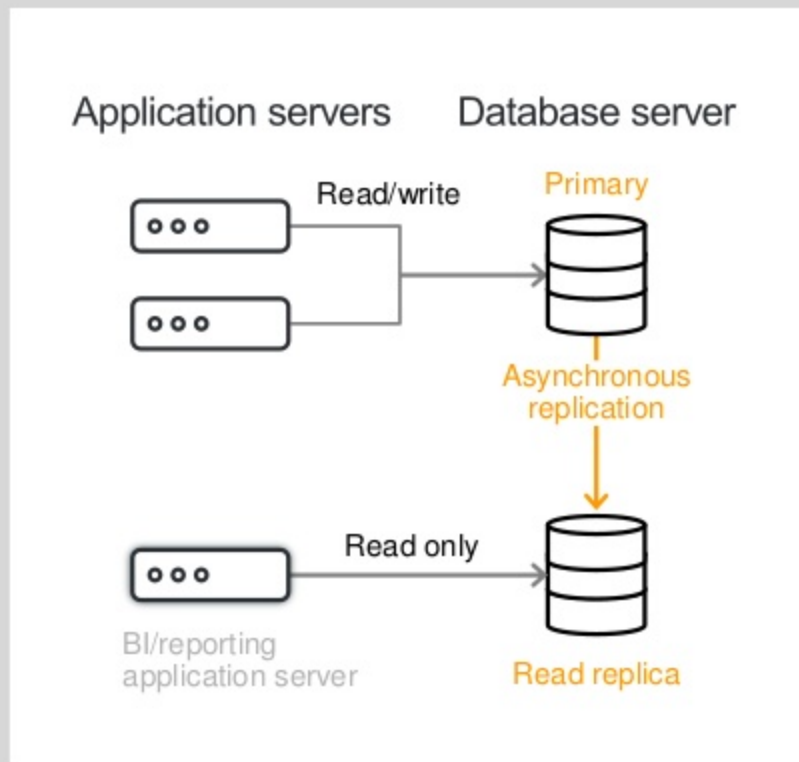
How long does it take?

- Each host manages set of Amazon EBS volumes with a full copy of the data
- Instances are monitored by an external observer to maintain consensus over quorum
- Failover initiated by automation or through the Amazon RDS API
- Redirection to the new primary instance is provided through DNS



Why would I use Read Replicas?

- Relieve pressure on your source database with additional read capacity
- Bring data close to your applications in different regions
- Promote a Read Replica to a master for faster recovery in the event of disaster
- Upgrade a Read Replica to a new engine version
- Supported for MySQL, MariaDB, and PostgreSQL



When should I use Multi-AZ as opposed to Read Replicas?

Multi-AZ

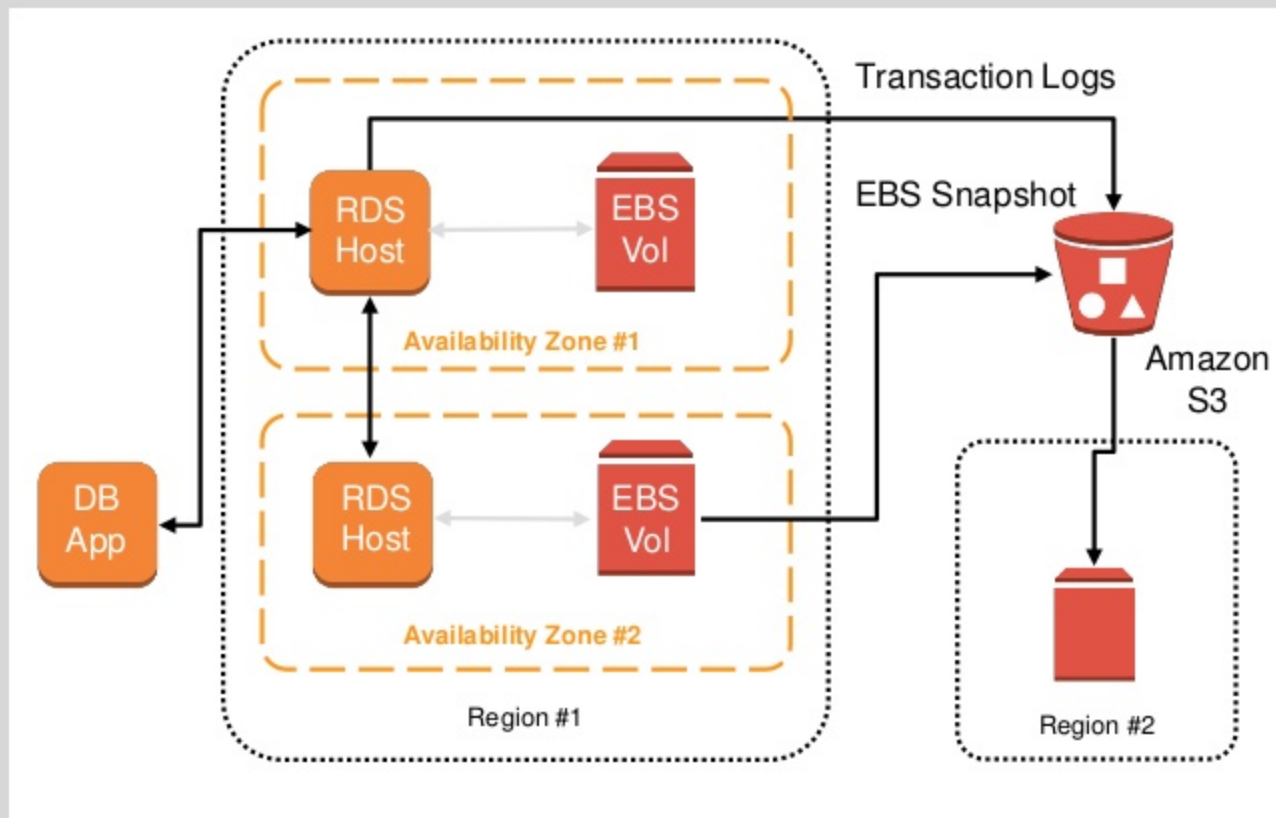
- Synchronous replication—highly durable
- Only primary instance is active at any point in time
- Backups can be taken from secondary
- Always in two Availability Zones within a Region
- Database engine version upgrades happen on primary
- Automatic failover when a problem is detected

Read Replicas

- Asynchronous replication—highly scalable
- All replicas are active and can be used for read scaling
- No backups configured by default
- Can be within an Availability Zone, cross-AZ, or cross-region
- Database engine version upgrades independently from source instance
- Can be manually promoted to a standalone database

How does Amazon RDS manage backups?

- Two options: automated backups and manual snapshots
- Amazon RDS backups use Amazon EBS snapshots stored in Amazon S3
- Transaction logs are stored every 5 minutes in Amazon S3 to support point-in-time recovery (PITR)
- No performance penalty for backups
- Snapshots can be copied across regions or shared with other accounts



When should I use automated backups as opposed to snapshots?

Automated backups

- Specify backup retention window per instance (7-day default)
- Kept until outside of window (35-day maximum) or instance is deleted
- Supports PITR
- Good for disaster recovery

Manual snapshots

- Manually created through the AWS Management Console, AWS CLI, or Amazon RDS API
- Kept until you delete them
- Restores to saved snapshot
- Use for checkpoint before making large changes, nonproduction or test environments, final copy before deleting a database

How do I restore a backup?

Why does it take so long?

- Restoring creates an entirely new database instance
 - Define the instance configuration just like a new instance
 - Will get the default parameter, security, and option groups
- New volumes are hydrated from Amazon S3
 - While the volume is usable immediately, full performance requires the volume to warm up until fully instantiated
 - Migrate to a DB instance class with high I/O capacity
 - Maximize I/O during restore process

Launch DB Instance

You are creating a new DB instance from a source DB instance at a specified time. This new DB instance will have the default DB security group and DB parameter groups.

Restore time

Point in time to restore from

☐ Latest restorable time

November 21, 2017 at 7:24:25 PM UTC-8

☒ Custom

Specify a custom date and time to restore from

Custom Date

November 17, 2017

Custom Time

02

:

00

:

00

UTC-8

Restore DB Instance

You are creating a new DB instance from a source DB instance at a specified time. This new DB instance will have the default DB Security Group and DB Parameter Groups. This feature is currently **supported for InnoDB storage engine only** if you are using MySQL, refer to details [here](#).

Instance specifications

DB Engine

Name of the Database Engine

mysql

License Model

License type associated with the database engine

general-public-license

DB Instance Class

Contains the compute and memory capacity of the DB instance.

db.t2.medium -- 2 vCPU, 4 GB RAM

Multi-AZ Deployment

Specifies if the DB instance should have a standby deployed in another Availability Zone.

☐ Yes

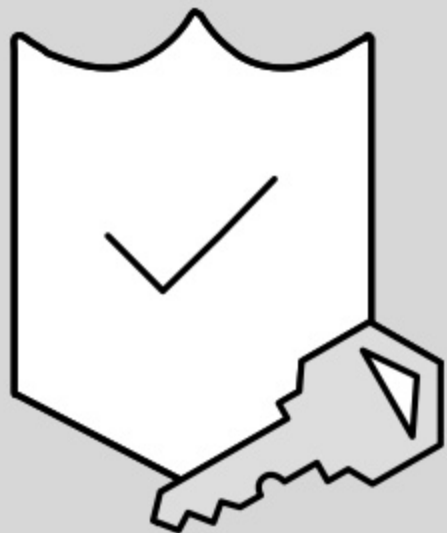
☒ No

Storage type [info](#)

General Purpose (SSD)

Securing your Amazon RDS database instance

How do I secure my Amazon RDS database?



- Amazon RDS is designed to be secure by default
- Network isolation with Amazon Virtual Private Cloud (Amazon VPC)
- AWS Identity and Access Management (IAM)-based resource-level permission controls
- Encryption at rest using AWS KMS (all engines) or Oracle/Microsoft TDE
- Use SSL protection for data in transit

What does Amazon VPC provide?

- Places your instance in a private subnet, making it secure from public routes on the internet
- Database instance IP firewall protection lets you securely control network configuration
- Turn off *Public Accessibility* in DB instance settings to restrict access outside Amazon VPC
- Use ClassicLink to network with non-VPC resources



Routing rules



AWS Direct Connect



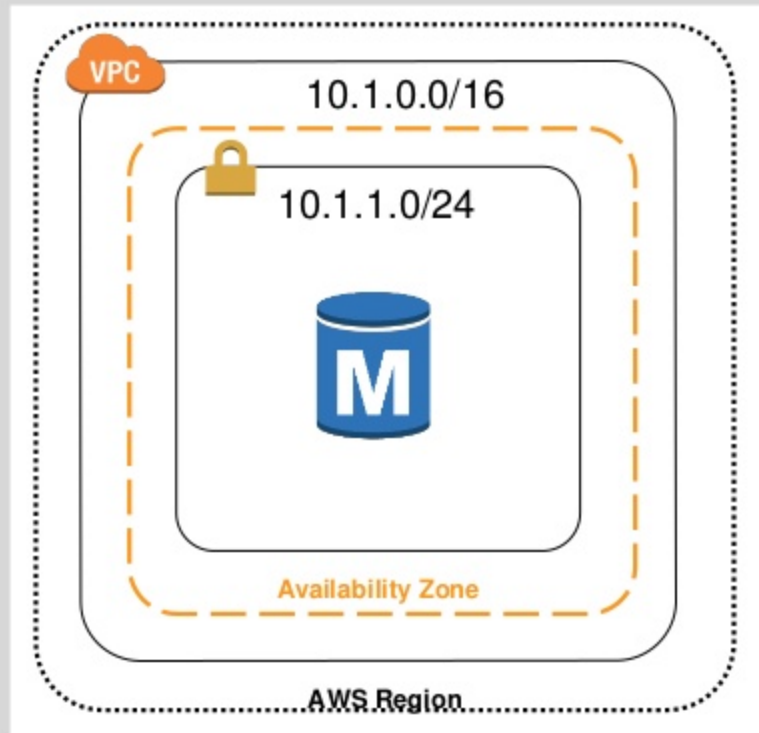
VPN connection



VPC peering



Internet gateway



How do I grant access to my database?

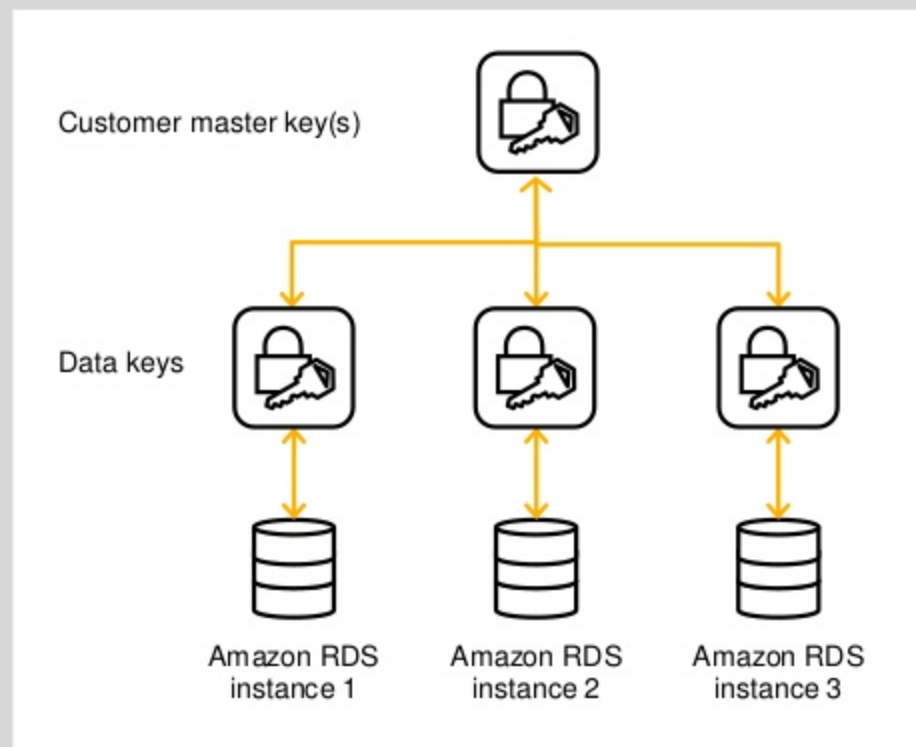
- Use IAM to control who can perform actions on Amazon RDS resources
- Do not use AWS root credentials to manage Amazon RDS resources—you should create an IAM user for everyone, including yourself
- Can use AWS multi-factor authentication (MFA) to provide extra level of protection



```
{
  "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"
        }
      }
    }
  ]
}
```

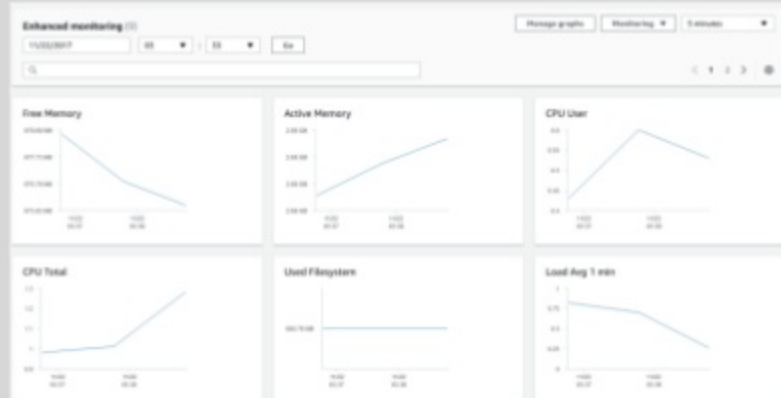
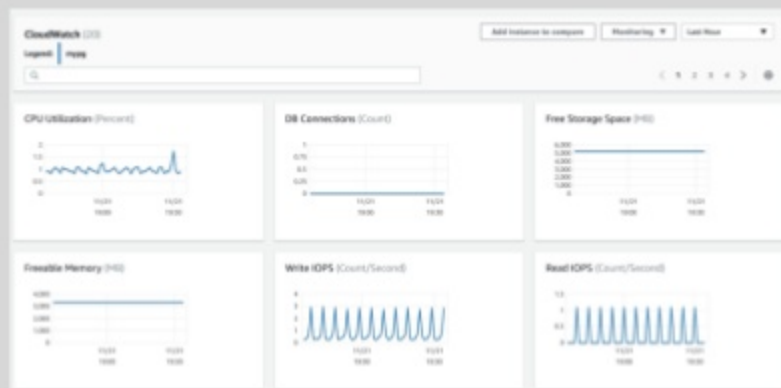
How do I encrypt my database?

- Use AWS KMS-based encryption in the AWS console
- No performance penalty for encrypting data
- Centralized access and audit of key activity
- Best practices
 - Encryption cannot be removed from DB instances
 - If source is encrypted, Read Replicas must be encrypted
 - Add encryption to an unencrypted DB instance by encrypting a snapshot copy



Monitoring your Amazon RDS database instance

How do I monitor my Amazon RDS database?



The screenshot shows the 'Operating system process list' window in Amazon CloudWatch. It contains a table of running processes on the RDS instance.

NAME	VRAM	RES	CPU%	MEM%
postgres [3213]	1.04 GB	52.74 MB	0	1.33
postgres: rhadmin rhadmin localhost(28320) site [177]	1.14 GB	8.04 MB	0	0.2
postgres: logger process [3214]	67.42 MB	1.69 MB	0	0.04
postgres: checkpoint process [3216]	1.04 GB	26.22 MB	0	0.66
postgres: writer process [3217]	1.04 GB	9.51 MB	0	0.34

Amazon CloudWatch metrics & alarms

Upload DB logs directly to CloudWatch Logs

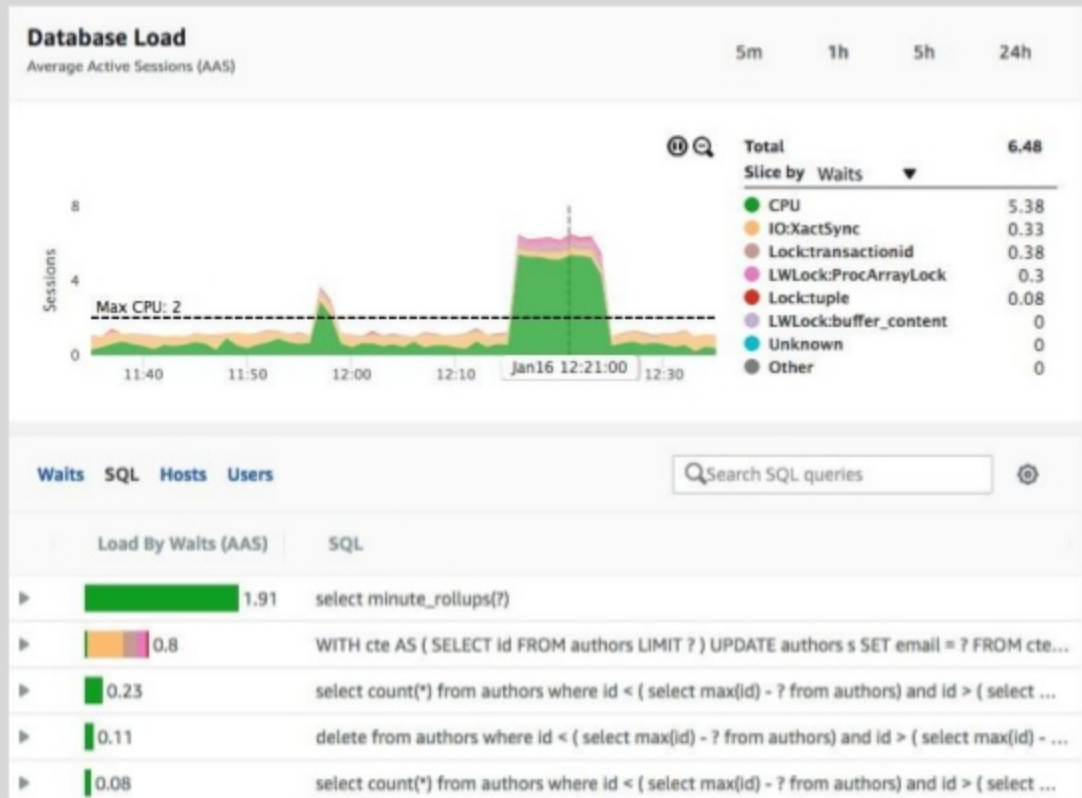
- Amazon RDS for MySQL/MariaDB

Enhanced Monitoring for Amazon RDS

- Access to over 50 CPU, memory, file system, and disk I/O metrics
- As low as 1-second intervals

Integration with third-party monitoring tools

How do I improve database performance?



- Introducing Amazon RDS Performance Insights
- DB load: average active sessions
- Identifies database bottlenecks
 - Easy
 - Powerful
 - Top SQL/most intensive queries
- Identifies source of bottlenecks
- Enables problem discovery
- Adjustable timeframe
 - Hour, day, week, and longer
- Available now for Aurora PostgreSQL
- Coming soon for all Amazon RDS engines

Can I know when service events happen?

- Amazon RDS uses Amazon SNS to receive notification when an event occurs
- Notifications can be in any form supported by Amazon SNS (email, a text message, or a call to an HTTP endpoint)
- Six different source types (DB instance, DB parameter group, DB security group, DB snapshot, DB cluster, DB cluster snapshot)
- 17 different event categories (availability, backup, deletion, configuration change, etc.)

The screenshot displays the configuration interface for an Amazon RDS subscription. It is divided into two main sections: 'Target' and 'Source'.

Target Section:

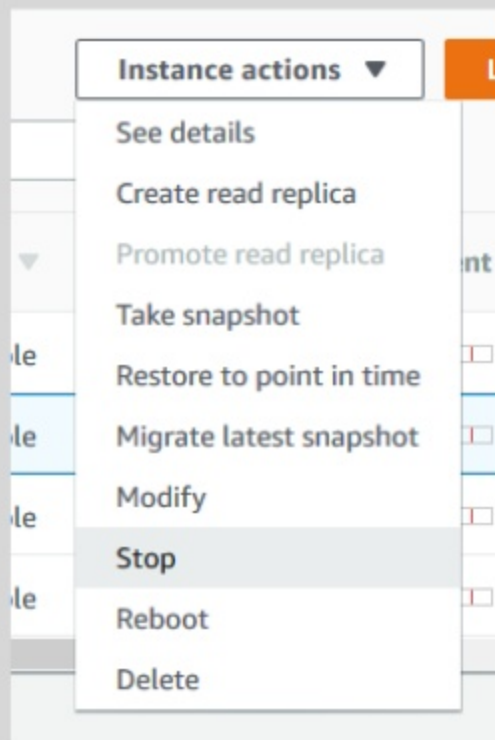
- Send notifications to:** Two radio buttons are present. 'ARN' is selected, while 'New email topic' is unselected.
- ARN:** A label 'ARN to send notifications to' is followed by a dropdown menu currently showing 'notifications'.

Source Section:

- Source type:** A label 'Source type of resource this subscription will consume event from' is followed by a dropdown menu currently showing 'Instances'.
- Instances to include:** A label 'Instances that this subscription will consume events from' is followed by two radio buttons. 'Select specific instances' is selected, while 'All instances' is unselected.
- Specific instances:** A label 'Specific instances' is followed by a dropdown menu showing 'select instances'. Below the dropdown are two tags: 'mysql' and 'mysql5619a', each with a close button (X).
- Event categories to include:** A label 'Event categories that this subscription will consume events from' is followed by two radio buttons. 'Select specific event categories' is selected, while 'All event categories' is unselected.
- Specific event:** A label 'Specific event' is followed by a dropdown menu showing 'select event categories'. Below the dropdown are two tags: 'backup' and 'maintenance', each with a close button (X).

Can I stop my database when it's not in use?

- Stop and start a running database instance from the console or AWS CLI
- Available for single-AZ DB instances
- While instance is stopped, you only pay for storage
- Backup retention window is maintained while stopped
- Instances are restarted after 7 days
 - Pending maintenance operations are applied
 - Instances can be stopped again if wanted



AWS Database Migration Service

What are AWS DMS and AWS SCT?

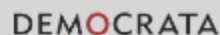
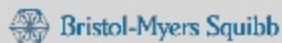
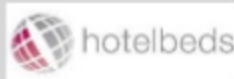
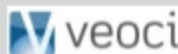
AWS Database Migration Service (AWS DMS) quickly and securely migrates or replicates your databases & data warehouses to AWS



AWS Schema Conversion Tool (AWS SCT) convert your database and data warehouse schemas to open source engines or AWS-native services (Amazon Aurora and Amazon Redshift)

We have migrated over 64,000 unique databases, and counting...

64,000 Databases Migrated with DMS



Key benefits of migrating with DMS



Get off expensive commercial databases & data warehouses

Avoid high fees and restrictive licenses! Switch to open-source based, pay-as-you-go services



Keep your applications running during the migration

Load and sync the target database, then switch over at your convenience



Low cost: pay only for the migration resources you use

Free DMS is available for 6 months when migrating to Aurora, Amazon Redshift, or Amazon DynamoDB

Other migrations are as low as \$3 per terabyte



Migration in both directions avoids lock-in

Replicate your data to a target outside of AWS. Or even migrate your data out of AWS, should your needs change

Database Migration the Easier Way

Step 1: Convert or copy your schema



Step 2: Move your data



Migration & Replication with DMS

Homogeneous or heterogeneous

Sources



Targets



Database Migration Use Cases

Modernize



- **Convert** and extract data from old database engines
- **Update** associated application code

Migrate



- **Migrate** business apps to Amazon RDS
- **Migrate** data warehouses to Amazon Redshift
- **Upgrade, consolidate & archive** your databases

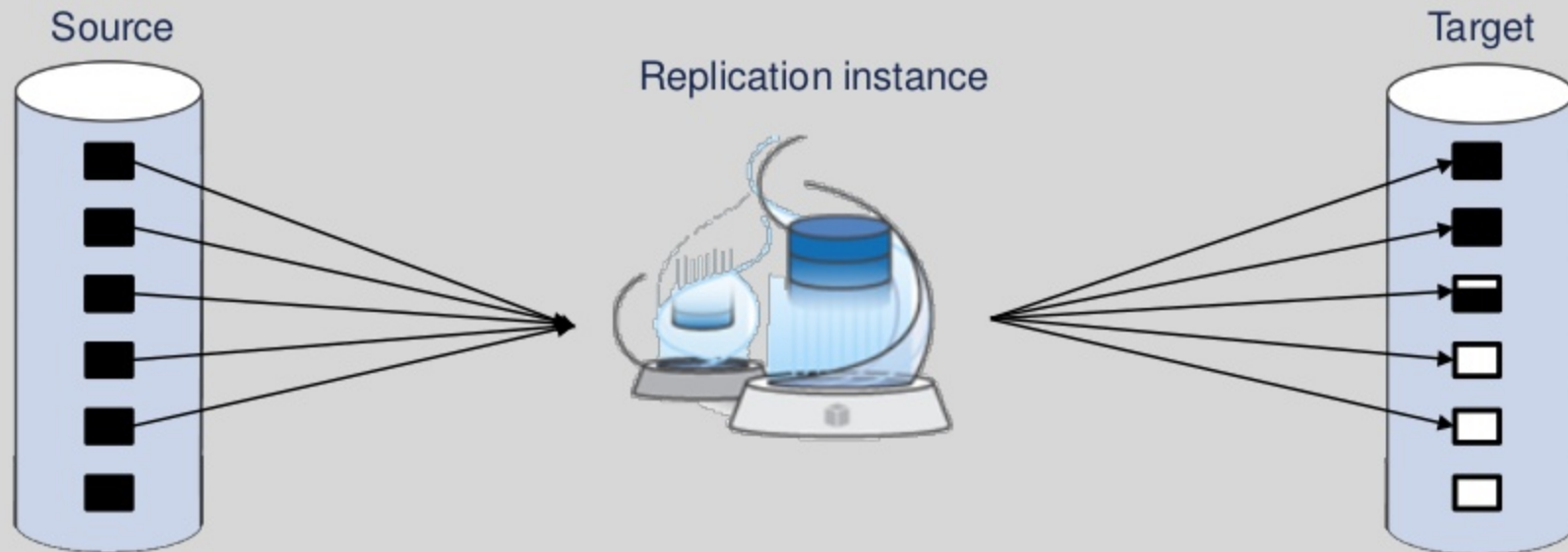
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Replicate

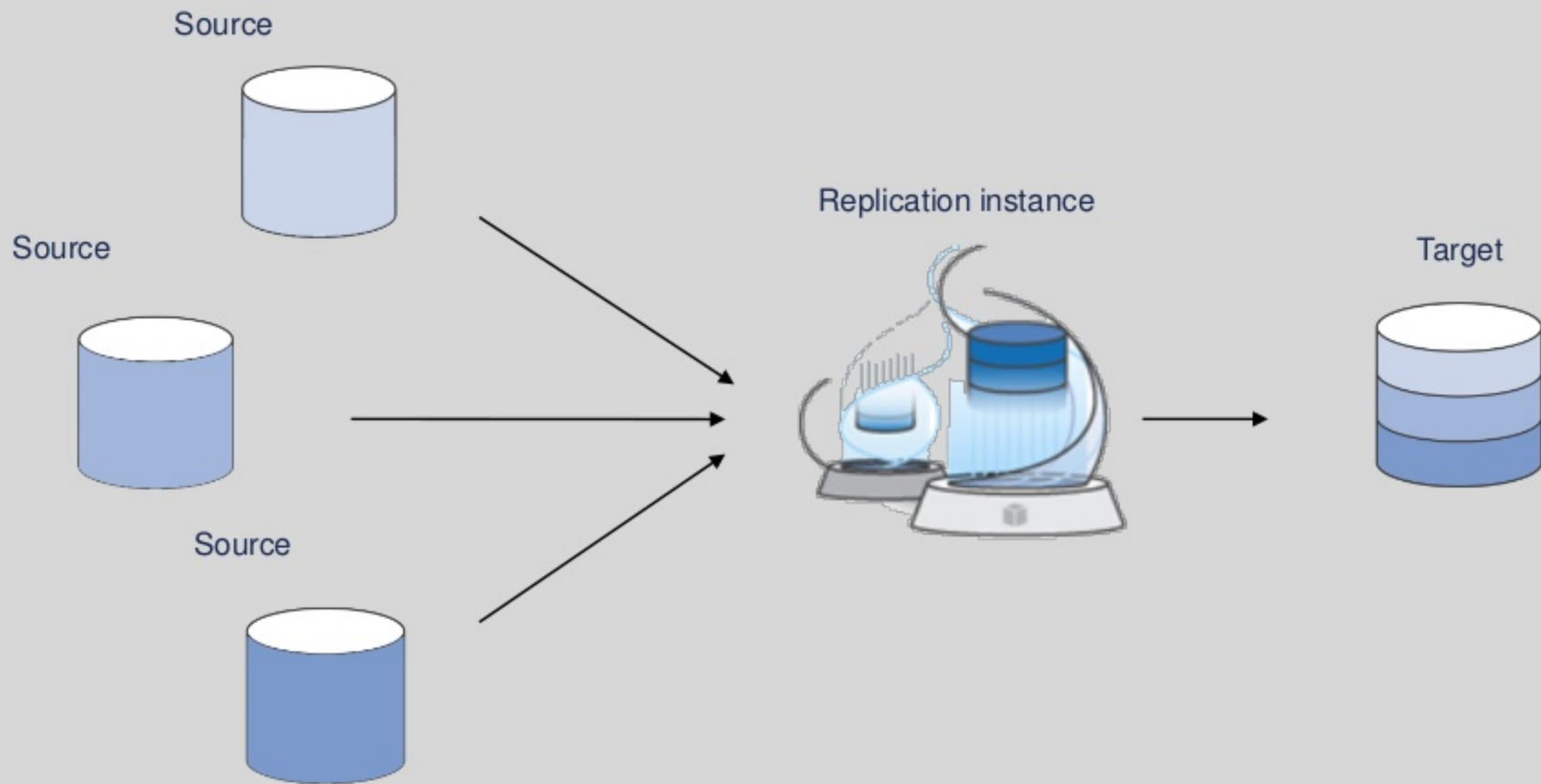


- **Create** cross-region Read Replicas
- **Run** analytics in the cloud
- **Keep** dev/test and production in sync

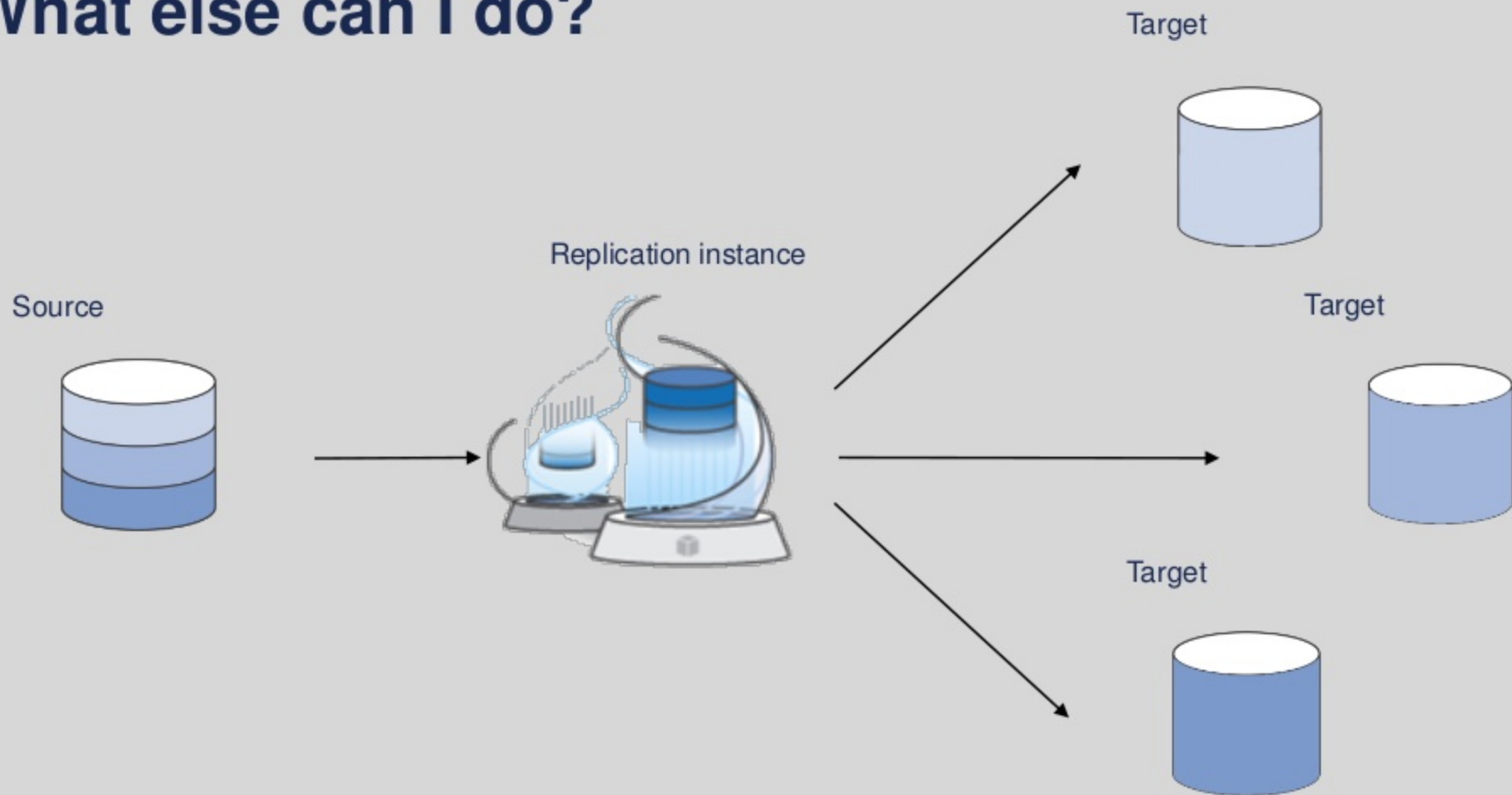
Load is table by table



What else can I do?



What else can I do?



AWS DMS & AWS Snowball



Got huge migration tasks?

Skip the network. Do a physical move with Snowball.

- Migrate large databases (over 5 TB)
- Migrate many databases at once
- Avoid migrations over slow network
- Push model instead of pull model

Schema Conversion with SCT

Modernize your database tier

ORACLE



Amazon
Aurora

Amazon Aurora



Modernize and **Migrate** your data
warehouse to Amazon Redshift

VERTICA



TERADATA



ORACLE



Amazon Redshift

SCT Helps with Converting Tables, Views, & Code

[illegible]

- Sequences
- User-defined types
- Synonyms
- Packages
- Stored procedures
- Functions
- Triggers
- Schemas
- Tables
- Indexes
- Views
- Sort and distribution keys

Database migration assessment

Database Migration Assessment Report

Source Database: RDS, ADMINISTRATION_db, administration-2 (4-17) 34-40 compute-1.amazonaws.com (1)
RDS (US)
Oracle Database 12c Standard Edition (12.1.0.2.0-64bit Production)

Executive Summary

We completed the analysis of your Oracle source database and estimate that 91% of the database storage objects and 100% of database code objects can be converted automatically or with minimal changes if you select Amazon Aurora as your migration target. Database storage objects include schemas, tables, columns, constraints, indexes, sequences, synonyms, user-defined types and types. Database code objects include functions, procedures, packages, triggers, views, materialized views, events, SQL-writer functions, SQL-inline functions, SQL-table functions, attributes, variables, constants, table types, public types, private types, cursors, exceptions, parameters and other objects. Based on our analysis of SQL syntax elements of your source database schema, we estimate that 99.9% of your entire database schema can be converted automatically to Amazon Aurora. To complete the migration, we recommend 297 conversion action(s) ranging from simple tasks to medium-complexity actions to significant conversion actions.

Database Objects with Conversion Actions for Amazon Aurora

Of the total 1,276 database storage object(s) and 157 database code object(s) in the source database, we were able to identify 1,427 (91%) database storage object(s) and 157 (100%) database code object(s) that can be converted automatically or with minimal changes to Amazon Aurora.

149 (9%) database storage object(s) required 149 significant user action(s) to complete the conversion.

Figure: Conversion statistics for database storage objects

Category	Count	Percentage
Objects automatically converted	1,427	91%
Objects with simple actions	149	9%
Objects with medium-complexity actions	0	0%
Objects with significant actions	0	0%

Figure: Conversion statistics for database code objects

Category	Count	Percentage
Objects automatically converted	157	100%
Objects with significant actions	0	0%

Detailed Recommendations for Amazon Aurora Migrations

If you choose to migrate your Oracle database to Amazon Aurora, we recommend the following actions.

Page 1 of 4

Connect SCT to
source and target
databases

Run assessment
report

Read executive
summary

Follow detailed
instructions

Database Migration Assessment Report

Source Database: RDS, ADMINISTRATION_db, administration-2 (4-17) 34-40 compute-1.amazonaws.com (1)
RDS (US)
Oracle Database 12c Standard Edition (12.1.0.2.0-64bit Production)

Storage Object Actions

Sequence Changes

Some changes are required to sequences that cannot be converted automatically. You'll need to address these issues manually.

Issue 341: MySQL doesn't support sequences
Recommended Action: Try developing a system for sequences in your application.
Issue Code: 341 | No. of Occurrences: 134 | Estimated Complexity: Significant
Schemas: RDS, ADMINISTRATION Sequences BACKUP_ID_SEQUENCE
Schemas: RDS, ADMINISTRATION Sequences CERTIFICATE_ID_SEQUENCE
Schemas: RDS, ADMINISTRATION Sequences CHARACTER_SET_ID_SEQ
Schemas: RDS, ADMINISTRATION Sequences CUSTOMER_TURNSET_GROUP_ID_SEQ
Schemas: RDS, ADMINISTRATION Sequences CUSTOMER_TURNSET_ID_SEQ
+129 more

Index Changes

Some changes are required to indexes that cannot be converted automatically. You'll need to address these issues manually.

Issue 397: MySQL doesn't support function indexes
Recommended Action: Review your code and try to use simple index.
Issue Code: 397 | No. of Occurrences: 1 | Estimated Complexity: Significant
Documentation References: <https://dev.mysql.com/doc/refman/7.0/create-table.html>
Schemas: RDS, ADMINISTRATION Tables DBL_ENGINE_SEEDS Indexes: 1,DBL_ENGINE_SEEDS,DBL_ENGINE_CONF_ID
Schemas: RDS, ADMINISTRATION Tables RDS_SYSTEM_ACCOUNTS Indexes: 1,SYS_ACCOUNT_DEFAULT
Schemas: RDS, ADMINISTRATION Tables RUNNABLE,DBL_CONFIG Indexes: 1,RNRL,DBL_CFG_PREFERRED

Constraint Changes

Some changes are required to constraints that cannot be converted automatically. You'll need to address these issues manually.

Issue 210: MySQL doesn't support FUNCTION AS DEFAULT VALUE
Recommended Action: Try using a trigger.
Issue Code: 210 | No. of Occurrences: 2 | Estimated Complexity: Simple
Documentation References: <https://dev.mysql.com/doc/refman/7.0/create-table.html>
Schemas: RDS, ADMINISTRATION Tables CUSTOMERS Constraints: CK_CUSTOMOR_TRUST_LEVEL_STATE: 0:10
Schemas: RDS, ADMINISTRATION Tables STORAGE_VOLUME_MBS Constraints: CK_STORAGE_VOLUME_MBS: 0:8

Issue 325: MySQL does not support check constraints. Enabling triggers created
Recommended Action: Please review generated code and modify it if it is necessary.
Issue Code: 325 | No. of Occurrences: 203 | Estimated Complexity: Simple
Documentation References: <https://dev.mysql.com/doc/refman/7.0/create-table.html>

Page 2 of 4

Jignesh Shah

Sr Product Manager,
Amazon RDS

jkshah@amazon.com

Yoav Eilat

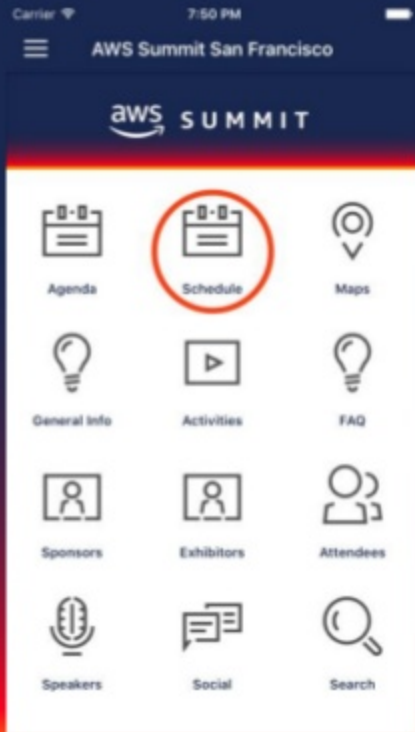
Sr Product Marketing Manager,
Amazon RDS

yeilat@amazon.com

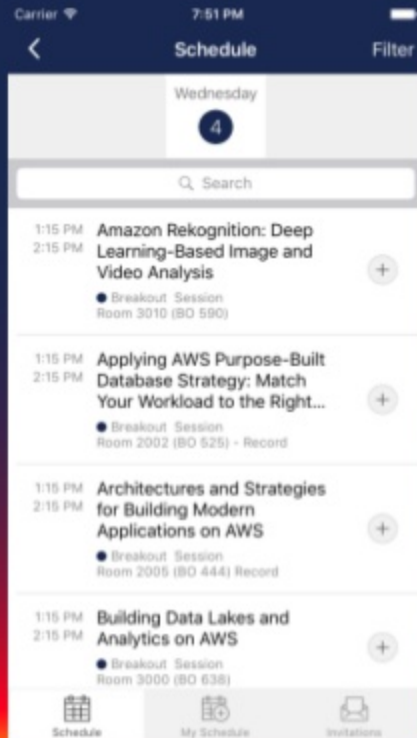
**Please complete the session
survey in the summit mobile app.**

Submit Session Feedback

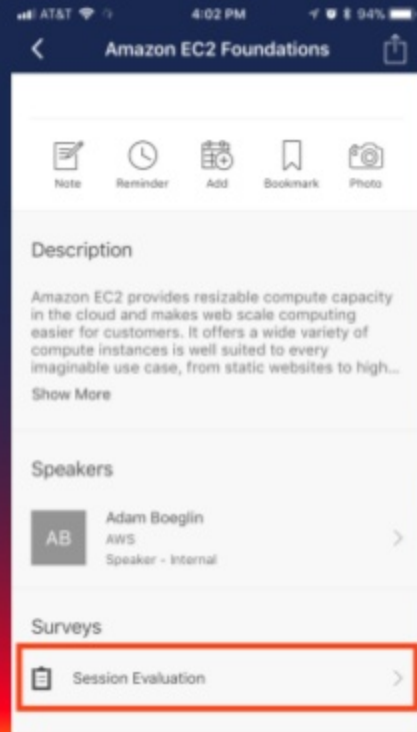
1. Tap the **Schedule** icon.



2. Select the session you attended.



3. Tap **Session Evaluation** to submit your feedback.



Thank you!

aws.amazon.com/rds

aws.amazon.com/dms