
Amazon Relational Database Service

사용 설명서

API 버전 2014-10-31



Amazon Relational Database Service: 사용 설명서

Copyright © 2019 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

Amazon RDS란 무엇입니까?	1
개요	1
DB 인스턴스	1
리전 및 가용 영역	2
보안	2
Amazon RDS DB 인스턴스 모니터링	3
Amazon RDS 인터페이스	3
AWS Management 콘솔	3
명령행 인터페이스	3
Amazon RDS를 사용한 프로그래밍	3
Amazon RDS에 대한 요금이 부과되는 방법	3
다음 단계	4
시작하기	4
데이터베이스 엔진별 항목	4
설정	5
AWS에 가입	5
IAM 사용자 생성	5
요구 사항 결정	7
보안 그룹을 생성하여 VPC 내부의 DB 인스턴스에 대한 액세스를 제공	8
시작하기	10
MariaDB DB 인스턴스 생성 및 데이터베이스에 연결	10
MariaDB DB 인스턴스 생성	10
MariaDB를 실행하는 DB 인스턴스의 데이터베이스에 연결	15
DB 인스턴스 삭제	17
Microsoft SQL Server DB 인스턴스 생성 및 DB 인스턴스에 연결	17
샘플 SQL Server DB 인스턴스 생성	17
샘플 DB 인스턴스에 연결	22
샘플 DB 인스턴스 탐색	24
샘플 DB 인스턴스 삭제	25
MySQL DB 인스턴스 생성 및 데이터베이스에 연결	26
MySQL DB 인스턴스 만들기	26
MySQL을 실행하는 DB 인스턴스의 데이터베이스에 연결	31
DB 인스턴스 삭제	33
Oracle DB 인스턴스 생성 및 데이터베이스에 연결	33
샘플 Oracle DB 인스턴스 생성	34
샘플 DB 인스턴스에 연결	37
샘플 DB 인스턴스 삭제	39
PostgreSQL DB 인스턴스 생성 및 데이터베이스에 연결	39
PostgreSQL DB 인스턴스 만들기	39
PostgreSQL DB 인스턴스에 연결	43
DB 인스턴스 삭제	46
자습서: 웹 서버 및 Amazon RDS 데이터베이스 생성	46
1단계: DB 인스턴스 만들기	47
2단계: 웹 서버 생성	53
자습서	65
Amazon RDS 모범 사례	66
Amazon RDS 기본 운영 지침	66
DB 인스턴스 RAM 권장 사항	67
Amazon RDS 보안 모범 사례	67
Enhanced Monitoring을 통한 운영 체제 문제 식별	67
지표를 통해 성능 문제 식별	68
성능 지표 보기	68
성능 지표 평가	69
쿼리 튜닝	70

MySQL 스토리지 엔진으로 작업하기 위한 모범 사례	71
MariaDB 스토리지 엔진으로 작업하기 위한 모범 사례	72
Oracle 작업의 모범 사례	72
PostgreSQL로 작업하기 위한 모범 사례	72
PostgreSQL DB 인스턴스에 데이터 로드	72
fsync 및 full_page_writes 데이터베이스 파라미터 작업	73
PostgreSQL Autovacuum 기능 사용	73
SQL Server로 작업하기 위한 모범 사례	74
DB 파라미터 그룹 작업	74
Amazon RDS 모범 사례 프레젠테이션 동영상	75
DB 인스턴스	76
DB 인스턴스 클래스 선택	77
DB 인스턴스 클래스 유형	77
용어	78
하드웨어 사양	79
지원되는 DB 엔진	82
DB 인스턴스 클래스 변경	87
프로세서 구성	87
DB 인스턴스 상태	96
리전 및 가용 영역	99
.....	99
DB 인스턴스 스토리지	101
스토리지 유형	101
범용 SSD 스토리지	101
프로비저닝된 IOPS 스토리지	103
마그네틱 스토리지	104
스토리지 성능 모니터링	105
스토리지 성능에 영향을 끼치는 요인	105
고가용성(다중 AZ)	109
다중 AZ 배포가 되도록 DB 인스턴스 수정	109
Amazon RDS 장애 조치 프로세스	110
관련 주제	110
DB 인스턴스 수명 주기	111
DB 인스턴스 생성	111
DB 인스턴스에 연결	111
DB 인스턴스 수정	111
DB 인스턴스 유지	113
엔진 버전 업그레이드	118
DB 인스턴스 이름 바꾸기	121
DB 인스턴스 재부팅	123
DB 인스턴스 종지	125
DB 인스턴스 시작하기	127
DB 인스턴스 삭제	128
RDS 리소스에 태그 지정	131
개요	131
AWS Management 콘솔	133
CLI	134
API	134
.....	135
읽기 전용 복제본 작업	135
개요	136
읽기 전용 복제본 만들기	138
읽기 전용 복제본 승격	139
다른 AWS 리전에서 읽기 전용 복제본 만들기	141
읽기 전용 복제본 모니터링	146
옵션 그룹 작업	148
옵션 그룹 개요	148

옵션 그룹 생성	150
옵션 그룹 복사	151
옵션을 옵션 그룹에 추가하기	152
옵션 그룹의 옵션 및 옵션 설정 표시하기	155
옵션 설정 변경	156
옵션을 옵션 그룹에서 제거하기	158
옵션 그룹 삭제	160
파라미터 그룹 작업	161
DB 파라미터 그룹 생성	163
DB 파라미터 그룹의 파라미터 수정	164
DB 파라미터 그룹 복사	166
DB 파라미터 그룹 나열	168
DB 파라미터 그룹의 파라미터 값 보기	169
DB 파라미터 그룹 비교	170
DB 파라미터 값	170
ARN 작업	172
ARN 생성	173
기존 ARN 가져오기	176
스토리 작업	178
DB 인스턴스 스토리지 용량 증가	178
스토리지 유형 변경	180
프로비저닝된 IOPS 수정	181
Amazon RDS에 대한 DB 인스턴스 결제	183
온 디맨드 DB 인스턴스	184
예약 DB 인스턴스	184
백업 및 복구	196
백업 작업	196
백업 스토리지	196
백업 기간	197
백업 보존 기간	198
자동 백업 비활성화	198
자동 백업 활성화	199
자동 백업 보존	201
지원되지 않는 MySQL 스토리지 엔진에 대한 자동 백업	202
지원되지 않는 MariaDB 스토리지 엔진에 대한 자동 백업	203
.....	203
DB 스냅샷 생성	203
DB 스냅샷에서 복원	205
파라미터 그룹	205
보안 그룹	205
옵션 그룹	205
Microsoft SQL Server	205
Oracle	206
스냅샷에서 복구	206
스냅샷 복사	207
제한 사항	207
스냅샷 보존	208
공유 스냅샷	208
암호화	208
AWS 리전 간 스냅샷 복사	208
옵션 그룹	209
파라미터 그룹	209
DB 스냅샷 복사	209
스냅샷 공유	215
암호화된 스냅샷 공유	216
스냅샷 공유	218
특정 시점으로 복구	221

스냅샷 삭제	222
DB 스냅샷 삭제	222
자습: DB 스냅샷에서 DB 인스턴스 복원	223
DB 스냅샷에서 DB 인스턴스 복원 필수 조건	224
DB 스냅샷에서 DB 인스턴스 복원	225
복원된 DB 인스턴스 수정	225
모니터링	228
모니터링 개요	228
모니터링 도구	229
CloudWatch를 사용하여 모니터링	230
CloudWatch Logs에 게시	235
확장 모니터링	238
Enhanced Monitoring 가능성	238
CloudWatch 측정치와 Enhanced Monitoring 측정치의 차이점	239
확장 모니터링 설정 및 활성화	239
확장 모니터링 보기	241
CloudWatch Logs를 사용하여 Enhanced Monitoring 보기	244
성능 개선 도우미	249
성능 개선 도우미 활성화	251
성능 개선 도우미의 액세스 제어	254
성능 개선 도우미 대시보드 사용	256
추가 사용자 인터페이스 기능	264
성능 개선 도우미 API	265
CloudWatch에 게시되는 지표	276
성능 개선 도우미 카운터	278
AWS CloudTrail을 사용하여 성능 개선 도우미 작업 로깅	286
Amazon RDS 권장 사항 사용	286
권장 사항 대응	288
Amazon RDS 이벤트 알림 서비스 사용	291
Amazon RDS 이벤트 카테고리 및 이벤트 메시지	292
Amazon RDS 이벤트 알림 구독	298
Amazon RDS 이벤트 알림 구독의 목록 표시	299
Amazon RDS 이벤트 알림 구독 변경	301
Amazon RDS 이벤트 알림 구독에 대한 소스 식별자 추가	302
Amazon RDS 이벤트 알림 구독의 소스 식별자 제거	303
Amazon RDS 이벤트 알림 카테고리의 목록 표시	303
Amazon RDS 이벤트 알림 구독 삭제	304
Amazon RDS 이벤트 보기	305
AWS Management 콘솔	305
CLI	306
API	306
데이터베이스 로그 파일	306
데이터베이스 로그 파일 보기 및 나열	307
데이터베이스 로그 파일 다운로드	307
데이터베이스 로그 파일 조사	308
CloudWatch Logs에 게시	309
REST를 사용하여 로그 파일 내용 읽기	309
MariaDB 데이터베이스 로그 파일	310
Microsoft SQL Server 데이터베이스 로그 파일	318
MySQL 데이터베이스 로그 파일	318
Oracle 데이터베이스 로그 파일	326
PostgreSQL 데이터베이스 로그 파일	331
AWS CloudTrail을 사용하여 Amazon RDS API 호출 로깅	335
CloudTrail의 Amazon RDS 정보	335
Amazon RDS 로그 파일 항목 이해	336
보안 구성	339

인증 및 액세스 제어	339
인증	340
액세스 제어	341
액세스 관리 개요	341
자격 증명 기반 정책(IAM 정책) 사용	344
Amazon RDS API 권한 참조	348
조건 사용	363
MySQL 및 PostgreSQL을 위한 IAM 데이터베이스 인증	369
Amazon RDS 리소스 암호화	385
Amazon RDS 리소스 암호화 개요	386
DB 인스턴스에 대해 Amazon RDS 암호화 활성화	386
Amazon RDS 암호화 가능성	387
Amazon RDS 암호화 키 관리	388
Amazon RDS 암호화된 DB 인스턴스의 제한	388
SSL을 사용하여 연결 암호화	388
중간 인증서	389
보안 그룹을 통한 액세스 제어	390
DB 보안 그룹	390
VPC 보안 그룹	390
DB 보안 그룹 및 VPC 보안 그룹	391
보안 그룹 시나리오	391
VPC 보안 그룹 생성	392
DB 인스턴스 연결	392
DB VPC 보안 그룹 삭제	392
EC2-Classic의 DB 보안 그룹	394
마스터 사용자 계정 권한	402
서비스 연결 역할	403
Amazon RDS에 대한 서비스 연결 역할 권한	404
Amazon RDS에 대한 서비스 연결 역할 생성	405
Amazon RDS에 대한 서비스 연결 역할 편집	405
Amazon RDS에 대한 서비스 연결 역할 삭제	405
Using Amazon RDS with Amazon VPC	406
EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인	407
VPC에서 DB 인스턴스에 액세스하는 시나리오	408
VPC에서 DB 인스턴스를 사용한 작업	415
DB 인스턴스에 대한 VPC 업데이트	420
자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성	421
Amazon RDS MariaDB	426
공통 관리 작업	426
MariaDB 버전	428
버전 및 기능 지원	429
MariaDB 10.3 지원	429
MariaDB 10.2 지원	429
MariaDB 10.1 지원	430
MariaDB 10.0 지원	430
지원되지 않는 기능	430
지원되는 스토리지 엔진	431
MariaDB 보안	431
SSL 지원	433
캐시 워밍	434
요청 시 버퍼풀 덤프 및 로딩	435
데이터베이스 파라미터	435
일반 DBA 작업	435
현지 시간대	435
MariaDB를 실행하는 DB 인스턴스 생성	437
AWS Management 콘솔	437
CLI	440

API	441
사용 가능한 설정	442
관련 주제	445
MariaDB를 실행 중인 DB 인스턴스에 연결	445
mysql 유ти리티를 통해 연결하기	447
SSL에 인스턴스 연결	447
최대 MariaDB 연결	448
관련 주제	448
MariaDB 기반 DB 인스턴스의 변경	449
사용 가능한 설정	450
관련 주제	455
MariaDB DB 엔진 업그레이드	455
개요	455
MariaDB DB 인스턴스 업그레이드	456
MySQL DB 스냅샷에서 MariaDB DB 인스턴스로 데이터 마이그레이션	456
MariaDB와 MySQL 간의 호환성 관련 문제	456
AWS Management 콘솔	457
CLI	459
API	460
MariaDB 복제 작업	460
MariaDB 읽기 전용 복제본 작업	461
GTID 기반 복제 구성	464
MariaDB DB 인스턴스로 데이터 가져오기	466
MariaDB를 위한 옵션	467
MariaDB 감사 플러그인 지원	467
MariaDB에 대한 파라미터	470
Amazon RDS SQL의 MariaDB 참조	474
mysql.rds_set_external_master_gtid	474
mysql.rds_kill_query_id	476
Amazon RDS의 Microsoft SQL Server	477
공통 관리 작업	477
제한	479
DB 인스턴스 클래스 지원	481
보안	481
규정 준수 프로그램	482
HIPAA	482
SSL 지원	483
버전 및 기능 지원	483
SQL Server 2017 지원	483
SQL Server 2016 지원	484
SQL Server 2014 지원	484
Amazon RDS에서 지원되는 SQL Server 2012	485
Amazon RDS에서 사용되지 않는 SQL Server 2008 R2 버전	486
MSSQL 버전 관리	487
데이터베이스 엔진 패치 및 버전	487
사용 중단 일정	487
CDC Support	488
지원되지 않는 기능과 지원이 제한된 기능	488
다중 AZ 배포	489
TDE 사용	489
현지 시간대	489
지원되는 시간대	490
Amazon RDS의 SQL Server 라이선싱	493
라이선스가 종료된 DB 인스턴스 복원	493
AWS에서 SQL Server Developer 버전 사용	493
관련 주제	493
SQL Server 기반 DB 인스턴스 생성	494

AWS Management 콘솔	494
CLI	498
API	499
사용 가능한 설정	500
관련 주제	503
SQL Server 기반 DB 인스턴스에 연기하기	503
SSMS로 DB 인스턴스에 연결	503
SQL Workbench/J로 DB 인스턴스에 연결	506
보안 그룹 고려 사항	508
문제 해결	509
.....	509
SQL Server 기반 DB 인스턴스의 설정 변경	509
사용 가능한 설정	511
관련 주제	516
SQL Server DB 엔진 업그레이드	516
개요	517
메이저 버전 업그레이드	517
다중 AZ와 인 메모리 최적화 활성화의 고려 사항	518
옵션 및 파라미터 그룹 고려 사항	518
업그레이드 테스트	519
SQL Server DB 인스턴스 업그레이드	519
지원 종료 전에 사용되지 않는 DB 인스턴스 업그레이드	520
SQL Server 데이터베이스 가져오기 및 내보내기	520
제한 및 권장 사항	521
설정	521
기본 백업 및 복원 사용	523
백업 파일 압축	527
문제 해결	528
관련 주제	529
다른 방법으로 SQL Server 데이터 가져오기 및 내보내기	529
SQL Server의 다중 AZ	537
다중 AZ를 SQL Server DB 인스턴스에 추가	538
참고 및 권장 사항	538
보조의 위치 확인	540
상시 작동 AG로 마이그레이션	541
SQL Server DB 인스턴스와 함께 SSL 사용	541
SSL 지정	541
특정 연결 암호화	542
SQL Server용 옵션	544
기본 백업 및 복원	545
투명한 데이터 암호화	547
SQL Server Audit	549
SQL Server에 대한 일반 DBA 작업	554
tempdb 데이터베이스에 액세스	555
SQL Server 튜닝 어드바이저를 사용한 데이터베이스 워크로드 분석	557
데이터 정렬 및 문자 세트	560
복구 모델 결정	560
다중 AZ 데이터베이스 삭제	561
CDC 사용	561
다중 AZ 데이터베이스의 이름 변경	563
db_owner 역할 암호 재설정	563
라이선스가 종료된 DB 인스턴스 복원	564
오프라인에서 온라인으로 데이터베이스 전환	564
SQL Server 에이전트의 사용	564
SQL Server 로그 작업	566
추적 및 덤프 파일 작업	566
SQL Server의 고급 관리 작업 및 개념	567

SQL Server DB 인스턴스를 통한 Windows 인증 사용	568
Amazon RDS의 MySQL	576
공통 관리 작업	576
MySQL 버전	578
Amazon RDS가 지원하지 않는 MySQL 기능	580
지원되는 스토리지 엔진	580
MySQL 보안	581
암호 확인 플러그인	582
SSL 지원	583
MySQL에서 memcached 및 기타 옵션 사용	584
InnoDB 캐시 워밍	585
요청 시 버퍼풀 덤팅 및 로딩	585
현지 시간대	585
알려진 문제 및 제한	587
MySQL 기반 DB 인스턴스 생성	587
AWS Management 콘솔	587
CLI	590
API	591
사용 가능한 설정	591
관련 주제	595
MySQL 기반 DB 인스턴스에 연결	595
MySQL 클라이언트를 통해 연결하기	597
SSL에 인스턴스 연결	597
MySQL Workbench에서 연결	598
최대 MySQL 연결	599
.....	600
MySQL 기반 DB 인스턴스의 변경	600
사용 가능한 설정	602
관련 주제	607
MySQL DB 엔진 업그레이드	607
개요	607
메이저 버전 업그레이드	608
업그레이드 테스트	610
MySQL DB 인스턴스 업그레이드	611
자동 중지 시간을 최소화한 MySQL 데이터베이스 업그레이드	611
MySQL DB 스냅샷 업그레이드	612
MySQL DB 스냅샷 업그레이드	612
CLI	612
API	613
MySQL DB 인스턴스로 데이터 가져오기	613
개요	613
데이터 가져오기 고려 사항	615
Amazon RDS MySQL DB 인스턴스로 백업 복원	619
MySQL 또는 MariaDB DB에서 Amazon RDS MySQL 또는 MariaDB DB 인스턴스로 데이터 가져오기	626
자동 중지 시간을 단축하여 Amazon RDS MySQL MariaDB DB 인스턴스로 데이터 가져오기	628
임의의 소스에서 MySQL 또는 MariaDB DB 인스턴스로 데이터 가져오기	640
MySQL 복제 작업	644
MySQL 읽기 전용 복제본 작업	644
GTID 기반 복제 사용	650
Amazon RDS 외부에서 실행 중인 MySQL 또는 MariaDB 인스턴스를 사용한 복제	655
MySQL DB 인스턴스에서 데이터 내보내기	660
Amazon RDS 외부에 MySQL의 인스턴스 준비	660
복제 원본 준비	661
데이터베이스 복사	661
내보내기 완료	663
관련 주제	663

MySQL을 위한 옵션	663
MariaDB 감사 플러그	664
memcached	666
MySQL을 위한 공통 DBA 작업	669
세션 또는 쿼리 종료	670
현재 복제 오류 넘어가기	670
충돌 복구 시간 개선을 위한 InnoDB 테이블스페이스 작업	671
전역적 상태 이력 관리	672
알려진 문제 및 제한	673
일관되지 않은 InnoDB 버퍼 풀 크기	674
인덱스 병합 최적화가 잘못된 결과를 반환	674
로그 파일 크기	675
Amazon RDS DB 인스턴스에 대한 MySQL 파라미터 예외	675
MySQL 파일 크기 제한	675
Amazon RDS SQL의 MySQL 참조	676
개요	676
SQL 참조 규칙	677
mysql.rds_set_master_auto_position	678
mysql.rds_set_external_master	678
mysql.rds_set_external_master_with_delay	680
mysql.rds_set_external_master_with_auto_position	683
mysql.rds_reset_external_master	685
mysql.rds_import_binlog_ssl_material	685
mysql.rds_remove_binlog_ssl_material	687
mysql.rds_set_source_delay	687
mysql.rds_start_replication	688
mysql.rds_start_replication_until	689
mysql.rds_start_replication_until_gtid	690
mysql.rds_stop_replication	691
mysql.rds_skip_transaction_with_gtid	691
mysql.rds_skip_repl_error	692
mysql.rds_next_master_log	693
mysql.rds_innodb_buffer_pool_dump_now	694
mysql.rds_innodb_buffer_pool_load_now	695
mysql.rds_innodb_buffer_pool_load_abort	695
mysql.rds_set_configuration	695
mysql.rds_show_configuration	697
mysql.rds_kill	698
mysql.rds_kill_query	698
mysql.rds_rotate_general_log	699
mysql.rds_rotate_slow_log	699
mysql.rds_enable_gsh_collector	699
mysql.rds_set_gsh_collector	699
mysql.rds_disable_gsh_collector	700
mysql.rds_collect_global_status_history	700
mysql.rds_enable_gsh_rotation	700
mysql.rds_set_gsh_rotation	700
mysql.rds_disable_gsh_rotation	701
mysql.rds_rotate_global_status_history	701
Amazon RDS의 Oracle	702
공통 관리 작업	702
라이선싱	704
라이선스 포함	704
Bring Your Own License(BYOL)	704
Oracle 다중 AZ 배포 라이선스	704
Oracle 버전 간 마이그레이션	705
DB 인스턴스 클래스 지원	705

db.m3 및 db.r3 DB 인스턴스 클래스 지원 종단	707
db.m1 및 db.m2 DB 인스턴스 클래스 지원 종단	708
보안	708
SSL 지원	708
Oracle 12c	709
Oracle 12c 버전 12.2.0.1	709
Oracle 12c 버전 12.1.0.2	712
Oracle 데이터베이스 기능 지원	718
Oracle 데이터베이스 파라미터 지원	720
엔진 버전 관리	720
Oracle 11.2.0.2 운영 종단	720
Oracle 11.2.0.3 운영 종단	721
Oracle 12.1.0.1 운영 종단	721
방대한 페이지 사용	721
utl_http, utl_tcp 및 utl_smtp 사용	724
OEM, APEX, TDE 및 기타 옵션 사용	726
확장 데이터 형식 사용	726
새 DB 인스턴스에 대한 확장 데이터 유형 활성화	726
기존 DB 인스턴스에 대한 확장 데이터 유형 활성화	727
공개 동의어	727
Oracle을 실행하는 DB 인스턴스 생성	727
AWS Management 콘솔	728
CLI	731
API	732
사용 가능한 설정	732
관련 주제	736
Oracle을 실행 중인 DB 인스턴스에 연결	736
엔드포인트 찾기	736
SQL Developer	738
SQL*Plus	740
보안 그룹 고려 사항	741
전용 및 공유 서버 프로세스	741
문제 해결	741
.....	742
Oracle 기반 DB 인스턴스의 변경	742
사용 가능한 설정	744
Oracle sqlnet.ora 파라미터 수정	750
Oracle DB 엔진 업그레이드	752
개요	752
메이저 버전 업그레이드	753
マイ너 버전 업그레이드	754
SE2 업그레이드 경로	754
옵션 및 파라미터 그룹 고려 사항	754
업그레이드 테스트	755
Oracle DB 인스턴스 업그레이드	756
Oracle DB 스냅샷 업그레이드	756
AWS Management 콘솔	756
CLI	756
API	757
관련 주제	758
Oracle로 데이터 가져오기	758
Oracle SQL Developer를 사용한 가져오기	758
Oracle Data Pump를 사용한 가져오기	759
Oracle Export/Import 유ти리티	766
Oracle SQL*Loader	766
Oracle 구체화 보기	767
Oracle 읽기 전용 복제본 작업	768

Oracle용 읽기 전용 복제본 구성	768
Oracle을 사용한 읽기 전용 복제본 제한	769
Oracle 읽기 전용 복제본 문제 해결	770
Oracle 문자 집합	770
Oracle 옵션	772
Amazon S3통합	773
Application Express(APEX)	783
Enterprise Manager	791
Java 가상 머신(JVM)	798
레이블 보안	801
로케이터	803
멀티미디어	805
기본 네트워크 암호화(NNE)	807
Secure Sockets Layer(SSL)	810
공간	815
SQLT	817
Statspack	822
시간대	824
TDE(Transparent Data Encryption)	827
UTL_MAIL	828
XML DB	831
Oracle을 위한 공통 DBA 작업	832
시스템 작업	836
데이터베이스 작업	845
로그 작업	852
RMAN 작업	858
기타 작업	873
Oracle용 도구 및 타사 소프트웨어	875
설정	875
Oracle GoldenGate 사용	885
Oracle Repository Creation Utility 사용	895
Amazon RDS에서 Oracle에 Siebel Database 설치	899
Oracle 데이터베이스 엔진 출시 정보	902
Oracle 버전 12.2.0.1	902
Oracle 버전 12.1.0.2 및 11.2.0.4	902
데이터베이스 엔진: 12.2.0.1	903
데이터베이스 엔진: 12.1.0.2	905
데이터베이스 엔진: 11.2.0.4	924
Amazon RDS의 PostgreSQL	945
Amazon RDS의 PostgreSQL을 위한 공통 관리 작업	945
PostgreSQL 기반 DB 인스턴스 만들기	947
PostgreSQL DB 인스턴스 생성	948
CLI	952
API	953
PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기	953
pgAdmin을 사용해 PostgreSQL DB 인스턴스에 연결	954
psql을 사용해 PostgreSQL DB 인스턴스에 연결	955
연결 문제 해결	956
PostgreSQL을 실행 중인 DB 인스턴스 수정	956
사용 가능한 설정	958
PostgreSQL DB 엔진 업그레이드	963
개요	963
메이저 버전 업그레이드	964
マイ너 버전 자동 업그레이드	967
PostgreSQL 읽기 전용 복제본 작업	967
PostgreSQL을 사용한 읽기 전용 복제본 구성	967
PostgreSQL 읽기 전용 복제본 모니터링	968

PostgreSQL을 사용한 읽기 전용 복제본 제한	968
PostgreSQL 읽기 전용 복제본을 통한 복제 중단	968
PostgreSQL 읽기 전용 복제본의 문제 해결	969
Amazon RDS에서 PostgreSQL로 데이터 가져오기	970
Amazon EC2 인스턴스에서 PostgreSQL 데이터베이스 가져오기	972
\copy 명령을 사용하여 PostgreSQL DB 인스턴스의 테이블로 데이터 가져오기	973
RDS PostgreSQL로 S3 데이터 가져오기	974
PostgreSQL을 위한 일반 DBA 작업	984
역할 생성	984
PostgreSQL 데이터베이스 액세스 관리	985
PostgreSQL 파라미터 작업	985
PostgreSQL Autovacuum 사용	994
PostgreSQL DB 인스턴스에 대한 감사 로깅	1002
pgaudit 확장 작업	1003
pg_repack 확장 작업	1004
PostGIS 작업	1005
pgBadger를 사용한 PostgreSQL의 로그 분석	1007
pg_config의 내용 보기	1007
orafce 확장 작업	1008
postgres_fdw 확장으로 외부 데이터 액세스	1009
아웃바운드 네트워크 액세스에 사용자 지정 DNS 서버 사용	1009
암호 관리 제한	1010
데이터베이스 미리 보기 환경 작업	1011
미리 보기 환경에서 지원하지 않는 기능	1012
미리 보기 환경에서 지원되는 PostgreSQL 확장	1012
미리 보기 환경에서 새 DB 인스턴스 생성	1013
PostgreSQL 버전과 확장	1014
지원되는 PostgreSQL 데이터베이스 버전	1014
Supported 기능 및 확장자	1030
제한	1062
Amazon RDS의 제한 값	1062
Amazon RDS의 제약 조건 명명	1063
Amazon RDS의 파일 크기 제한	1065
Amazon RDS의 MySQL 파일 크기 제한	1065
Amazon RDS의 MariaDB 파일 크기 제한	1066
문제 해결	1068
DB 인스턴스에 연결할 수 없음	1068
DB 인스턴스 연결 테스트	1068
연결 인증 문제 해결	1069
보안 문제	1069
오류 메시지 "계정 속성을 불러오지 못했습니다. 일부 콘솔 기능이 손상되었을 수 있습니다."	1069
DB 인스턴스 소유자 역할 암호 재설정	1070
DB 인스턴스 종단 또는 재부팅	1070
파라미터 변경 사항이 적용 안 됨	1071
DB 인스턴스 스토리지 부족	1071
부족한 DB 인스턴스 용량	1072
MySQL 문제	1072
인덱스 병합 최적화가 잘못된 결과를 반환	1072
Read Replica 사이의 지연 문제 진단 및 해결	1073
MySQL 또는 MariaDB 읽기 복제 오류 진단 및 해결	1074
이진 로깅이 활성화된 상태에서 트리거를 생성하려면 SUPER 권한 필요	1075
특정 시점으로 복원 오류 진단 및 해결	1077
Slave Down 또는 Disabled 오류	1077
읽기 전용 복제본 만들기 실패 또는 치명적 오류 1236으로 복제 중단	1078
Oracle GoldenGate 문제	1078
충분한 시간 동안 로그 보존	1078
SQL Server DB 인스턴스에 연결할 수 없음	1078

PostgreSQL DB 인스턴스에 연결할 수 없음	1079
백업 보존을 0으로 설정 불가	1079
Amazon RDS API 참조	1080
Query API 사용	1080
쿼리 파라미터	1080
쿼리 요청 인증	1080
애플리케이션 문제 해결	1080
오류 검색	1081
문제 해결 팁	1081
문서 이력	1082
이전 업데이트	1094

Amazon Relational Database Service(Amazon RDS)란 무엇입니까?

Amazon Relational Database Service(Amazon RDS)는 클라우드에서 관계형 데이터베이스를 더 쉽게 설치, 운영 및 확장할 수 있는 웹 서비스입니다. 이 서비스는 산업 표준 관계형 데이터베이스를 위한 경제적이고 크기 조절이 가능한 용량을 제공하고 공통 데이터베이스 관리 작업을 관리합니다.

Note

이 지침은 Aurora 이외의 다른 Amazon RDS 데이터베이스 엔진에도 적용됩니다. Amazon Aurora 사용에 대한 자세한 정보는 [Amazon Aurora 사용 설명서](#)를 참조하십시오.

Amazon RDS의 개요

관리되는 관계형 데이터베이스 서비스가 필요한 이유는 무엇일까요? Amazon RDS가 어렵거나 지루한 관계형 데이터베이스 관리 작업을 대다수 대신하기 때문입니다.

- 서버를 구입하면 CPU, 메모리, 스토리지 및 IOPS가 모두 한데 묶여 제공됩니다. Amazon RDS를 사용하면 이 모두가 따로 분할되므로 독립적으로 확장할 수 있습니다. CPU가 더 많이 필요하거나 IOPS가 더 적게 필요하거나 스토리지가 더 많이 필요할 경우 쉽게 할당할 수 있습니다.
- Amazon RDS는 백업, 소프트웨어 패치, 자동 장애 감지 및 복구를 관리합니다.
- 관리형 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 shell 액세스를 제공하지 않으며, 고급 권한을 필요로 하는 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다.
- 필요할 때 자동화된 백업을 수행하거나 고유한 백업 스냅샷을 수동으로 만들 수 있습니다. 이러한 백업을 사용하여 데이터베이스를 복원할 수 있습니다. Amazon RDS 복원 프로세스는 안정적이고 효율적입니다.
- 기본 인스턴스 및 문제 발생 시 장애 조치를 수행할 수 있는 동기식 보조 인스턴스에서 가용성을 높일 수 있습니다. MySQL, MariaDB 또는 PostgreSQL 읽기 전용 복제본을 사용해 읽기 조정을 높일 수도 있습니다.
- 이미 친숙한 MySQL, MariaDB, PostgreSQL, Oracle 및 Microsoft SQL Server 같은 데이터베이스 제품을 사용할 수 있습니다.
- 데이터베이스 패키지의 보안 외에도 AWS Identity and Access Management(IAM)을 사용해 사용자 및 권한을 정의하는 방법으로 RDS 데이터베이스에 액세스할 수 있는 사용자를 제어할 수 있습니다. 데이터베이스를 가상 사설 클라우드에 넣어 데이터베이스를 보호할 수도 있습니다.

AWS 제품 및 서비스를 처음 사용하는 경우 자세한 정보는 다음 자료를 참조하십시오.

- 모든 AWS 제품의 개요는 [클라우드 컴퓨팅이란?](#)을 참조하십시오.
- Amazon Web Services는 수많은 데이터베이스 서비스를 제공합니다. 환경에 가장 적합한 서비스에 대한 지침은 [AWS에서 데이터베이스 실행](#)을 참조하십시오.

DB 인스턴스

Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. DB 인스턴스는 클라우드에 있는 격리된 데이터베이스 환경입니다. DB 인스턴스에는 사용자가 만든 여러 데이터베이스가 포함될 수 있으며, 독립 실행형 데

이터베이스 인스턴스와 함께 사용하는 동일한 도구 및 애플리케이션을 사용해 액세스할 수 있습니다. AWS Command Line Interface, Amazon RDS API 또는 AWS Management 콘솔을 사용해 DB 인스턴스를 생성하고 수정할 수 있습니다.

각 DB 인스턴스는 DB 엔진을 실행합니다. Amazon RDS는 현재 MySQL, MariaDB, PostgreSQL, Oracle 및 Microsoft SQL Server DB 엔진을 지원합니다. 각 DB 엔진에는 지원되는 고유한 기능이 있으며, DB 엔진의 각 버전에는 특정 기능이 포함될 수 있습니다. 또한 DB 엔진마다 관리하는 데이터베이스의 동작을 제어하는 DB 파라미터 그룹에 파라미터 집합이 있습니다.

DB 인스턴스의 계산 및 메모리 용량은 해당 DB 인스턴스 클래스에 의해 결정됩니다. 사용자의 요구 사항에 가장 잘 맞는 DB 인스턴스를 선택할 수 있습니다. 시간이 지나면서 요구 사항이 바뀌면 DB 인스턴스를 변경할 수 있습니다. 자세한 정보는 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.

Note

DB 인스턴스 클래스에 대한 요금 정보는 [Amazon RDS](#) 제품 페이지의 요금 단원을 참조하십시오.

DB 인스턴스 스토리지는 마그네트, 범용(SSD) 및 프로비저닝된 IOPS(PIOPS) 등 세 가지 유형이 제공됩니다. 이 세 가지 유형은 성능 특성과 가격이 다르므로 데이터베이스 요건에 따라 스토리지 성능과 비용을 조정할 수 있습니다. DB 인스턴스는 각각 스토리지 유형과 지원하는 데이터베이스 엔진에 따라 최소/최대 스토리지 요구 사항이 있습니다. 데이터베이스에 확장할 수 있는 공간이 있고 DB 엔진의 기능에 콘텐츠를 쓰고 항목을 기록할 수 있는 공간이 있도록 충분한 스토리지를 확보하는 것이 중요합니다. 자세한 정보는 [Amazon RDS DB 인스턴스 스토리지 \(p. 101\)](#) 단원을 참조하십시오.

Amazon Virtual Private Cloud(VPC) 서비스를 사용해 가상 사설 클라우드에서 DB 인스턴스를 실행할 수 있습니다. 가상 프라이빗 클라우드를 사용하여 가상 네트워킹 환경을 제어하는 경우 고유한 IP 주소 범위를 선택하고, 서브넷을 만들고, 라우팅 및 액세스 제어 목록을 구성할 수 있습니다. Amazon RDS의 기본 기능은 VPC를 사용하는지, Amazon RDS가 백업, 소프트웨어 패칭, 자동 오류 감지 및 복구를 관리하는지 여부에 상관없이 동일합니다. DB 인스턴스를 VPC에서 실행하는 데는 추가 비용이 들지 않습니다. VPC 및 RDS에 대한 자세한 정보는 [Amazon Virtual Private Cloud\(VPC\)](#) 및 [Amazon RDS \(p. 406\)](#) 단원을 참조하십시오.

Amazon RDS는 DB 인스턴스를 기준으로 시간을 동기화하는 데 NTP(Network Time Protocol)를 사용합니다.

리전 및 가용 영역

Amazon 클라우드 컴퓨팅 리소스는 전 세계 여러 리전의 가용성이 높은 데이터 센터 시설에 하우징됩니다 (예: 북미, 유럽 또는 아시아). 각 데이터 센터 위치를 지역이라고 합니다.

리전마다 가용 영역 또는 AZ라는 고유한 위치가 여러 개 포함됩니다. 각 가용 영역은 다른 가용 영역의 오류로부터 격리되고 같은 리전의 다른 가용 영역에 경제적이고 지연 시간이 낮은 네트워크 연결을 제공하도록 엔지니어링됩니다. 별도의 가용 영역에서 인스턴스를 시작함으로써 단일 위치에서 장애가 발생할 경우 애플리케이션을 보호할 수 있습니다. 자세한 정보는 [리전 및 가용 영역 \(p. 99\)](#) 단원을 참조하십시오.

여러 가용 영역에서 DB 인스턴스를 실행할 수 있습니다. 다중 AZ 배포라는 옵션입니다. 이 옵션을 선택하면 Amazon이 다른 가용 영역에서 보조 예비 DB 인스턴스를 자동으로 프로비저닝하고 유지합니다. 기본 DB 인스턴스는 가용 영역에서 보조 인스턴스로 동기식으로 복제되어 데이터 중복, 장애 조치 지원을 제공하고 I/O 중지를 제거하고 시스템 백업 중에 지연 시간 스파이크를 최소화합니다. 자세한 정보는 [Amazon RDS를 위한 고가용성\(다중 AZ\) \(p. 109\)](#) 단원을 참조하십시오.

보안

보안 그룹은 DB 인스턴스에 대한 액세스를 제어합니다. 사용자가 지정한 IP 주소 범위 또는 Amazon EC2 인스턴스에 액세스할 수 있도록 허용하는 방법으로 제어합니다.

Amazon RDS는 DB 보안 그룹, VPC 보안 그룹 및 EC2 보안 그룹을 사용합니다. 간단히 말해 DB 보안 그룹은 VPC 외부의 DB 인스턴스에 대한 액세스를 제어하고, VPC 보안 그룹은 VPC 내부의 DB 인스턴스에 대한 액세스를 제어하고, Amazon EC2 보안 그룹은 EC2 인스턴스에 대한 액세스를 제어하며, DB 인스턴스와 함께 사용할 수 있습니다. 보안 그룹에 대한 자세한 정보는 [Amazon RDS의 보안 구성 \(p. 339\)](#) 단원을 참조하십시오.

Amazon RDS DB 인스턴스 모니터링

DB 인스턴스의 성능과 상태를 추적할 수 있는 여러 가지 방법이 있습니다. 무료 Amazon CloudWatch 서비스를 사용해 DB 인스턴스의 성능과 상태를 모니터링할 수 있습니다. 성능 차트는 Amazon RDS 콘솔에 나와 있습니다. Amazon RDS 이벤트를 구독해 DB 인스턴스, DB 스냅샷, DB 파라미터 그룹 또는 DB 보안 그룹이 변경될 때마다 알림을 받을 수 있습니다. 자세한 정보는 [Amazon RDS 모니터링 \(p. 228\)](#) 단원을 참조하십시오.

Amazon RDS 인터페이스

Amazon RDS와 상호 작용하는 방법에는 여러 가지가 있습니다.

AWS Management 콘솔

AWS Management 콘솔은 간단한 웹 기반 사용자 인터페이스입니다. 콘솔에서 프로그래밍 없이 DB 인스턴스를 관리할 수 있습니다. Amazon RDS 콘솔에 액세스하려면 <https://console.aws.amazon.com/rds/>에서 AWS Management 콘솔에 로그인한 후 Amazon RDS 콘솔을 엽니다.

명령행 인터페이스

Amazon RDS API는 AWS Command Line Interface(AWS CLI)를 사용하여 대화식으로 액세스할 수 있습니다. AWS CLI를 설치하려면 [AWS 명령줄 인터페이스 설치](#)를 참조하십시오. RDS용 AWS CLI를 사용하려면 [Amazon RDS용 AWS Command Line Interface 레퍼런스](#)를 참조하십시오.

Amazon RDS를 사용한 프로그래밍

개발자라면 프로그래밍 방식으로 Amazon RDS에 액세스할 수 있습니다. 자세한 정보는 [Amazon RDS 애플리케이션 프로그래밍 인터페이스\(API\) 레퍼런스 \(p. 1080\)](#) 단원을 참조하십시오.

애플리케이션을 개발하는 경우에는 AWS 소프트웨어 개발 키트(SDK) 중 하나를 사용하는 것이 좋습니다. AWS SDK가 인증, 재시도 로직, 오류 처리 등 저레벨 정보를 처리하기 때문에 개발자는 애플리케이션 로직에 더욱 집중할 수 있습니다. SDK는 다양한 언어로 제공됩니다. 자세한 정보는 [Amazon Web Services용 도구](#) 단원을 참조하십시오.

그 밖에도 AWS는 라이브러리, 샘플 코드, 자습서 등 더욱 쉽게 시작하는 데 도움이 될 수 있는 리소스를 제공합니다. 자세한 정보는 [샘플 코드 및 라이브러리](#) 단원을 참조하십시오.

Amazon RDS에 대한 요금이 부과되는 방법

Amazon RDS를 사용하는 경우 온디맨드 DB 인스턴스 또는 예약된 DB 인스턴스를 선택하여 사용할 수 있습니다. 자세한 정보는 [Amazon RDS에 대한 DB 인스턴스 결제 \(p. 183\)](#) 단원을 참조하십시오.

Amazon RDS 요금에 대한 자세한 정보는 [Amazon RDS 제품 페이지](#)를 참조하십시오.

다음 단계

이전 단계에서는 RDS에서 제공하는 기본 인프라 구성 요소를 소개했습니다. 다음으로 무엇을 해야 할까요?

시작하기

[Amazon RDS 시작하기 \(p. 10\)](#) 섹션의 지침을 사용해 DB 인스턴스를 만드십시오.

데이터베이스 엔진별 항목

다음 섹션에서 특정 DB 엔진별 정보를 검토할 수 있습니다.

- [Amazon RDS MariaDB \(p. 426\)](#)
- [Amazon RDS의 Microsoft SQL Server \(p. 477\)](#)
- [Amazon RDS의 MySQL \(p. 576\)](#)
- [Amazon RDS의 Oracle \(p. 702\)](#)
- [Amazon RDS의 PostgreSQL \(p. 945\)](#)

Amazon RDS 설정

아래에는 처음으로 Amazon Relational Database Service(Amazon RDS)를 설정하는 방법에 대한 자세한 정보가 나와 있습니다. AWS 계정이 이미 있고 Amazon RDS 요구 사항을 알고 있으며 IAM 및 VPC 보안 그룹에 모두 기본값을 사용하려는 경우에는 [시작하기 \(p. 4\)](#) 단원으로 건너뛰십시오.

Amazon Web Services(AWS)에 대해 알아야 할 사항:

- AWS에 가입하면 AWS 계정은 Amazon RDS를 포함한 AWS의 모든 서비스에 자동으로 액세스합니다. 하지만 사용한 서비스에 대해서만 청구됩니다.
- Amazon RDS에서는 활성인 RDS 인스턴스에 대해서만 비용을 지불합니다. 생성한 DB 인스턴스는 실시간으로 활성화됩니다(샌드박스에서 실행되지 않음). 종료하기 전까지 해당 인스턴스에 대해 표준 사용 요금이 발생합니다. Amazon RDS 사용 요금에 대한 자세한 정보는 [Amazon RDS 제품 페이지](#)를 참조하십시오.

주제

- [AWS에 가입 \(p. 5\)](#)
- [IAM 사용자 생성 \(p. 5\)](#)
- [요구 사항 결정 \(p. 7\)](#)
- [보안 그룹을 생성하여 VPC 내부의 DB 인스턴스에 대한 액세스를 제공 \(p. 8\)](#)

AWS에 가입

이미 AWS 계정이 이미 있는 경우 다음 [IAM 사용자 생성 \(p. 5\)](#) 단원으로 건너뛰십시오.

AWS 계정이 없는 경우에는 아래 절차를 사용하여 계정을 만들 수 있습니다. AWS를 처음 사용하는 고객에게는 Amazon RDS를 무료로 사용해볼 수 있습니다. 자세한 정보는 [AWS 프리 티어](#) 단원을 참조하십시오.

새 AWS 계정을 만들려면

1. <https://aws.amazon.com/>을 열고 Create an AWS Account(AWS 계정 생성)를 선택합니다.

Note

전에 AWS 계정 루트 사용자 자격 증명을 사용하여 AWS Management 콘솔에 로그인한 적이 있는 경우 Sign in to a different account(다른 계정으로 로그인)를 선택합니다. 전에 IAM 자격 증명을 사용하여 콘솔에 로그인한 적이 있는 경우 Sign-in using root account credentials(루트 계정 자격 증명으로 로그인)를 선택합니다. 그런 다음 Create a new AWS account(새 AWS 계정 생성)를 선택합니다.

2. 온라인 지시 사항을 따릅니다.

등록 절차 중 전화를 받고 전화 키패드를 사용하여 확인 코드를 입력하는 과정이 있습니다.

IAM 사용자 생성

AWS 계정을 생성하고 AWS Management 콘솔에 연결한 후에는 AWS Identity and Access Management(IAM) 사용자를 생성할 수 있습니다. AWS 루트 계정을 사용하여 로그인하는 대신에, Amazon RDS에서 IAM 관리 사용자를 사용하는 것이 좋습니다.

이렇게 하는 한 가지 방법은 새 IAM 사용자를 생성하고 이 사용자에게 관리자 권한을 부여하는 것입니다. 또는 Amazon RDS 관리 권한을 사용하여 IAM 그룹에 기존 IAM 사용자를 추가할 수 있습니다. 그러면 사용자의 자격 증명을 사용하여 특수 URL에서 AWS에 액세스할 수 있습니다.

AWS에 가입했지만 IAM 사용자를 생성하지 않았다면 IAM 콘솔에서 생성할 수 있습니다.

IAM 사용자를 직접 생성하여 관리자 그룹에 추가하려면

1. <https://console.aws.amazon.com/iam/>에서 AWS 계정 루트 사용자 이메일 주소 및 암호를 사용하여 IAM 콘솔에 AWS 계정 루트 사용자 로그인합니다.

Note

Administrator IAM 사용자를 사용하는 아래 모범 사례를 준수하고, 루트 사용자 자격 증명을 안전하게 보관해 두는 것이 좋습니다. 몇 가지 [계정 및 서비스 관리 작업](#)을 수행하려면 반드시 루트 사용자로 로그인해야 합니다.

2. 콘솔의 탐색 창에서 사용자를 선택한 후 사용자 추가를 선택합니다.
3. [User name]에 **Administrator**를 입력합니다.
4. AWS Management 콘솔 액세스 옆의 확인란을 선택하고 Custom password(사용자 지정 암호)를 선택한 다음 텍스트 상자에 새로운 사용자의 암호를 입력합니다. 필요하다면 [Require password reset]을 선택하여 다음에 사용자가 로그인할 때 새 암호를 생성하도록 설정할 수 있습니다.
5. Next: Permissions(다음: 권한)를 선택합니다.
6. 권한 설정 페이지에서 그룹에 사용자 추가(Add user to group)를 선택합니다.
7. Create group을 선택합니다.
8. 그룹 생성 대화 상자의 그룹 이름에 **Administrators**를 입력합니다.
9. 필터 정책에서 AWS managed - job function(AWS 관리 - 직무)의 확인란을 선택합니다.
10. 정책 목록에서 AdministratorAccess 옆의 확인란을 선택합니다. 그런 다음 Create group을 선택합니다.
11. 그룹 목록으로 돌아가 새로 만든 그룹 옆의 확인란을 선택합니다. 목록에서 그룹을 확인하기 위해 필요한 경우 Refresh를 선택합니다.
12. 다음: 태그를 선택하고 키-값 페어로 태그를 첨부하여 사용자에게 메타데이터를 추가합니다.
13. Next: Review를 선택하여 새 사용자에 추가될 그룹 멤버십의 목록을 확인합니다. 계속 진행할 준비가 되었으면 Create user를 선택합니다.

이제 동일한 절차에 따라 그룹이나 사용자를 추가 생성하여 AWS 계정 리소스에 액세스할 수 있는 권한을 사용자에게 부여할 수 있게 되었습니다. 특정 AWS 리소스에 대한 사용자의 권한을 제한하는 정책을 사용하는 방법을 알아보려면 [액세스 관리](#) 및 [정책 예제](#)를 참조하십시오.

이 새로운 IAM 사용자로 로그인하려면 먼저 AWS Management 콘솔에서 로그아웃합니다. 그런 다음 아래 URL을 사용합니다. 여기서 **your_aws_account_id**는 하이픈이 없는 AWS 계정 번호입니다. 예를 들어 사용자의 AWS 계정 번호가 1234-5678-9012인 경우 AWS 계정 ID는 123456789012입니다.

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

방금 생성한 IAM 사용자 이름과 암호를 입력합니다. 로그인하면 탐색 모음에 "your_user_name @ your_aws_account_id"가 표시됩니다.

로그인 페이지의 URL에 AWS 계정 ID가 포함되지 않게 하려면 계정 별칭을 생성할 수 있습니다. IAM 대시보드에서 사용자 지정을 선택하고 회사 이름 등의 별칭을 입력합니다. 계정 별칭 생성 후 로그인할 때는 다음 URL을 사용합니다.

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

본인 계정의 IAM 사용자 로그인 링크를 확인하려면 IAM 콘솔을 열고 대시보드의 [AWS Account Alias] 아래를 체크합니다.

또한 AWS 계정에 대한 액세스 키를 생성할 수 있습니다. 이러한 액세스 키를 사용하여 AWS Command Line Interface(AWS CLI) 또는 Amazon RDS API를 통해 AWS에 액세스할 수 있습니다. 자세한 정보는 [AWS 계정에 대한 액세스 키 관리](#), [AWS 명령줄 인터페이스 설치 및 Amazon RDS API 참조](#) 단원을 참조하십시오.

요구 사항 결정

Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. DB 인스턴스에서는 데이터베이스를 생성합니다. DB 인스턴스는 엔드포인트라고 하는 네트워크 주소를 할당합니다. 애플리케이션은 이 엔드포인트를 사용하여 DB 인스턴스에 연결합니다. DB 인스턴스를 생성할 때 스토리지, 메모리, 데이터베이스 엔진 및 버전, 네트워크 구성, 보안, 유지 관리 기간 등의 세부 정보를 지정합니다. 보안 그룹을 통해 DB 인스턴스에 대한 네트워크 액세스를 제어할 수 있습니다.

DB 인스턴스와 보안 그룹을 생성하기 전에 DB 인스턴스 및 네트워크 필요를 알아야 합니다. 고려해야 할 몇 가지 중요 사항은 다음과 같습니다.

- 리소스 요구 사항 – 애플리케이션 또는 서비스의 메모리 및 프로세서 요구 사항은 무엇입니까? 이러한 설정을 사용하면 어떤 DB 인스턴스 클래스를 사용할지를 결정하는 데 도움이 됩니다. DB 인스턴스 클래스에 대한 사양은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.
- VPC, 서브넷 및 보안 그룹 – DB 인스턴스는 대부분의 경우 가상 사설 클라우드(VPC) 안에 있습니다. DB 인스턴스에 연결하려면 보안 그룹 규칙을 설정해야 합니다. 이러한 규칙은 사용하는 VPC 종류와 사용 방식(기본 VPC, 사용자 정의 VPC 또는 VPC 외부)에 따라 다르게 설정됩니다.

Note

일부 레거시 계정은 VPC를 사용하지 않습니다. 새 AWS 리전에 액세스하고 있거나 새 RDS 사용자인 경우(2013년 이후) 대부분 VPC 내부에서 DB 인스턴스를 생성합니다.

계정이 특정 AWS 리전에서 기본 VPC를 가지고 있는지 확인하는 방법은 [EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 \(p. 407\)](#) 단원을 참조하십시오.

다음은 각 VPC 옵션의 규칙을 설명한 목록입니다.

- 기본 VPC – 현재 리전에서 AWS 계정에 현재 AWS 리전의 기본 VPC가 있는 경우 해당 VPC를 구성하여 DB 인스턴스를 지원할 수 있습니다. DB 인스턴스 생성 시 기본 VPC를 지정할 경우 다음을 수행합니다.
 - VPC 보안 그룹을 생성하여 데이터베이스를 이용해 애플리케이션 또는 서비스에서 Amazon RDS DB 인스턴스까지 연결 권한을 부여합니다. [Amazon EC2 API](#)를 사용하거나, 혹은 VPC 콘솔에서 보안 그룹 옵션을 선택하여 VPC 보안 그룹을 생성합니다. 자세한 정보는 [4단계: VPC 보안 그룹 만들기 \(p. 419\)](#) 단원을 참조하십시오.
 - 기본 DB 서브넷 그룹을 지정합니다. 이 AWS 리전에서 처음 DB 인스턴스를 생성하는 경우에는 Amazon RDS가 DB 인스턴스 생성과 함께 기본 DB 서브넷 그룹을 생성합니다.
- 사용자 정의 VPC – DB 인스턴스 생성 시 사용자 정의 VPC를 지정할 경우 다음 사항에 유의해야 합니다.
 - VPC 보안 그룹을 생성하여 데이터베이스를 이용해 애플리케이션 또는 서비스에서 Amazon RDS DB 인스턴스까지 연결 권한을 부여해야 합니다. [Amazon EC2 API](#)를 사용하거나, 혹은 VPC 콘솔에서 보안 그룹 옵션을 선택하여 VPC 보안 그룹을 생성합니다. 자세한 정보는 [4단계: VPC 보안 그룹 만들기 \(p. 419\)](#) 단원을 참조하십시오.
 - VPC가 DB 인스턴스를 호스팅하려면 별도의 가용 영역에서 각각 최소 2개 이상씩 서브넷을 구성하는 등 특정 요구 사항을 충족해야 합니다. 자세한 정보는 [Amazon Virtual Private Cloud\(VPC\) 및 Amazon RDS \(p. 406\)](#) 단원을 참조하십시오.
 - DB 서브넷 그룹을 지정하여 DB 인스턴스에서 VPC를 사용할 서브넷을 정의해야 합니다. 자세한 정보는 [VPC에서 DB 인스턴스를 사용한 작업 \(p. 415\)](#)에서 DB 서브넷 그룹을 참조하십시오.

- VPC 없음- AWS 계정에 기본 VPC가 없고 사용자 정의 VPC를 지정하지 않는 경우 DB 보안 그룹을 생성합니다. DB 보안 그룹은 애플리케이션 또는 유ти리티를 실행하는 디바이스와 Amazon RDS 인스턴스에서 연결하여 DB 인스턴스의 데이터베이스에 액세스할 수 있는 권한을 부여합니다. 자세한 정보는 [DB 보안 그룹 작업\(EC2-Classic 플랫폼\) \(p. 394\)](#) 단원을 참조하십시오.
- 높은 가용성: 장애 조치 지원이 필요습니까? 에서 다중 AZ 배포는 기본 DB 인스턴스를 생성하고 장애 조치 지원을 위해 다른 가용 영역에 보조 예비 DB 인스턴스를 생성합니다. 고가용성을 유지하기 위해 프로덕션 워크로드에는 다중 AZ 배포를 권장합니다. 개발 및 테스트 목적으로는 비 다중 AZ 배포를 사용할 수 있습니다. 자세한 정보는 [Amazon RDS를 위한 고가용성\(다중 AZ\) \(p. 109\)](#) 단원을 참조하십시오.
- IAM 정책: AWS 계정에 Amazon RDS 작업을 수행하는 데 필요한 권한을 부여하는 정책이 있습니까? IAM 자격 증명을 사용하여 AWS에 연결하는 경우 IAM 계정에는 Amazon RDS 작업을 수행하는 데 필요한 권한을 부여하는 IAM 정책이 있어야 합니다. 자세한 정보는 [인증 및 액세스 제어 \(p. 339\)](#) 단원을 참조하십시오.
- 개방 포트: 데이터베이스가 어떤 TCP/IP 포트에서 수신 대기합니까? 일부 기업에서는 방화벽이 데이터베이스 엔진의 기본 포트 연결을 차단하는 경우도 있습니다. 이처럼 기업 방화벽이 기본 포트를 차단할 경우 새로운 DB 인스턴스를 생성할 때 다른 포트를 선택해야 합니다. 단, 지정 포트에서 수신 대기할 DB 인스턴스를 생성하는 경우 해당 DB 인스턴스를 수정하여 포트를 변경할 수 있습니다.
- AWS 리전: 데이터베이스를 구성하려고 하는 AWS 리전은 어디입니까? 애플리케이션이나 웹 서비스에 가깝게 데이터베이스를 구성하면 네트워크 지연 시간을 줄일 수 있습니다.
- DB 디스크 하위 시스템: 스토리지 요구 사항은 무엇입니까? Amazon RDS는 다음 세 가지 스토리지 유형을 제공합니다.
 - 마그네틱(표준 스토리지)
 - 범용(SSD)
 - 프로비저닝된 IOPS(PIOPS)

마그네틱 스토리지는 I/O 요구 사항이 보통 수준이거나 가끔씩 집중적으로 발생하는 애플리케이션에 적합한 비용 효율적인 스토리지입니다. gp2라고도 불리는 범용 SSD 기반 스토리지는 디스크 기반 스토리지보다 액세스 속도가 더욱 빠릅니다. 프로비저닝된 IOPS 스토리지는 성능에 민감하고 임의 액세스 I/O 처리량이 일정한 I/O 집약적 워크로드, 특히 데이터베이스 워크로드 요구 사항을 충족하도록 설계되었습니다. Amazon RDS 스토리지에 대한 자세한 정보는 [Amazon RDS DB 인스턴스 스토리지 \(p. 101\)](#) 단원을 참조하십시오.

보안 그룹과 DB 인스턴스 생성에 필요한 정보를 확인하였으면 다음 단계로 진행합니다.

보안 그룹을 생성하여 VPC 내부의 DB 인스턴스에 대한 액세스를 제공

VPC 보안 그룹은 VPC에서 실행되는 DB 인스턴스에 대한 액세스를 제공합니다. 이들은 연결된 DB 인스턴스에 대한 방화벽 역할을 하여 인스턴스 수준에서 인바운드 트래픽과 아웃바운드 트래픽을 모두 제어합니다. 기본적으로 DB 인스턴스는 DB 인스턴스를 보호하는 방화벽 및 기본 보안 그룹과 함께 생성됩니다.

DB 인스턴스에 연결하려면 먼저 연결하는 데 사용할 수 있는 규칙을 보안 그룹에 추가해야 합니다. 네트워크 및 구성 정보를 사용하여 DB 인스턴스에 액세스하는 데 사용할 수 있는 규칙을 생성합니다.

Note

레거시 DB 인스턴스가 2013년 3월 이전에 생성되었고 VPC 안에 있지 않은 경우 연결된 보안 그룹이 없을 수 있습니다. DB 인스턴스가 이 날짜 이후에 생성된 경우 기본 VPC 내부에 있을 수 있습니다.

예를 들어 VPC의 DB 인스턴스에서 데이터베이스에 액세스하는 애플리케이션이 있다고 가정해 보겠습니다. 이 경우, 애플리케이션이 데이터베이스에 액세스하는 데 사용할 포트 범위와 IP 주소를 지정하는 사용자

지정 TCP 규칙을 추가해야 합니다. Amazon EC2 인스턴스에서 실행하는 애플리케이션이 있다면 Amazon EC2 인스턴스에 대해 설정한 VPC 또는 EC2 보안 그룹을 사용할 수 있습니다.

VPC 보안 그룹의 생성 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/vpc>에서 Amazon VPC 콘솔을 엽니다.
2. AWS Management 콘솔의 오른쪽 위 모서리에서 VPC 보안 그룹 및 DB 인스턴스를 만들 AWS 리전을 선택합니다. 해당 AWS 리전의 리소스 목록에 1개 이상의 VPC와 몇 개의 서브넷이 표시되어야 합니다. 그렇지 않으면 해당 AWS 리전에 기본 VPC가 없는 것입니다.
3. 탐색 창에서 [Security Groups]를 선택합니다.
4. [Create Security Group]를 선택합니다.
5. [Create Security Group] 창에서 보안 그룹에 대한 [Name tag], [Group name] 및 [Description] 값을 입력합니다. [VPC]에서 DB 인스턴스를 생성할 VPC를 선택합니다. [Yes, Create]를 선택합니다.
6. 생성한 VPC 보안 그룹이라고 해도 여전히 선택할 수 있어야 합니다. 그렇지 않으면 목록에서 찾아서 선택합니다. 콘솔 화면 하단 세부 정보 창에 보안 그룹 세부 정보를 비롯해 인바운드 및 아웃바운드 규칙 작업에 대한 탭이 표시됩니다. 인바운드 규칙 탭을 선택합니다.
7. Inbound Rules 탭에서 [Edit]를 선택합니다.
 - a. [Type]의 경우 [Custom TCP Rule]을 선택합니다.
 - b. [Port Range]에는 DB 인스턴스에 사용할 포트 값을 입력합니다.
 - c. [Source]에서 인스턴스에 액세스할 IP 주소 범위(CIDR 값)를 입력하거나 보안 그룹 이름을 선택합니다.
8. IP 주소를 추가하거나 다른 포트 범위를 입력해야 한다면 [Add another rule]을 클릭합니다.
9. (선택 사항) [Outbound Rules] 탭을 사용하여 아웃바운드 트래픽 규칙을 추가합니다. 기본적으로 모든 아웃바운드 트래픽이 허용됩니다.

방금 생성한 VPC 보안 그룹을 DB 인스턴스 생성 시 보안 그룹으로 사용할 수 있습니다. DB 인스턴스를 VPC 내부에 배포하지 않는 경우에는 [DB 보안 그룹 작업\(EC2-Classic 플랫폼\) \(p. 394\)](#)를 참조하여 DB 인스턴스를 생성하면서 사용할 DB 보안 그룹까지 생성합니다.

Note

기본 VPC를 사용하는 경우 VPC의 모든 서브넷을 포함하는 기본 서브넷 그룹이 자동으로 생성됩니다. DB 인스턴스를 생성할 때 기본 VPC를 선택하고 DB 서브넷 그룹의 기본값을 사용할 수 있습니다.

설정 요구 사항을 완료한 후에는 요구 사항과 보안 그룹을 사용하여 DB 인스턴스를 시작할 수 있습니다. DB 인스턴스를 만드는 방법에 대한 자세한 정보는 다음 표의 관련 문서를 참조하십시오.

데이터베이스 엔진	설명서
MariaDB	MariaDB DB 인스턴스 생성 및 MariaDB DB 인스턴스의 데이터베이스에 연결 (p. 10)
Microsoft SQL Server	Microsoft SQL Server DB 인스턴스 생성 및 DB 인스턴스에 연결 (p. 17)
MySQL	MySQL DB 인스턴스를 만들고 MySQL DB 인스턴스의 데이터베이스에 연결 (p. 26)
Oracle	Oracle DB 인스턴스를 만들고 Oracle DB 인스턴스의 데이터베이스에 연결 (p. 33)
PostgreSQL	PostgreSQL DB 인스턴스 생성 및 PostgreSQL DB 인스턴스의 데이터베이스에 연결 (p. 39)

Amazon RDS 시작하기

이 섹션에서는 Amazon Relational Database Service(Amazon RDS)를 사용하여 DB 인터페이스를 만들고 여기에 연결하는 방법을 보여 줍니다. MySQL, Oracle, PostgreSQL, Microsoft SQL Server 또는 MariaDB를 사용하는 DB 인스턴스를 생성하거나 시작할 수 있습니다.

Important

DB 인스턴스를 생성하거나 DB 인스턴스에 연결하려면 먼저 [Amazon RDS 설정 \(p. 5\)](#) 섹션의 작업을 완료해야 합니다.

DB 인스턴스를 생성한 후 DB 인스턴스의 데이터베이스에 연결하는 작업은 DB 엔진마다 조금씩 다릅니다. 따라서 DB 인스턴스를 생성하여 연결하는 데 필요한 세부 정보를 감안하여 사용할 DB 엔진을 선택해야 합니다. DB 인스턴스를 생성하여 연결하고 나면 DB 인스턴스 삭제를 도와주는 지침을 이용할 수 있습니다.

주제

- [MariaDB DB 인스턴스 생성 및 MariaDB DB 인스턴스의 데이터베이스에 연결 \(p. 10\)](#)
- [Microsoft SQL Server DB 인스턴스 생성 및 DB 인스턴스에 연결 \(p. 17\)](#)
- [MySQL DB 인스턴스를 만들고 MySQL DB 인스턴스의 데이터베이스에 연결 \(p. 26\)](#)
- [Oracle DB 인스턴스를 만들고 Oracle DB 인스턴스의 데이터베이스에 연결 \(p. 33\)](#)
- [PostgreSQL DB 인스턴스 생성 및 PostgreSQL DB 인스턴스의 데이터베이스에 연결 \(p. 39\)](#)
- [자습서: 웹 서버 및 Amazon RDS 데이터베이스 생성 \(p. 46\)](#)

MariaDB DB 인스턴스 생성 및 MariaDB DB 인스턴스의 데이터베이스에 연결

MariaDB DB 인스턴스를 만드는 가장 쉬운 방법은 Amazon RDS 콘솔을 사용하는 것입니다. DB 인스턴스를 만든 후에는 mysql 같은 명령줄 도구나 HeidiSQL 같은 표준 그래픽 도구 사용해 DB 인스턴스에 있는 데이터베이스에 연결할 수 있습니다.

Important

DB 인스턴스를 생성하거나 DB 인스턴스에 연결하려면 먼저 [Amazon RDS 설정 \(p. 5\)](#) 섹션의 작업을 완료해야 합니다.

주제

- [MariaDB DB 인스턴스 생성 \(p. 10\)](#)
- [MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스의 데이터베이스에 연결 \(p. 15\)](#)
- [DB 인스턴스 삭제 \(p. 17\)](#)

MariaDB DB 인스턴스 생성

Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. MariaDB 데이터베이스를 실행하는 환경입니다.

이 예시에서는 db.t2.small DB 인스턴스 클래스, 20GiB의 스토리지, 하루 동안의 보존 기간으로 활성화된 자동 백업을 사용해 mariadb-instance1이라는 MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스를 만듭니다.

MariaDB DB 인스턴스를 만들려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
 2. Amazon RDS 콘솔의 오른쪽 위 모서리에서 DB 인스턴스를 생성하려는 AWS 리전을 선택합니다.
 3. 탐색 창에서 데이터베이스를 선택합니다.
- 탐색 창이 닫혀 있는 경우 왼쪽 위의 메뉴 아이콘을 선택하여 여십시오.
4. 데이터베이스 생성을 선택합니다. 엔진 선택 페이지가 열립니다.

Select engine

Engine options

Amazon Aurora
Amazon Aurora


MySQL


MariaDB


PostgreSQL


Oracle
ORACLE


Microsoft SQL Server


MariaDB
MariaDB Community Edition is a MySQL-compatible database with strong support from the open source community, and extra features and performance optimizations.

- Supports database size up to 16 TB.
- Instances offer up to 32 vCPUs and 244 GiB Memory.
- Supports automated backup and point-in-time recovery.
- Supports cross-region read replicas.
- Supports global transaction ID (GTID) and thread pooling.
- Developed and supported by the MariaDB open source community.

Only enable options eligible for RDS Free Usage Tier [info](#)

[Cancel](#) **Next**

5. [MariaDB]를 선택한 후 [Next]를 선택합니다.
6. [Choose use case] 페이지에 생성하려는 DB 인스턴스를 프로덕션에 사용할지 묻는 메시지가 나타납니다. 예제 인스턴스이므로 [Dev/Test - MariaDB]를 선택합니다. 그런 다음 [Next]를 선택합니다.

Note

프로덕션 인스턴스를 생성하는 경우에는 일반적으로 이 페이지에서 [Production - MariaDB]를 선택하여 장애 조치 옵션 다중 AZ 및 프로비저닝된 IOPS 스토리지 옵션을 활성화합니다.

7. DB 세부 정보 지정 페이지에서 DB 인스턴스 정보를 지정합니다. 다음 표는 예제 DB 인스턴스의 설정을 보여 줍니다. 원하는 대로 설정되었으면 [Next]를 선택합니다.

이 파라미터의 경우...	수행할 작업
라이선스 모델	기본값 [general-public-license]를 선택해 MariaDB에 대한 GNU 일반 공개 라이선스, 버전 2를 사용합니다. MariaDB에는 라이선스 모델이 하나밖에 없습니다.
DB 엔진 버전	사용하려는 MariaDB 버전을 선택합니다.
DB 인스턴스 클래스	2GiB 메모리, 1ECU(1ECU의 가상 코어 하나), 64비트 플랫폼 및 보통 I/O 용량과 동일한 구성에 대해 [db.t2.small]을 선택합니다.
다중 AZ 배포	장애 조치를 위해 다른 가용 영역에 DB 인스턴스의 예비 복제본을 생성하려면 Create replica in a different zone 를 선택합니다. 이때 고가용성을 유지하려면 프로덕션 워크로드를 위한 다중 AZ를 권장합니다. 개발 및 테스팅을 위해 No 를 선택할 수 있습니다. 자세한 내용은 Amazon RDS를 위한 고가용성(다중 AZ) (p. 109) 단원을 참조하십시오.
스토리지 유형	스토리지 유형으로 범용(SSD)을 선택합니다. 스토리지에 대한 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.
할당된 스토리지	데이터베이스에 대해 스토리지 20GiB를 할당하려면 20 을 입력합니다. 경우에 따라 DB 인스턴스에 대해 데이터베이스의 크기보다 많은 양의 스토리지를 할당하면 I/O 성능을 개선할 수 있습니다. 스토리지 할당에 대한 자세한 내용은 Amazon RDS 기능 을 참조하십시오.
DB 인스턴스 식별자	선택한 AWS 리전의 계정에 대해 고유한 DB 인스턴스의 이름을 입력합니다. 예를 들어, mariadb-instance1 처럼 선택한 AWS 리전과 DB 엔진을 포함하는 등 알기 쉬운 이름을 지정할 수 있습니다.
Master username	1-16자의 영숫자 문자를 사용해 DB 인스턴스에 로그온하기 위해 마스터 사용자 이름으로 사용할 이름을 입력합니다. 이 사용자 이름은 처음으로 DB 인스턴스의 데이터베이스에 로그온하기 위해 사용합니다.
Master password 및 Confirm password	마스터 사용자 암호로 인쇄 가능한 ASCII 문자(/, " 및 @ 제외) 8-41자를 포함하는 암호를 입력합니다. 이 암호는 데이터베이스에 로그온할 때 사용자 이름과 함께 사용합니다. 암호 확인에 암호를 다시 입력합니다.

Specify DB details

Instance specifications

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

DB engine

MariaDB Community Edition

License model [Info](#)

general-public-license

DB engine version [Info](#)

MariaDB 10.3.8



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

8. 고급 설정 구성 페이지에서 RDS가 MariaDB DB 인스턴스를 시작하는 데 필요한 추가 정보를 입력합니다. 표는 예제 DB 인스턴스의 설정을 보여 줍니다. DB 인스턴스 정보를 지정한 다음 데이터베이스 생성을 선택합니다.

이 파라미터의 경우...	수행할 작업
Virtual Private Cloud(VPC)	MariaDB DB 인스턴스를 호스팅할 Amazon Virtual Private Cloud(Amazon VPC)의 이름을 선택합니다. VPC 사용에 관한 자세한 내용은 Amazon Virtual Private Cloud(VPC) 및 Amazon RDS (p. 406) 단원을 참조하십시오.
Subnet Group	[Create new DB subnet group]을 선택합니다.
퍼블릭 액세스 가능성	Yes(예)를 선택합니다.
[Availability zone]	특정 가용 영역의 지정 여부를 결정합니다. 가용 영역에 대한 자세한 내용은 리전 및 가용 영역 (p. 99) 단원을 참조하십시오.
VPC 보안 그룹	[Create new VPC security group]을 선택합니다.
데이터베이스 이름	기본 데이터베이스의 이름을 1~64자의 영숫자 문자로 입력합니다. 이름을 제공하지 않으면 만들고 있는 DB 인스턴스

이 파라미터의 경우...	수행할 작업
	에서 Amazon RDS가 데이터베이스를 자동으로 만들지 않습니다. 추가 데이터베이스를 생성하려면 DB 인스턴스에 연결한 다음 SQL 명령어 CREATE DATABASE 를 사용하십시오. DB 인스턴스 연결에 대한 자세한 내용은 MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 (p. 445) 단원을 참조하십시오.
데이터베이스 포트	데이터베이스에 액세스할 때 사용할 특정 포트가 없으면 기본값 3306을 유지합니다. MariaDB 설치의 기본값은 포트 3306로 설정됩니다.
DB 파라미터 그룹	고유한 DB 파라미터 그룹을 만들지 않았다면 기본값 [default.MySQL10.0]을 유지합니다. 파라미터 그룹에 대한 자세한 내용은 DB 파라미터 그룹 작업 (p. 161) 단원을 참조하십시오.
옵션 그룹	기본값을 수락합니다.
스냅샷으로 태그 복사	스냅샷을 생성할 때 DB 인스턴스 태그를 DB 스냅샷에 복사하려면 이 옵션을 선택합니다. 자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.
암호화	[Disable encryption]을 선택합니다. Note 프로덕션 인스턴스에 대해서는 일반적으로 [Enable encryption]을 선택하여 이 DB 인스턴스에 대해 유 휴 암호화를 활성화합니다. 자세한 내용은 Amazon RDS 리소스 암호화 (p. 385) 단원을 참조하십시오.
백업 보존 기간	데이터베이스에서 자동 백업을 지속할 일수를 설정합니다. 테스트를 위해 이 값을 [1 day]로 설정할 수 있습니다.
백업 기간	데이터베이스를 백업할 특정 시간을 지정하지 않으려면 기본값으로 [No Preference]를 사용합니다.
확장 모니터링	DB 인스턴스가 실행되는 운영 체제에 대한 실시간 수집 측 정치를 활성화하지 않으려면 기본값인 [Disable enhanced monitoring]을 사용합니다.
로그 내보내기	[General log]를 선택합니다. 자세한 내용은 MariaDB 데이터베이스 로그 파일 (p. 310) 단원을 참조하십시오.
マイ너 버전 자동 업그레이드	DB 엔진의 기본 마이너 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다.
유지 관리 기간	대기 중인 DB 인스턴스의 설정 변경을 적용하려면 30분으로 선택합니다. 기간이 중요하지 않은 경우 기본 설정 없음을 선택합니다.

9. 데이터베이스 생성을 선택합니다.
10. [View DB instance details]를 선택합니다.

RDS 콘솔에 새 DB 인스턴스의 세부 정보가 표시됩니다. DB 인스턴스를 사용할 준비가 될 때까지 DB 인스턴스의 상태는 [creating]입니다. 상태가 available로 변경되면 DB 인스턴스에 연결할 수 있습니다. DB 인스턴스 클래스와 스토리지의 양에 따라 새 인스턴스를 사용할 수 있을 때까지 최대 20분이 걸릴 수 있습니다.

The screenshot shows the 'Summary' section for a new MariaDB instance named 'mariadb-instance1'. The instance is currently in the 'creating' state. The engine is MariaDB 10.1.26, and the DB instance class is db.t2.small. The CloudWatch metrics section shows 17 metrics for this instance. A legend indicates that the blue line represents 'mariadb-instance1'.

MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스의 데이터베이스에 연결

Amazon RDS가 DB 인스턴스를 프로비저닝한 후에는 표준 SQL 클라이언트 애플리케이션을 사용해 DB 인스턴스에 있는 데이터베이스에 연결할 수 있습니다. 이 예에서는 mysql 명령줄 도구를 사용해 MariaDB DB 인스턴스의 데이터베이스에 연결합니다. 연결하는 데 사용할 수 있는 GUI 기반 애플리케이션 하나는 HeidiSQL입니다. 자세한 내용은 [HeidiSQL 다운로드](#) 페이지를 참조하십시오. MariaDB를 사용하는 방법에 관한 자세한 내용은 [MariaDB 설명서](#)를 참조하십시오.

mysql 명령줄 도구를 사용하여 DB 인스턴스의 데이터베이스에 연결하는 방법

1. DB 인스턴스의 엔드포인트(DNS 이름)와 포트 번호를 찾습니다.
 - a. RDS 콘솔을 연 다음 데이터베이스를 선택해 DB 인스턴스의 목록을 표시합니다.
 - b. 세부 정보를 표시하고자 하는 MariaDB DB 인스턴스 이름을 선택합니다.
 - c. 연결 탭에서 엔드포인트를 복사합니다. 또한 포트 번호를 적어둡니다. DB 인스턴스에 연결하려면 엔드포인트와 포트 번호가 모두 필요합니다.

mymariadb

Summary

DB Name
mymariadb

Role
Instance

CPU
Current

Connectivity Monitoring Logs & events Configuration Maintenance

Connectivity

Endpoint & port

Endpoint
mymariadb [REDACTED] rds.amazonaws.com

Port
3306

1. MariaDB DB 인스턴스의 데이터베이스에 연결하려면 클라이언트의 명령 프롬프트에 다음 명령을 입력합니다. `<endpoint>`를 DB 인스턴스의 DNS 이름(엔드포인트)으로 대체하고, `<mymasteruser>`를 마스터 사용자 이름으로 대체한 후, 암호를 입력하라는 요청을 받으면 사용한 마스터 암호를 입력합니다.

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

사용자에 대한 암호를 입력하면 다음과 유사한 출력이 나타납니다.

```
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 >

DB 인스턴스 삭제

만든 샘플 DB 인스턴스에 연결한 후에는 이에 대한 요금이 더 이상 부과되지 않도록 DB 인스턴스를 삭제해야 합니다.

최종 DB 스냅샷이 없는 DB 인스턴스를 삭제하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
 탐색 창이 닫혀 있는 경우 왼쪽 상단의 메뉴 아이콘을 선택하여 여십시오.
3. 삭제할 DB 인스턴스를 선택합니다.
4. [Actions]에 대해 [Delete]를 선택합니다.
5. 최종 스냅샷 생성?에서 아니요를 선택하고 확인을 선택합니다.
6. 삭제를 선택합니다.

Microsoft SQL Server DB 인스턴스 생성 및 DB 인스턴스에 연결

Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. Amazon RDS DB 인스턴스가 온프레미스 Microsoft SQL Server와 유사합니다. SQL Server DB 인스턴스를 만든 후에 1개 이상의 사용자 지정 데이터베이스를 추가할 수 있습니다.

Important

DB 인스턴스를 생성하려면 우선 AWS 계정을 보유해야 합니다. AWS 계정이 없다면 <https://aws.amazon.com/>을 연 후에 [Create an AWS Account]를 선택합니다.

이 항목에서는 샘플 SQL Server DB 인스턴스를 생성합니다. 그런 다음 DB 인스턴스에 연결해 간단한 쿼리를 실행합니다. 마지막으로 샘플 DB 인스턴스를 삭제합니다.

샘플 SQL Server DB 인스턴스 생성

이 절차에서는 AWS Management 콘솔을 이용해 샘플 DB 인스턴스를 만듭니다. 샘플 DB 인스턴스만 생성하므로 각 설정을 완전하게 설명하지는 않습니다. 각 설정에 관한 완전한 설명을 보려면 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 생성 \(p. 494\)](#) 단원을 참조하십시오.

Microsoft SQL Server DB 엔진을 실행하는 DB 인스턴스를 생성하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. Amazon RDS 콘솔의 오른쪽 위 모서리에서 DB 인스턴스를 생성하려는 AWS 리전을 선택합니다.

3. 탐색 창에서 데이터베이스를 선택합니다.
4. 데이터베이스 생성을 선택합니다.

[Select engine] 페이지가 표시됩니다.

Select engine

Engine options

Amazon Aurora
Amazon Aurora

MySQL


MariaDB


PostgreSQL


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**

5. SQL Server 아이콘을 선택한 다음 SQL Server Express 에디션의 [Select]를 선택합니다.

[Specify DB Details] 페이지가 나타납니다.

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](#)

6. DB 세부 정보 지정 페이지에서 다음 표에 나와 있는 것처럼 DB 인스턴스에 대한 정보를 제공합니다.

이 파라미터의 경우...	수행할 작업
라이선스 모델	[license-included]를 선택하여 Microsoft SQL Server에 대한 일반 라이선스 계약을 사용합니다.
DB 엔진 버전	목록에서 제공하는 SQL Server의 최신 버전을 선택합니다.
DB 인스턴스 클래스	db.t2.micro를 선택합니다. 이 인스턴스 클래스는 테스트에 적합합니다.
시간대	시간대를 선택하지 마십시오. 표준 시간대를 선택하지 않으면 DB 인스턴스는 기본 표준 시간대를 사용합니다.
[Storage Type]	스토리지 유형으로 범용(SSD)을 선택합니다.

이 파라미터의 경우...	수행할 작업
할당된 스토리지	데이터베이스에 대해 스토리지 20GiB를 할당하려면 20 을 입력합니다. 더 많은 스토리지의 할당을 고려해야 한다는 경고가 표시되지만 샘플 DB 인스턴스이므로 20GiB이면 충분합니다.
DB 인스턴스 식별자	sample-instance 를 입력합니다.
마스터 사용자 이름	모든 데이터베이스 권한을 사용해 DB 인스턴스에 로그인하기 위해 마스터 사용자 이름으로 사용할 이름을 입력합니다. 마스터 사용자 이름은 SQL Server Authentication 로그인입니다.
[Master Password] 및 [Confirm Password]	마스터 사용자 암호를 입력합니다. 암호는 8~128자 인쇄 가능한 ASCII 문자를 포함해야 합니다(/, :, 및 @ 제외).

7. 다음을 선택하여 계속 진행합니다.

[Configure Advanced Settings] 페이지가 나타납니다.

Configure advanced settings

Network & Security

Virtual Private Cloud (VPC) [Info](#)
VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-2aed394c) [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.

default

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](#)

No preference

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

8. Configure Advanced Settings(고급 설정 구성) 페이지에서 다음 표에 나와 있는 것처럼 DB 인스턴스에 대한 정보를 입력합니다.

이 파라미터의 경우...	수행할 작업
VPC	[Create new VPC]를 선택합니다.
[Subnet Group]	[Create new DB Subnet Group]을 선택합니다.
Publicly Accessible	예를 선택합니다.
가용 영역	[No Preference]를 선택합니다.
[VPC Security Group]	[Create new Security Group]을 선택합니다.
데이터베이스 포트	데이터베이스에 액세스할 때 사용할 특정 포트가 없으면 기본값 1433을 유지합니다. SQL Server 설치에는 기본적으로 포트 1433이 사용되지만, 경우에 따라 방화벽에서 이 포트를 차단할 수도 있습니다. 의심스러운 경우에는 어느 포트를 사용해야 하는지 네트워크 관리자에게 물어보십시오.
DB 파라미터 그룹	기본값을 유지합니다.
옵션 그룹	기본값을 유지합니다.
스냅샷으로 태그 복사	이 설정을 선택하지 않은 상태로 유지합니다.
백업 보존 기간	7을 선택합니다.
백업 기간	[No Preference]를 선택합니다.
확장 모니터링 활성화	[No]를 선택합니다.
마이너 버전 자동 업그레이드	DB 엔진의 기본 마이너 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다.
유지 관리 기간	[No Preference]를 선택합니다.

9. 데이터베이스 생성을 선택합니다.
10. [View Your DB Instances]를 선택합니다.

RDS 콘솔의 DB 인터페이스 목록에 새 DB 인스턴스가 나타납니다. DB 인스턴스를 사용할 준비가 될 때 까지 DB 인스턴스의 상태는 [creating]입니다. 상태가 available로 변경되면 DB 인스턴스에 연결할 수 있습니다. DB 인스턴스 클래스와 스토리지의 양에 따라 새 인스턴스를 사용할 수 있을 때까지 최대 20분 이 걸릴 수 있습니다.

The screenshot shows the 'Databases' section of the AWS RDS console. At the top, there is a breadcrumb navigation: 'RDS > Databases'. Below the header, there is a search bar labeled 'Filter databases'. The main area displays a table with the following columns: 'DB Name', 'Role', 'Engine', and 'Region'. The table contains the following data:

DB Name	Role	Engine	Region
mymariadb	Instance	MariaDB	US-
myoracledb	Instance	Oracle Enterprise Edition	US-
mypostgresql	Instance	PostgreSQL	US-
mysqldb	Instance	SQL Server Express Edition	US-
testauroramysql-cl	Regional	Aurora MySQL	US-

샘플 SQL Server DB 인스턴스에 연결

이 절차에서는 Microsoft SQL Server Management Studio(SSMS)를 사용하여 샘플 DB 인스턴스에 연결합니다. 이 유ти리티의 독립 실행형 버전을 다운로드하려면 Microsoft 설명서의 [SQL Server Management Studio\(SSMS\)](#)를 참조하십시오.

SSMS를 사용하여 DB 인스턴스에 연결하려면

1. DB 인스턴스에 대한 DNS 이름과 포트 번호를 찾습니다.
 - a. RDS 콘솔을 연 다음 데이터베이스를 선택해 DB 인스턴스의 목록을 표시합니다.
 - b. SQL Server DB 인스턴스의 이름을 선택해 해당 DB 인스턴스에 대한 요약 정보를 표시합니다.

mysqldb

Summary

DB Name mysqldb	CPU	Info Backing-up
Role Instance	Current activity 0 Connections	Engine SQL Server Express Edition

Connectivity Monitoring Logs & events Configuration Maintenance & backups Tags

Connectivity

Endpoint & port	Networking
Endpoint mysqldb rds.amazonaws.com	Availability zone us-west-2b
Port 1433	VPC vpc-2aed394c

- c. 연결 탭에서 엔드포인트를 복사합니다. Endpoint 필드에는 콜론(:)으로 구분된 두 부분이 있습니다. 콜론 이전 부분은 인스턴스의 DNS 이름이고, 콜론 이후 부분은 포트 번호입니다. 두 파트를 모두 복사합니다.
2. SQL Server Management Studio를 시작합니다.

Connect to Server 대화 상자가 나타납니다.



3. 샘플 DB 인스턴스에 대한 정보를 제공합니다.

- a. [Server type]에서 [Database Engine]을 선택합니다.
- b. 서버 이름에서 쉼표로 구분한 샘플 DB 인스턴스의 DNS 이름과 포트 번호를 입력합니다.

Important

DNS 이름과 포트 번호 사이의 콜론을 쉼표로 바꿉니다.

예를 들면 서버 이름은 다음과 같은 형식이어야 합니다.

```
sample-instance.cg034hpkmmt.us-east-1.rds.amazonaws.com,1433
```

- c. [Authentication]의 경우 [SQL Server Authentication]을 선택합니다.
- d. 로그인에 샘플 DB 인스턴스에 대해 이전에 선택한 마스터 사용자 이름을 입력합니다.
- e. 암호에 샘플 DB 인스턴스에 대해 이전에 선택한 암호를 입력합니다.

4. [Connect]를 선택합니다.

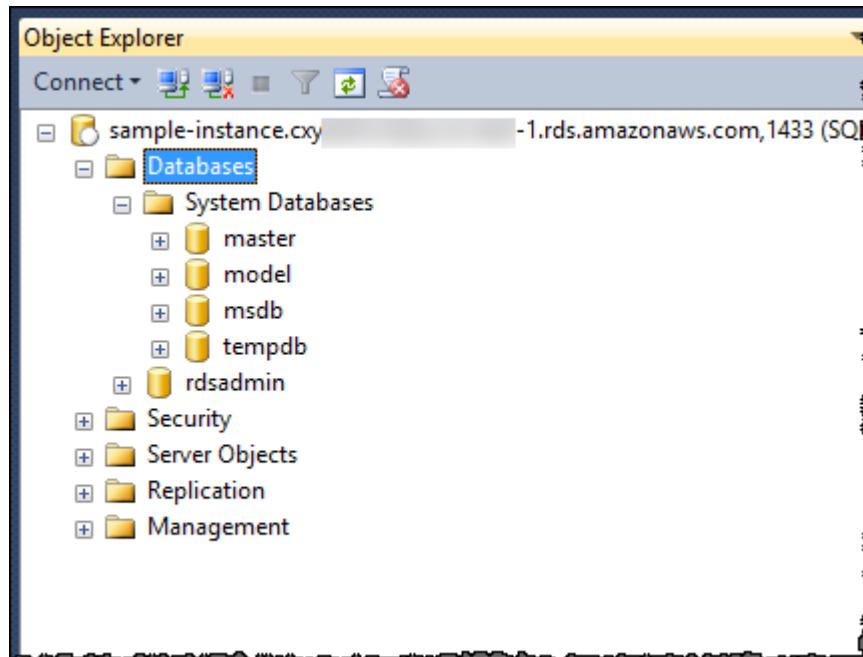
몇 분 정도 지나면 SSMS가 DB 인스턴스에 연결됩니다. DB 인스턴스에 연결할 수 없는 경우 [SQL Server DB 인스턴스 연결 문제 해결 \(p. 509\)](#) 단원을 참조하십시오.

샘플 SQL Server DB 인스턴스 탐색

이 절차에서는 이전 절차를 유지하고 Microsoft SQL Server Management Studio(SSMS)를 사용하여 샘플 DB 인스턴스를 탐색합니다.

SSMS를 이용해 DB 인스턴스를 탐색하려면

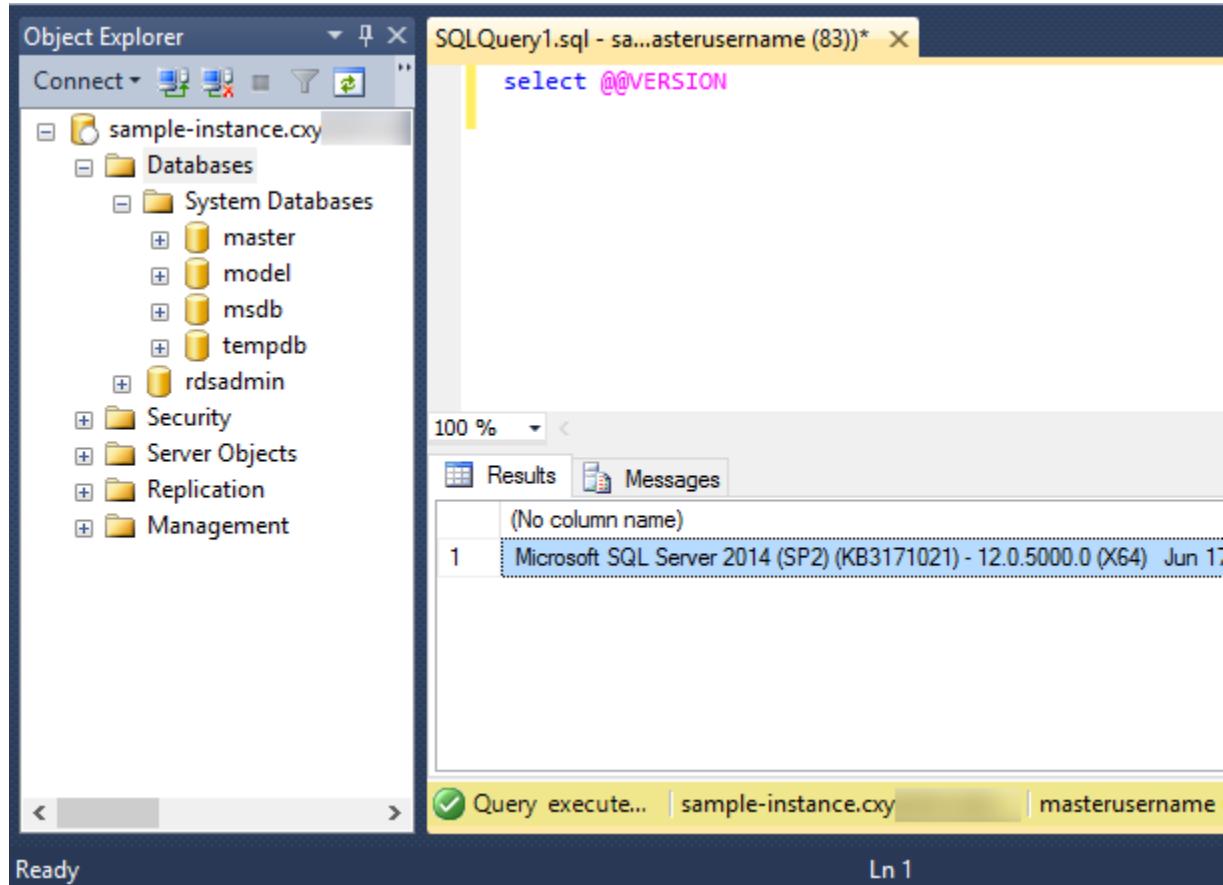
1. SQL 서버 DB 인스턴스는 SQL 서버의 표준 기본 제공 시스템 데이터베이스(마스터, 모델, msdb 및 tempdb)와 함께 제공됩니다. 시스템 데이터베이스를 탐색하려면 다음을 수행하십시오.
 - a. SSMS의 [View] 메뉴에서 [Object Explorer]를 선택합니다.
 - b. DB 인스턴스와 [Databases]를 확장하고, 다음과 같이 [System Databases]를 확장합니다.



2. SQL Server DB 인스턴스는 rdsadmin이라는 데이터베이스와 함께 제공됩니다. Amazon RDS는 이 데이터베이스를 사용하여 데이터베이스를 관리하는 데 사용하는 객체를 저장합니다. rdsadmin 데이터베이스에도 고급 작업을 수행하기 위해 실행할 수 있는 저장 프로시저가 포함됩니다.
3. 이제 자체 데이터베이스 생성을 시작하고 평소대로 DB 인스턴스와 데이터베이스에 대한 쿼리 실행을 시작할 수 있습니다. 샘플 DB 인스턴스에 대한 테스트 쿼리를 실행하려면 다음을 수행합니다.
 - a. SSMS의 [File] 메뉴에서 [New]를 가리킨 후 [Query with Current Connection]을 선택합니다.
 - b. 다음 SQL 쿼리를 입력합니다.

```
select @@VERSION
```

- c. 쿼리를 실행합니다. SSMS가 Amazon RDS DB 인스턴스의 SQL Server 버전을 반환합니다.



샘플 DB 인스턴스 삭제

만든 샘플 DB 인스턴스를 탐색한 후에는 이에 대한 요금이 더 이상 부과되지 않도록 DB 인스턴스를 삭제해야 합니다.

DB 인스턴스를 삭제하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 샘플 DB 인스턴스를 선택합니다.
4. [Actions]에 대해 [Delete]를 선택합니다.

5. [Create final Snapshot]에서 [No]를 선택합니다.

Note

삭제하는 모든 프로덕션 DB 인스턴스에 대한 최종 스냅샷을 만들어야 합니다.

6. 삭제를 선택합니다.

MySQL DB 인스턴스를 만들고 MySQL DB 인스턴스의 데이터베이스에 연결

DB 인스턴스를 만드는 가장 쉬운 방법은 AWS Management 콘솔을 사용하는 것입니다. DB 인스턴스를 만든 후에는 MySQL Workbench 같은 표준 MySQL 유ти리티를 사용해 DB 인스턴스에 있는 데이터베이스에 연결할 수 있습니다.

Important

DB 인스턴스를 생성하거나 DB 인스턴스에 연결하려면 먼저 [Amazon RDS 설정 \(p. 5\)](#) 섹션의 작업을 완료해야 합니다.

주제

- [MySQL DB 인스턴스 만들기 \(p. 26\)](#)
- [MySQL 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 \(p. 31\)](#)
- [DB 인스턴스 삭제 \(p. 33\)](#)

MySQL DB 인스턴스 만들기

Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. MySQL 데이터베이스를 실행하는 환경입니다.

이 예시에서는 db.m1.small DB 인스턴스 클래스, 20GiB의 스토리지, 하루 동안의 보존 기간으로 활성화된 자동화된 백업을 사용해 mysql-instance1이라는 MySQL 데이터베이스 엔진을 실행하는 DB 인스턴스를 만들습니다.

MySQL DB 인스턴스를 만들려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. Amazon RDS 콘솔의 오른쪽 위 모서리에서 DB 인스턴스를 생성하려는 AWS 리전을 선택합니다.
3. 탐색 창에서 데이터베이스를 선택합니다.
탐색 창이 닫혀 있는 경우 왼쪽 상단의 메뉴 아이콘을 선택하여 여십시오.
4. 데이터베이스 생성을 선택합니다. 엔진 선택 페이지가 열립니다.

Select engine

Engine options

Amazon Aurora
Amazon Aurora

MySQL


MariaDB


PostgreSQL


Oracle
ORACLE

Microsoft SQL Server


MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 16 TB.
- Instances offer up to 32 vCPUs and 244 GiB Memory.
- Supports automated backup and point-in-time recovery.
- Supports cross-region read replicas.

Only enable options eligible for RDS Free Usage Tier [info](#)

[Cancel](#) **Next**

5. [MySQL]을 선택한 후 [Next]를 선택합니다.
6. [Choose use case] 페이지에 생성하려는 DB 인스턴스를 프로덕션에 사용할지 묻는 메시지가 나타납니다. [Dev/Test]를 선택하고 [Next]를 선택합니다.
7. DB 세부 정보 지정 페이지에서 DB 인스턴스 정보를 지정합니다. 다음 표는 예제 DB 인스턴스의 설정을 보여 줍니다. 원하는 대로 설정되었으면 [Next]를 선택합니다.

이 파라미터의 경우...	수행할 작업
라이선스 모델	기본값인 general-public-license 를 선택하여 MySQL에 대한 일반 라이선스 계약을 사용합니다. MySQL에는 라이선스 모델이 하나밖에 없습니다.
DB 엔진 버전	MySQL 기본 버전을 선택합니다. Amazon RDS는 일부 AWS 리전에서 MySQL의 여러 버전을 지원합니다.
DB 인스턴스 클래스	db.m1.small 을 선택합니다.
다중 AZ 배포	장애 조치를 위해 다른 가용 영역에 DB 인스턴스의 예비 복제본을 생성하려면 Yes 를 선택합니다. 이때 고가용성을 유지하려면 프로덕션 워크로드를 위한 다중 AZ를 권장합니다. 개발 및 테스팅을 위해 No 를 선택할 수 있습니다.

이 파라미터의 경우...	수행할 작업
	자세한 내용은 Amazon RDS를 위한 고가용성(다중 AZ) (p. 109) 단원을 참조하십시오.
스토리지 유형	스토리지 유형 General Purpose (SSD) 를 선택합니다. 스토리지에 대한 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.
할당된 스토리지	데이터베이스에 대해 스토리지 20GiB를 할당하려면 20 을 입력합니다. 경우에 따라 DB 인스턴스에 대해 데이터베이스의 크기보다 많은 양의 스토리지를 할당하면 I/O 성능을 개선할 수 있습니다. 스토리지 할당에 대한 자세한 내용은 Amazon RDS 기능 을 참조하십시오.
DB 인스턴스 식별자	선택한 AWS 리전의 계정에 대해 고유한 DB 인스턴스의 이름을 입력합니다. 예를 들어, <code>mysql-instance1</code> 처럼 선택한 AWS 리전과 DB 엔진을 포함하는 등 알기 쉬운 이름을 지정할 수 있습니다.
Master username	영숫자 문자를 사용해 DB 인스턴스에 로그온하기 위해 마스터 사용자 이름으로 사용할 이름을 입력합니다. 이 이름은 처음으로 DB 인스턴스의 데이터베이스에 로그온하기 위해 사용하는 사용자 이름입니다.
Master password 및 Confirm password	마스터 사용자 암호로 인쇄 가능한 ASCII 문자(/, " 및 @ 제외) 8-41자를 포함하는 암호를 입력합니다. 이 암호는 사용자 이름을 사용해 데이터베이스에 로그온하는 데 사용하는 암호입니다. 그런 다음 [Confirm Password] 상자에 암호를 다시 한 번 입력합니다.

Specify DB details

Instance specifications

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

DB engine

MySQL Community Edition

License model [Info](#)

general-public-license

DB engine version [Info](#)

MySQL 5.6.40



Known Issues/Limitations

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



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.r4.xlarge — 4 vCPU, 30.5 GiB RAM

Multi-AZ deployment [Info](#)

8. [Next]를 선택합니다.
9. 고급 설정 구성 페이지에서 RDS가 MySQL DB 인스턴스를 시작하는 데 필요한 추가 정보를 입력합니다. 표는 예제 DB 인스턴스의 설정을 보여 줍니다. DB 인스턴스 정보를 지정한 다음 데이터베이스 생성을 선택합니다.

이 파라미터의 경우...	수행할 작업
Virtual Private Cloud(VPC)	[Create new VPC]를 선택합니다.
Subnet Group	[Create new DB subnet group]을 선택합니다.
퍼블릭 액세스 가능성	Yes(예)를 선택합니다.
[Availability zone]	[No Preference]를 선택합니다.

이 파라미터의 경우...	수행할 작업
VPC 보안 그룹	[Create new VPC security group]을 선택합니다.
데이터베이스 이름	기본 데이터베이스의 이름을 1~64자의 영숫자 문자로 입력합니다. 이름을 제공하지 않으면 만들고 있는 DB 인스턴스에서 Amazon RDS가 데이터베이스를 자동으로 만들지 않습니다.
	추가 데이터베이스를 생성하려면 DB 인스턴스에 연결한 다음 SQL 명령어 <code>CREATE DATABASE</code> 를 사용하십시오. DB 인스턴스 연결에 대한 자세한 내용은 MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 (p. 595) 단원을 참조하십시오.
데이터베이스 포트	데이터베이스에 액세스할 때 사용할 특정 포트가 없으면 기본값 3306 을 유지합니다. MySQL 설치의 기본값은 포트 3306으로 설정됩니다.
DB 파라미터 그룹	고유한 DB 파라미터 그룹을 만들지 않았다면 기본값을 유지합니다. 파라미터 그룹에 대한 자세한 내용은 DB 파라미터 그룹 작업 (p. 161) 단원을 참조하십시오.
옵션 그룹	이 옵션 그룹이 이전 페이지에서 선택한 MySQL 버전과 함께 사용되므로 기본값을 선택합니다.
스냅샷으로 태그 복사	스냅샷을 생성할 때 DB 인스턴스 태그를 DB 스냅샷에 복사하려면 이 옵션을 선택합니다. 자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.
IAM DB 인증	No 을 선택합니다. 자세한 내용은 인증 및 액세스 제어 (p. 339) 단원을 참조하십시오.
암호화	이 DB 인스턴스에 대해 비활성화되어 있는 암호화를 활성화하려면 [Enable encryption]를 선택합니다. 자세한 내용은 Amazon RDS 리소스 암호화 (p. 385) 단원을 참조하십시오.
백업 보존 기간	데이터베이스에서 자동 백업을 지속할 일수를 설정합니다. 테스트를 위해 이 값을 1 로 설정할 수 있습니다.
백업 기간	데이터베이스를 백업할 특정 시간을 지정하지 않으려면 기본값으로 No Preference 를 사용합니다.
확장 모니터링	DB 인스턴스가 실행되는 운영 체제에 대한 실시간 수집 측정치를 활성화하지 않으려면 기본값인 [Disable enhanced monitoring]을 사용합니다.
로그 내보내기	[General log]를 선택합니다. 자세한 내용은 MySQL 데이터베이스 로그 파일 (p. 318) 단원을 참조하십시오.
マイ너 버전 자동 업그레이드	DB 엔진의 기본 마이너 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다.
유지 관리 기간	[No preference]를 선택합니다.

10. 데이터베이스 생성을 선택합니다.
11. [View DB instance details]를 선택합니다.

RDS 콘솔에 새 DB 인스턴스의 세부 정보가 표시됩니다. DB 인스턴스를 사용할 준비가 될 때까지 DB 인스턴스의 상태는 [creating]입니다. 상태가 available로 변경되면 DB 인스턴스에 연결할 수 있습니다. DB 인스턴스 클래스와 스토리지의 양에 따라 새 인스턴스를 사용할 수 있을 때까지 최대 20분이 걸릴 수 있습니다.

The screenshot shows the AWS RDS console interface for managing MySQL databases. At the top, it displays the instance name 'mysql-instance1'. Below this is a 'Summary' section with three main items: 'Engine' (MySQL 5.6.37), 'DB instance class' (db.t2.small), and 'DB instance status' (creating). Further down, there's a 'CloudWatch' section showing 17 metrics for the instance, with a legend indicating the data series. A search bar and a 'Monitoring' button are also visible.

MySQL 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결

Amazon RDS가 DB 인스턴스를 프로비저닝한 후에는 표준 SQL 클라이언트 애플리케이션을 사용해 DB 인스턴스에 있는 데이터베이스에 연결할 수 있습니다. 이 예에서는 MySQL 모니터 명령을 사용해 MySQL DB 인스턴스의 데이터베이스에 연결합니다. 연결하는 데 사용할 수 있는 GUI 기반 애플리케이션 하나는 MySQL Workbench입니다. 자세한 내용은 [MySQL Workbench 다운로드](#) 페이지를 참조하십시오. MySQL을 사용하는 방법에 대한 자세한 내용은 [MySQL 설명서](#) 단원을 참조하십시오. MySQL 설치(MySQL 클라이언트 포함)에 대한 정보는 [MySQL 설치 및 업그레이드](#)를 참조하십시오.

MySQL 모니터를 사용해 DB 인스턴스의 데이터베이스에 연결하는 방법

1. DB 인스턴스의 엔드포인트(DNS 이름)와 포트 번호를 찾습니다.
 - a. RDS 콘솔을 연 다음 데이터베이스를 선택해 DB 인스턴스의 목록을 표시합니다.
 - b. 세부 정보를 표시하고자 하는 MySQL DB 인스턴스 이름을 선택합니다.
 - c. 연결 탭에서 엔드포인트를 복사합니다. 또한 포트 번호를 적어둡니다. DB 인스턴스에 연결하려면 엔드포인트와 포트 번호가 모두 필요합니다.

mysql-instance1

Summary

DB Name
mysql-instance1

Role
Master

CPU
[Progress Bar]

Current
[Progress Bar]

Connectivity Monitoring Logs & events Configuration Maintenance

Connectivity

Endpoint & port

Endpoint
mysql-instance1 [Red Box]rds.amazonaws.com

Port
3306

2. DB 인스턴스에 연결하는 데 사용할 수 있는 SQL 클라이언트를 다운로드하십시오.

Amazon RDS MySQL DB 인스턴스는 MySQL 명령줄 유ти리티 같은 도구를 사용하여 연결할 수 있습니다. MySQL 클라이언트 사용에 대한 자세한 내용은 MySQL 문서의 [mysql - The MySQL 명령줄 도구](#) 단원을 참조하십시오. 연결에 사용할 수 있는 GUI 기반 애플리케이션 중 하나는 MySQL Workbench입니다. 자세한 내용은 [MySQL Workbench 다운로드](#) 페이지 단원을 참조하십시오.

3. MySQL DB 인스턴스의 데이터베이스에 연결하십시오. 예를 들어 MySQL 클라이언트를 사용해 MySQL DB 인스턴스의 데이터베이스에 연결하려면 클라이언트 컴퓨터의 명령 프롬프트에 다음 명령을 입력합니다. `<endpoint>`를 DB 인스턴스의 DNS 이름으로 대체하고 `<mymasteruser>`를 마스터 사용자 이름으로 대체한 다음, 암호를 입력하라는 요청을 받으면 사용한 마스터 암호를 입력합니다.

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

사용자에 대한 암호를 입력하면 다음과 유사한 출력이 나타납니다.

```
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>
```

MySQL DB 인스턴스에 연결할 수 없는 경우 새 DB 인스턴스에 연결하지 못하는 두 가지 공통 원인은 다음과 같습니다.

- 보안 그룹을 사용하여 DB 인스턴스를 만들었지만, 이 보안 그룹이 MySQL 애플리케이션 또는 유ти리티를 실행 중인 디바이스나 Amazon EC2 인스턴스에서 연결할 수 있는 권한을 부여하지 않은 경우. DB 인스턴스가 VPC 내부에서 생성된 경우 이 인스턴스의 VPC 보안 그룹에서 연결 권한을 부여해야 합니다. DB 인스턴스가 VPC 외부에서 생성된 경우 이 인스턴스의 DB 보안 그룹에서 연결 권한을 부여해야 합니다. 자세한 내용은 [Amazon Virtual Private Cloud\(VPC\)](#) 및 [Amazon RDS \(p. 406\)](#) 단원을 참조하십시오.
- 포트 3306을 사용해 DB 인스턴스를 만들었는데 기업 방화벽 규칙에 따라 기업 네트워크의 디바이스에서 해당 포트에 연결하는 것이 차단된 경우. 이 오류를 수정하려면 인스턴스를 다른 포트로 다시 만들어야 합니다.

MySQL DB 인스턴스 연결에 대한 자세한 내용은 [MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 595\)](#) 단원을 참조하십시오.

DB 인스턴스 삭제

만든 샘플 DB 인스턴스에 연결한 후에는 이에 대한 요금이 더 이상 부과되지 않도록 DB 인스턴스를 삭제해야 합니다.

최종 DB 스냅샷이 없는 DB 인스턴스를 삭제하려면

- AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
 - 탐색 창에서 데이터베이스를 선택합니다.
- 탐색 창이 닫혀 있는 경우 왼쪽 상단의 메뉴 아이콘을 선택하여 여십시오.
- 삭제할 DB 인스턴스를 선택합니다.
 - [Actions]에 대해 [Delete]를 선택합니다.
 - 최종 스냅샷 생성?에서 아니요를 선택하고 확인을 선택합니다.
 - 삭제를 선택합니다.

Oracle DB 인스턴스를 만들고 Oracle DB 인스턴스의 데이터베이스에 연결

Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. Amazon RDS DB 인스턴스는 온프레미스 Oracle 데이터베이스와 유사합니다.

Important

DB 인스턴스를 생성하려면 우선 AWS 계정을 보유해야 합니다. AWS 계정이 없다면 <https://aws.amazon.com/>을 연 후에 [Create an AWS Account]를 선택합니다.

이 항목에서는 샘플 Oracle DB 인스턴스를 생성합니다. 그런 다음 DB 인스턴스에 연결해 간단한 쿼리를 실행합니다. 마지막으로 샘플 DB 인스턴스를 삭제합니다.

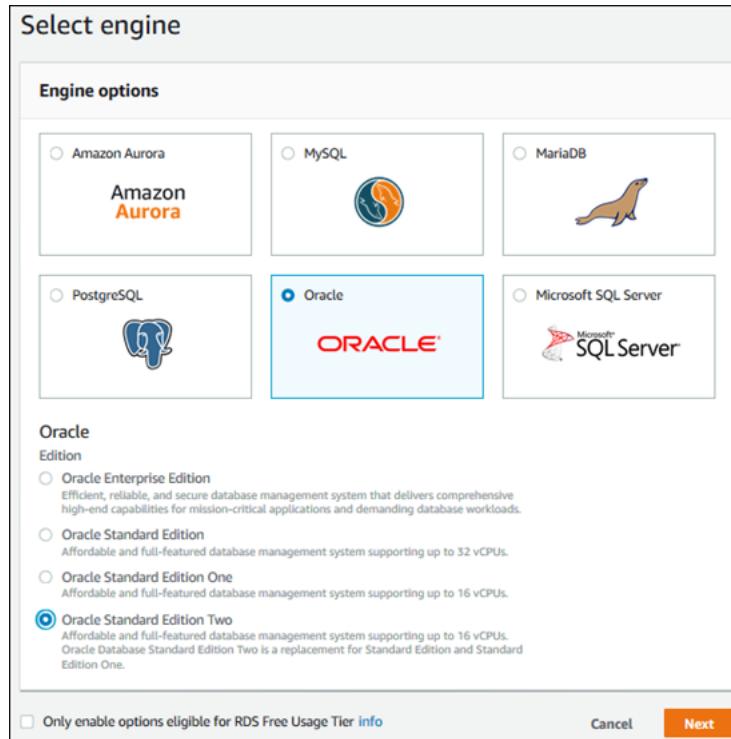
샘플 Oracle DB 인스턴스 생성

이 절차에서는 AWS Management 콘솔을 이용해 샘플 DB 인스턴스를 만듭니다. 샘플 DB 인스턴스만 생성 하므로 각 설정을 완전하게 설명하지는 않습니다. 각 설정에 관한 완전한 설명을 보려면 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.

Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스를 생성하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. Amazon RDS 콘솔의 오른쪽 위 모서리에서 DB 인스턴스를 생성하려는 AWS 리전을 선택합니다.
3. 탐색 창에서 데이터베이스를 선택합니다.
4. 데이터베이스 생성을 선택합니다.

[Select engine] 페이지가 표시됩니다.



5. Oracle 아이콘을 선택한 다음 Oracle Standard Edition Two 에디션의 [Select]를 선택합니다.
6. [Choose use case] 페이지에 생성하려는 DB 인스턴스를 프로덕션에 사용할지 묻는 메시지가 나타납니다. [Dev/Test]를 선택하고 [Next]를 선택합니다.

[Specify DB details] 페이지가 나타납니다.

Specify DB details

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

DB engine: Oracle Database Standard Edition Two

License model info: bring-your-own-license

DB engine version info: Oracle 12.1.0.2.v10

Free tier
The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GB 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: - Select one -

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: General Purpose (SSD)

Allocated storage: 20 GB
(Minimum: 20 GB, Maximum: 6144 GB) Higher allocated storage may improve IOPS performance.

Provisioning less than 100 GB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. Click here for more details.

7. DB 세부 정보 지정 페이지에서 다음 표에 나와 있는 것처럼 DB 인스턴스에 대한 정보를 제공합니다.

이 파라미터의 경우...	수행할 작업
라이선스 모델	[license-included]를 선택하여 Oracle 일반 라이선스 계약을 사용합니다.
DB 엔진 버전	목록에서 제공하는 Oracle의 최신 버전을 선택합니다.
DB 인스턴스 클래스	[db.t2.small]을 선택합니다. 이 인스턴스 클래스는 테스트에 적합합니다.
다중 AZ 배포	개발 및 테스트용이면 [No]를 선택합니다.
스토리지 유형	스토리지 유형으로 범용(SSD)을 선택합니다.
할당된 스토리지	데이터베이스에 대해 스토리지 20GiB를 할당하려면 20을 입력합니다.
DB 인스턴스 식별자	oracle-instance1 를 입력합니다.
Master username	모든 데이터베이스 권한을 사용해 DB 인스턴스에 로그인하기 위해 마스터 사용자 이름으로 사용할 이름을 입력합니다.

이 파라미터의 경우...	수행할 작업
	마스터 사용자 이름은 SQL Server Authentication 로그인입니다.
Master password 및 confirm Password	마스터 사용자 암호를 입력합니다. 암호는 8~128자 인쇄 가능한 ASCII 문자를 포함해야 합니다(/, 및 @ 제외).

8. 다음을 선택하여 계속 진행합니다.

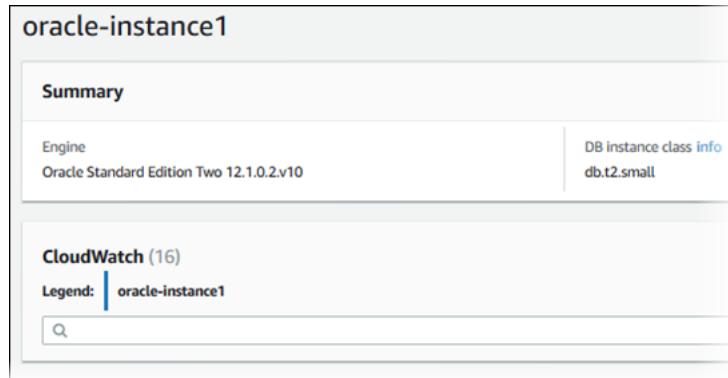
[Configure Advanced Settings] 페이지가 나타납니다.

9. Configure Advanced Settings(고급 설정 구성) 페이지에서 다음 표에 나와 있는 것처럼 DB 인스턴스에 대한 정보를 입력합니다.

이 파라미터의 경우...	수행할 작업
Virtual Private Cloud(VPC)	[Create new VPC]를 선택합니다.
Subnet Group	[Create new DB subnet group]을 선택합니다.
퍼블릭 액세스 가능성	Yes(예)를 선택합니다.
[Availability zone]	[No Preference]를 선택합니다.
VPC 보안 그룹	[Create new VPC security group]을 선택합니다.
데이터베이스 이름	ORCL를 입력합니다.
데이터베이스 포트	데이터베이스에 액세스할 때 사용할 특정 포트가 없으면 기본값 1521을 유지합니다. Oracle 설치에는 기본적으로 포트 1521이 사용되지만, 경우에 따라 방화벽에서 이 포트를 차단할 수도 있습니다. 의심스러운 경우에는 어느 포트를 사용해야 하는지 네트워크 관리자에게 물어보십시오.
DB 파라미터 그룹	기본값을 유지합니다.
옵션 그룹	기본값을 유지합니다.
스냅샷으로 태그 복사	이 설정을 선택하지 않은 상태로 유지합니다.
Character set name	유니코드 5.0 UTF-8 범용 문자 집합에 대해 기본값 [AL32UTF8]을 선택합니다.
Enable encryption	이 DB 인스턴스에 대해 비활성화되어 있는 암호화를 활성화하려면 [No]를 선택합니다.
백업 보존 기간	7 days 템을 선택합니다.
백업 기간	[No preference]를 선택합니다.
확장 모니터링	[Disable enhanced monitoring]을 선택합니다.
マイ너 버전 자동 업그레이드	DB 엔진의 기본 마이너 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다.
유지 관리 기간	[No preference]를 선택합니다.

10. 데이터베이스 생성을 선택합니다.
11. [View DB instance details]를 선택합니다.

RDS 콘솔에 새 DB 인스턴스의 세부 정보가 표시됩니다. DB 인스턴스를 사용할 준비가 될 때까지 DB 인스턴스의 상태는 [creating]입니다. 상태가 available로 변경되면 DB 인스턴스에 연결할 수 있습니다. DB 인스턴스 클래스와 스토리지의 양에 따라 새 인스턴스를 사용할 수 있을 때까지 최대 20분이 걸릴 수 있습니다.



샘플 Oracle DB 인스턴스에 연결

Amazon RDS가 DB 인스턴스를 프로비저닝한 후에는 표준 SQL 클라이언트 애플리케이션을 사용해 DB 인스턴스에 연결할 수 있습니다. 이 절차에서는 Oracle sqlplus 명령줄 유ти리티를 사용하여 샘플 DB 인스턴스에 연결합니다. 이 유ти리티의 독립 실행형 버전을 다운로드하려면 [SQL*Plus 사용 설명서 및 참조](#)를 참조하십시오.

SQL*Plus를 사용하여 DB 인스턴스에 연결하려면

1. DB 인스턴스의 앤드포인트(DNS 이름)와 포트 번호를 찾습니다.
 - a. RDS 콘솔을 연 다음 데이터베이스를 선택해 DB 인스턴스의 목록을 표시합니다.
 - b. 세부 정보를 표시하고자 하는 Oracle DB 인스턴스 이름을 선택합니다.
 - c. 연결 탭에서 앤드포인트를 복사합니다. 또한 포트 번호를 적어둡니다. DB 인스턴스에 연결하려면 앤드포인트와 포트 번호가 모두 필요합니다.

oracle-instance1

Summary

DB Name
oracle-instance1

Role
Instance

Connectivity Monitoring Logs & events Configuration

Connectivity

Endpoint & port

Endpoint
oracle-instance1. [REDACTED].rds.amazonaws.com

Port
1521

- sqlplus 유ти리티를 사용하여 DB 인스턴스에 연결하려면 명령 프롬프트에서 한 줄에 다음 명령을 입력합니다. Host의 값은 DB 인스턴스의 앤드포인트이고, Port의 값은 DB 인스턴스를 할당한 포트입니다. Oracle SID의 값은 DB 인스턴스의 이름이 아니라, DB 인스턴스를 만들 때 지정한 DB 인스턴스의 데이터베이스 이름입니다.

```
PROMPT>sqlplus 'mydbusr@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=endpoint)(PORT=1521))(CONNECT_DATA=(SID=ORCL)))'
```

다음과 유사한 출력 화면이 표시되어야 합니다.

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

SQL>

Oracle DB 인스턴스 연결에 대한 자세한 내용은 단원을 참조하십시오.

샘플 DB 인스턴스 삭제

만든 샘플 DB 인스턴스를 탐색한 후에는 이에 대한 요금이 더 이상 부과되지 않도록 DB 인스턴스를 삭제해야 합니다.

최종 DB 스냅샷이 없는 DB 인스턴스를 삭제하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 삭제할 DB 인스턴스를 선택합니다.
4. [Actions]에 대해 [Delete]를 선택합니다.
5. 최종 스냅샷 생성 여부에서 아니오를 선택하고 확인을 선택합니다.
6. 삭제를 선택합니다.

PostgreSQL DB 인스턴스 생성 및 PostgreSQL DB 인스턴스의 데이터베이스에 연결

DB 인스턴스를 만드는 가장 쉬운 방법은 RDS 콘솔을 사용하는 것입니다. DB 인스턴스를 만든 후에는 pgAdmin 유ти리티 같은 표준 SQL 클라이언트 유ти리티를 사용해 DB 인스턴스에 연결할 수 있습니다. 이 예에서는 db.m1.small DB 인스턴스 클래스, 10GiB의 스토리지 및 하루 동안의 보존 기간으로 활성화된 자동화된 백업을 사용해 west2-postgres1이라는 PostgreSQL 데이터베이스 엔진을 실행하는 DB 인스턴스를 만듭니다.

Important

DB 인스턴스를 생성하거나 DB 인스턴스에 연결하려면 먼저 [Amazon RDS 설정 \(p. 5\)](#) 섹션의 작업을 완료해야 합니다.

주제

- [PostgreSQL DB 인스턴스 만들기 \(p. 39\)](#)
- [PostgreSQL DB 인스턴스에 연결 \(p. 43\)](#)
- [DB 인스턴스 삭제 \(p. 46\)](#)

PostgreSQL DB 인스턴스 만들기

PostgreSQL DB 엔진을 실행하는 DB 인스턴스를 만드는 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. AWS Management 콘솔의 오른쪽 위 모서리에서 DB 인스턴스를 만들 AWS 리전을 선택합니다.
3. 탐색 창에서 데이터베이스를 선택합니다.

탐색 창이 닫혀 있는 경우 왼쪽 상단의 메뉴 아이콘을 선택하여 여십시오.

4. 데이터베이스 생성을 선택하여 엔진 선택 페이지를 여십시오.

Select engine

Engine options

Amazon Aurora
Amazon Aurora

MySQL


MariaDB


PostgreSQL


Oracle
ORACLE

Microsoft SQL Server


PostgreSQL

PostgreSQL is a powerful, open-source object-relational database system with a strong reputation of reliability, stability, and correctness.

- High reliability and stability in a variety of workloads.
- Advanced features to perform in high-volume environments.
- Vibrant open-source community that releases new features multiple times per year.
- Supports multiple extensions that add even more functionality to the database.
- Supports up to 5 Read Replicas per instance, within a single Region or cross-region.
- The most Oracle-compatible open-source database.

Info If you want to create PostgreSQL 11 in the Preview environment, click [here](#)

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**

5. 엔진 선택 페이지에서 PostgreSQL 아이콘을 선택한 후 다음을 선택합니다.
6. 다음으로 [Use case] 페이지에 생성하려는 DB 인스턴스를 프로덕션에 사용할지 묻는 메시지가 나타납니다. 사용할 계획이라면 [Production]을 선택합니다. 이 옵션을 선택하면 장애 조치 옵션인 [Multi-AZ]와 스토리지 옵션인 [Provisioned IOPS]가 다음 단계에서 사전 선택됩니다. 마쳤으면 [Next]를 선택합니다.
7. [Specify DB Details] 페이지에서 DB 인스턴스 정보를 지정합니다. 마쳤으면 [Next]를 선택합니다.

이 파라미터의 경우...	수행할 작업
라이선스 모델	PostgreSQL에는 라이선스 모델이 하나밖에 없습니다. postgresql-license 를 선택하여 PostgreSQL 일반 라이선스 계약을 사용합니다.
DB 엔진 버전	사용할 PostgreSQL 버전을 선택합니다.
DB 인스턴스 클래스	2GiB 메모리, 1ECU(1ECU의 가상 코어 하나), 64비트 플랫폼 및 보통 I/O 용량과 동일한 구성에 대해 db.t2.small 을 선택합니다. 모든 DB 인스턴스 클래스 옵션에 대한 자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 단원을 참조하십시오.
다중 AZ 배포	장애 조치를 위해 다른 가용 영역에 DB 인스턴스의 예비 복제본을 생성하려면 Yes 를 선택합니다. 이때 고가용성을 유지하려면 프로덕션 워크로드를 위한 다중 AZ를 권장합니다. 개발 및 테스팅을 위해 No 를 선택할 수 있습니다. 자세한 내용은 Amazon RDS를 위한 고가용성(다중 AZ) (p. 109) 단원을 참조하십시오.
[Storage Type]	스토리지 유형 General Purpose (SSD) 를 선택합니다. 스토리지에 대한 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.
활당된 스토리지	데이터베이스에 대해 스토리지 20GiB를 할당하려면 20 을 입력합니다. 경우에 따라 DB 인스턴스에 대해 데이터베이스의 크기보다 많은 양의 스토리지를 할당하면 I/O 성능을 개선할 수 있습니다. 스토리지 할당에 대한 자세한 내용은 Amazon RDS 기능 을 참조하십시오.
DB 인스턴스 식별자	선택한 AWS 리전의 계정에 대해 고유한 DB 인스턴스의 이름을 입력합니다. 예를 들어, postgresql-test 처럼 선택한 AWS 리전과 DB 엔진을 포함하는 등 알기 쉬운 이름을 지정할 수 있습니다.
마스터 사용자 이름	영숫자 문자를 사용해 DB 인스턴스에 로그온하기 위해 마스터 사용자 이름으로 사용할 이름을 입력합니다. 마스터 사용자 이름에 부여된 기본 권한에 대한 자세한 내용은 PostgreSQL용 Amazon RDS 버전과 확장 (p. 1014) 단원을 참조하십시오.
[Master Password] 및 [Confirm Password]	마스터 암호에 인쇄 가능한 ASCII 문자('," 및 @ 제외)로 구성된 암호(8~128자)를 입력하고 암호 확인 상자에 해당 암호를 다시 입력합니다.

8. [Configure Advanced Settings] 페이지에서 Amazon RDS가 PostgreSQL DB 인스턴스를 시작하는 데 필요한 추가 정보를 입력합니다. 표는 예제 DB 인스턴스의 설정을 보여 줍니다. DB 인스턴스 정보를 지정한 다음 데이터베이스 생성을 선택합니다.

이 파라미터의 경우...	수행할 작업
VPC	이 설정은 현재 사용 중인 플랫폼에 따라 다릅니다. AWS를 처음 사용하는 경우에는 기본 VPC를 선택합니다. 이전 E2-Classic 플랫폼에서 VPC를 사용하지 않는 DB 인스턴스를 만드는 경우에는 [Not in VPC] 를 선택합니다. VPC에 대

이 파라미터의 경우...	수행할 작업
	한 자세한 내용은 Amazon Virtual Private Cloud(VPC) 및 Amazon RDS (p. 406) 단원을 참조하십시오.
[Subnet Group]	이 설정은 현재 사용 중인 플랫폼에 따라 다릅니다. AWS를 처음 사용하는 고객은 default 를 선택합니다. 이 DB 서브넷 그룹은 계정에 대해 생성된 기본 DB 서브넷 그룹입니다. 이전 E2-Classic 플랫폼에서 DB 인스턴스를 만들어서 특정 VPC에 배포하려면 해당 VPC로 생성된 DB 서브넷 그룹을 선택합니다. VPC에 대한 자세한 내용은 Amazon Virtual Private Cloud(VPC) 및 Amazon RDS (p. 406) 단원을 참조하십시오.
Publicly Accessible	Yes 를 선택하여 VPC 외부에서도 액세스할 수 있게 DB 인스턴스에 퍼블릭 IP 주소를 할당합니다. VPC 내부에서만 DB 인스턴스에 액세스할 수 있게 하려면 No 를 선택합니다. 퍼블릭 액세스가 불가능하도록 DB 인스턴스를 숨기는 방법에 대한 자세한 내용은 VPC에 있는 DB 인스턴스를 인터넷에서 숨기기 (p. 416) 단원을 참조하십시오.
가용 영역	가용 영역을 지정하지 않으려면 기본값으로 No Preference 를 사용합니다.
[VPC Security Group]	AWS를 처음 사용하는 경우에는 기본 VPC를 선택합니다. 이전에 VPC 보안 그룹을 만들었으면 해당 VPC 보안 그룹을 선택합니다.
데이터베이스 이름	데이터베이스의 이름을 최대 63자의 영숫자 문자로 입력합니다. 이름을 제공하지 않으면 기본 "postgres" 데이터베이스가 생성됩니다. 추가 데이터베이스를 생성하려면 DB 인스턴스에 연결한 다음 SQL 명령어 <code>CREATE DATABASE</code> 를 사용하십시오. DB 인스턴스 연결에 대한 자세한 내용은 PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 (p. 953) 단원을 참조하십시오.
데이터베이스 포트	데이터베이스에 액세스하는 데 사용할 포트를 지정합니다. PostgreSQL 설치의 기본값은 포트 5432로 설정됩니다.
DB 파라미터 그룹	고유한 파라미터 그룹을 만들지 않았다면 기본값을 사용합니다.
옵션 그룹	고유한 옵션 그룹을 만든 경우가 아니면 기본값을 사용합니다.
스냅샷으로 태그 복사	스냅샷을 생성할 때 DB 인스턴스 태그를 DB 스냅샷에 복사하려면 이 옵션을 선택합니다. 자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.
[Enable Encryption]	이 DB 인스턴스에 대해 비활성화되어 있는 암호화를 활성화하려면 [Yes] 를 선택합니다. 자세한 내용은 Amazon RDS 리소스 암호화 (p. 385) 단원을 참조하십시오.
백업 보존 기간	데이터베이스에서 자동 백업을 지속할 일수를 설정합니다. 테스트를 위해 이 값을 1로 설정할 수 있습니다.
백업 기간	데이터베이스를 백업할 특정 시간을 지정하지 않으려면 기본값으로 No Preference 를 사용합니다.

이 파라미터의 경우...	수행할 작업
확장 모니터링 활성화	실시간 OS 모니터링을 사용하려면 Yes 를 선택합니다. Amazon RDS는 DB 인스턴스가 실행되는 운영 체제 (OS)에 대한 측정치를 실시간으로 제공합니다. Amazon CloudWatch Logs이 제공하는 프리 티어를 초과하는 Enhanced Monitoring에 대해서만 비용이 청구됩니다.
역할 모니터링	기본 IAM 역할을 사용하려면 Default 를 선택합니다.
Granularity	매분마다 인스턴스를 모니터링하려면 60 을 선택합니다.
마이너 버전 자동 업그레이드	DB 엔진의 기본 마이너 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다.
유지 관리 기간	대기 중인 DB 인스턴스의 설정 변경을 적용하려면 30 분으로 선택합니다. 시간이 중요하지 않은 경우 No Preference 를 선택합니다.

9. 최종 페이지에서 데이터베이스 생성을 선택합니다.
10. Amazon RDS 콘솔의 DB 인스턴스 목록에 새로운 DB 인스턴스가 나타납니다. DB 인스턴스를 만들고 사용할 준비가 될 때까지 DB 인스턴스의 상태는 [creating]입니다. 상태가 available로 변경되면 DB 인스턴스에 연결할 수 있습니다. DB 인스턴스 클래스와 할당된 저장소에 따라 새 인스턴스를 사용할 수 있을 때까지 몇 분 정도 걸릴 수 있습니다.

The screenshot shows the 'Databases' section of the AWS RDS console. At the top, there are buttons for 'Group resources', 'Actions', 'Restore from S3', and a prominent orange 'Create database' button. Below this is a search bar labeled 'Filter databases'. The main area displays a table with columns: DB Name, Role, Engine, Region & AZ, Size, Status, and CPU. The table lists several databases, including 'mymariadb', 'mysql-instance1', 'testmysql-rr', 'mysqlserver-instance1', 'oracle-instance1', 'postgresql-test', 'testaurorysql-cl', and 'testauroramysql'. The 'testauroramysql-cl' entry is expanded to show its 'Writer' child database, 'testauroramysql'. All databases are currently listed as 'Available'.

PostgreSQL DB 인스턴스에 연결

Amazon RDS가 DB 인스턴스를 프로비저닝한 후에는 표준 SQL 클라이언트 애플리케이션을 사용해 인스턴스에 연결할 수 있습니다. DB 인스턴스를 만들 때 여기에 할당한 보안 그룹에서는 DB 인스턴스에 대한 액세스를 허용해야 합니다. DB 인스턴스에 연결하는 데 문제가 발생할 경우 그 원인은 대부분 DB 인스턴스에 할당한 보안 그룹에서 설정한 액세스 규칙에 있습니다.

이 섹션에서는 PostgreSQL DB 인스턴스에 연결하는 두 가지 방법을 보여 줍니다. 첫 번째 예에서는 PostgreSQL의 잘 알려진 오픈 소스 관리 및 개발 도구인 pgAdmin을 사용합니다. 클라이언트 컴퓨터에 PostgreSQL의 로컬 인스턴스가 없어도 pgAdmin을 다운로드하고 사용할 수 있습니다. 두 번째 예에서는 PostgreSQL 설치에 속하는 명령형 유ти리티인 psql을 사용합니다. psql을 사용하려면 클라이언트 컴퓨터에 PostgreSQL을 설치하거나 시스템에 psql 클라이언트를 설치해야 합니다.

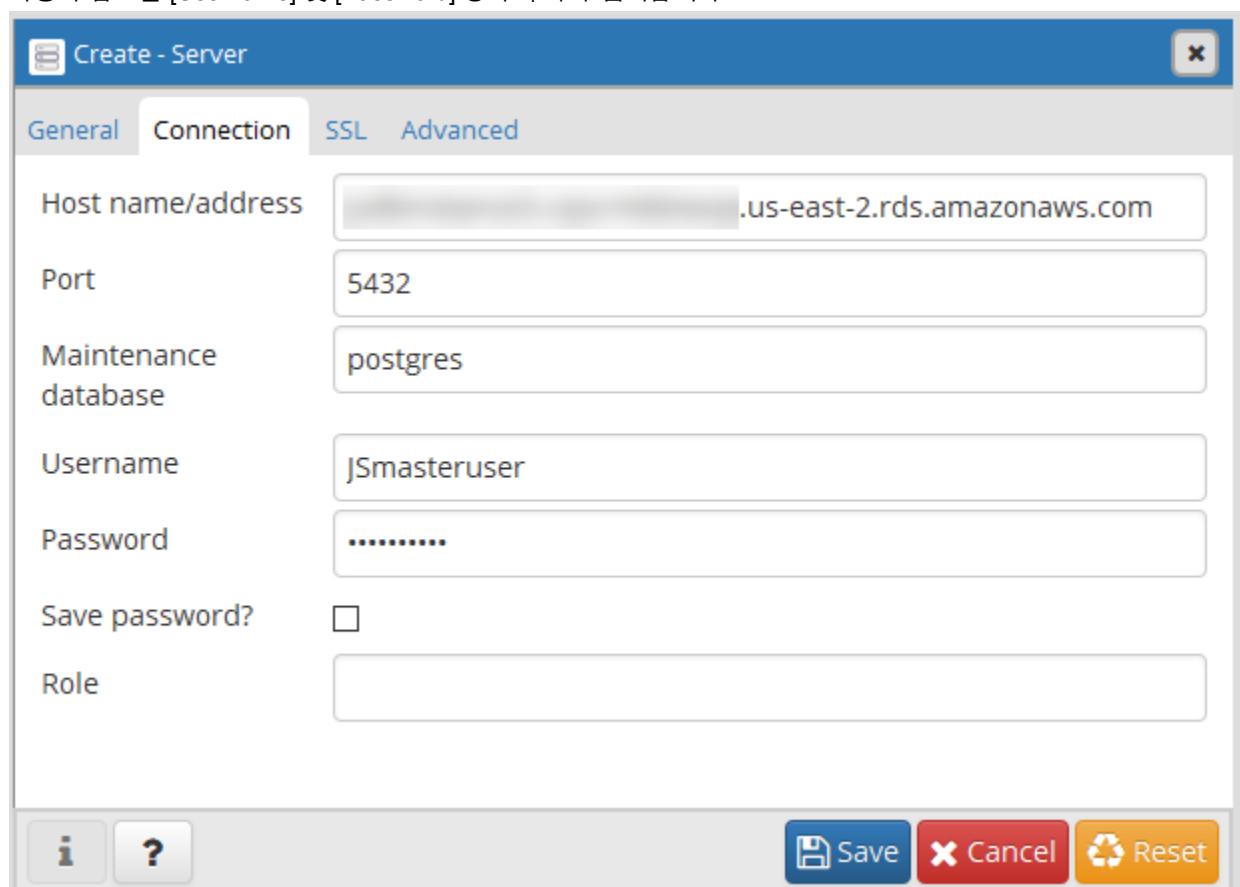
이 예에서는 pgAdmin을 사용해 PostgreSQL DB 인스턴스에 연결합니다.

pgAdmin을 사용해 PostgreSQL DB 인스턴스에 연결

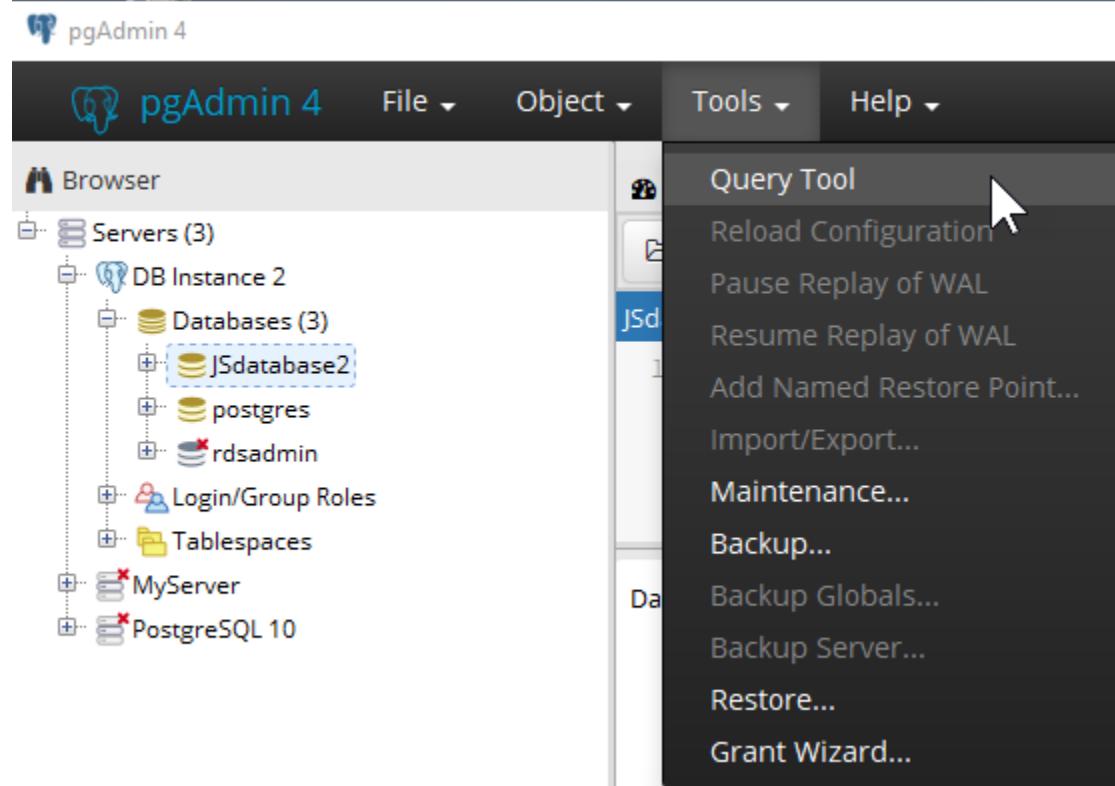
pgAdmin을 사용해 PostgreSQL DB 인스턴스에 연결하는 방법

1. 클라이언트 컴퓨터에서 pgAdmin 애플리케이션을 실행합니다. pgAdmin은 <http://www.pgadmin.org/>에서 설치할 수 있습니다.
2. [File] 메뉴에서 [Add Server]를 선택합니다.
3. New Server Registration(새 서버 등록) 대화 상자에서 호스트 상자에 DB 인스턴스 엔드포인트(예: mypostgresql.c6c8dntfzzhgv0.us-west-2.rds.amazonaws.com)를 입력합니다. Amazon RDS 콘솔에 나와 있는 대로 콜론이나 포트 번호를 포함하지 마십시오(mypostgresql.c6c8dntfzzhgv0.us-west-2.rds.amazonaws.com:5432).

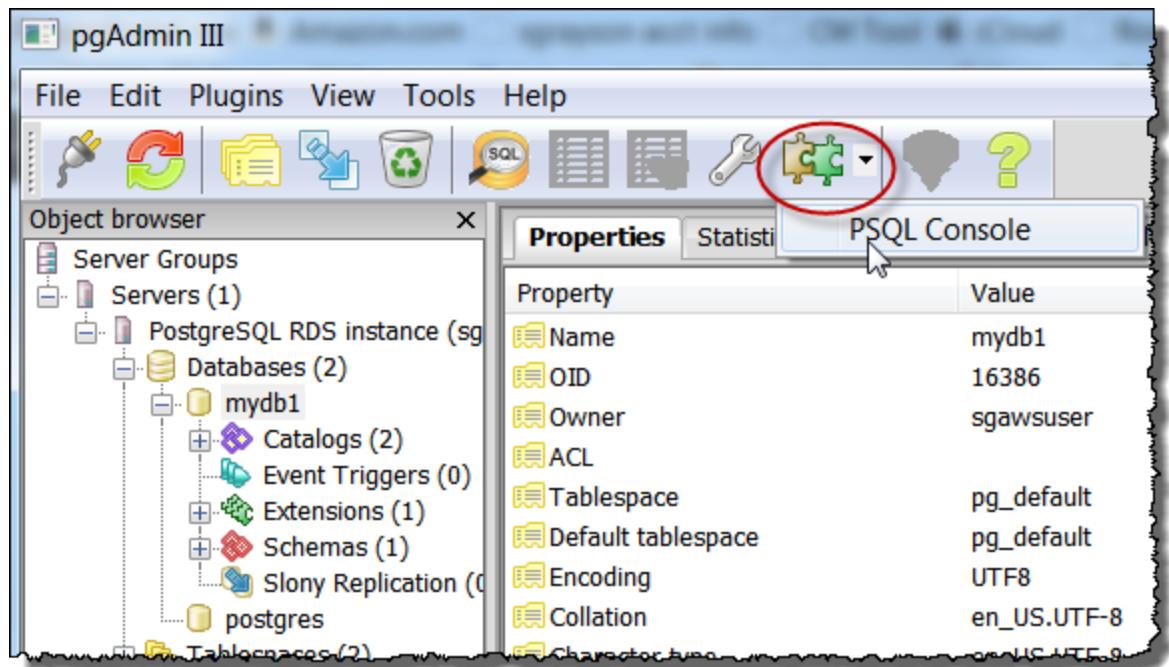
DB 인스턴스에 할당한 포트를 [Port] 상자에 입력합니다. DB 인스턴스를 만들 때 입력한 사용자 이름과 사용자 암호를 [Username] 및 [Password] 상자에 각각 입력합니다.



4. [OK]를 선택합니다.
5. Object browser에서 Server Groups를 확장합니다. 만든 서버(DB 인스턴스)를 선택한 다음 데이터베이스 이름을 선택합니다.



6. 플러그인 아이콘을 선택하고 [PSQL Console]을 선택합니다. 만든 기본 데이터베이스에 대해 psql 명령 창이 열립니다.



7. 명령 창을 사용해 SQL 또는 psql 명령을 입력합니다. \q를 입력해 창을 닫습니다.

psql을 사용해 PostgreSQL DB 인스턴스에 연결

클라이언트 컴퓨터에 PostgreSQL이 설치되어 있으면 psql의 로컬 인스턴스를 사용해 PostgreSQL DB 인스턴스에 연결할 수 있습니다. psql을 사용해 PostgreSQL DB 인스턴스에 연결하려면 호스트 정보 및 액세스 자격 증명을 제공해야 합니다.

다음 형식을 사용해 Amazon RDS에서 PostgreSQL DB 인스턴스에 연결합니다.

```
psql --host=<DB instance endpoint> --port=<port> --username=<master user name> --password  
--dbname=<database name>
```

예를 들어 다음 명령은 가상 자격 증명을 사용해 mypostgresql이라는 PostgreSQL DB 인스턴스에서 mypgdb라는 데이터베이스에 연결합니다.

```
psql --host=mypostgresql.c6c8mwvfdgv0.us-west-2.rds.amazonaws.com --port=5432 --  
username=awsuser --password --dbname=mypgdb
```

연결 문제 해결

DB 인스턴스의 데이터베이스에 연결하려 할 때 가장 일반적으로 발생하는 문제는 DB 인스턴스에 할당된 보안 그룹의 액세스 규칙입니다. DB 인스턴스를 만들 때 기본 DB 보안 그룹을 사용한 경우 인스턴스에 액세스 할 수 있도록 허용하는 규칙이 보안 그룹에 없을 확률이 큽니다. Amazon RDS 보안 그룹에 대한 자세한 내용은 [보안 그룹을 통한 액세스 제어 \(p. 390\)](#) 단원을 참조하십시오.

가장 일반적인 오류는 서버에 연결할 수 없습니다. 연결이 시간 초과되었습니다.입니다. 이 오류가 발생할 경우 호스트 이름이 DB 인스턴스 앤드포인트이고 포트 번호가 올바른지 확인하십시오. DB 인스턴스에 할당된 보안 그룹에 모든 연결이 통과할 수 있는 방화벽을 통한 액세스를 허용하는 데 필요한 규칙이 있는지 확인하십시오.

DB 인스턴스 삭제

만든 샘플 DB 인스턴스에 연결한 후에는 이에 대한 요금이 더 이상 부과되지 않도록 DB 인스턴스를 삭제해야 합니다.

최종 DB 스냅샷이 없는 DB 인스턴스를 삭제하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
탐색 창이 닫혀 있는 경우 왼쪽 상단의 메뉴 아이콘을 선택하여 여십시오.
3. 삭제할 DB 인스턴스를 선택합니다.
4. [Actions]에 대해 [Delete]를 선택합니다.
5. 최종 스냅샷 생성?에서 아니요를 선택하고 확인을 선택합니다.
6. 삭제를 선택합니다.

자습서: 웹 서버 및 Amazon RDS 데이터베이스 생성

이 자습서를 참조하여 PHP가 있는 Apache 웹 서버를 설치하고 MySQL 데이터베이스를 생성할 수 있습니다. 이 웹 서버는 Amazon Linux를 사용하여 Amazon EC2 인스턴스에서 실행되며, MySQL 데이터베이스

는 Amazon RDS MySQL DB 인스턴스입니다. Amazon EC2 인스턴스와 Amazon RDS DB 인스턴스가 모두 Amazon Virtual Private Cloud 서비스(Amazon VPC) 기반 VPC에서 실행됩니다.

Note

이 자습서에서는 Amazon Linux을 사용하여 작업하며 Ubuntu와 같은 다른 버전의 Linux는 적용되지 않을 수 있습니다.

이 자습서를 시작하기 전에 퍼블릭 및 프라이빗 서브넷과 해당 보안 그룹이 있는 VPC가 있어야 합니다. 없는 경우 [자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 \(p. 421\)](#)에서 다음 작업을 완료하십시오.

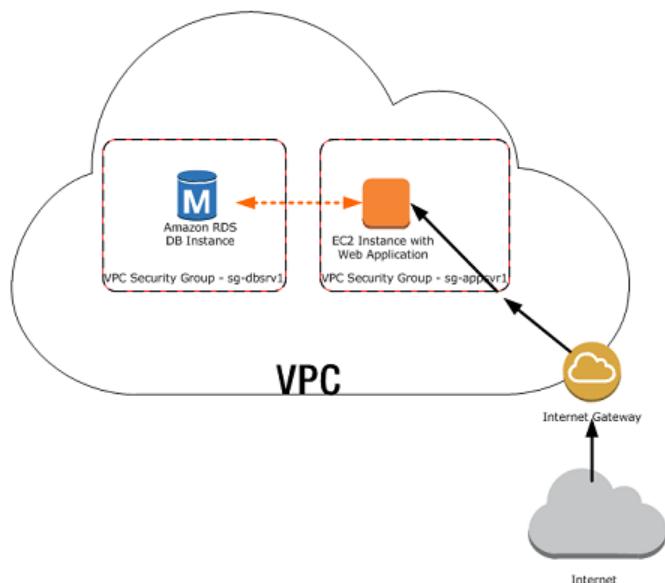
1. [프라이빗 서브넷과 퍼블릭 서브넷을 포함하는 VPC 생성 \(p. 422\)](#)
2. [추가 서브넷 생성 \(p. 422\)](#)
3. [퍼블릭 웹 서버에 대해 VPC 보안 그룹 생성 \(p. 423\)](#)
4. [프라이빗 Amazon RDS DB 인스턴스에 대한 VPC 보안 그룹 생성 \(p. 424\)](#)
5. [DB 서브넷 그룹 만들기 \(p. 425\)](#)

이어지는 자습서에서는 DB 인스턴스를 만들 때 VPC, 서브넷, 보안 그룹을 지정합니다. 또한 웹 서버를 호스팅하는 EC2 인스턴스를 만들 때도 이러한 항목을 지정합니다. DB 인스턴스와 웹 서버가 통신하려면 VPC, 서브넷, 보안 그룹이 필요합니다. VPC를 설정한 후 이 자습서에서는 DB 인스턴스를 만들고 웹 서버를 설치하는 방법을 보여 줍니다. DB 인스턴스 엔드포인트를 사용하여 VPC의 RDS DB 인스턴스에 웹 서버를 연결합니다.

이 자습서에서는 다음 절차를 수행합니다.

- [1단계: RDS DB 인스턴스 만들기 \(p. 47\)](#)
- [2단계: EC2 인스턴스 생성 및 웹 서버 설치 \(p. 53\)](#)

다음 다이어그램은 이 자습서를 완료했을 때 구성을 보여 줍니다.



1단계: RDS DB 인스턴스 만들기

이 단계에서는 웹 애플리케이션에서 사용되는 데이터를 유지 관리하는 Amazon RDS MySQL DB 인스턴스를 만듭니다.

Important

이 단계를 시작하기 전에 퍼블릭 및 프라이빗 서브넷과 해당 보안 그룹이 있는 VPC가 있어야 합니다. 없는 경우 [자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 \(p. 421\)](#) 단원을 참조하십시오. [프라이빗 서브넷과 퍼블릭 서브넷을 포함하는 VPC 생성 \(p. 422\)](#), [추가 서브넷 생성 \(p. 422\)](#), [퍼블릭 웹 서버에 대해 VPC 보안 그룹 생성 \(p. 423\)](#), [프라이빗 Amazon RDS DB 인스턴스에 대한 VPC 보안 그룹 생성 \(p. 424\)](#)의 단계를 수행합니다.

MySQL DB 인스턴스를 시작하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. AWS Management 콘솔의 오른쪽 위 모서리에서 DB 인스턴스를 만들 AWS 리전을 선택합니다. 이 예에서는 미국 서부(오레곤) 리전을 사용합니다.
3. 탐색 창에서 데이터베이스를 선택합니다.
탐색 창이 닫혀 있는 경우 왼쪽 상단의 메뉴 아이콘을 선택하여 여십시오.
4. 데이터베이스 생성을 선택하여 엔진 선택 페이지를 여십시오.
5. [Select engine] 페이지에서 다음과 같이 [MySQL]을 선택한 후 [Next]를 선택합니다.

Select engine

Engine options

Amazon Aurora **MySQL** MariaDB

PostgreSQL Oracle Microsoft SQL Server

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 16 TB.
- Instances offer up to 32 vCPUs and 244 GiB Memory.
- Supports automated backup and point-in-time recovery.
- Supports cross-region read replicas.

Only enable options eligible for RDS Free Usage Tier [info](#)

Cancel **Next**

6. Choose use case(사용 사례 선택) 페이지에서 Dev/Test – MySQL(개발/테스트 - MySQL)을 선택한 후 Next(다음)를 선택합니다.

7. [Specify DB details] 페이지에서 다음과 같이 값을 설정합니다.

- 라이선스 모델: 기본값을 사용합니다.
- DB 엔진 버전: 기본값을 사용합니다.
- DB 인스턴스 클래스: db.t2.small
- 다중 AZ 배포: No
- 스토리지 유형: General Purpose (SSD)
- 할당된 스토리지: 20 GiB
- DB 인스턴스 식별자: tutorial-db-instance
- Master username: tutorial_user
- Master password: 암호를 선택합니다.
- Confirm password: 암호를 다시 입력합니다.

Specify DB details

Instance specifications
Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator.

DB engine: MySQL Community Edition

License model info: general-public-license

DB engine version info: mysql 5.6.37

Known Issues/Limitations
Review the Known Issues/Limitations to learn about potential compatibility issues with specific database versions.

Free tier
The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GB 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

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: General Purpose (SSD)

Allocated storage: 20 GB (Minimum: 20 GB, Maximum: 16384 GB) Higher allocated storage may improve IOPS performance.

Settings

DB instance identifier info: tutorial-db-instance
Specify a name that is unique for all DB instances owned by your AWS account in the current region.

Master username info: tutorial_user
Specify an alphanumeric string that defines the login ID for the master user.

Master password info: Confirm password info:
Master Password must be at least eight characters long, as in "mypassword".

Cancel **Previous** **Next**

8. [Next]를 선택하고 [Configure advanced settings] 페이지에서 다음과 같이 값을 설정합니다.

- Virtual Private Cloud (VPC): [프라이빗 서브넷과 퍼블릭 서브넷을 포함하는 VPC 생성 \(p. 422\)](#)에서 생성된 `tutorial-vpc(vpc-identifier)`와 같이 퍼블릭 및 프라이빗 서브넷이 있는 기존 VPC를 선택합니다.

Note

VPC는 서로 다른 가용 영역에 서브넷이 있어야 합니다.

- [Subnet group:] [DB 서브넷 그룹 만들기 \(p. 425\)](#)에서 생성한 `tutorial-db-subnet-group` 같은 VPC용 DB 서브넷 그룹.
- 퍼블릭 액세스 가능성: 아니요
- 가용 영역: 기본 설정 없음
- VPC 보안 그룹: [프라이빗 Amazon RDS DB 인스턴스에 대한 VPC 보안 그룹 생성 \(p. 424\)](#)에서 생성된 `tutorial-db-securitygroup`과 같이 프라이빗 액세스에 맞게 구성된 기존 VPC 보안 그룹을 선택합니다.

각각에 연결된 X를 선택해 기본 보안 그룹 같은 다른 보안 그룹을 제거합니다.

- 데이터베이스 이름: `sample`

다른 옵션은 기본 설정을 유지합니다.

Configure advanced settings

Network & Security

Virtual Private Cloud (VPC) [Info](#)

VPC defines the virtual networking environment for this DB instance.

tutorial-vpc ([redacted])



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 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](#)

No preference



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

Choose VPC security groups



tutorial-db-securitygroup X

Database options

Database name

sample

Note: if no database name is specified then no initial MySQL database will be created on the DB Instance.

9. Amazon RDS MySQL DB 인스턴스를 생성하려면 데이터베이스 생성을 선택합니다.
10. 다음 페이지에서 [View DB instances details]를 선택하여 RDS MySQL DB 인스턴스를 확인합니다.
11. 새 DB 인스턴스의 DB 인스턴스 상태가 사용 가능으로 표시될 때까지 기다립니다. 그런 다음 [Connect] 섹션으로 스크롤하면 다음이 표시됩니다.

The screenshot shows the 'Connect' section of the AWS RDS console. It displays the endpoint (tutorial-db-instance.rds.amazonaws.com), port (3306), and a note that it is not publicly accessible. Below this, the 'Security group rules (2)' section is shown, listing two inbound rules from the 'tutorial-db-securitygroup'. The first rule is for the MySQL port (3306) and the second is for all traffic (0.0.0.0/0).

Security group	Type	Rule
tutorial-db-securitygroup	Security Group - Inbound	
tutorial-db-securitygroup	CIDR/IP - Outbound	0.0.0.0/0

DB 인스턴스의 엔드포인트와 포트를 적어둡니다. 이 정보를 사용하여 웹 서버를 RDS DB 인스턴스에 연결하게 됩니다.

RDS MySQL DB 인스턴스를 가능한 최고로 보안하려면 VPC 외부의 소스가 RDS MySQL DB 인스턴스에 연결할 수 없도록 하십시오.

다음 단계

[2단계: EC2 인스턴스 생성 및 웹 서버 설치 \(p. 53\)](#)

2단계: EC2 인스턴스 생성 및 웹 서버 설치

이 단계에서는 [1단계: RDS DB 인스턴스 만들기 \(p. 47\)](#)에서 생성한 Amazon RDS DB 인스턴스에 연결할 웹 서버를 생성합니다.

EC2 인스턴스 시작

먼저 VPC의 퍼블릭 서브넷에서 Amazon EC2 인스턴스를 생성합니다.

EC2 인스턴스를 시작하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 다음과 같이 [EC2 Dashboard]를 선택한 다음, [Launch Instance]를 선택합니다.

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

2 Running Instances
0 Dedicated Hosts
2 Volumes
2 Key Pairs
0 Placement Groups

1 Elastic IPs
0 Snapshots
0 Load Balancers
24 Security Groups

EC2 Spot. Save up to 90% off On-Demand Prices. Turbo Boost your Workloads. Get started with Amazon EC2 Spot Instances.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

Note: Your instances will launch in the US East (N. Virginia) region

Service Health

Scheduled Events

3. 다음과 같이 Amazon Linux AMI를 선택합니다.

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.

Cancel and Exit

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

Quick Start

My AMIs

AWS Marketplace

Community AMIs

Free tier only ⓘ

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-061392db613a6357b (64-bit x86) / ami-062ce7f8c1e7ffd3c (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.

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

Select

64-bit (x86)
64-bit (Arm)

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-01e24be29428c15b2

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

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

Select

64-bit (x86)

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-079596bf7a949ddf8 (64-bit x86) / ami-07a968a2c17fb48b (64-bit Arm)

Select

64-bit (x86)

Important

Amazon Linux 2 AMI에는 이 자습서에 필요한 소프트웨어 패키지가 없으므로 선택하지 마십시오.

4. 다음과 같이 [t2.small] 인스턴스 유형을 선택한 다음, [Next: Configure Instance Details]를 선택합니다.

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.

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3a.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3a.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3a.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes

Review and Launch

5. [Configure Instance Details] 페이지에서 다음과 같이 값을 설정하고 다른 값은 기본값으로 유지합니다.

- Network: [프라이빗 서브넷과 퍼블릭 서브넷을 포함하는 VPC 생성 \(p. 422\)](#)에서 생성된 `tutorial-vpc(vpc-identifier)`와 같이 DB 인스턴스에 대해 선택한 퍼블릭 및 프라이빗 서브넷이 있는 VPC를 선택합니다.
- Subnet: [퍼블릭 웹 서버에 대해 VPC 보안 그룹 생성 \(p. 423\)](#)에서 생성된 `subnet-identifier | Tutorial public | us-west-2a`와 같이 기존 퍼블릭 서브넷을 선택합니다.
- 퍼블릭 IP 자동 할당: `Enable(활성화)`을 선택합니다.

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 1 Launch into Auto Scaling Group

Purchasing option Request Spot Instances

Network vpc-tutorial-vpc Create new VPC

Subnet subnet-Tutorial public | Tutorial public | us-west-2a Create new subnet
249 IP Addresses available

Auto-assign Public IP Enable

Placement group Add instance to placement group

Capacity Reservation Open Create new Capacity Reservation

IAM role None Create new IAM role

Shutdown behavior Stop

Enable termination protection Protect against accidental termination

Monitoring Enable CloudWatch detailed monitoring Additional charges apply

Tenancy Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy

Elastic Inference Add an Elastic Inference accelerator Additional charges apply

T2/T3 Unlimited Enable Additional charges may apply

Cancel Previous Review and Launch Next: Add Storage

6. [Next: Add Storage]를 선택합니다.
7. 스토리지 추가 페이지에서 기본값을 유지하고 다음: 태그 추가를 선택합니다.
8. 태그 추가 페이지에서 다음과 같이 태그 추가를 선택한 다음 키에 **Name**을 입력하고 값에 **tutorial-web-server**를 입력합니다.

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)

Name tutorial-web-server Instances Volumes

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

9. [Next: Configure Security Group]를 선택합니다.
10. 다음과 같이 [Configure Security Group] 페이지에서 [Select an existing security group]을 선택한 다음 퍼블릭 웹 서버에 대해 VPC 보안 그룹 생성 (p. 423)에서 생성된 **tutorial-securitygroup**과 같이 기존 보안 그룹을 선택합니다. 보안 그룹은 SSH 및 HTTP 액세스를 위한 인바운드 규칙을 포함해야 합니다.

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
sg-[REDACTED]	default	default VPC security group	Copy to new
sg-[REDACTED]	tutorial-db-securitygroup	Tutorial DB Instance Security Group	Copy to new
sg-[REDACTED]	tutorial-securitygroup	Tutorial Security Group	Copy to new

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

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	[REDACTED]	
SSH	TCP	22	[REDACTED]	

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

11. Review and Launch(검토 및 시작)을 선택합니다.
12. 다음과 같이 [Review Instance Launch] 페이지에서 설정을 확인한 다음 [Launch]를 선택합니다.

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.

AMI Details [Edit AMI](#)

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-061392db613a6357b

Free tier eligible 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
sg-[REDACTED]	tutorial-securitygroup	Tutorial Security Group

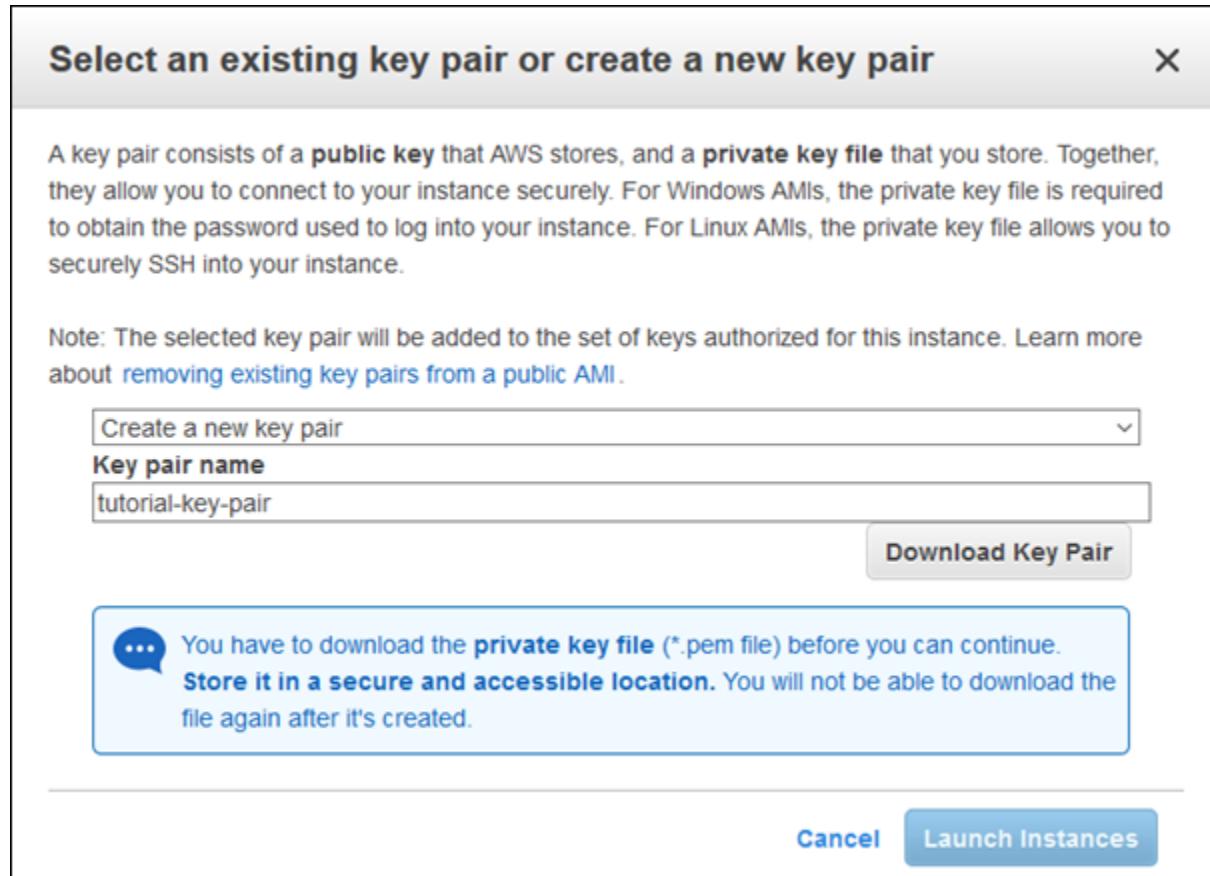
All selected security groups inbound rules

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	[REDACTED]	
SSH	TCP	22	[REDACTED]	

Instance Details [Edit instance details](#)

[Cancel](#) [Previous](#) **Launch**

13. 다음과 같이 기존 키 페어 선택 또는 새 키 페어 생성 페이지에서 새 키 페어 만들기를 선택하고 키 페어 이름을 tutorial-key-pair로 설정합니다. [Download Key Pair]를 선택하고 로컬 시스템에 키 페어 파일을 저장합니다. 이 키 페어 파일을 사용하여 EC2 인스턴스에 연결하게 됩니다.



14. EC2 인스턴스를 시작하려면 [Launch Instances]를 선택합니다. 다음과 같이 [Launch Status] 페이지에서, 새 EC2 인스턴스의 식별자(예: i-0288d65fd4470b6a9)를 기록해둡니다.

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](#)

- 인스턴스를 찾으려면 [View Instances]를 선택합니다.
- 인스턴스의 [Instance Status]가 **running**으로 읽힐 때까지 기다린 다음 계속합니다.

PHP가 포함된 Apache 웹 서버 설치

이제 EC2 인스턴스에 연결하고 웹 서버를 설치합니다.

EC2 인스턴스에 연결하고 PHP가 포함된 Apache 웹 서버를 설치하는 방법

- 앞에서 만든 EC2 인스턴스에 연결하려면 [Linux 인스턴스에 연결](#)에 있는 단계를 따릅니다.
- 최신 버그 수정 및 보안 업데이트를 얻으려면 다음 명령을 사용하여 EC2 인스턴스의 소프트웨어를 업데이트하십시오.

Note

-y 옵션을 사용하면 확인 여부를 물지 않고 업데이트를 설치합니다. 설치 전에 업데이트 정보를 확인하려면 이 옵션을 생략합니다.

```
[ec2-user ~]$ sudo yum update -y
```

- 업데이트 완료 후 yum install 명령을 사용하여 PHP 소프트웨어 패키지가 포함된 Apache 웹 서버를 설치합니다. 이 명령은 여러 소프트웨어 패키지와 관련 종속 프로그램을 동시에 설치합니다.

```
[ec2-user ~]$ sudo yum install -y httpd24 php56 php56-mysqlnd
```

Note

No package *package-name* available 오류가 발생하면 인스턴스가 Amazon Linux AMI로 실행되지 않은 것입니다(Amazon Linux 2 AMI를 사용하고 있는 것일 수 있음). 다음 명령을 사용하여 Amazon Linux 버전을 볼 수 있습니다.

```
cat /etc/system-release
```

자세한 내용은 [인스턴스 소프트웨어 업데이트](#) 단원을 참조하십시오.

4. 다음 명령을 사용하여 웹 서버를 시작합니다.

```
[ec2-user ~]$ sudo service httpd start
```

EC2 인스턴스의 퍼블릭 DNS 이름을 웹 브라우저의 주소 표시줄에 입력(예: `http://ec2-42-8-168-21.us-west-1.compute.amazonaws.com`)해 보면 웹 서버가 제대로 설치되어 시작되었는지 테스트할 수 있습니다. 웹 서버가 실행되고 있으면 Apache 테스트 페이지가 표시됩니다. Apache 테스트 페이지가 표시되지 않으면 [자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 \(p. 421\)](#)에서 생성한 VPC 보안 그룹의 인바운드 규칙에, 웹 서버에 연결할 때 사용하는 IP 주소에 대해 HTTP(포트 80) 액세스를 허용하는 규칙이 포함되어 있는지 확인하십시오.

Note

Apache 테스트 페이지는 문서의 루트 디렉터리 `/var/www/html`에 콘텐츠가 없는 경우에만 표시됩니다. 문서의 루트 디렉터리에 콘텐츠를 추가한 후에는 콘텐츠가 Apache 테스트 페이지 대신 EC2 인스턴스의 퍼블릭 DNS 주소에 나타납니다.

5. `chkconfig` 명령을 사용하여 웹 서버가 시스템 부팅 때마다 시작되도록 구성합니다.

```
[ec2-user ~]$ sudo chkconfig httpd on
```

`ec2-user`가 Apache 웹 서버의 기본 루트 디렉터리에 있는 파일을 관리할 수 있도록 하려면 `/var/www` 디렉터리의 소유권 및 권한을 변경해야 합니다. 본 자습서에서는 `www`라는 그룹을 EC2 인스턴스에 추가한 다음, `/var/www` 디렉터리의 해당 그룹 소유권을 부여하고 해당 그룹의 쓰기 권한을 추가하게 됩니다. 해당 그룹의 모든 멤버는 웹 서버에 대해서 파일의 추가, 삭제, 수정을 할 수 있습니다.

Apache 웹 서버에 대한 파일 권한 설정 방법

1. 다음 명령을 사용하여 `www` 그룹을 EC2 인스턴스에 추가합니다.

```
[ec2-user ~]$ sudo groupadd www
```

2. `ec2-user` 사용자를 `www` 그룹에 추가합니다.

```
[ec2-user ~]$ sudo usermod -a -G www ec2-user
```

3. 권한을 새로 고치고 새 `www` 그룹을 포함하려면 로그아웃합니다.

```
[ec2-user ~]$ exit
```

4. 다시 로그인한 다음, groups 명령을 사용하여 www 그룹이 있는지 확인합니다.

```
[ec2-user ~]$ groups  
ec2-user wheel www
```

5. /var/www 디렉터리 및 해당 콘텐츠의 그룹 소유권을 www 그룹으로 변경합니다.

```
[ec2-user ~]$ sudo chown -R root:www /var/www
```

6. /var/www 및 그 하위 디렉터리의 디렉터리 권한을 변경해서 그룹 쓰기 권한을 추가하고 나중에 생성될 하위 디렉터리에서 그룹 ID를 설정합니다.

```
[ec2-user ~]$ sudo chmod 2775 /var/www  
[ec2-user ~]$ find /var/www -type d -exec sudo chmod 2775 {} +
```

7. /var/www 디렉터리 및 하위 디렉터리의 파일 권한을 계속 변경해서 그룹 쓰기 권한을 추가합니다.

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} +
```

RDS DB 인스턴스에 Apache 웹 서버 연결

이제 Amazon RDS DB 인스턴스에 연결되는 Apache 웹 서버에 콘텐츠를 추가합니다.

RDS DB 인스턴스에 연결되는 Apache 웹 서버에 콘텐츠를 추가하는 방법

1. EC2 인스턴스에 계속 연결되어 있을 때 디렉터리를 /var/www로 변경하고 inc라는 새로운 하위 디렉터리를 생성합니다.

```
[ec2-user ~]$ cd /var/www  
[ec2-user ~]$ mkdir inc  
[ec2-user ~]$ cd inc
```

2. dbinfo.inc라는 inc 디렉터리에서 새 파일을 생성한 다음 nano 또는 선택한 편집기를 호출하여 파일을 편집합니다.

```
[ec2-user ~]$ >dbinfo.inc  
[ec2-user ~]$ nano dbinfo.inc
```

3. 다음 콘텐츠를 dbinfo.inc 파일에 추가합니다. 이때, **db_instance_endpoint**는 RDS MySQL DB 인스턴스의 엔드포인트(포트 없음)이며, **master_password**는 RDS MySQL DB 인스턴스의 마스터 암호입니다.

Note

웹 서버의 문서 루트에 포함되지 않는 폴더에 사용자 이름 및 암호 정보를 저장해야 보안 정보가 노출될 가능성을 줄일 수 있습니다.

```
<?php  
  
define('DB_SERVER', 'db_instance_endpoint');  
define('DB_USERNAME', 'tutorial_user');  
define('DB_PASSWORD', 'master password');  
define('DB_DATABASE', 'sample');  
  
?>
```

4. dbinfo.inc 파일을 저장하고 닫습니다.
5. 디렉터리를 /var/www/html로 변경합니다.

```
[ec2-user ~]$ cd /var/www/html
```

6. SamplePage.php라는 html 디렉터리에서 새 파일을 생성한 다음 nano 또는 선택한 편집기를 호출하여 파일을 편집합니다.

```
[ec2-user ~]$ >SamplePage.php  
[ec2-user ~]$ nano SamplePage.php
```

7. 다음 콘텐츠를 SamplePage.php 파일에 추가합니다.

Note

웹 서버의 문서 루트에 포함되지 않는 폴더에 사용자 이름 및 암호 정보를 저장해야 보안 정보가 노출될 가능성을 줄일 수 있습니다.

```
<?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. SamplePage.php 파일을 저장하고 닫습니다.
9. 웹 서버를 열고 <http://EC2 instance endpoint/SamplePage.php>(예: <http://ec2-55-122-41-31.us-west-2.compute.amazonaws.com/SamplePage.php>)를 검색하여 웹 서버에서 RDS MySQL DB 인스턴스에 제대로 연결되는지 확인합니다.

SamplePage.php를 사용하여 RDS MySQL DB 인스턴스에 데이터를 추가할 수 있습니다. 그러면 추가한 데이터가 페이지에 표시됩니다.

RDS MySQL DB 인스턴스를 가능한 최고로 보안하려면 VPC 외부의 소스가 RDS MySQL DB 인스턴스에 연결할 수 없도록 하십시오.

자습서

다음은 Amazon RDS를 사용하는 일반적인 작업의 수행 방법을 보여주는 자습서입니다.

- [자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 \(p. 421\)](#)
- [자습서: 웹 서버 및 Amazon RDS 데이터베이스 생성 \(p. 46\)](#)
- [자습: DB 스냅샷에서 DB 인스턴스 복원 \(p. 223\)](#)

동영상은 [AWS 설명 비디오 및 실습](#)을 참조하십시오.

Amazon RDS 모범 사례

Amazon RDS 작업 모범 사례에 대해서 알아봅니다. 새로운 모범 사례가 확인되는 대로 이 섹션을 업데이트 할 예정입니다.

주제

- [Amazon RDS 기본 운영 지침 \(p. 66\)](#)
- [DB 인스턴스 RAM 권장 사항 \(p. 67\)](#)
- [Amazon RDS 보안 모범 사례 \(p. 67\)](#)
- [Enhanced Monitoring을 통한 운영 체제 문제 식별 \(p. 67\)](#)
- [지표를 통해 성능 문제 식별 \(p. 68\)](#)
- [MySQL 스토리지 엔진으로 작업하기 위한 모범 사례 \(p. 71\)](#)
- [MariaDB 스토리지 엔진으로 작업하기 위한 모범 사례 \(p. 72\)](#)
- [Oracle 작업의 모범 사례 \(p. 72\)](#)
- [PostgreSQL로 작업하기 위한 모범 사례 \(p. 72\)](#)
- [SQL Server로 작업하기 위한 모범 사례 \(p. 74\)](#)
- [DB 파라미터 그룹 작업 \(p. 74\)](#)
- [Amazon RDS 모범 사례 프레젠테이션 동영상 \(p. 75\)](#)

Note

Amazon RDS에 대한 일반적인 권장 사항은 [Amazon RDS 권장 사항 사용 \(p. 286\)](#) 단원을 참조하십시오.

Amazon RDS 기본 운영 지침

다음은 Amazon RDS로 작업할 때 모든 사용자가 따라야 하는 기본 운영 지침입니다. Amazon RDS 서비스 수준 계약에 다음 지침을 따르도록 명시되어 있습니다.

- 메모리, CPU 및 스토리지 사용을 모니터링합니다. Amazon CloudWatch에서 사용 패턴이 변경되거나 사용자가 배포 용량에 도달했을 때 알림을 받도록 설정할 수 있으므로 시스템 성능과 가용성을 유지할 수 있습니다.
- 스토리지 용량 한도에 도달할 경우 DB 인스턴스를 확장합니다. 스토리지 및 메모리에 어느 정도 버퍼가 있어야만 애플리케이션에서 수요가 예기치 않게 늘어날 경우 이를 수용할 수 있습니다.
- 자동 백업을 활성화하고 일일 쓰기 IOPS가 낮은 동안 백업 창이 열리도록 설정합니다.
- 데이터베이스 작업량으로 인해 프로비저닝한 I/O보다 많이 필요할 경우 장애 조치 또는 데이터베이스 오류가 발생한 후에 복구 속도가 느려집니다. DB 인스턴스의 I/O 용량을 늘리려면 다음 중 일부 항목이나 모든 항목을 수행하십시오.
 - I/O 용량이 높은 DB 인스턴스 클래스로 마이그레이션합니다.
 - 필요한 종분 양에 따라 표준 스토리지를 범용 또는 프로비저닝된 IOPS 스토리지로 변환합니다. 사용 가능한 스토리지 유형에 대한 자세한 내용은 [Amazon RDS 스토리지 유형 \(p. 101\)](#) 단원을 참조하십시오.

프로비저닝된 IOPS 스토리지로 변환할 경우 프로비저닝된 IOPS에 최적화된 DB 인스턴스 클래스를 사용해야 합니다. 프로비저닝된 IOPS에 대한 자세한 내용은 [프로비저닝된 IOPS SSD 스토리지 \(p. 103\)](#) 단원을 참조하십시오.

- 이미 프로비저닝된 IOPS 스토리지를 사용하고 있는 경우 추가 처리 용량을 프로비저닝합니다.
- 클라이언트 애플리케이션이 DB 인스턴스의 DNS(Domain Name Service) 데이터를 캐시하는 경우 TTL(Time-to-Live) 값을 30초 미만으로 설정합니다. 장애 조치 이후에 DB 인스턴스의 기본 IP 주소가 변경

될 수 있기 때문에 DNS 데이터를 오랜 시간 동안 캐시하면 애플리케이션이 더 이상 서비스되지 않는 IP 주소에 연결하려 할 경우 연결 오류로 이어질 수 있습니다.

- 사용 사례에 대한 프로세스가 얼마나 오래 걸리는지 이해하고 DB 인스턴스에 액세스하는 애플리케이션이 장애 조치 이후에 새 DB 인스턴스에 자동으로 연결할 수 있는지 확인하려면 DB 인스턴스에 대한 장애조치를 테스트합니다.

DB 인스턴스 RAM 권장 사항

작업 집합이 거의 완전히 메모리에 상주하도록 RAM을 충분히 할당하는 것이 Amazon RDS 성능 모범 사례에 따르는 길입니다. 작업 집합이 거의 전부 메모리에 있는지 확인하려면 (Amazon CloudWatch를 사용하여) DB 인스턴스에 작업 부하가 걸려 있는 동안 ReadIOPS 메트릭을 확인하십시오. ReadIOPS의 값은 작고 안정적이어야 합니다. DB 인스턴스 클래스를 RAM이 더 많은 클래스로 확장하여 ReadIOPS가 대폭 떨어질 경우, 작업 집합이 거의 완전히 메모리에 상주하지는 못합니다. 규모 조정 작업 후 ReadIOPS가 더 이상 대폭 떨어지지 않을 때까지 계속 확장합니다. 그렇지 않으면 ReadIOPS가 매우 작은 양으로 감소됩니다. DB 인스턴스의 메트릭 모니터링에 대한 자세한 내용은 [DB 인스턴스 측정치 보기 \(p. 236\)](#) 단원을 참조하십시오.

Amazon RDS 보안 모범 사례

AWS IAM 계정을 사용해 Amazon RDS API 작업 중에서도 DB 인스턴스, 보안 그룹, 옵션 그룹 또는 파라미터 그룹 같은 RDS 리소스를 만들거나 수정하거나 삭제하는 작업과 DB 인스턴스를 백업하고 복원하거나 프로비저닝된 IOPS 스토리지를 구성하는 등 일반적인 관리 작업을 수행하는 작업에 대한 액세스를 제어하십시오.

- RDS 리소스를 관리하는 각 사용자에게 개별 IAM 계정을 할당합니다. AWS 루트 자격 증명을 사용해 Amazon RDS 리소스를 관리하지 마십시오. 자기 자신을 포함한 모두를 위해 IAM 사용자를 만들어야 합니다.
- 각 사용자에게 각자의 임무를 수행하는 데 필요한 최소 권한 집합을 부여합니다.
- IAM 그룹을 사용해 여러 사용자에 대한 권한을 효과적으로 관리합니다.
- IAM 자격 증명을 정기적으로 순환합니다.

IAM에 대한 자세한 내용은 [AWS Identity and Access Management](#) 단원을 참조하십시오. IAM 모범 사례에 대한 자세한 내용은 [IAM 모범 사례](#)를 참조하십시오.

AWS Management 콘솔, AWS CLI 또는 Amazon RDS API를 사용하여 마스터 사용자의 암호를 변경합니다. SQL 클라이언트 등과 같은 다른 도구를 사용하여 마스터 사용자 암호를 변경할 경우 의도치 않게 사용자에 대해 권한이 취소될 수 있습니다.

Enhanced Monitoring을 통한 운영 체제 문제 식별

Amazon RDS는 DB 인스턴스가 실행되는 운영 체제(OS)에 대한 측정치를 실시간으로 제공합니다. 콘솔을 사용하여 DB 인스턴스에 대한 측정치를 보거나, 선택한 모니터링 시스템의 Amazon CloudWatch Logs에서 Enhanced Monitoring JSON 출력을 사용할 수 있습니다. Enhanced Monitoring에 대한 자세한 내용은 [확장 모니터링 \(p. 238\)](#) 단원을 참조하십시오.

Enhanced Monitoring은 다음 데이터베이스 엔진에 사용할 수 있습니다.

- MariaDB
- Microsoft SQL Server
- MySQL 버전 5.5 이상
- Oracle
- PostgreSQL

향상된 모니터링은 db.m1.small을 제외한 모든 DB 인스턴스 클래스에서 사용할 수 있습니다. Enhanced Monitoring은 AWS GovCloud (US-West)을 제외한 모든 리전에서 사용할 수 있습니다.

지표를 통해 성능 문제 식별

리소스 부족이나 기타 일반적인 병목으로 인해 발생하는 성능 문제를 식별하기 위해 Amazon RDS DB 인스턴스에서 사용할 수 있는 지표를 모니터링할 수 있습니다.

성능 지표 보기

다양한 기간 동안의 평균, 최대, 최소 측정값을 보려면 성능 지표를 정기적으로 모니터링해야 합니다. 이렇게 하면 성능이 저하된 시점을 식별할 수 있습니다. 또한 특정 지표 임계값에 대해 Amazon CloudWatch 경보를 설정하여 해당 임계값에 이를 경우 알리도록 할 수 있습니다.

성능 문제를 해결하기 위해서는 시스템의 기준 성능을 파악해야 합니다. 새로운 DB 인스턴스를 설정하여 일반적인 워크로드 실행할 경우 다양한 간격(예: 1시간, 24시간, 1주, 2주)으로 모든 성능 측정의 평균값, 최댓값, 최솟값을 수집하여 정상 상태를 파악해야 합니다. 이렇게 하면 작업의 최고 피크와 최저 피크 시간을 비교할 수 있습니다. 그런 다음 이 정보를 사용하여 성능이 표준 수준 이하로 떨어진 때를 식별할 수 있습니다.

성능 지표를 보려면

1. AWS Management 콘솔에 로그인한 후 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 후 DB 인스턴스를 선택합니다.
3. 모니터링 표시를 선택합니다. 처음 8개의 성능 지표가 표시됩니다. 지표는 기본적으로 현재 날짜의 데이터가 표시됩니다.
4. 상단 오른쪽의 번호 버튼을 사용하여 다른 페이지의 지표를 보거나, 모두 표시를 선택하여 모든 지표를 봅니다.
5. 다른 날짜의 데이터를 보려면 성능 지표의 시간 범위를 조정합니다. [Statistic], [Time Range] 및 [Period] 값을 변경하여 표시되는 정보를 조정할 수 있습니다. 예를 들어 지난 2주 동안 각 날짜의 지표에 대한 피크 값을 보려면 [Statistic]를 [Maximum]으로, [Time Range]를 [Last 2 Weeks]로, [Period]를 [Day]로 설정합니다.

Note

[Statistic], [Time Range], [Period] 값을 변경하면 모든 지표에 대해 변경됩니다. 업데이트된 값은 현재 세션 동안 또는 다시 값을 변경할 때까지 유지됩니다.

CLI 또는 API를 사용하여 성능 지표를 볼 수도 있습니다. 자세한 내용은 [DB 인스턴스 측정치 보기 \(p. 236\)](#) 단원을 참조하십시오.

CloudWatch 경보를 설정하려면

1. AWS Management 콘솔에 로그인한 후 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 후 DB 인스턴스를 선택합니다.
3. 모니터링 표시를 선택한 후 성능 지표를 선택하여 확장된 보기 를 불러옵니다.
4. [Create Alarm]을 선택합니다.
5. Create Alarm(경보 생성) 페이지에서 알림 받을 대상의 값을 선택하여 알림을 받을 이메일 주소를 지정합니다. 필요할 경우 해당 상자 오른쪽에 있는 주제 생성을 선택하여 경보 수신자를 새로 만듭니다.
6. Whenever(다음 경우 항상)에서 설정할 경보 통계를 선택합니다.
7. of(/)에서 경보 지표를 선택합니다.
8. Is 상자와 그 오른쪽에 있는 상자에서 다음과 같이 경보 임계값을 설정합니다.

Create Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a threshold. To edit an alarm, first choose whom to notify and then define when the notification should be sent.

Send a notification to: [create topic](#)

Whenever: of
Is: > Percent

For at least: consecutive period(s) of

Name of alarm:

9. For at least(최소 다음의 경우) 상자에 지정한 임계값에 몇 번 도달해야 경보를 발생할지 횟수를 입력합니다.
10. consecutive period(s) of(연속 기간) 상자에서 임계값이 얼마나 유지되어야 경보를 발생할지 기간을 선택합니다.
11. 경보 이름에 경보 이름을 입력합니다.
12. [Create Alarm]을 선택합니다.

성능 지표 페이지가 나타나고 CloudWatch 경보 상태 표시줄에 새로운 경보가 나타납니다. 상태 표시줄이 보이지 않으면 페이지를 새로 고치십시오.

성능 지표 평가

한 개의 DB 인스턴스에는 서로 다른 많은 카테고리의 지표가 있으며, 허용되는 값을 결정하는 방법은 지표에 따라 다릅니다.

CPU

- CPU 사용률 - 사용된 컴퓨터 처리 용량의 백분율입니다.

메모리

- 여유 메모리 - DB 인스턴스에서 사용 가능한 RAM을 메가바이트 단위로 나타냅니다. [Monitoring] 탭 지표의 붉은색 선이 CPU, 메모리 및 스토리지 지표의 75%에 표시됩니다. 인스턴스 메모리 소비가 주기적으로 이 선을 넘는 경우 이는 워크로드를 확인하거나 인스턴스를 업그레이드해야 함을 나타냅니다.
- 스왑 사용량 - DB 인스턴스에서 스왑 공간을 얼마나 많이 사용했는지를 메가바이트 단위로 나타냅니다.

디스크 공간

- 여유 스토리지 공간 - DB 인스턴스에서 현재 사용하지 않고 있는 디스크 공간의 크기(MB)

입력/출력 작업

- IOPS 읽기, IOPS 쓰기 - 초당 디스크 읽기 또는 쓰기 작업의 평균 횟수입니다.

- 읽기 지연 시간, 쓰기 지연 시간 - 읽기 또는 쓰기 작업의 평균 시간(밀리초)입니다.
- 읽기 처리량, 쓰기 처리량 - 초당 디스크에서 읽거나 디스크에 쓴 평균 크기(메가바이트)입니다.
- 대기열 길이 - 디스크에 쓰기 위해 또는 디스크에서 읽기 위해 대기 중인 I/O 작업의 수입니다.

네트워크 트래픽

- 네트워크 수신 처리량, 네트워크 전송 처리량 - DB 인스턴스에 대한 수신 또는 전송 네트워크 트래픽 속도 (초당 메가바이트)입니다.

데이터베이스 연결

- DB 연결 - DB 인스턴스에 연결된 클라이언트 세션의 수입니다.

사용 가능한 각 성능 지표에 대한 자세한 내용은 [Amazon RDS 차원 및 측정치](#)를 참조하십시오.

일반적으로 성능 지표에 허용되는 값은 기준이 무엇인지 그리고 애플리케이션 무엇을 수행하는지에 따라 달립니다. 기준과의 일관된 차이 또는 추세를 조사하십시오. 특정 지표 유형에 대한 참고 정보는 다음과 같습니다.

- CPU 또는 RAM 사용량이 많음 - CPU 또는 RAM 사용량이 많을 경우 해당 애플리케이션의 목표와 일치하고 예상되는 결과라면 문제가 되지 않을 수 있습니다.
- 디스크 공간 사용량 - 총 디스크 용량의 85퍼센트 이상이 계속 사용될 경우 디스크 공간 사용량을 검사합니다. 인스턴스에서 데이터를 삭제할 수 있는지 또는 다른 시스템에 데이터를 아카이브하여 공간을 확보할 수 있는지 확인합니다.
- 네트워크 트래픽 - 네트워크 트래픽의 경우 시스템 관리자에게 문의하여 해당 도메인 네트워크 및 인터넷 연결의 기대 처리량을 확인합니다. 처리량이 기대값보다 항상 낮으면 네트워크 트래픽을 검사합니다.
- 데이터베이스 연결 - 인스턴스 성능 저하 및 응답 시간 지연과 함께 사용자 연결 수가 많을 경우 데이터베이스 연결 제한을 고려해 봅니다. DB 인스턴스에 대한 최적의 사용자 연결 수는 해당 인스턴스 클래스와, 수행하는 작업의 복잡성에 따라 달립니다. DB 인스턴스를 User Connections 파라미터가 0(무제한)이 아닌 다른 값으로 설정된 파라미터 그룹과 연결하여 데이터베이스 연결 수를 지정할 수 있습니다. 기존 파라미터 그룹을 사용하거나 새로 하나 만들 수 있습니다. 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.
- IOPS 지표 - IOPS 지표의 기대값은 디스크 사양 및 서버 구성에 따라 다르므로 해당 기준에 일반적인 값을 파악합니다. 값이 기준과 계속 차이가 나는지 검사합니다. 최적의 IOPS 성능을 위해, 일반적인 작업 세트가 메모리에 적합하고 읽기 및 쓰기 작업을 최소화하는지 확인합니다.

성능 지표와 관련하여 문제가 있을 경우 성능을 향상하기 위해 제일 먼저 할 수 있는 것은, 가장 많이 사용되고 가장 비용이 높은 쿼리를 튜닝하여 시스템 리소스에 대한 부하를 줄일 수 있는지 확인하는 것입니다. 자세한 내용은 [쿼리 튜닝 \(p. 70\)](#) 단원을 참조하십시오.

쿼리를 튜닝해도 문제가 지속된다면, 문제와 관련된 리소스(CPU, RAM, 디스크 공간, 네트워크 대역폭, I/O 용량)를 추가하여 Amazon RDS [DB 인스턴스 클래스 선택 \(p. 77\)](#)를 업그레이드하십시오.

쿼리 튜닝

DB 인스턴스 성능을 향상하는 제일 좋은 방법 중 하나는 일반적으로 가장 많이 사용하는 쿼리와 리소스를 가장 많이 사용하는 쿼리를 튜닝하여 실행 비용을 낮추는 것입니다.

MySQL 쿼리 튜닝

성능을 높이기 위한 쿼리 작성에 대한 자세한 내용은 MySQL 문서에서 [Optimizing SELECT Statements](#)를 참조하십시오. 다른 쿼리 튜닝 리소스에 대한 내용은 [MySQL Performance Tuning and Optimization Resources](#)를 참조하십시오.

Oracle 쿼리 튜닝

성능을 높이기 위한 쿼리 쓰기 및 분석에 대한 자세한 내용은 Oracle 문서에서 [Database SQL Tuning Guide](#)를 참조하십시오.

SQL Server 쿼리 튜닝

SQL Server DB 인스턴스의 쿼리를 향상시키려면 SQL Server 문서에서 [Analyzing a Query](#)를 참조하십시오. 또한 [동적 관리 뷰 및 함수\(Transact-SQL\)](#) 문서에서 설명한 실행 관련, 인덱스 관련 및 I/O 관련 데이터 보기(DMV)를 사용하여 SQL Server 쿼리 문제를 해결할 수도 있습니다.

쿼리 튜닝의 공통적인 부분은 효율적인 인덱스를 만드는 것입니다. [데이터베이스 엔진 튜닝 관리자](#)를 사용하여 DB 인스턴스에 대한 인덱스 성능을 높일 수 있습니다. 자세한 내용은 [SQL Server 튜닝 어드바이저를 사용한 Amazon RDS DB 인스턴스의 데이터베이스 워크로드 분석 \(p. 557\)](#) 단원을 참조하십시오.

PostgreSQL 쿼리 튜닝

쿼리 계획을 분석하는 방법은 PostgreSQL 문서에서 [Using EXPLAIN](#)을 참조하십시오. 이 정보를 참조하여 쿼리 성능을 높이기 위해 쿼리나 기본 테이블을 수정할 수 있습니다. 또한 [Controlling the Planner with Explicit JOIN Clauses](#)에서 최적의 성능을 위해 쿼리에 조인을 지정하는 방법에 대한 도움말을 볼 수 있습니다.

MariaDB 쿼리 튜닝

성능을 높이기 위한 쿼리 쓰기에 대한 자세한 내용은 MariaDB 설명서에서 [Query Optimizations](#)를 참조하십시오.

MySQL 스토리지 엔진으로 작업하기 위한 모범 사례

MySQL DB 인스턴스에서 다음 테이블 생성 제한을 준수합니다.

- [프로비저닝된 IOPS 스토리지를 사용하거나, 범용 스토리지를 사용하고 DB 인스턴스가 200GiB 이상인 경우 10,000개의 테이블로 제한됩니다.](#)
- [표준 스토리지를 사용하거나 범용 스토리지를 사용하고 DB 인스턴스가 200GiB 미만인 경우 1,000개의 테이블로 제한됩니다.](#)

테이블 수가 많을 경우 장애 조치 또는 데이터베이스 충돌 후 데이터베이스 복구 시간이 크게 증가되므로 이러한 제한을 사용하는 것이 좋습니다. 권장 개수보다 많은 테이블을 만들 필요가 있는 경우에는 [innodb_file_per_table](#) 파라미터를 0으로 설정합니다. 자세한 내용은 [충돌 복구 시간 개선을 위한 InnoDB 테이블스페이스 작업 \(p. 671\)](#) 및 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

버전 5.7 이상을 사용하는 MySQL DB 인스턴스의 경우 InnoDB 충돌 복구 기능의 향상으로 이러한 테이블 생성 제한을 초과할 수 있습니다. 하지만 매우 많은 테이블을 생성할 경우 성능에 영향을 줄 수 있으므로 주의해야 합니다.

MySQL DB 인스턴스에서 데이터베이스의 테이블이 너무 크게 늘어나지 않도록 합니다. 프로비저닝 스토리지 제한에 따라 MySQL 테이블 파일의 최대 크기를 16TB로 제한합니다. 대신 파일 크기가 16TB 제한을 넘지 않도록 라지 테이블을 분할합니다. 이 접근 방식을 수행하면 성능 및 복구 시간도 향상할 수 있습니다. 자세한 내용은 [MySQL 파일 크기 제한 \(p. 675\)](#) 단원을 참조하십시오.

MySQL용 Amazon RDS의 특정 시점으로 복원 및 스냅샷 복원 기능을 사용하려면 중단 복구 가능 스토리지 엔진이 필요하며, 이러한 기능은 InnoDB 스토리지 엔진에서만 지원됩니다. MySQL은 다양한 기능을 가진 여러 스토리지 엔진을 지원하지만, 모든 엔진이 충돌 복구와 데이터 내구성에 최적화되어 있지는 않습니다. 예를 들어 MyISAM 스토리지 엔진은 안정적인 충돌 복구를 지원하지 않으며, 이로 인해 특정 시점으로 복원 또는 스냅샷 복원이 의도한 대로 작동하지 못할 수 있습니다. 그 결과 충돌 후 MySQL을 다시 시작하면 데이터가 손실되거나 손상될 수 있습니다.

InnoDB는 Amazon RDS에서 MySQL DB 인스턴스용으로 권장되고 지원되는 스토리지 엔진입니다. InnoDB 인스턴스는 Aurora로 마이그레이션할 수 있지만, MyISAM 인스턴스는 마이그레이션할 수 없습니다. 하지만 MyISAM은 강력한 전체 텍스트 검색 기능이 필요한 경우 InnoDB보다 나은 성능을 발휘합니다. 그래도 Amazon RDS와 함께 MyISAM을 사용하도록 선택할 경우 [지원되지 않는 MySQL 스토리지 엔진에 대한 자동 백업 \(p. 202\)](#)에 개략적으로 설명되어 있는 단계를 따르면 특정한 시나리오에서 스냅샷 복원 기능에 유용할 수 있습니다.

기존 MyISAM 테이블을 InnoDB 테이블로 변환하려는 경우 [MySQL 설명서](#)에 나와 있는 프로세스를 사용하면 됩니다. MyISAM과 InnoDB는 각기 다른 장점과 단점을 갖고 있으므로 전환하기 전에 이 전환이 애플리케이션에 미치는 영향을 충분히 평가해야 합니다.

또한 MySQL을 위한 외부 스토리지 엔진은 현재 Amazon RDS에서 지원되지 않습니다.

MariaDB 스토리지 엔진으로 작업하기 위한 모범 사례

MariaDB용 Amazon RDS의 특정 시점으로 복원 및 스냅샷 복원 기능을 사용하려면 충돌 복구 가능 스토리지 엔진이 필요합니다. MariaDB는 다양한 기능을 가진 여러 스토리지 엔진을 지원하지만, 모든 엔진이 충돌 복구와 데이터 내구성에 최적화되어 있지는 않습니다. 예를 들어, Aria가 충돌 안정성을 개선한 MyISAM 대체 스토리지 엔진이지만, 여전히 특정 시점으로 복원 또는 스냅샷 복원이 의도한 대로 작동하지 못할 수 있습니다. 그 결과 충돌 후 MariaDB를 다시 시작하면 데이터가 손실되거나 손상될 수 있습니다. InnoDB(버전 10.2 이상) 및 XtraDB(버전 10.0 및 10.1)는 의 MariaDB DB 인스턴스에 대한 권장 및 지원되는 스토리지 엔진입니다. 그래도 Amazon RDS와 함께 Aria를 사용하도록 선택할 경우 [지원되지 않는 MariaDB 스토리지 엔진에 대한 자동 백업 \(p. 203\)](#)에 개략적으로 설명되어 있는 단계를 따르면 특정한 시나리오에서 스냅샷 복원 기능에 유용할 수 있습니다.

Oracle 작업의 모범 사례

Amazon RDS for Oracle 작업의 모범 사례에 대한 자세한 내용은 [Amazon Web Services에서 Oracle Database 실행의 모범 사례](#) 및 [Amazon RDS에서 Oracle 데이터베이스 실행 비디오](#)를 참조하십시오.

PostgreSQL로 작업하기 위한 모범 사례

Amazon RDS에서 PostgreSQL로 성능을 향상 시킬 수 있는 두 가지 중요한 영역은 DB 인스턴스로 데이터를 로드할 때와 PostgreSQL autovacuum 기능을 사용할 때입니다. 다음 섹션에서는 이런 영역에 대해 권장하는 모범 사례를 다룹니다.

PostgreSQL DB 인스턴스에 데이터 로드

Amazon RDS PostgreSQL DB 인스턴스로 데이터를 로드할 때, 데이터를 DB 인스턴스로 가장 효율적으로 가져올 수 있도록 DB 인스턴스 설정과 DB 파라미터 그룹 값을 수정해야 합니다.

DB 인스턴스 설정을 다음과 같이 수정합니다.

- DB 인스턴스 백업 비활성화(`backup_retention`을 0으로 설정)
- Multi-AZ 비활성화

다음 설정을 포함하도록 DB 파라미터 그룹을 수정합니다. 파라미터 설정을 테스트하여 DB 인스턴스에 가장 효율적인 설정을 찾아야 합니다.

- `maintenance_work_mem` 파라미터의 값을 늘립니다. PostgreSQL 리소스 사용 파라미터에 대한 자세한 내용은 [PostgreSQL 문서](#)를 참조하십시오.
- wal 로그에 대한 쓰기 수를 줄이도록 `checkpoint_segments` 및 `checkpoint_timeout` 파라미터의 값을 늘립니다.
- `synchronous_commit` 파라미터를 비활성화합니다(FSYNC를 해제하지는 말 것).
- PostgreSQL autovacuum 파라미터를 비활성화합니다.
- 가져오려는 테이블 모두가 로깅되는지 확인합니다. 로깅되지 않는 테이블에 저장된 데이터는 장애 조치 중에 손실될 수 있습니다. 자세한 내용은 [CREATE TABLE UNLOGGED](#)를 참조하십시오.

이런 설정과 함께 `pg_dump -Fc(압축)` 또는 `pg_restore -j(병렬)` 명령을 사용합니다.

fsync 및 full_page_writes 데이터베이스 파라미터 작업

Amazon RDS의 PostgreSQL 9.4.1에서는 `fsync` 및 `full_page_writes` 데이터베이스 파라미터를 수정할 수 없습니다. `fsync` 및 `full_page_writes` 데이터베이스 파라미터를 비활성화하면 데이터가 손상될 수 있으므로 사용자 파라미터가 자동으로 활성화되었습니다. PostgreSQL의 다른 9.3 DB 엔진 버전을 사용하는 고객은 `fsync` 및 `full_page_writes` 파라미터를 비활성화하지 않는 것이 좋습니다.

PostgreSQL Autovacuum 기능 사용

PostgreSQL 데이터베이스용 autovacuum 기능은 PostgreSQL DB 인스턴스의 상태를 유지 관리하는 데 사용할 것을 강력히 권장하는 중요한 기능입니다. Autovacuum은 VACUUM 및 ANALYZE 명령의 실행을 자동화합니다. PostgreSQL에서는 autovacuum을 반드시 사용해야 하며 Amazon RDS에서는 필수 사항은 아니지만 훌륭한 성능을 달성하는 데 매우 중요한 기능입니다. 이 기능은 모든 새로운 Amazon RDS PostgreSQL DB 인스턴스에 대해 기본적으로 활성화되며, 관련 구성 파라미터는 적절히 기본적으로 설정됩니다.

데이터베이스 관리자는 이 유지 관리 작업을 파악하고 이해할 필요가 있습니다. Autovacuum에 관한 PostgreSQL 문서는 <http://www.postgresql.org/docs/current/static/routine-vacuuming.html#AUTOVACUUM>을 참조하십시오.

Autovacuum은 "리소스"를 일정 부분 사용하는 작업하지만, 백그라운드에서 작동하며 사용자 작업에 최대한 많은 리소스가 할당되도록 합니다. Autovacuum이 활성화되어 있을 때는 업데이트되거나 삭제된 튜플 수가 많은 테이블이 있는지 확인합니다. 또한, 이 기능은 [트랜잭션 ID 랙어라운드](#)로 인해 매우 오래된 데이터가 손실되지 않도록 보호합니다.

Autovacuum을 더 나은 성능을 위해 줄일 수 있는 높은 오버헤드 작업으로 생각하면 안 됩니다. 오히려, autovacuum을 실행하지 않으면 업데이트 및 삭제 속도가 빠른 테이블이 시간이 흐르면서 빠르게 성능이 저하됩니다.

Important

Autovacuum을 실행하지 않으면 훨씬 더 많은 주입식 vacuum 작업을 수행하기 위해 결국은 필연적으로 종단될 수 있습니다. Autovacuum을 지나치게 보수적으로 사용하는 바람에 Amazon RDS PostgreSQL DB 인스턴스를 사용할 수 없게 되면, PostgreSQL 데이터베이스는 스스로를 보호하기 위해 종료됩니다. 그 시점에서 Amazon RDS는 DB 인스턴스에서 직접 단일 사용자 모드 전체 vacuum을 수행해야 하며, 이때 여러 시간 동안 시스템이 종단될 수 있습니다. 따라서 기본적으로 활성화되는 autovaccum은 아예 해제하지 않는 것이 좋습니다.

Autovacuum 파라미터에 따라 autovacuum의 작동 시점과 정도가 결정됩니다.

`autovacuum_vacuum_threshold` 및 `autovacuum_vacuum_scale_factor` 파라미터에 따라 autovacuum 실행 시점이 결정됩니다. `autovacuum_max_workers`, `autovacuum_nap_time`, `autovacuum_cost_limit` 및 `autovacuum_cost_delay` 파라미터에 따라 autovacuum의 작동 정도가 결정됩니다. Autovacuum에 대한 자세한 정보, autovacuum의 작동 시점, 필요한 파라미터에 관해서는 [PostgreSQL 문서](#)를 참조하십시오.

다음 쿼리를 통해 table1로 명명된 테이블에 있는 "죽은" 튜플의 수를 알 수 있습니다.

```
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' order by n_dead_tup desc;
```

쿼리의 결과는 다음과 같은 내용일 것입니다.

relname	n_dead_tup	last_vacuum	last_autovacuum
tasks	81430522		

(1 row)

SQL Server로 작업하기 위한 모범 사례

SQL Server DB 인스턴스를 포함한 Multi-AZ 배포를 위한 모범 사례에는 다음이 포함됩니다.

- Amazon RDS RDS DB 이벤트를 사용하여 장애 조치를 모니터링합니다. 예를 들어 DB 인스턴스가 장애 조치할 때 문자 메시지 또는 이메일로 알림 서비스를 받을 수 있습니다. Amazon RDS 이벤트에 대한 자세한 내용은 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오.
- 애플리케이션이 DNS 값을 캐시하는 경우 TTL(time to live)을 30초 미만으로 설정합니다. 장애 조치가 발생하여 IP 주소가 바뀔 수 있고 캐시된 값은 더 이상 사용하지 못할 수도 있는 경우에 TTL을 그렇게 설정하는 것이 좋습니다.
- 다음 모드에서는 Multi-AZ에 필수적인 트랜잭션 로깅이 해제되므로, 이들 모드를 활성화하지 않는 것이 좋습니다.
 - 단순 복구 모드
 - 오프라인 모드
 - 읽기 전용 모드
- 테스트를 통해 DB 인스턴스를 장애 조치하는 데 얼마나 오래 걸리는지 확인합니다. 장애 조치 시간은 데이터베이스의 유형, 인스턴스 클래스, 사용하는 스토리지 유형에 따라 변할 수 있습니다. 장애 조치가 발생할 경우 작업을 계속 수행할 수 있는 애플리케이션의 능력도 테스트해야 합니다.
- 장애 조치 시간을 단축하려면 다음을 수행해야 합니다.
 - 워크로드에 대해 충분한 프로비저닝된 IOPS가 할당되어 있는지 확인하십시오. I/O가 부적합하면 장애 조치 시간이 길어질 수 있습니다. 데이터베이스 복구에 I/O가 필요합니다.
 - 더 작은 트랜잭션을 사용하십시오. 데이터베이스 복구는 트랜잭션에 의존하므로, 큰 트랜잭션을 여러 개의 작은 트랜잭션으로 나눌 수 있다면 장애 조치 시간이 단축될 것입니다.
 - 장애 조치 중에 지연 시간이 증가할 것이라는 점을 고려하십시오. 장애 조치 프로세스의 일부로서, Amazon RDS는 데이터를 새 대기 인스턴스로 자동으로 복제합니다. 이 복제는 새 데이터가 두 개의 서로 다른 DB 인스턴스로 커밋되는 종이라는 의미이므로, 대기 DB 인스턴스가 새 기본 DB 인스턴스를 따라잡을 때까지 약간의 지연 시간이 발생할 수 있습니다.
 - 모든 가용 영역에 애플리케이션을 배포합니다. 한 가용 영역이 다운되더라도 다른 가용 영역에 있는 애플리케이션을 계속 사용할 수 있습니다.

SQL Server의 다중 AZ 배포를 사용하여 작업할 때, Amazon RDS가 인스턴스에 있는 모든 SQL Server 데이터베이스의 복제본을 생성합니다. 특정 데이터베이스의 보조 복제본이 생기지 않도록 하려면 해당 데이터베이스에 대해 다중 AZ를 사용하지 않는 별개의 DB 인스턴스를 설정합니다.

DB 파라미터 그룹 작업

DB 파라미터 그룹 변경 내용을 프로덕션 DB 인스턴스에 적용하기 전에 테스트 DB 인스턴스에 적용해 보는 것이 좋습니다. DB 파라미터 그룹에 DB 엔진 파라미터를 잘못 설정하면 성능 저하나 시스템 불안정 등의 의

도하지 않은 부작용이 있을 수 있습니다. DB 엔진 파라미터를 수정할 때 항상 주의를 기울이고 DB 파라미터 그룹을 수정하기 전에 DB 인스턴스를 백업하십시오.

DB 인스턴스 백업에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 백업 및 복원 \(p. 196\)](#) 단원을 참조하십시오.

Amazon RDS 모범 사례 프레젠테이션 동영상

2016 AWS Summit 컨퍼런스(시카고)에는 Amazon RDS를 사용하여 안전하고 가용성이 높은 데이터베이스 인스턴스를 생성 및 구성하는 모범 사례에 대한 프레젠테이션이 포함되었습니다. 프레젠테이션 비디오는 [여기](#)에서 볼 수 있습니다.

Amazon RDS DB 인스턴스

DB 인스턴스는 클라우드에서 실행하는 격리된 데이터베이스 환경입니다. 이것은 Amazon RDS의 기본 구성 요소입니다. DB 인스턴스에는 여러 사용자가 만든 데이터베이스가 포함될 수 있으며, 독립 실행형 데이터베이스 인스턴스에 액세스할 때 사용하는 도구 및 애플리케이션을 사용해 액세스할 수 있습니다. Amazon AWS 명령줄 도구, Amazon RDS API 작업 또는 AWS Management 콘솔을 사용해 간단히 DB 인스턴스를 만들고 수정할 수 있습니다.

Note

Amazon RDS는 표준 SQL 클라이언트 애플리케이션을 사용하여 DB 인스턴스의 데이터베이스에 대한 액세스를 지원합니다. Amazon RDS는 호스트에 대한 직접적인 액세스를 허용하지 않습니다.

Amazon RDS DB 인스턴스는 최대 40개까지 보유할 수 있습니다. 이러한 인스턴스 40개 중 최대 10개는 '라이선스 포함' 모델에 따른 Oracle 또는 SQL Server DB 인스턴스가 될 수 있습니다. DB 인스턴스 40개 모두 MySQL, MariaDB 또는 PostgreSQL용으로 이용할 수 있습니다. 또한 DB 인스턴스 40개를 'BYOL' 라이선스 모델에 따른 SQL Server 또는 Oracle용으로 이용할 수도 있습니다. 애플리케이션에 더 많은 DB 인스턴스가 요구되는 경우, <https://console.aws.amazon.com/support/home#/case/create?issueType=service-limit-increase&limitType=service-code-rds-instances>에 있는 양식을 사용하여 추가 DB 인스턴스를 요청할 수 있습니다.

DB 인스턴스마다 DB 인스턴스 식별자가 있습니다. 이 식별자는 고객이 제공하는 이름으로, Amazon RDS API 및 AWS CLI 명령과 상호 작용하는 경우 DB 인스턴스를 고유하게 식별합니다. DB 인스턴스 식별자는 한 AWS 리전 내의 해당 고객에 대해 고유해야 합니다.

각 DB 인스턴스는 데이터베이스 엔진을 지원합니다. Amazon RDS는 현재 MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server, Amazon Aurora 데이터베이스 엔진을 지원합니다.

DB 인스턴스를 만들 때, 일부 데이터베이스 엔진의 경우 데이터베이스 이름을 지정해야 합니다. 한 DB 인스턴스에서 여러 개의 데이터베이스 또는 여러 스키마를 포함하는 단일 Oracle 데이터베이스를 호스팅할 수 있습니다. 데이터베이스의 이름 값은 데이터베이스 엔진에 따라 다음과 같이 달라집니다.

- MySQL 및 MariaDB 데이터베이스 엔진의 경우, 데이터베이스 이름은 DB 인스턴스에서 호스팅하는 데이터베이스의 이름입니다. 동일한 DB 인스턴스가 호스팅하는 데이터베이스는 해당 인스턴스 내부에서 고유 이름이 부여되어야 합니다.
- Oracle 데이터베이스 엔진의 경우, 데이터베이스 이름을 사용하여 ORACLE_SID의 값을 설정하며, 이는 Oracle RDS 인스턴스에 연결할 때 제공해야 합니다.
- Microsoft SQL Server 데이터베이스 엔진의 경우, 데이터베이스 이름은 지원되는 파라미터가 아닙니다.
- PostgreSQL 데이터베이스 엔진의 경우, 데이터베이스 이름은 DB 인스턴스에서 호스팅되는 데이터베이스의 이름입니다. DB 인스턴스를 만들 때 데이터베이스 이름을 지정할 필요가 없습니다. 동일한 DB 인스턴스가 호스팅하는 데이터베이스는 해당 인스턴스 내부에서 고유 이름이 부여되어야 합니다.

Amazon RDS는 DB 인스턴스를 생성하는 중에 DB 인스턴스의 마스터 사용자 계정을 생성합니다. 이 마스터 사용자는 데이터베이스 생성을 비롯하여 마스터 사용자가 생성하는 테이블의 생성, 삭제, 선택 및 업데이트 및 삽입 작업을 수행할 수 있는 권한을 가집니다. DB 인스턴스를 만들 때 마스터 사용자 암호를 설정해야 합니다. 마스터 사용자 암호는 Amazon AWS 명령줄 도구, Amazon RDS API 작업 또는 AWS Management 콘솔을 사용하여 언제든지 변경할 수 있습니다. 또한 SQL 표준 명령을 사용하여 마스터 사용자 암호를 변경하고 사용자를 관리할 수도 있습니다.

Note

이 지침은 Aurora 이외의 다른 Amazon RDS 데이터베이스 엔진에도 적용됩니다. Amazon Aurora 사용에 대한 자세한 정보는 [Amazon Aurora 사용 설명서](#)를 참조하십시오.

주제

- [DB 인스턴스 클래스 선택 \(p. 77\)](#)

- DB 인스턴스 상태 (p. 96)
- 리전 및 가용 영역 (p. 99)
- Amazon RDS DB 인스턴스 스토리지 (p. 101)
- Amazon RDS를 위한 고가용성(다중 AZ) (p. 109)
- Amazon RDS DB 인스턴스 수명 주기 (p. 111)
- Amazon RDS 리소스에 태그 지정 (p. 131)
- 읽기 전용 복제본 작업 (p. 135)
- 옵션 그룹 작업 (p. 148)
- DB 파라미터 그룹 작업 (p. 161)
- Amazon RDS의 Amazon 리소스 이름(ARN)을 사용한 작업 (p. 172)
- Amazon RDS용 스토리지 DB 인스턴스 작업 (p. 178)
- Amazon RDS에 대한 DB 인스턴스 결제 (p. 183)

DB 인스턴스 클래스 선택

DB 인스턴스 클래스는 Amazon RDS DB 인스턴스의 계산 및 메모리 용량을 결정합니다. 필요한 DB 인스턴스 클래스는 DB 인스턴스의 처리력 및 메모리 요구 사항에 따라 다릅니다.

인스턴스 클래스 요금에 대한 자세한 내용은 [Amazon RDS 요금](#)을 참조하십시오.

주제

- DB 인스턴스 클래스 유형 (p. 77)
- DB 인스턴스 클래스 하드웨어 사양 관련 용어 (p. 78)
- 사용 가능한 모든 DB 인스턴스 클래스의 하드웨어 사양 (p. 79)
- 사용 가능한 모든 DB 인스턴스 클래스에 대해 지원되는 DB 엔진 (p. 82)
- DB 인스턴스 클래스 변경 (p. 87)
- DB 인스턴스 클래스의 프로세서 구성 (p. 87)

DB 인스턴스 클래스 유형

Amazon RDS는 표준, 메모리 최적화 및 버스트 성능의 3가지 인스턴스 클래스 유형을 지원합니다. Amazon EC2 인스턴스 유형에 대한 자세한 내용은 Amazon EC2 설명서의 [인스턴스 유형](#)을 참조하십시오.

다음은 사용 가능한 표준 DB 인스턴스 클래스입니다.

- db.m5 – 컴퓨팅, 메모리 및 네트워크 리소스가 균형잡힌 최신 세대 범용 인스턴스 클래스로, 대부분의 애플리케이션에 적합합니다. db.m5 인스턴스 클래스는 이전의 db.m4 인스턴스 클래스보다 더 많은 컴퓨팅 용량을 제공합니다.
- db.m4 – 이전 db.m3 인스턴스 클래스에 비해 더 많은 컴퓨팅 용량을 제공하는 최신 범용 인스턴스 클래스입니다.
- db.m3 – 이전 db.m1 인스턴스 클래스에 비해 더 많은 컴퓨팅 용량을 제공하는 이전 범용 인스턴스 클래스입니다.
- db.m1 – 전세대 범용 인스턴스 클래스입니다.

다음은 사용 가능한 메모리 최적화 DB 인스턴스 클래스입니다.

- db.x1e – 메모리 집약적 애플리케이션에 최적화된 최신 세대 인스턴스 클래스입니다. 이러한 점은 DB 인스턴스 클래스의 RAM GiB 및 DRAM 기반 인스턴스 메모리의 최대 3,904GiB 당 최저 비용을 듭니다. db.x1e 인스턴스 클래스는 미국 동부(버지니아 북부), 미국 서부(오레곤), EU(아일랜드), 아시아 태평양(도쿄), 아시아 태평양(시드니) 리전에서만 사용할 수 있습니다.

- db.x1 – 메모리 집약적 애플리케이션에 최적화된 현재 세대 인스턴스 클래스 이러한 클래스는 DB 인스턴스 클래스의 RAM GiB 및 DRAM 기반 인스턴스 메모리의 최대 3,904GiB 당 최저 비용을 듭니다.
- db.r5 – 메모리 집약적 애플리케이션에 최적화된 최신 세대 인스턴스 클래스입니다. 향상된 네트워킹과 Amazon Elastic Block Store(Amazon EBS) 성능을 제공합니다. 전용 하드웨어 및 경량 하이파이바이저 결합된 AWS Nitro System을 기반으로 합니다.
- db.r4 – 메모리 집약적 애플리케이션에 최적화된 현재 세대 인스턴스 클래스 향상된 네트워킹과 Amazon EBS 성능을 제공합니다.
- db.r3 – 메모리를 최적화하는 전세대 인스턴스 클래스 db.r3 인스턴스 클래스는 EU(파리) 리전 및 남아메리카(상파울루) 리전에서는 사용할 수 없습니다.
- db.m2 – 전세대 메모리 최적화 인스턴스 클래스

다음은 사용 가능한 버스트 가능 성능 DB 인스턴스 클래스입니다.

- db.t3 – CPU 사용률을 최대로 버스트할 수 있는 기능을 통해 기준 성능 수준을 제공하는 최신 세대 인스턴스 클래스입니다. 이 인스턴스 클래스는 이전의 db.t2 인스턴스 클래스보다 더 많은 컴퓨팅 용량을 제공합니다.
- db.t2 – CPU 사용률을 최대로 버스트할 수 있는 기능을 통해 기준 성능 수준을 제공하는 현재 세대 인스턴스 클래스입니다.

DB 인스턴스 클래스 하드웨어 사양 관련 용어

다음 용어는 DB 인스턴스 클래스의 하드웨어 사양을 기술하는 데 사용됩니다.

- vCPU – 가상 CPU(중앙 처리 디바이스)의 수입니다. 가상 CPU는 DB 인스턴스 클래스를 비교하는 데 사용 할 수 있는 용량 단위입니다. 특정 프로세서를 구매하거나 임차해 몇 개월 또는 몇 년간 사용하는 것이 아니라, 시간 단위로 용량을 임대합니다. 목표는 실제 기본 하드웨어의 제한 내에서 일정하고 구체적인 CPU 용량을 제공하는 것입니다.
- ECU – Amazon EC2 인스턴스의 정수 처리 파워에 대한 상대적인 척도입니다. 개발자들이 다양한 인스턴스 클래스 간에 CPU 용량을 손쉽게 비교할 수 있도록 컴퓨팅 유닛(ECU)을 정의했습니다. 특정 인스턴스에 할당된 CPU의 용량은 이러한 ECU로 표현됩니다. 현재 ECU 한 개당 제공하는 CPU 용량은 1.0-1.2GHz 2007 Opteron 또는 2007 Xeon 프로세서와 동일합니다.
- 메모리(GiB) – DB 인스턴스에 할당되는 RAM(단위: 기비바이트)입니다. 메모리와 vCPU 간 일정한 비율이 존재하는 경우가 많다는 점에 유의하십시오. db.r5 인스턴스 클래스와 비슷한 vCPU 비율에 대한 메모리가 있는 db.r4 인스턴스 클래스를 예로 들 수 있습니다. 그러나 대부분의 사용 db.r5 인스턴스 클래스는 db.r4 인스턴스 클래스보다 더 낮고 더 일관성 있는 성능을 제공합니다.
- VPC 전용 – 인스턴스 클래스는 Amazon VPC 서비스에 기반을 둔 VPC에 있는 DB 인스턴스에 대해서만 지원됩니다. VPC가 필요한 인스턴스 클래스를 사용하고 싶지만 현재 DB 인스턴스가 VPC에 없는 경우가 있을 수 있습니다. 이러한 경우에는 DB 인스턴스를 VPC로 이동하는 것부터 시작하십시오. 자세한 내용은 [VPC에 있지 않은 DB 인스턴스를 VPC로 이동 \(p. 420\)](#) 단원을 참조하십시오.
- EBS 최적화 – DB 인스턴스는 최적화된 구성 스택을 사용하며 I/O를 위한 추가 전용 용량을 제공합니다. 이러한 최적화를 통해 인스턴스에서 I/O와 기타 트래픽 간의 경합이 최소화되어 성능이 극대화됩니다. Amazon EBS-최적화 인스턴스에 관한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EBS-최적화 인스턴스를 참조하십시오.](#)
- 최대 대역폭(Mbps) – 초당 메가비트 단위로 최대 대역폭입니다. 이 값을 8로 나누면 초당 메가바이트 단위로 예상되는 처리량을 구할 수 있습니다.

Important

Amazon RDS DB 인스턴스의 범용 SSD(gp2) 볼륨은 대부분의 경우 처리량 제한이 250MiB/s입니다. 그러나 처리량 제한은 볼륨 크기에 따라 달라질 수 있습니다. 자세한 내용은 Linux 인스턴스 용 Amazon EC2 사용 설명서의 [Amazon EBS 볼륨 유형](#)을 참조하십시오. gp2 스토리지의 대역폭 추정에 대한 자세한 내용은 [범용 SSD 스토리지 \(p. 101\)](#) 단원을 참조하십시오.

- 네트워크 성능 – 다른 DB 인스턴스 클래스 대비 네트워크 속도입니다.

사용 가능한 모든 DB 인스턴스 클래스의 하드웨어 사양

아래 표에서 Amazon RDS DB 인스턴스 클래스의 세부 정보를 확인하실 수 있습니다. 표 열에 있는 용어에 대한 자세한 설명은 [DB 인스턴스 클래스 하드웨어 사양 관련 용어 \(p. 78\)](#) 단원을 참조하십시오. 각 DB 인스턴스 클래스에 대한 Amazon RDS DB 엔진 지원에 관한 자세한 내용은 [사용 가능한 모든 DB 인스턴스 클래스에 대해 지원되는 DB 엔진 \(p. 82\)](#) 단원을 참조하십시오.

인스턴스 클래스	vCPU	ECU	메모리 (GiB)	VPC 전용	EBS 최적	최대 대역폭 (Mbps)	네트워크 성능
db.m5 – 최신 세대 표준 인스턴스 클래스							
db.m5.24xlarge	96	345	384	예	예	14,000	25Gbps
db.m5.12xlarge	48	173	192	예	예	7,000	10Gbps
db.m5.4xlarge	16	61	64	예	예	3,500	최대 10기가비트
db.m5.2xlarge	8	31	32	예	예	3,500	최대 10기가비트
db.m5.xlarge	4	15	16	예	예	3,500	최대 10기가비트
db.m5.large	2	10	8	예	예	3,500	최대 10기가비트
db.m4 – 현재 세대 표준 인스턴스 클래스							
db.m4.16xlarge	64	188	256	예	예	10,000	25Gbps
db.m4.10xlarge	40	124.5	160	예	예	4,000	10Gbps
db.m4.4xlarge	16	53.5	64	예	예	2,000	높음
db.m4.2xlarge	8	25.5	32	예	예	1,000	높음
db.m4.xlarge	4	13	16	예	예	750	높음
db.m4.large	2	6.5	8	예	예	450	보통
db.m3 – 전세대 표준 인스턴스 클래스							
db.m3.2xlarge	8	26	30	아니요	예	1,000	높음
db.m3.xlarge	4	13	15	아니요	예	500	높음
db.m3.large	2	6.5	7.5	아니요	아니요	—	보통
db.m3.medium	1	3	3.75	아니요	아니요	—	보통
db.m1 – 전세대 표준 인스턴스 클래스							
db.m1.xlarge	4	4	15	아니요	예	450	높음
db.m1.large	2	2	7.5	아니요	예	450	보통
db.m1.medium	1	1	3.75	아니요	아니요	—	보통
db.m1.small	1	1	1.7	아니요	아니요	—	매우 낮음

인스턴스 클래스	vCPU	ECU	메모리 (GiB)	VPC 전용	EBS 최적	최대 대역폭(Mbps)	네트워크 성능
db.x1e – 최신 세대 메모리 최적화 인스턴스 클래스							
db.x1e.32xlarge	128	340	3,904	예	예	14,000	25Gbps
db.x1e.16xlarge	64	179	1,952	예	예	7,000	10Gbps
db.x1e.8xlarge	32	91	976	예	예	3,500	최대 10Gbps
db.x1e.4xlarge	16	47	488	예	예	1,750	최대 10Gbps
db.x1e.2xlarge	8	23	244	예	예	1,000	최대 10Gbps
db.x1e.xlarge	4	12	122	예	예	500	최대 10Gbps
db.x1 – 현재 세대 메모리 최적화 인스턴스 클래스							
db.x1.32xlarge	128	349	1,952	예	예	14,000	25Gbps
db.x1.16xlarge	64	174.5	976	예	예	7,000	10Gbps
db.r5 – 최신 세대 메모리 최적화 인스턴스 클래스							
db.r5.24xlarge	96	347	768	예	예	14,000	25Gbps
db.r5.12xlarge	48	173	384	예	예	7,000	10Gbps
db.r5.4xlarge	16	71	128	예	예	3,500	최대 10Gbps
db.r5.2xlarge	8	38	64	예	예	최대 3,500	최대 10Gbps
db.r5.xlarge	4	19	32	예	예	최대 3,500	최대 10Gbps
db.r5.large	2	10	16	예	예	최대 3,500	최대 10Gbps
db.r4 – 현재 세대 메모리 최적화 인스턴스 클래스							
db.r4.16xlarge	64	195	488	예	예	14,000	25Gbps
db.r4.8xlarge	32	99	244	예	예	7,000	10Gbps
db.r4.4xlarge	16	53	122	예	예	3,500	최대 10Gbps
db.r4.2xlarge	8	27	61	예	예	1,750	최대 10Gbps
db.r4.xlarge	4	13.5	30.5	예	예	875	최대 10Gbps
db.r4.large	2	7	15.25	예	예	437	최대 10Gbps

인스턴스 클래스	vCPU	ECU	메모리 (GiB)	VPC 전용	EBS 최적	최대 대역폭(Mbps)	네트워크 성능
db.r3 – 전세대 메모리 최적화 인스턴스 클래스							
db.r3.8xlarge	32	104	244	아니요	아니요	—	10Gbps
db.r3.4xlarge	16	52	122	아니요	예	2,000	높음
db.r3.2xlarge	8	26	61	아니요	예	1,000	높음
db.r3.xlarge	4	13	30.5	아니요	예	500	보통
db.r3.large	2	6.5	15.25	아니요	아니요	—	보통
db.m2 – 전세대 메모리 최적화 인스턴스 클래스							
db.m2.4xlarge	8	26	68.4	아니요	예	1,000	높음
db.m2.2xlarge	4	13	34.2	아니요	예	500	보통
db.m2.xlarge	2	6.5	17.1	아니요	아니요	—	보통
db.t3 – 최신 세대 성능 순간 확장 가능 인스턴스 클래스							
db.t3.2xlarge	8	변수	32	예	예	2,050	최대 5기가비트
db.t3.xlarge	4	변수	16	예	예	2,050	최대 5기가비트
db.t3.large	2	변수	8	예	예	2,050	최대 5기가비트
db.t3.medium	2	변수	4	예	예	1,500	최대 5기가비트
db.t3.small	2	변수	2	예	예	1,500	최대 5기가비트
db.t3.micro	2	변수	1	예	예	1,500	최대 5기가비트
db.t2 – 현재 세대 버스트 성능 인스턴스 클래스							
db.t2.2xlarge	8	변수	32	예	아니요	—	보통
db.t2.xlarge	4	변수	16	예	아니요	—	보통
db.t2.large	2	변수	8	예	아니요	—	보통
db.t2.medium	2	변수	4	예	아니요	—	보통
db.t2.small	1	변수	2	예	아니요	—	낮음
db.t2.micro	1	변수	1	예	아니요	—	낮음

사용 가능한 모든 DB 인스턴스 클래스에 대해 지원되는 DB 엔진

아래 표에서 각 Amazon RDS DB 엔진에 지원되는 Amazon RDS DB 인스턴스 클래스의 세부 정보를 확인하실 수 있습니다. DB 인스턴스 클래스 사양은 [사용 가능한 모든 DB 인스턴스 클래스의 하드웨어 사양 \(p. 79\)](#) 단원을 참조하십시오.

다음은 DB 인스턴스 클래스에 대한 DB 엔진 고려 사항입니다.

- Microsoft SQL Server – 인스턴스 클래스 지원은 SQL Server 버전과 에디션에 따라 달라집니다. 버전 및 에디션별 인스턴스 클래스 지원은 [Microsoft SQL Server를 위한 DB 인스턴스 클래스 지원 \(p. 481\)](#) 단원을 참조하십시오.
- Oracle – 인스턴스 클래스 지원은 Oracle 버전과 에디션에 따라 달라집니다. 버전 및 에디션별 인스턴스 클래스 지원은 [Oracle을 위한 DB 인스턴스 클래스 지원 \(p. 705\)](#) 단원을 참조하십시오.
- PostgreSQL – db.m5, db.r5 및 db.t3 DB 인스턴스 클래스는 다음 Amazon RDS PostgreSQL 버전에 대해 지원됩니다.
 - PostgreSQL 9.6.9 및 상위 9.6 버전
 - PostgreSQL 10.4 및 상위 10 버전
 - PostgreSQL 11.1 및 상위 11 버전

인스턴스 클래스	Mar	Mic	MyS	Ora	Pos										
db.m5 – 최신 세대 표준 인스턴스 클래스															
db.m5.24xlarge	예	예	예	예	예										
db.m5.12xlarge	예	예	예	예	예										
db.m5.4xlarge	예	예	예	예	예										
db.m5.2xlarge	예	예	예	예	예										
db.m5.xlarge	예	예	예	예	예										
db.m5.large	예	예	예	예	예										
db.m4 – 현재 세대 표준 인스턴스 클래스															
db.m4.16xlarge	예	예	MySQL 8.0, 5.7, 5.6	예											
db.m4.10xlarge	예	예	예	예	예										
db.m4.4xlarge	예	예	예	예	예										
db.m4.2xlarge	예	예	예	예	예										
db.m4.xlarge	예	예	예	예	예										
db.m4.large	예	예	예	예	예										

인스턴스 클래스	Mar	Mic	MySQL	Ora	PostgreSQL							
db.m3 – 전세대 표준 인스턴스 클래스												
db.m3.2xlarge	아 니 요	예	예	예	예							
db.m3.xlarge	아 니 요	예	예	예	예							
db.m3.large	아 니 요	예	예	예	예							
db.m3.medium	아 니 요	예	예	예	예							
db.m1 – 전세대 표준 인스턴스 클래스												
db.m1.xlarge	아 니 요	예	MySQL 5.6, 5.5	사용 되 지 않 음	PostgreSQL 9.4, 9.3							
db.m1.large	아 니 요	예	MySQL 5.6, 5.5	사용 되 지 않 음	PostgreSQL 9.4, 9.3							
db.m1.medium	아 니 요	예	MySQL 5.6, 5.5	사용 되 지 않 음	PostgreSQL 9.4, 9.3							
db.m1.small	아 니 요	예	MySQL 5.6, 5.5	사용 되 지 않 음	PostgreSQL 9.4, 9.3							
db.x1e – 최신 세대 메모리 최적화 인스턴스 클래스												

인스턴스 클래스	Mar	Mic	MyS	Ora	Pos								
db.x1e.32xlarge	아 니 요	아 니 요	아 니 요	예	아 니 요								
db.x1e.16xlarge	아 니 요	아 니 요	아 니 요	예	아 니 요								
db.x1e.8xlarge	아 니 요	아 니 요	아 니 요	예	아 니 요								
db.x1e.4xlarge	아 니 요	아 니 요	아 니 요	예	아 니 요								
db.x1e.2xlarge	아 니 요	아 니 요	아 니 요	예	아 니 요								
db.x1e.xlarge	아 니 요	아 니 요	아 니 요	예	아 니 요								
db.x1 – 현재 세대 메모리 최적화 인스턴스 클래스													
db.x1.32xlarge	아 니 요	아 니 요	아 니 요	예	아 니 요								
db.x1.16xlarge	아 니 요	아 니 요	아 니 요	예	아 니 요								
db.r5 – 최신 세대 메모리 최적화 인스턴스 클래스													
db.r5.24xlarge	예	아 니 요		예	예	예							
db.r5.12xlarge	예	아 니 요		예	예	예							
db.r5.4xlarge	예	아 니 요		예	예	예							
db.r5.2xlarge	예	아 니 요		예	예	예							

인스턴스 클래스	Mar	Mic	MySQL Ser	MySQ	Ora	Pos								
db.r5.xlarge	예	아 니 요	예	예	예	예								
db.r5.large	예	아 니 요	예	예	예	예								
db.r4 – 현재 세대 메모리 최적화 인스턴스 클래스														
db.r4.16xlarge	예	예	MySQL 8.0, 5.7, 5.6	MySQL 8.0, 5.7, 5.6	PostgreSQL 9.6, 9.5, 9.4	PostgreSQL 9.6, 9.5, 9.4								
db.r4.8xlarge	예	예	MySQL 8.0, 5.7, 5.6	MySQL 8.0, 5.7, 5.6	PostgreSQL 9.6, 9.5, 9.4	PostgreSQL 9.6, 9.5, 9.4								
db.r4.4xlarge	예	예	MySQL 8.0, 5.7, 5.6	MySQL 8.0, 5.7, 5.6	PostgreSQL 9.6, 9.5, 9.4	PostgreSQL 9.6, 9.5, 9.4								
db.r4.2xlarge	예	예	MySQL 8.0, 5.7, 5.6	MySQL 8.0, 5.7, 5.6	PostgreSQL 9.6, 9.5, 9.4	PostgreSQL 9.6, 9.5, 9.4								
db.r4.xlarge	예	예	MySQL 8.0, 5.7, 5.6	MySQL 8.0, 5.7, 5.6	PostgreSQL 9.6, 9.5, 9.4	PostgreSQL 9.6, 9.5, 9.4								
db.r4.large	예	예	MySQL 8.0, 5.7, 5.6	MySQL 8.0, 5.7, 5.6	PostgreSQL 9.6, 9.5, 9.4	PostgreSQL 9.6, 9.5, 9.4								
db.r3 – 전세대 메모리 최적화 인스턴스 클래스														
db.r3.8xlarge	예	예	예	예	예	예								
db.r3.4xlarge	예	예	예	예	예	예								
db.r3.2xlarge	예	예	예	예	예	예								
db.r3.xlarge	예	예	예	예	예	예								
db.r3.large	예	예	예	예	예	예								
db.m2 – 전세대 메모리 최적화 인스턴스 클래스														

인스턴스 클래스	Mar	Mic	MyS	Ora	Pos								
db.m2.4xlarge	아 니 요	예	MySQL 5.6, 5.5	MySQL 5.6, 5.5	MySQL 5.6, 5.5	PostgreSQL 9.4, 9.3							
db.m2.2xlarge	아 니 요	예	MySQL 5.6, 5.5	MySQL 5.6, 5.5	MySQL 5.6, 5.5	PostgreSQL 9.4, 9.3							
db.m2.xlarge	아 니 요	예	MySQL 5.6, 5.5	MySQL 5.6, 5.5	MySQL 5.6, 5.5	PostgreSQL 9.4, 9.3							
db.t3 – 최신 세대 성능 순간 확장 가능한 인스턴스 클래스													
db.t3.2xlarge	예	아 니 요	예	예	예	예							
db.t3.xlarge	예	아 니 요	예	예	예	예							
db.t3.large	예	아 니 요	예	예	예	예							
db.t3.medium	예	아 니 요	예	예	예	예							
db.t3.small	예	아 니 요	예	예	예	예							
db.t3.micro	예	아 니 요	예	예	예	예							
db.t2 – 현재 세대 버스트 성능 인스턴스 클래스													
db.t2.2xlarge	예	아 니 요	MySQL 8.0, 5.7, 5.6	MySQL 8.0, 5.7, 5.6	MySQL 8.0, 5.7, 5.6	PostgreSQL 9.6, 9.5, 9.4							

인스턴스 클래스	Mari	Micro	MySQL	Oracle	PostgreSQL								
db.t2.xlarge	예	아 니 요	MySQL 8.0, 5.7, 5.6	PostgreSQL 9.6, 9.5, 9.4									
db.t2.large	예	예	예	예	예								
db.t2.medium	예	예	예	예	예								
db.t2.small	예	예	예	예	예								
db.t2.micro	예	예	예	예	예								

DB 인스턴스 클래스 변경

DB 인스턴스 클래스를 변경하여 DB 인스턴스에서 사용 가능한 CPU 및 메모리를 변경할 수 있습니다. DB 인스턴스 클래스를 변경하려면 특정 데이터베이스 엔진에 대한 지침에 따라 DB 인스턴스를 수정하십시오.

- [MariaDB 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 449\)](#)
- [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#)
- [MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 600\)](#)
- [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#)
- [PostgreSQL 데이터베이스 엔진을 실행 중인 DB 인스턴스 수정 \(p. 956\)](#)

일부 인스턴스 클래스에서는 DB 인스턴스가 VPC에 있어야 합니다. 현재 DB 인스턴스가 VPC에 없는데 VPC가 필요한 인스턴스 클래스를 사용하려는 경우 먼저 DB 인스턴스를 VPC로 옮기십시오. 자세한 내용은 [VPC에 있지 않은 DB 인스턴스를 VPC로 이동 \(p. 420\)](#) 단원을 참조하십시오.

DB 인스턴스 클래스의 프로세서 구성

Amazon RDS DB 인스턴스 클래스는 여러 개의 스레드를 하나의 Intel Xeon CPU 코어에서 동시에 실행할 수 있는 하이퍼 스레딩 기술을 지원합니다. 각 스레드는 DB 인스턴스에서 가상 CPU(vCPU)로 표현됩니다. DB 인스턴스에는 DB 인스턴스 유형에 따라 다른 기본 CPU 코어 수가 있습니다. 예를 들어, db.m4.xlarge DB 인스턴스 유형에는 기본적으로 2개의 CPU 코어 및 코어당 2개의 스레드, 총 4개의 vCPU가 있습니다.

Note

각 vCPU는 Intel Xeon CPU 코어의 하이퍼 스레딩입니다.

주제

- [프로세서 구성 개요 \(p. 87\)](#)
- [DB 인스턴스 클래스별 CPU 코어 및 CPU 코어당 스레드 \(p. 88\)](#)
- [DB 인스턴스 클래스의 CPU 코어 및 CPU 코어당 스레드 설정 \(p. 90\)](#)

프로세서 구성 개요

대부분의 경우 워크로드에 적합하게 메모리와 vCPU 수가 결합된 DB 인스턴스 클래스를 확인할 수 있습니다. 하지만 다음 프로세서 기능을 지정하여 특정 워크로드 또는 비즈니스 필요에 맞게 DB 인스턴스를 최적화 할 수도 있습니다.

- CPU 코어 수 – DB 인스턴스에 대한 CPU 코어 수를 사용자 지정할 수 있습니다. 이를 통해, 메모리 집약 워크로드용 RAM이 충분하면서도 CPU 코어를 적게 사용하는 DB 인스턴스의 소프트웨어 라이선스 비용을 잠재적으로 최적화할 수 있습니다.
- 코어당 스레드 – CPU 코어당 단일 스레드를 지정하여 인텔 하이퍼 스레딩 기술을 비활성화할 수 있습니다. HPC(고성능 컴퓨팅) 워크로드와 같은 특정 워크로드에 대해 이 작업을 수행할 수 있습니다.

각 코어에 대해 개별적으로 CPU 코어 및 스레드 수를 제어할 수 있습니다. 요청에서 하나의 값 또는 두 값을 설정할 수 있습니다. 설정이 DB 인스턴스와 연결된 후에는 변경할 때까지 설정이 지속됩니다.

DB 인스턴스에 대한 프로세서 설정은 DB 인스턴스의 스냅샷과 연결됩니다. 스냅샷이 복원되면 복원된 DB 인스턴스는 스냅샷을 생성할 때 사용된 프로세서 기능 설정을 사용합니다.

DB 인스턴스의 DB 인스턴스 클래스를 기본값이 아닌 프로세서 설정으로 수정하는 경우 수정 시 기본 프로세서 설정을 지정하거나 프로세서 설정을 명시적으로 지정하십시오. 이 요구 사항을 준수하면 DB 인스턴스를 수정할 때 발생할 수 있는 타사 라이선스 비용을 알 수 있습니다.

Amazon RDS DB 인스턴스에서 프로세서 기능 지정에 대한 추가 요금이나 비용 경감은 없습니다. 기본 CPU 구성으로 시작한 DB 인스턴스와 동일한 요금이 청구됩니다.

DB 인스턴스 클래스별 CPU 코어 및 CPU 코어당 스레드

다음 표에서는 다수의 CPU 코어 및 코어당 CPU 스레드 설정을 지원하는 DB 인스턴스 클래스를 확인할 수 있습니다. 각 DB 인스턴스 클래스에 대한 CPU 코어 및 코어당 CPU 스레드 수의 기본값과 유효한 값도 확인할 수 있습니다.

DB 인스턴스 클래스	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
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.m5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m5.24xlarge	96	48	2	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 인스턴스 클래스	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
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
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.r5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r5.24xlarge	96	48	2	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 인스턴스 클래스	기본 vCPU	기본 CPU 코어	코어당 기본 스레드	유효한 CPU 코어 수	코어당 유효한 스레드 수
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.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

Note

현재는 Oracle DB 인스턴스에 대해서만 CPU 코어 및 코어당 스레드 수를 구성할 수 있습니다. 다른 Oracle 데이터베이스 에디션에서 지원되는 DB 인스턴스 클래스에 대한 자세한 내용은 [Oracle을 위한 DB 인스턴스 클래스 지원 \(p. 705\)](#) 단원을 참조하십시오.

Oracle DB 인스턴스의 경우 CPU 코어 및 코어당 스레드 수 구성은 BYOL(기존 보유 라이선스 사용) 라이선스 옵션에서만 지원됩니다. Oracle 라이선스 옵션에 대한 자세한 내용은 [Oracle 라이선싱 \(p. 704\)](#) 단원을 참조하십시오.

AWS CloudTrail을 사용해 Oracle DB 인스턴스용 Amazon RDS의 프로세스 구성 관련 변경 사항에 대한 모니터링 및 감사를 수행할 수 있습니다. CloudTrail 사용에 관한 자세한 내용은 [AWS CloudTrail을 사용하여 Amazon RDS API 호출 로깅 \(p. 335\)](#) 단원을 참조하십시오.

DB 인스턴스 클래스의 CPU 코어 및 CPU 코어당 스레드 설정

다음 작업을 수행할 때 DB 인스턴스 클래스의 CPU 코어 및 코어당 스레드 수를 구성할 수 있습니다.

- [Amazon RDS DB 인스턴스 생성 \(p. 111\)](#)
- [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#)
- [DB 스냅샷에서 복원 \(p. 205\)](#)
- [DB 인스턴스를 지정된 시간으로 복원 \(p. 221\)](#)

Note

CPU 코어 또는 코어당 스레드 수를 구성하기 위해 DB 인스턴스를 수정할 때 DB 인스턴스가 잠시 종단됩니다.

AWS Management 콘솔, AWS CLI 또는 RDS API를 사용하여 DB 인스턴스 클래스의 CPU 코어 및 CPU 코어당 스레드를 설정할 수 있습니다.

AWS Management 콘솔

DB 인스턴스를 생성, 수정 또는 복원할 때 AWS Management 콘솔에서 DB 인스턴스 클래스를 설정합니다. 인스턴스 사양 단원에는 프로세서에 대한 옵션이 표시됩니다. 다음 이미지는 프로세서 기능 옵션을 보여줍니다.

Modify DB Instance: mytestdb

Instance specifications

License model

License type associated with the database engine

bring-your-own-license

DB engine version

Version number of the database engine to be used for this instance.

Oracle 12.1.0.2.v12 (default)

DB instance class

Contains the compute and memory capacity of the DB instance.

db.r4.large — 2 vCPU, 15.25 GiB RAM

Processor features

Core count [Info](#)

1

Threads per core [Info](#)

2

Use default processor settings for the selected instance class

Total vCPU enabled - 2

Multi-AZ deployment

Specifies if the DB instance should have a standby deployed in another availability zone.

Processor features(프로세서 기능)에서 다음 옵션을 DB 인스턴스 클래스에 적절한 값으로 설정합니다.

- Core count(코어 수) – 이 옵션을 사용하여 CPU 코어 수를 설정합니다. 이 값은 DB 인스턴스의 최대 CPU 코어 수보다 작거나 같아야 합니다.
- Threads per core(코어당 스레드) – 코어당 여러 스레드를 활성화하려면 2를 지정하거나, 코어다 여러 스레드를 비활성화하려면 1을 설정합니다.

DB 인스턴스를 수정하거나 복원할 때 CPU 코어 및 CPU 코어당 스레드를 인스턴스 클래스의 기본값으로 설정할 수도 있습니다.

콘솔에서 DB 인스턴스에 대한 세부 정보를 볼 때 구성 탭에서 DB 인스턴스 클래스에 대한 프로세서 정보를 볼 수 있습니다. 다음 이미지는 CPU 코어 하나 및 코어당 여러 스레드가 있는 DB 인스턴스 클래스를 보여 줍니다.

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

Oracle DB 인스턴스의 경우 BYOL(기존 보유 라이선스 사용) DB 인스턴스에 대한 프로세서 정보만 나타납니다.

CLI

다음 AWS CLI 명령 중 하나를 실행할 때 DB 인스턴스의 프로세서 기능을 설정할 수 있습니다.

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

AWS CLI를 사용하여 DB 인스턴스에 대한 DB 인스턴스 클래스의 프로세서를 구성하려면 `--processor-features` 옵션을 명령에 포함시킵니다. `coreCount` 기능 이름을 사용하여 CPU 코어 수를 지정하고, `threadsPerCore` 기능 이름을 사용하여 코어당 여러 스레드가 활성화되는지 여부를 지정합니다.

옵션에는 다음과 같은 구문이 있습니다.

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

다음은 프로세서를 구성하는 예제입니다.

예제

- [DB 인스턴스의 CPU 코어 수 설정 \(p. 93\)](#)
- [DB 인스턴스의 CPU 코어 수 설정 및 여러 스레드 비활성화 \(p. 93\)](#)
- [DB 인스턴스 클래스에 유효한 프로세서 값 보기 \(p. 94\)](#)
- [DB 인스턴스의 기본 프로세서 설정으로 돌아가기 \(p. 95\)](#)
- [DB 인스턴스의 기본 CPU 코어 수로 돌아가기 \(p. 95\)](#)
- [DB 인스턴스의 코어당 기본 스레드 수로 돌아가기 \(p. 95\)](#)

DB 인스턴스의 CPU 코어 수 설정

Example

다음 예에서는 CPU 코어 수를 4로 설정하여 `mydbinstance`를 수정합니다. `--apply-immediately`를 사용하면 변경 내용이 즉시 적용됩니다. 다음 예약 유지 관리 기간 중에 변경 내용을 적용하려는 경우 `--apply-immediately` 옵션을 생략합니다.

Linux, OS X, Unix의 경우:

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

Windows의 경우:

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

DB 인스턴스의 CPU 코어 수 설정 및 여러 스레드 비활성화

Example

다음 예제에서는 CPU 코어 수를 4로 설정하고 코어당 여러 스레드를 비활성화하여 `mydbinstance`를 수정합니다. `--apply-immediately`를 사용하면 변경 내용이 즉시 적용됩니다. 다음 예약 유지 관리 기간 중에 변경 내용을 적용하려는 경우 `--apply-immediately` 옵션을 생략합니다.

Linux, OS X, Unix의 경우:

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

Windows의 경우:

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

DB 인스턴스 클래스에 유효한 프로세서 값 보기

Example

`describe-orderable-db-instance-options` 명령을 실행하고 `--db-instance-class` 옵션의 인스턴스 클래스를 지정하여 특정 DB 인스턴스에 유효한 프로세서 값을 볼 수 있습니다. 예를 들어, 다음 명령의 출력은 db.r3.large 인스턴스 클래스의 프로세서 옵션을 보여 줍니다.

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

다음은 JSON 형식 명령의 샘플 출력입니다.

```
{
    "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
}
```

```
}
```

또한 DB 인스턴스 클래스 프로세서 정보에 대해 다음 명령을 실행할 수 있습니다.

- **describe-db-instances** – 지정된 DB 인스턴스에 대한 프로세서 정보를 보여 줍니다.
- **describe-db-snapshots** – 지정된 DB 스냅샷에 대한 프로세서 정보를 보여 줍니다.
- **describe-valid-db-instance-modifications** – 지정된 DB 인스턴스에 유효한 프로세서 수정 사항을 보여 줍니다.

DB 인스턴스의 기본 프로세서 설정으로 돌아가기

Example

다음 예제에서는 DB 인스턴스 클래스를 기본 프로세서 값으로 되돌려서 mydbinstance를 수정합니다. **--apply-immediately**를 사용하면 변경 내용이 즉시 적용됩니다. 다음 예약 유지 기간 중에 변경 내용을 적용하려는 경우 **--apply-immediately** 옵션을 생략합니다.

Linux, OS X, Unix의 경우:

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

Windows의 경우:

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

DB 인스턴스의 기본 CPU 코어 수로 돌아가기

Example

다음 예제에서는 DB 인스턴스 클래스를 기본 CPU 코어 수로 되돌려서 mydbinstance를 수정합니다. 코어당 스레드 설정은 변경되지 않습니다. **--apply-immediately**를 사용하면 변경 내용이 즉시 적용됩니다. 다음 예약 유지 기간 중에 변경 내용을 적용하려는 경우 **--apply-immediately** 옵션을 생략합니다.

Linux, OS X, Unix의 경우:

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

Windows의 경우:

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

DB 인스턴스의 코어당 기본 스레드 수로 돌아가기

Example

다음 예제에서는 DB 인스턴스 클래스를 코어당 기본 스레드 수로 되돌려서 mydbinstance를 수정합니다. CPU 코어 수 설정은 변경되지 않습니다. **--apply-immediately**를 사용하면 변경 내용이 즉시 적용됩니다. 다음 예약 유지 기간 중에 변경 내용을 적용하려는 경우 **--apply-immediately** 옵션을 생략합니다.

Linux, OS X, Unix의 경우:

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

Windows의 경우:

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

API

다음 Amazon RDS API 작업 중 하나를 호출할 때 DB 인스턴스의 프로세서 기능을 설정할 수 있습니다.

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

Amazon RDS API를 사용하여 DB 인스턴스에 대한 DB 인스턴스 클래스의 프로세서 기능을 구성하려면 `ProcessFeatures` 파라미터를 호출에 포함시킵니다.

파라미터의 구문은 다음과 같습니다.

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

`coreCount` 기능 이름을 사용하여 CPU 코어 수를 지정하고, `threadsPerCore` 기능 이름을 사용하여 코어 당 여러 스레드가 활성화되는지 여부를 지정합니다.

[DescribeOrderableDBInstancesOptions](#) 작업을 실행하고 `DBInstanceClass` 파라미터의 인스턴스 클래스를 지정하여 특정 DB 인스턴스에 유효한 프로세서 값을 볼 수 있습니다.

또한 DB 인스턴스 클래스 프로세서 정보에 대해 다음 작업을 사용할 수 있습니다.

- [DescribeDBInstances](#) – 지정된 DB 인스턴스에 대한 프로세서 정보를 보여 줍니다.
- [DescribeDBSnapshots](#) – 지정된 DB 스냅샷에 대한 프로세서 정보를 보여 줍니다.
- [DescribeValidDBInstanceModifications](#) – 지정된 DB 인스턴스에 유효한 프로세서 수정 사항을 보여 줍니다.

DB 인스턴스 상태

DB 인스턴스의 상태는 DB 인스턴스의 상태를 나타냅니다. Amazon RDS 콘솔, AWS CLI 명령 `describe-db-instances` 또는 [DescribeDBInstances](#) API 작업을 사용하여 DB 인스턴스 상태를 확인할 수 있습니다.

Note

또한 Amazon RDS는 유지 관리 상태라는 상태를 한 가지 더 사용하며, 이는 Amazon RDS 콘솔의 유지 관리 열에 표시됩니다. 이 값은 DB 인스턴스에 적용해야 하는 모든 유지 관리 패치 상태를 나타냅니다. 유지 관리 상태는 DB 인스턴스 상태와 무관합니다. 유지 관리 상태에 대한 자세한 내용은 [DB 인스턴스의 업데이트 적용 \(p. 114\)](#) 단원을 참조하십시오.

다음 표에서 DB 인스턴스에 가능한 상태 값을 찾으십시오. 각 상태별 요금도 표시됩니다. DB 인스턴스와 스토리지에만 요금이 청구되는지, 스토리지에만 청구되는지, 혹은 청구되지 않는지 표시됩니다. 모든 DB 인스턴스 상태의 백업 사용에는 항상 청구됩니다.

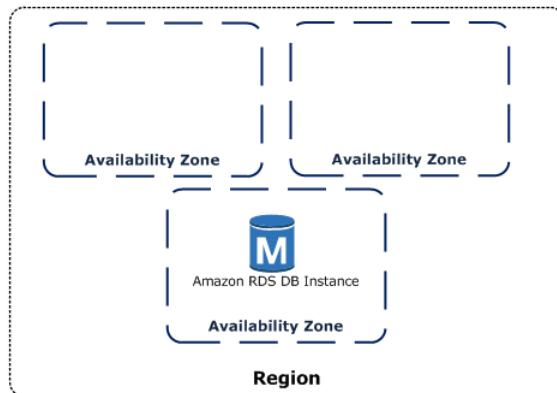
DB 인스턴스 상태	청구	설명
사용 가능	청구	DB 인스턴스에 문제가 없으며 사용할 수 있습니다.
백업 중	청구	현재 DB 인스턴스를 백업 중입니다.
역추적	청구	현재 DB 인스턴스를 역추적 중입니다. 이 상태는 Aurora MySQL에만 적용됩니다.
Enhanced Monitoring 구성 중	청구	이 DB 인스턴스에 대해 확장 모니터링이 활성화 또는 비활성화 상태입니다.
configuring-iam-database-auth	청구	이 DB 인스턴스에 대해 AWS Identity and Access Management(IAM) 데이터베이스 인증이 활성화 또는 비활성화 상태입니다.
configuring-log-exports	청구	Amazon CloudWatch Logs에 로그 파일을 게시하는 기능이 이 DB 인스턴스에 대해 활성화 또는 비활성화 상태입니다.
converting-to-vpc	청구	DB 인스턴스를 Amazon Virtual Private Cloud(Amazon VPC)에 없는 DB 인스턴스에서 Amazon VPC에 있는 DB 인스턴스로 변환 중입니다.
creating	미청구	DB 인스턴스를 생성 중입니다. 생성 중인 DB 인스턴스에는 액세스할 수 없습니다.
deleting	미청구	DB 인스턴스를 삭제 중입니다.
실패	미청구	DB 인스턴스 오류가 발생하여 Amazon RDS로 복구할 수 없습니다. DB 인스턴스의 복원 가능한 가장 최근 시간을 기준으로 특정 시점으로 복원을 수행하여 데이터를 복구합니다.
inaccessible-encryption-credentials	미청구	DB 인스턴스를 암호화 또는 해독하는데 사용되는 AWS KMS 키는 액세스할 수 없습니다.
네트워크 호환 장애	미청구	Amazon RDS가 DB 인스턴스에 대한 복구 작업을 시도했으나 VPC의 상태로 인하여 작업을 완료할 수 없어 복구하지 못했습니다. 예를 들어 서브넷에 사용 가능한 IP 주소가 모두 사용 중이어서 Amazon RDS가 DB 인스턴스에 대한 IP 주소를 받을 수 없는 경우 이러한 상태가 발생할 수 있습니다.
옵션 그룹 호환 장애	청구	Amazon RDS 옵션 그룹 변경을 적용하려고 시도했으나 적용하지 못했습니다. 또한 Amazon RDS는 이전 옵션 그룹 상태를 롤백하지 못했습니다. 자세한 내용은 DB 인스턴스에 대한 최근 이벤트 목록을 참조하십시오. 예를 들어, 옵션 그룹은 TDE 등의 옵션을 포함하는데 DB 인스턴스는 암호화된 정보를 포함하지 않는 경우 이러한 상태가 발생할 수 있습니다.
파라미터 호환 장애	청구	Amazon RDS가 DB 인스턴스의 DB 파라미터 그룹에서 지정된 파라미터가 DB 인스턴스와 호환되지 않아 DB 인스턴스를 시작할 수 없습니다. DB 인스턴스에 대한 액세스 권한을 다시 얻으려면 변경된 파라미터를 되돌리거나 DB 인스턴스와 호환되는 파라미터를 정의해야 합니다. 호환되지 않는 파라미터에 대한 자세한 내용은 DB 인스턴스에 대한 최근 이벤트 목록을 참조하십시오.

DB 인스턴스 상태	청구	설명
복원 호환 장애	미청구	Amazon RDS가 특정 시점으로 복원을 수행할 수 없습니다. 임시 테이블 사용, MySQL에서 MyISAM 테이블 사용 또는 MariaDB에서 Aria 테이블 사용은 이러한 상태가 발생하는 일반적 원인 중 하나입니다.
유지 관리	청구	Amazon RDS는 DB 인스턴스에 유지 관리 업데이트를 적용합니다. 이 상태는 RDS가 한참 전에 미리 예약하는 인스턴스 수준의 유지 관리에 사용됩니다.
수정 중	청구	고객의 DB 인스턴스 수정 요청으로 인해 DB 인스턴스를 수정 중입니다.
moving-to-vpc	청구	DB 인스턴스가 새 Amazon Virtual Private Cloud(Amazon VPC)로 이동하는 중입니다.
재부팅 중	청구	고객의 요청 또는 DB 인스턴스 재부팅이 필요한 Amazon RDS 프로세스로 인해 DB 인스턴스를 재부팅 중입니다.
이름을 바꾸는 중	청구	고객의 이름 바꾸기 요청으로 인해 DB 인스턴스 이름을 바꾸는 중입니다.
마스터 자격 증명 재설정 중	청구	고객의 재설정 요청으로 인해 DB 인스턴스 사용자 자격 증명을 재설정하는 중입니다.
restore-error	청구	특정 시점으로 복원을 시도하거나 스냅샷에서 복원을 시도할 때 DB 인스턴스에서 오류가 발생했습니다.
시작 중	요금 부과 스토리지	DB 인스턴스를 시작하는 중입니다.
중단됨	요금 부과 스토리지	DB 인스턴스가 종지되었습니다.
종지 중	요금 부과 스토리지	DB 인스턴스를 종지 중입니다.
스토리지가 꽉 참	청구	DB 인스턴스에 할당된 스토리지 용량이 거의 다 찼습니다. 이것은 중요한 상태이므로 즉각 이 문제를 수정하는 것이 좋습니다. 그러면 DB 인스턴스를 수정하여 스토리지를 확장하십시오. 이러한 상황을 피하려면 스토리지 공간이 부족해지면 경고하도록 Amazon CloudWatch 경보를 설정하십시오.
스토리지 최적화	청구	DB 인스턴스가 스토리지 크기 또는 유형을 변경하도록 수정 중입니다. DB 인스턴스가 완전히 작동 중입니다. 그러나 DB 인스턴스가 스토리지 최적화 상태에 있는 동안 DB 인스턴스의 스토리지에 대한 변경을 요청할 수 없습니다. 스토리지 최적화 프로세스는 보통 짧지만, 가끔 최대 24시간까지 걸릴 수도 있습니다.
upgrading	청구	데이터베이스 엔진 버전은 현재 업그레이드 중입니다.

리전 및 가용 영역

Amazon 클라우드 컴퓨팅 리소스는 세계 각지의 여러 곳에서 호스팅됩니다. 이 위치들은 AWS 리전과 가용 영역으로 구성됩니다. 각 AWS 리전은 개별 지리 영역입니다. 각 AWS 리전에는 가용 영역이라는 여러 개의 격리된 위치가 있습니다. Amazon RDS는 여러 위치에 인스턴스와 같은 리소스와 데이터를 배치할 수 있는 기능을 제공합니다. 리소스는 특별하게 이를 지정하지 않을 경우 AWS 리전에 복제되지 않습니다.

Amazon은 최신 기술을 탑재한 고가용성 데이터 센터를 운영하고 있습니다. 드물기는 하지만 동일한 위치에 있는 인스턴스의 가용성에 영향을 미치는 장애가 발생할 수도 있습니다. 그런 장애의 영향을 받는 단일한 위치에서 모든 인스턴스를 호스팅하는 경우에는 모든 인스턴스가 사용이 불가능해질 수 있습니다.



각 AWS 리전이 서로 완전히 독립적이라는 점에 유의하십시오. 사용자의 모든 Amazon RDS 활동(데이터베이스 인스턴스 또는 사용할 수 있는 데이터베이스 인스턴스 목록 생성 등)은 현재 기본 AWS 리전에서만 실행됩니다. 기본 AWS 리전은 EC2_REGION 환경 변수를 설정하여 콘솔에서 변경하거나 AWS Command Line Interface에서 --region 파라미터를 사용해 재정의할 수 있습니다. 자세한 내용은 [AWS Command Line Interface 구성](#), 특히 환경 변수 및 명령줄 옵션을 참조하십시오.

Amazon RDS는 미국 정부 기관 및 고객이 더욱 민감한 워크로드를 클라우드로 이전할 수 있도록 설계된 AWS GovCloud (US-West)라는 특별한 AWS 리전을 지원합니다. AWS GovCloud (US-West)은 미국의 특정 규제 및 규정 준수 요구를 처리합니다. AWS GovCloud (US-West)에 대한 자세한 내용은 [AWS GovCloud \(US-West\)란?](#)을 참조하십시오.

특정 AWS 리전에서 Amazon RDS DB 인스턴스를 생성하거나 사용하려면 해당 리전 서비스 엔드포인트를 사용하십시오.

Amazon RDS는 다음의 표에 나와 있는 엔드포인트를 지원합니다.

Region Name	Region	Endpoint	Protocol
미국 동부 (오하이오)	us-east-2	rds.us-east-2.amazonaws.com	HTTPS
미국 동부 (버지니아 북부)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS
미국 서부 (캘리포니아 북부)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
미국 서부 (오레곤)	us-west-2	rds.us-west-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol	
아시아 태평양(홍콩)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS	
아시아 태평양(뭄바이)	ap-south-1	rds.ap-south-1.amazonaws.com	HTTPS	
아시아 태평양(오사카-로컬)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com	HTTPS	
아시아 태평양(서울)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com	HTTPS	
아시아 태평양(싱가포르)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com	HTTPS	
아시아 태평양(시드니)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com	HTTPS	
아시아 태평양(도쿄)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS	
캐나다(중부)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS	
중국(베이징)	cn-north-1	rds.cn-north-1.amazonaws.com.cn	HTTPS	
중국(닝샤)	cn-northwest-1	rds.cn-northwest-1.amazonaws.com.cn	HTTPS	
EU(프랑크푸르트)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS	
EU(아일랜드)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS	
EU(런던)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS	
EU(파리)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS	
EU(스톡홀름)	eu-north-1	rds.eu-north-1.amazonaws.com	HTTPS	
남아메리카(상파울루)	sa-east-1	rds.sa-east-1.amazonaws.com	HTTPS	
AWS GovCloud(미국 동부)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com	HTTPS	
AWS GovCloud(US)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com	HTTPS	

엔드포인트를 명시적으로 지정하지 않는 경우 기본 엔드포인트는 미국 서부(오레곤)입니다.

Amazon RDS DB 인스턴스 스토리지

MySQL, MariaDB, PostgreSQL, Oracle, 및 Microsoft SQL Server용 Amazon RDS의 DB 인스턴스는 데이터베이스 및 로그 스토리지의 크기 때문에 Amazon Elastic Block Store(Amazon EBS)를 사용합니다. Amazon RDS는 필요한 스토리지 용량에 따라 자동으로 데이터를 여러 Amazon EBS 볼륨에 나누어 저장하여 성능을 강화합니다.

Amazon RDS 스토리지 유형

Amazon RDS에서는 범용 SSD(gp2라고도 함), 프로비저닝된 IOPS SSD(io1이라고도 함), 마그네틱 등 세 가지 스토리지 유형을 제공합니다. 이러한 3가지 유형은 성능 특성과 가격이 다르므로 데이터베이스 워크로드 요건에 따라 스토리지 성능과 비용을 조정할 수 있습니다. 이제 MySQL, MariaDB, PostgreSQL RDS DB 인스턴스는 스토리지의 최대 64TiB까지 생성할 수 있습니다. RDS DB 인스턴스는 스토리지의 최대 64 TiB까지 생성할 수 있습니다. SQL Server RDS DB 인스턴스는 스토리지의 최대 16TiB까지 생성할 수 있습니다. 이 스토리지 크기 때문에 프로비저닝된 IOPS SSD 및 범용 SSD 스토리지 유형을 사용합니다.

다음은 세 가지 스토리지 유형에 대한 간략한 설명입니다.

- 범용 SSD – gp2라고도 하는 범용 SSD 볼륨은 광범위한 워크로드에 이상적인 비용 효율적 스토리지를 제공합니다. 이러한 볼륨은 시간을 연장할 경우 3,000IOPS의 버스트 기능까지 지원되어 지연 시간이 한 자릿수 밀리초에 불과합니다. 이러한 볼륨에 대한 기준 성능은 볼륨의 크기에 의해 결정됩니다.

스토리지 크기 범위를 포함하여 범용 SSD 스토리지에 대한 자세한 내용은 [범용 SSD 스토리지 \(p. 101\)](#) 단원을 참조하십시오.

- 프로비저닝된 IOPS – 프로비저닝된 IOPS 스토리지는 I/O 지연 시간이 짧고 I/O 처리량이 일정한 I/O 집약적 워크로드, 특히 데이터베이스 워크로드 요구 사항을 충족하도록 설계되었습니다.

스토리지 크기 범위를 포함하여 프로비저닝된 IOPS 스토리지에 대한 자세한 내용은 [프로비저닝된 IOPS SSD 스토리지 \(p. 103\)](#) 단원을 참조하십시오.

- 마그네틱 – Amazon RDS는 역호환성을 위해 마그네틱 스토리지도 지원합니다. 새 스토리지가 필요할 경우 범용 SSD 또는 프로비저닝된 IOPS를 사용하는 것이 좋습니다. 마그네틱 스토리지에서는 DB 인스턴스에 허용되는 최대 스토리지 크기가 나머지 스토리지 유형의 크기보다 작습니다. 자세한 내용은 [마그네틱 스토리지 \(p. 104\)](#) 단원을 참조하십시오.

Amazon EBS 볼륨은 인스턴스 구성, I/O 특성, 워크로드 수요 등 성능에 영향을 끼치는 몇 가지 요인이 있습니다. 프로비저닝된 IOPS 볼륨을 최대한 활용할 수 있는 자세한 내용은 [Amazon EBS 볼륨 성능](#)을 참조하십시오.

범용 SSD 스토리지

범용 SSD 스토리지는 비용 효율적이어서 대부분의 데이터베이스 워크로드에 적합합니다. 다음은 범용 SSD DB 인스턴스의 스토리지 크기 범위입니다.

- MariaDB, MySQL 및 PostgreSQL 데이터베이스 인스턴스: 20GiB–64TiB
- Enterprise, Standard, Web 및 Express 에디션용 SQL Server: 20GiB–16TiB
- Oracle 인스턴스: 20GiB- 64 TiB

범용 SSD 스토리지의 기본 I/O 성능은 1GiB당 3IOPS입니다. 이 관계는 볼륨이 클수록 성능이 높아진다는 의미입니다. 예를 들어, 100GiB 볼륨에 대한 기준 성능은 300IOPS입니다. 1TiB 볼륨에 대한 기본 성능은 GB 당 3,000IOPS입니다. 그리고 5.34TiB 볼륨에 대한 기본 성능은 16,000IOPS입니다.

1TiB보다 작은 크기는 늘어난 시간에 대해 3,000IOPS로 확장됩니다. 버스트는 1TiB 초과 볼륨에는 적합하지 않습니다. 버스트 성능은 인스턴스 I/O 크레딧 밸런스에 의해 결정됩니다. 인스턴스 I/O 크레딧에 대한 자세한 내용은 [I/O 크레딧 및 버스트 성능 \(p. 102\)](#) 단원을 참조하십시오.

많은 워크로드는 버스트 밸런스를 절대 고갈시키지 않아 범용 SSD가 많은 워크로드의 이상적인 스토리지가 될 수 있습니다. 하지만, 일부 워크로드가 3,000IOPS 버스트 스토리지 크레딧 밸런스를 소비할 수 있으므로 워크로드의 요건에 맞게 스토리지 용량을 계획해야 합니다.

I/O 크레딧 및 버스트 성능

범용 SSD 스토리지 성능은 볼륨 크기에 따라 결정됩니다. 그 이유는 볼륨 크기가 볼륨의 기본 성능과 I/O 크레딧의 누적 속도에 영향을 미치기 때문입니다. 즉, 볼륨이 클수록 기본 성능이 높아지고 I/O 크레딧의 누적 속도도 빨라집니다. I/O 크레딧이란 범용 SSD 스토리지에서 기본 성능 이상이 필요할 때 대용량 I/O를 버스트하는 데 사용할 수 있는 가용 대역폭을 의미합니다. 스토리지에 I/O에 대한 I/O 크레딧이 많을수록 워크로드가 더 높은 성능이 필요할 때 기본 성능 이상으로 버스트하는 시간이 길어질 뿐만 아니라 성능도 좋아집니다.

범용 SSD 스토리지를 사용할 경우, DB 인스턴스에서 초기 I/O 크레딧 밸런스는 540만 I/O 크레딧과 동일합니다. 이 초기 I/O 크레딧 잔고는 30분 동안 3,000 IOPS의 버스트 성능을 유지할 수 있는 수준입니다. 이 밸런스는 부트 볼륨에 빠른 초기 부팅 주기를 제공하고 기타 애플리케이션에 좋은 부트스트래핑 환경을 제공하도록 설계되었습니다. 볼륨은 볼륨 크기의 GiB 당 3 IOPS의 기준 성능 비율로 I/O 크레딧을 획득합니다. 예를 들어, 100-GiB SSD 볼륨은 300IOPS의 기준 성능을 갖습니다.

스토리지에 기본 성능 I/O 이상이 필요한 경우 I/O 크레딧 밸런스에서 I/O 크레딧을 사용하여 필요한 성능을 버스트할 수 있습니다. 버스트는 최대 3,000IOPS까지 올라갑니다. 1,000GiB 이상의 스토리지는 최대 버스트 성능 이상의 기준 성능을 갖습니다. 스토리지가 초당 획득한 I/O 크레딧 이하를 사용하는 경우 미사용 I/O 크레딧은 I/O 크레딧 밸런스에 가산됩니다. 범용 SSD 스토리지를 사용하는 DB 인스턴스에서 최대 I/O 크레딧 밸런스는 초기 I/O 크레딧 밸런스(540만 I/O 크레딧)와 동일합니다.

스토리지가 I/O 크레딧 밸런스를 모두 사용한다고 가정하십시오. 만약 그렇다면, I/O 수요가 기본 성능 밑으로 떨어져서 미사용 I/O 크레딧이 I/O 크레딧 밸런스에 가산될 때까지 최대 성능은 기본 성능과 동일한 수준을 유지합니다. (여기에서 기본 성능 수준이란 스토리지가 I/O 크레딧을 획득하는 속도를 말합니다.) 스토리지가 클수록 기본 성능이 높아지고 I/O 크레딧 밸런스의 보충 속도도 빨라집니다.

Note

마그네티к 스토리지와 범용 SSD 스토리지를 서로 전환할 때는 잠재적으로 I/O 크레딧 밸런스가 고갈될 수도 있어 전환 시간이 더 오래 걸릴 수 있습니다. 스토리지 조정에 대한 자세한 내용은 [Amazon RDS용 스토리지 DB 인스턴스 작업 \(p. 102\)](#) 단원을 참조하십시오.

다음 표에는 몇 가지 스토리지 크기가 나와 있습니다. 각 스토리지 크기마다 스토리지의 기본 성능, 즉 I/O 크레딧이 누적되는 속도가 표시되어 있습니다. 또한 I/O 크레딧 밸런스가 가득 찬 상태에서 시작할 경우 최대 3,000IOPS일 때 버스트 지속 시간도 나열되어 있습니다. 그 밖에 비어있는 I/O 크레딧 밸런스를 스토리지가 다시 채우는 데 걸리는 시간(초)도 확인할 수 있습니다.

Note

IOPS 수치는 5,334GiB의 볼륨 스토리지 크기에서 최대값에 도달합니다.

스토리지 크기(GiB)	기본 성능(IOPS)	3,000IOPS(초)에서 최대 버스트 지속 시간	고갈된 I/O 크레딧 밸런스를 보충하는 데 소요되는 시간(초)
1	100	1,862	54,000
100	300	2,000	18,000
250	750	2,400	7,200

스토리지 크기(GiB)	기본 성능(IOPS)	3,000IOPS(초)에서 최대 버스트 지속 시간	고갈된 I/O 크레딧 밸런스를 보충하는 데 소요되는 시간(초)
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	무제한	해당 사항 없음
5,334	16,000	해당 사항 없음	해당 사항 없음

스토리지의 버스트 지속 시간은 스토리지 크기, 필요한 버스트 IOPS 및 버스트 실행 시 I/O 크레딧 밸런스에 따라 달라집니다. 이 관계는 다음 방정식과 같습니다.

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

I/O 크레딧 밸런스의 고갈로 인해 스토리지 성능이 종종 기준 수준으로 제한됩니다. 그렇다면, 더 높은 기준 성능 수준의 범용 SSD 추가 할당을 고려하십시오. 또는 IOPS 성능을 유지해야 하는 워크로드인 경우에는 Provisioned IOPS 스토리지로 전환하는 것도 좋습니다.

I/O 성능이 꾸준하게 요구되는 워크로드인 경우 100GiB 미만의 범용 SSD 스토리지를 프로비저닝하면 I/O 크레딧 밸런스가 고갈되어 오히려 지연 시간이 늘어날 수 있습니다.

Note

일반적으로 대부분의 워크로드는 I/O 크레딧 밸런스를 초과하지 않습니다.

기준 성능과 I/O 크레딧 밸런스가 성능에 어떻게 영향을 주는지에 대한 자세한 내용은 [Amazon RDS 및 GP2에서 버스트와 기준 성능 비교 이해](#)를 참조하십시오.

프로비저닝된 IOPS SSD 스토리지

빠르고 일관적인 I/O 성능이 필요한 프로덕션 애플리케이션의 경우에는 프로비저닝된 IOPS(초당 입/출력 연산 수) 스토리지를 권장합니다. 프로비저닝된 IOPS 스토리지는 성능이 예측 가능하며, 일관적으로 지연 시간이 짧은 스토리지 유형입니다. 프로비저닝된 IOPS 스토리지는 일관적인 성능이 필요한 온라인 트랜잭션 프로세싱(OLTP) 워크로드에 이상적일 뿐만 아니라 프로비저닝된 IOPS는 이런 워크로드의 성능 투영에도 효과적입니다.

DB 인스턴스를 생성할 때 IOPS 속도와 볼륨 크기를 지정합니다. Amazon RDS는 사용자가 변경할 때까지 DB 인스턴스의 IOPS 속도를 제공합니다.

Note

데이터베이스 워크로드는 프로비저닝한 100% IOPS를 달성할 수 없을 수 있습니다.

다음 표는 각 데이터베이스 엔진에 따른 프로비저닝된 IOPS 범위 및 스토리지 크기 범위를 나타냅니다.

데이터베이스 엔진	프로비저닝된 IOPS 범위	스토리지 범위
MariaDB	1,000–80,000IOPS	100GiB–64TiB
SQL Server, Enterprise 및 Standard Edition	1000–64,000IOPS*	20GiB–16TiB

데이터베이스 엔진	프로비저닝된 IOPS 범위	스토리지 범위
SQL Server, Web 및 Express Edition	1000–64,000IOPS*	100GiB–16TiB
MySQL	1,000–80,000IOPS	100GiB–64TiB
Oracle	1,000–80,000IOPS	100GiB– 64 TiB
PostgreSQL	1,000–80,000IOPS	100GiB–64TiB

* 최대 IOPS 64,000은 m5 인스턴스 유형의 Nitro 기반 인스턴스에서만 보장됩니다. 다른 인스턴스 패밀리는 최대 32,000 IOPS의 성능을 보장합니다.

프로비저닝된 IOPS 스토리지를 다중 AZ 배포 또는 읽기 전용 복제본과 결합합니다.

프로덕션 OLTP 사용 사례에서 내결함성을 높이는 것이 목적일 때는 다중 AZ 배포를, 빠르고 예측 가능한 성능이 목적일 때는 프로비저닝된 IOPS 스토리지를 사용하는 것이 바람직합니다.

MySQL, MariaDB 또는 PostgreSQL 읽기 전용 복제본과 함께 프로비저닝된 IOPS SSD 스토리지를 사용합니다. 읽기 전용 복제본의 스토리지 유형은 마스터 DB 인스턴스의 스토리지와 독립되어 있습니다. 예를 들어, 프로비저닝된 IOPS SSD 스토리지를 사용하는 마스터 DB 인스턴스에 읽기 전용 복제본의 범용 SSD를 사용하여 비용을 절감할 수 있습니다. 그러나 이 경우의 읽기 전용 복제본 성능은 프로비저닝된 IOPS SSD 스토리지를 사용하는 마스터 DB 인스턴스와 읽기 전용 복제본 구성 성능과 차이를 보일 수 있습니다.

프로비저닝된 IOPS 스토리지 비용

프로비저닝된 IOPS 스토리지는 한 달간 사용 여부에 상관없이 프로비저닝된 리소스에 대한 요금이 청구됩니다.

요금에 대한 자세한 내용은 [Amazon RDS 요금](#)을 참조하십시오.

Amazon RDS 프로비저닝된 IOPS SSD 스토리지 최대한 활용하기

워크로드가 I/O 제한되면 프로비저닝된 IOPS SSD 스토리지의 사용에 따라 시스템이 동시에 처리할 수 있는 I/O 요청 수가 증가합니다. 이처럼 동시 처리 가능한 수가 증가하면 대기열의 I/O 요청 시간이 줄어들어 지연 시간이 감소하는 효과가 있습니다. 지연 시간 감소는 데이터베이스 커밋 속도의 향상으로 이어져 응답 시간이 개선되고 데이터베이스 처리 속도가 빨라집니다.

프로비저닝된 IOPS SSD 스토리지는 IOPS를 지정하여 I/O 용량을 예비할 수 있습니다. 하지만, 다른 시스템 용량 속성과 마찬가지로 재하 시 최대 처리 속도는 처음 사용된 리소스에 따라 제약을 받게 됩니다. 이러한 리소스로는 네트워크 대역폭, CPU 메모리 또는 데이터베이스 내부 리소스가 있습니다.

마그네틱 스토리지

또한 Amazon RDS는 이전 버전과의 호환성을 위해 마그네틱 스토리지를 지원합니다. 새 스토리지가 필요할 경우 범용 SSD 또는 프로비저닝된 IOPS SSD를 사용하는 것이 좋습니다. 다음은 마그네틱 스토리지의 몇 가지 제한입니다.

- SQL 서버 데이터베이스 엔진을 사용할 경우 스토리지를 확장할 수 없습니다.
- 탄력적 볼륨을 지원하지 않습니다.
- 최대 4TiB 크기로 제한됩니다.

- 최대 1,000IOPS로 제한됩니다.

스토리지 성능 모니터링

Amazon RDS는 DB 인스턴스의 성능을 측정할 수 있는 몇 가지 측정치가 있습니다. Amazon RDS 관리 콘솔의 인스턴스에 대한 요약 페이지에서 지표를 볼 수 있습니다. 또한 Amazon CloudWatch를 사용하여 이 측정치를 모니터링할 수도 있습니다. 자세한 내용은 [DB 인스턴스 측정치 보기 \(p. 236\)](#) 단원을 참조하십시오. 향상된 모니터링은 더 자세한 I/O 측정치를 제공합니다. 자세한 내용은 [확장 모니터링 \(p. 238\)](#) 단원을 참조하십시오.

다음 지표는 DB 인스턴스용 스토리지를 모니터링하는 데 유용합니다.

- IOPS – 완료한 초당 I/O 작업 횟수입니다. 이 지표는 지정된 시간 간격 동안의 평균 IOPS로 보고됩니다. Amazon RDS는 1분 간격으로 읽기 IOPS와 쓰기 IOPS를 따로 보고합니다. 총 IOPS는 읽기 IOPS와 쓰기 IOPS의 합계입니다. 일반적인 IOPS 값 범위는 초당 0에서 수만에 이릅니다.
- Latency(지연 시간) – I/O 요청을 제출하여 완료될 때까지 걸리는 시간입니다. 이 지표는 지정된 시간 간격 동안의 평균 지연 시간으로 보고됩니다. Amazon RDS는 초 단위를 사용하여 1분 간격으로 읽기 IOPS와 쓰기 IOPS를 따로 보고합니다. 일반적인 지연 시간 값은 밀리초(ms) 단위로 표시됩니다. 예를 들어 0.002초의 경우 Amazon RDS는 2ms를 보고합니다.
- Throughput(처리량) – 디스크와 송/수신하는 초당 바이트 수입니다. 이 지표는 지정된 시간 간격 동안의 평균 처리량으로 보고됩니다. Amazon RDS는 초당 메가바이트(MB/s) 단위를 사용하여 1분 간격으로 읽기 처리량과 쓰기 처리량을 따로 보고합니다. 일반적인 처리량 값 범위는 0에서 I/O 채널의 최대 대역폭까지입니다.
- Queue Depth(대기열 깊이) – 대기열에서 처리를 기다리는 I/O 요청 수입니다. 애플리케이션에서 I/O 요청을 제출하였지만 디바이스가 다른 I/O 요청을 처리하고 있을 때는 전송되지 않는 경우가 있습니다. 이때 대기열에서 기다리게 되는데 이 대기 시간이 지연 시간과 서비스 시간(측정치로 사용하지 않음)을 구성합니다. 이 지표는 지정된 시간 간격 동안의 평균 대기열 깊이로 보고됩니다. Amazon RDS는 1분 간격으로 대기열 깊이를 보고합니다. 일반적인 대기열 깊이 값은 0에서 수백에 이릅니다.

측정된 IOPS 값은 개별 I/O 작업 크기와 독립적입니다. I/O 성능을 측정할 경우, 단순한 I/O 작업 수가 아닌 인스턴스 처리량에 관심을 두어야 한다는 의미입니다.

스토리지 성능에 영향을 끼치는 요인

시스템 활동과 데이터베이스 워크로드 모두 스토리지 성능에 영향을 미칠 수 있습니다.

시스템 활동

다음 시스템 관련 활동은 I/O 용량을 사용하기 때문에 진행되는 동안에는 데이터베이스 인스턴스 성능이 떨어질 수 있습니다.

- 다중 AZ 대기 생성
- 읽기 전용 복제본 생성
- 스토리지 유형 변경

데이터베이스 워크로드

어떤 경우, 데이터베이스 또는 애플리케이션 설계에 따라 동시 처리 가능한 연산 문제, 잠금 또는 그 밖에 데이터베이스 논쟁을 불러일으킬 수 있습니다. 이런 경우, 프로비저닝된 대역폭 모두를 직접 사용하지 못할 수 있습니다. 또한, 다음과 같은 워크로드 관련 상황이 발생할 수 있습니다.

- 기본 인스턴스 유형의 제한된 처리량에 도달했습니다.
- 애플리케이션이 충분한 I/O 작업을 실행하지 않기 때문에 대기열 깊이는 지속적으로 1 미만입니다.
- 몇 가지 I/O 용량을 사용하지 않아도 데이터베이스의 쿼리 문제가 있을 수 있습니다.

최대치에 이르거나 가까운 시스템 리소스가 단 하나도 없고, 스레드를 추가하더라도 데이터베이스의 트랜잭션 터리 속도가 증가하지 않는 경우에는 병목 현상의 원인이 데이터베이스에 있을 가능성이 높습니다. 가장 공통적인 원인은 행 잠금과 인덱스 페이지 잠금이지만 그 밖에 다른 가능성도 있습니다. 이런 경우에는 데이터베이스 성능 튜닝 전문가에게 조언을 구해야 합니다.

DB 인스턴스 클래스

Amazon RDS 데이터베이스 인스턴스의 최대 성능을 얻으려면 자체 스토리지 유형을 지원하는 충분한 대역폭이 있는 현재 작업 인스턴스 유형을 선택합니다. 예를 들어, 10Gb 네트워크에 연결된 EBS에 최적화된 인스턴스를 선택해야 합니다.

EBS 최적화를 지원하는 Amazon EC2 인스턴스의 전체 목록은 [EBS 최적화를 지원하는 인스턴스 유형](#)을 참조하십시오.

최상의 성능을 얻으려면 최신 세대 인스턴스를 사용할 것을 권장합니다. 이전 세대 DB 인스턴스는 인스턴스 스토리지 제한이 더 낮습니다. 다음 표에서는 각 데이터베이스 인스턴스에 대해 각 DB 인스턴스 클래스가 확장할 수 있는 최대 스토리지를 보여줍니다. 모든 값은 테라바이트(TiB) 단위입니다.

인스턴스 클래스	Mar	Mic	MyS	Ora	Pos								
db.m5 – 최신 세대 표준 인스턴스 클래스													
db.m5.24xlarge	64	16	64	64	64								
db.m5.12xlarge	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 – 현재 세대 표준 인스턴스 클래스													
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 – 전세대 표준 인스턴스 클래스													
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								

인스턴스 클래스	Mari	MySQL	Oracle	PostgreSQL								
Instance Class	MariaDB	MySQL	Oracle	PostgreSQL								
db.r5 – 최신 세대 메모리 최적화 인스턴스 클래스												
db.r5.24xlarge	16		16	64	64							
db.r5.12xlarge	16		16	64	64							
db.r5.4xlarge	16		16	64	64							
db.r5.2xlarge	16		16	64	64							
db.r5.xlarge	16		16	64	64							
db.r5.large	16		16	64	64							
db.r4 – 현재 세대 메모리 최적화 인스턴스 클래스												
db.r4.16xlarge	64	16	64	64	64							
db.r4.8xlarge	64	16	64	64	64							
db.r4.4xlarge	64	16	64	64	64							
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 – 전세대 메모리 최적화 인스턴스 클래스												
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							
Instance Class	MariaDB	MySQL	Oracle	PostgreSQL								
db.t3 – 최신 세대 버스트 성능 인스턴스 클래스												
db.t3.2xlarge	16		16	64	64							
db.t3.xlarge	16		16	64	64							

인스턴스 클래스	Mar	Mic	MySQL	Ora	PostgreSQL								
db.t3.large	16		16	64	64								
db.t3.medium	16		16	32	32								
db.t3.small	16		16	32	16								
db.t3.micro	16		16	32	16								
db.t2 – 현재 세대 버스트 성능 인스턴스 클래스													
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								
Instance Class	MariaDB	MySQL	Oracle	PostgreSQL									
db.x1e – 최신 세대 메모리 최적화 인스턴스 클래스													
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									
db.x1 – 현재 세대 메모리 최적화 인스턴스 클래스													
db.x1.32xlarge		16		64									
db.x1.16xlarge		16		64									

Oracle의 경우 다음 인스턴스 클래스에서만 최대 80,000IOPS로 확장할 수 있습니다.

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

지원되는 모든 인스턴스 클래스에 대한 자세한 내용은 [이전 세대 DB 인스턴스](#)를 참조하십시오.

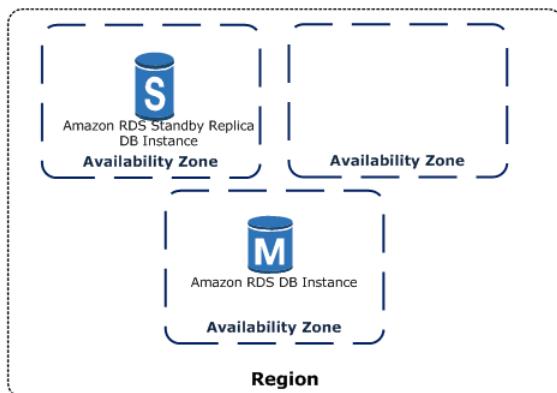
Amazon RDS를 위한 고가용성(다중 AZ)

Amazon RDS는 다중 AZ 배포를 사용해 DB 인스턴스에 고가용성과 장애 조치 기능을 지원합니다. Amazon RDS는 다양한 기술을 이용해 장애 조치 지원을 제공합니다. Oracle, PostgreSQL, MySQL 및 MariaDB DB 인스턴스용 다중 AZ 배포는 Amazon의 장애 조치 기술을 사용합니다. SQL Server DB 인스턴스는 SQL Server 데이터베이스 미러링(DBM)을 사용합니다.

다중 AZ 배포에서 Amazon RDS는 자동으로 서로 다른 가용 영역에 동기식 예비 복제본을 프로비저닝하고 유지합니다. 기본 DB 인스턴스는 가용 영역에서 예비 복제본으로 동기식으로 복제되어 데이터 이중화를 제공하고 I/O 중지를 제거하며 시스템 백업 시 지연 시간 스파이크를 최소화합니다. DB 인스턴스를 고가용성으로 실행하면 계획된 시스템 유지 관리 중 가용성을 향상시킬 수 있으며, 데이터베이스는 DB 인스턴스 오류 및 가용 영역 중단이 일어나는 것을 방지할 수 있습니다. 가용 영역에 대한 자세한 정보는 [리전 및 가용 영역 \(p. 99\)](#) 단원을 참조하십시오.

Note

고가용성 기능은 읽기 전용 시나리오의 확장 솔루션이 아니므로 예비 복제본을 사용하여 읽기 트래픽을 지원할 수 없습니다. 읽기 트래픽을 지원하려면 읽기 전용 복제본을 사용해야 합니다. 자세한 정보는 [읽기 전용 복제본 작업 \(p. 135\)](#) 단원을 참조하십시오.



RDS 콘솔을 사용하여 DB 인스턴스 생성 시 다중 AZ를 지정함으로써 간편하게 다중 AZ 배포를 생성할 수 있습니다. 또한 콘솔을 사용하여 DB 인스턴스를 수정하고 다중 AZ 옵션을 지정함으로써 기존 DB 인스턴스를 다중 AZ 배포로 변환할 수 있습니다. RDS 콘솔에 보조 AZ라는 예비 복제본의 가용 영역이 표시됩니다.

CLI를 사용하여 다중 AZ 배포를 지정할 수도 있습니다. AWS CLI [describe-db-instances](#) 명령 또는 Amazon RDS API [DescribeDBInstances](#) 작업을 사용하여 예비 복제본의 가용 영역(보조 AZ)을 볼 수 있습니다.

RDS 콘솔에는 예비 복제본의 가용 영역(보조 AZ)이 표시됩니다. 또한 AWS CLI [rds-describe-db-instances](#) 명령 또는 Amazon RDS API [DescribeDBInstances](#) 작업을 사용하여 보조 AZ를 찾을 수 있습니다.

다중 AZ 배포를 사용하는 DB 인스턴스는 동기식 데이터 복제 실행으로 인하여 단일 AZ 배포에 비해 쓰기 및 저장 지연 시간이 길어질 수 있습니다. AWS는 가용 영역 간 지연 시간이 짧은 네트워크 연결을 제공하도록 엔지니어링되어 있지만, 예비 복제본으로 배포 장애 조치를 취한 경우 지연 시간이 변경될 수 있습니다. 프로덕션 워크로드의 경우, 빠르고 일관된 성능을 구현하도록 프로비저닝된 IOPS에 최적화된 프로비저닝된 IOPS 및 DB 인스턴스 클래스를 사용하는 것이 좋습니다. DB 인스턴스 클래스에 대한 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.

다중 AZ 배포가 되도록 DB 인스턴스 설정

단일 AZ 배포에 DB 인스턴스가 있고 이 배포를 다중 AZ 배포로 수정하는 경우(SQL Server 또는 Amazon Aurora 이외의 다른 엔진에 대해) Amazon RDS는 몇 가지 조치를 취합니다. 우선 Amazon RDS는 배포에서

기본 DB 인스턴스의 스냅샷을 캡처한 후 이 스냅샷을 또 다른 가용 영역으로 복원합니다. 그런 다음 Amazon RDS는 기본 DB 인스턴스와 새 인스턴스 간에 동기 복제를 설정합니다. 이로써 단일 AZ에서 다중 AZ로 전환할 때 가동 중지를 피할 수 있지만 처음 다중 AZ로 전환할 때는 성능에 상당한 영향을 미칠 수 있습니다. 대용량의 쓰기 집약적인 DB 인스턴스일 경우에는 이러한 영향이 더욱 커집니다.

일단 수정이 완료되면 Amazon RDS는 과정 완료를 표시하는 이벤트(RDS-EVENT-0025)를 트리거합니다. Amazon RDS 이벤트를 모니터링할 수 있습니다. 이벤트에 대한 자세한 정보는 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오.

Amazon RDS 장애 조치 프로세스

다중 AZ를 활성화한 경우, DB 인스턴스에 계획되거나 계획되지 않은 중단이 발생하면 Amazon RDS는 자동으로 다른 가용 영역에 있는 예비 복제본으로 전환합니다. 장애 조치가 완료되는 데 소요되는 시간은 기본 DB 인스턴스를 사용할 수 없게 된 시점의 데이터베이스 활동 및 기타 조건에 따라 달라집니다. 장애 조치에 소요되는 시간은 일반적으로 60-120초입니다. 그러나 트랜잭션의 규모가 크거나 복구 프로세스가 복잡한 경우 장애 조치에 소요되는 시간이 증가할 수 있습니다. 장애 조치가 완료되면 RDS 콘솔 UI가 새로운 가용 영역에 반영하는 데 추가적으로 시간이 소요됩니다.

장애 조치 메커니즘은 자동으로 DB 인스턴스의 DNS 기록을 예비 DB 인스턴스 지점으로 변경합니다. 그 결과 DB 인스턴스의 기존 연결을 모두 재설정해야 합니다. Java DNS 캐싱 메커니즘의 작동 방법으로 인해, JVM 환경을 재구성해야 할 수 있습니다. 장애 조치의 경우에 DNS 값을 캐싱하는 Java 애플리케이션의 관리 방법에 대한 자세한 정보는 [AWS SDK for Java](#) 단원을 참조하십시오.

Note

DB 인스턴스를 재부팅할 때 장애 조치를 수동으로 강제할 수 있습니다. 자세한 내용은 [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#) 단원을 참조하십시오.

Amazon RDS는 자동으로 장애 조치를 취하여 관리자의 개입 없이 데이터베이스 작업을 신속하게 재개할 수 있도록 합니다. 다음 상태가 발생하는 경우 기본 DB 인스턴스는 자동으로 예비 복제본으로 전환됩니다.

- 가용 영역 중단
- 기본 DB 인스턴스 오류
- DB 인스턴스 서버 유형 변경
- DB 인스턴스의 운영 체제에서 소프트웨어 패치를 수행하는 중
- 장애 조치 재부팅을 사용하여 DB 인스턴스에 수동 장애 조치가 취해진 경우

몇 가지 방법을 통해 다중 AZ DB 인스턴스에 장애 조치가 취해졌는지 확인할 수 있습니다.

- DB 이벤트 구독을 설정하여 장애 조치가 취해졌음을 이메일이나 SMS를 통해 알릴 수 있습니다. 이벤트에 대한 자세한 정보는 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오.
- Amazon RDS 콘솔 또는 API 작업을 사용하여 DB 이벤트를 확인할 수 있습니다.
- Amazon RDS 콘솔 및 API 작업을 통해 다중 AZ 배포의 현재 상태를 확인할 수 있습니다.

장애 조치, 복구 시간 절감 및 기타 Amazon RDS 모범 사례에 대한 자세한 정보는 [Amazon RDS 모범 사례 \(p. 66\)](#) 단원을 참조하십시오.

관련 주제

- Microsoft SQL Server의 다중 AZ 배포 (p. 537)
- Oracle 다중 AZ 배포 라이선스 (p. 704)

Amazon RDS DB 인스턴스 수명 주기

Amazon RDS DB 인스턴스의 수명 주기는 인스턴스의 생성, 변경, 유지 관리 및 업그레이드, 백업 및 복구 실행, 재부팅, 그리고 삭제로 구성되어 있습니다. 이 섹션에서는 이러한 프로세스에 대한 정보를 비롯해 관련된 추가 링크에 대해 살펴보겠습니다.

주제

- [Amazon RDS DB 인스턴스 생성 \(p. 111\)](#)
- [Amazon RDS DB 인스턴스에 연결 \(p. 111\)](#)
- [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#)
- [DB 인스턴스 유지 관리 \(p. 113\)](#)
- [DB 인스턴스 엔진 버전 업그레이드 \(p. 118\)](#)
- [DB 인스턴스 이름 바꾸기 \(p. 121\)](#)
- [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#)
- [Amazon RDS DB 인스턴스의 일시적 중지 \(p. 125\)](#)
- [이전에 중지된 Amazon RDS DB 인스턴스 시작 \(p. 127\)](#)
- [DB 인스턴스 삭제 \(p. 128\)](#)

Amazon RDS DB 인스턴스 생성

Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. Amazon RDS DB 인스턴스를 생성하려면 해당 데이터베이스 엔진에 대한 지침을 따르십시오.

- [MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 437\)](#)
- [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 생성 \(p. 494\)](#)
- [MySQL 데이터베이스 엔진 기반 DB 인스턴스 생성 \(p. 587\)](#)
- [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#)
- [PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스 만들기 \(p. 947\)](#)

Amazon RDS DB 인스턴스에 연결

Amazon RDS DB 인스턴스를 생성한 후에는 표준 SQL 클라이언트 애플리케이션을 사용해 DB 인스턴스에 연결할 수 있습니다. Amazon RDS DB 인스턴스에 연결하려면 해당 데이터베이스 엔진에 대한 지침을 따르십시오.

- [MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 \(p. 445\)](#)
- [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 503\)](#)
- [MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 595\)](#)
- [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 \(p. 736\)](#)
- [PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 953\)](#)

Amazon RDS DB 인스턴스 수정

대부분의 DB 인스턴스 수정 사항은 즉시 적용하거나, 다음 유지 관리 기간까지 연기할 수 있습니다. 파라미터 그룹 변경 등의 수정 사항을 적용하려면 DB 인스턴스를 수동으로 재부팅해야 합니다.

Important

일부 수정 사항을 적용하려면 Amazon RDS가 DB 인스턴스를 재부팅해야 하므로 인스턴스가 잠시 종단될 수도 있습니다. DB 인스턴스 설정을 수정하기 전에 데이터베이스 및 애플리케이션에 미치는 영향을 확인하십시오.

Amazon RDS DB 인스턴스를 변경하려면 해당 데이터베이스 엔진에 대한 지침을 따르십시오.

- MariaDB 데이터베이스 엔진 기반 DB 인스턴스의 변경 (p. 449)
- Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 (p. 509)
- MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경 (p. 600)
- Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 (p. 742)
- PostgreSQL 데이터베이스 엔진을 실행 중인 DB 인스턴스 수정 (p. 956)

즉시 적용 파라미터 사용

DB 인스턴스를 수정하는 경우 변경 사항을 즉시 적용할 수 있습니다. 변경 사항을 즉시 적용하려면 AWS Management 콘솔에서 Apply Immediately(즉시 적용) 옵션을 선택하거나, AWS CLI를 호출하는 경우 `--apply-immediately` 파라미터를 사용하거나, Amazon RDS API를 사용하는 경우 `ApplyImmediately` 파라미터를 `true`로 설정합니다.

변경 사항을 즉시 적용하지 않기로 선택하면 변경 사항이 보류 중 수정 사항 대기열로 보내집니다. 다음 유지 관리 기간에 대기열에 있는 보류 중 변경 사항이 적용됩니다. 변경 사항을 즉시 적용하기로 선택하면 새로운 변경 사항과 보류 중인 수정 사항 대기열에 있는 모든 변경 사항이 적용됩니다.

Important

보류 중 수정 사항이 하나라도 가동 중지를 필요로 하는 경우 즉시 적용을 선택하면 예기치 못한 가동 중지가 발생할 수 있습니다.

변경 사항을 즉시 적용하도록 선택하면, 보류 중인 수정 사항도 다음 유지 관리 기간이 아니라 즉시 적용됩니다.

보류 중인 변경 사항을 다음 유지 관리 기간에 적용하지 않으려면 AWS CLI를 사용하여 변경 사항을 되돌리도록 DB 인스턴스를 수정하고 `--apply-immediately` 옵션을 지정할 수 있습니다.

변경을 지연하기로 선택하더라도 일부 데이터베이스 설정 변경 사항은 즉시 적용됩니다. 서로 다른 데이터베이스 설정이 즉시 적용 설정과 상호 작용하는 방법을 보려면 특정 데이터베이스 엔진의 설정을 참조하십시오.

- MariaDB DB 인스턴스 설정 (p. 450)
- Microsoft SQL Server DB 인스턴스 설정 (p. 511)
- MySQL DB 인스턴스 설정 (p. 602)
- Oracle DB 인스턴스 설정 (p. 744)
- PostgreSQL DB 인스턴스 설정 (p. 958)

관련 주제

- DB 인스턴스 이름 바꾸기 (p. 121)
- DB 클러스터에서 DB 인스턴스 (p. 123)
- Amazon RDS DB 인스턴스의 일시적 중지 (p. 125)
- `modify-db-instance`
- `ModifyDBInstance`

DB 인스턴스 유지 관리

Amazon RDS는 Amazon RDS 리소스를 정기적으로 유지 관리합니다. 이러한 유지 관리에는 DB 인스턴스의 기본 하드웨어, 기본 운영 체제(OS) 또는 데이터베이스 엔진 버전에 대한 업데이트가 수반되는 경우가 많습니다. 운영 체제 업데이트는 보안상 가장 빈번하게 발생하며 최대한 빠른 시간 내에 실행해야 합니다.

일부 유지 관리 항목을 사용하려면 Amazon RDS에서 DB 인스턴스를 잠시 동안 오프라인 상태로 전환해야 합니다. 리소스가 오프라인 상태에 있어야 하는 유지 관리 항목에는 필수 운영 체제 또는 데이터베이스 패치 이 포함됩니다. 이때 보안 및 인스턴스 안정성과 관련된 패치에 대해 필수 패치 작업으로 자동 예약됩니다. 이러한 패치 작업은 찾지는 않아서(수 개월에 한 번 정도) 유지 관리 기간에 비하면 매우 적은 편입니다.

즉시 적용되지 않도록 연기한 DB 인스턴스 수정은 유지 관리 기간에 적용됩니다. 예를 들어 DB 인스턴스 클래스 또는 파라미터 그룹을 유지 관리 기간에 변경하도록 선택할 수 있습니다. DB 인스턴스 수정에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#) 단원을 참조하십시오.

RDS 콘솔, AWS CLI 또는 Amazon RDS API를 사용하여 DB 인스턴스에 대해 유지 관리 업데이트를 사용할 수 있는지 여부를 확인할 수 있습니다. 업데이트가 있는 경우에는 다음과 같이 Amazon RDS 콘솔에서 DB 인스턴스의 유지 관리 열에 사용 가능 여부가 표시됩니다.

The screenshot shows the 'Instances' section of the Amazon RDS console. At the top, there's a toolbar with various icons. Below it is a search bar labeled 'DB Instances...'. A message 'Viewing 130 of 130 DB Instances' is displayed. The main area is a table with three rows of data. The columns are: VPC, Multi-AZ, Class, Status, and Maintenance. The first row has a VPC value of 'm1-large-67757-ygij', a Multi-AZ value of 'See Details', a Class value of 'db.m1.small', a Status value of 'available' (green), and a Maintenance value of 'Available' (orange). The second row has a VPC value of 'vpc-d12950ba', a Multi-AZ value of 'See Details', a Class value of 'db.m1.large', a Status value of 'available' (green), and a Maintenance value of 'None' (gray). The third row has a VPC value of 'vpc-7f5ab617', a Multi-AZ value of 'No', a Class value of 'db.m1.small', a Status value of 'available' (green), and a Maintenance value of 'None' (gray). A red box highlights the 'Maintenance' column header and the 'Available' cell in the first row.

VPC	Multi-AZ	Class	Status	Maintenance
m1-large-67757-ygij	See Details	db.m1.small	available	Available
vpc-d12950ba	See Details	db.m1.large	available	None
vpc-7f5ab617	No	db.m1.small	available	None

DB 인스턴스에 대해 유지 관리 업데이트가 제공되지 않는 경우 그에 대한 열 값은 없음입니다.

DB 인스턴스에 대해 유지 관리 업데이트가 제공되지 않는 경우 다음과 같은 열 값이 가능합니다.

- 필수 – 유지 관리 작업은 리소스에 적용되며 보류할 수 없습니다.
- 사용 가능 – 유지 관리 작업을 사용할 수 있습니다. 그러나 리소스에 자동으로 적용되지 않고 수동으로 적용할 수 있습니다.
- 다음 기간 – 유지 관리 작업은 다음 유지 관리 기간 중에 리소스에 적용됩니다.
- 진행 중 – 유지 관리 작업이 리소스에 적용되고 있는 중입니다.

업데이트가 있을 경우에는 다음 테이블의 작업 중 하나를 실행할 수 있습니다.

- 유지 관리 값이 다음 기간인 경우 작업에서 업그레이드 보류를 선택하여 유지 관리 항목을 보류하십시오.
- 유지 관리 항목을 즉시 적용합니다.
- 다음 유지 관리 기간 중 시작할 유지 관리 항목을 예약합니다.
- 작업 없음

Note

일부 OS 업데이트가 필수로 표시됩니다. 필수 업데이트를 연기할 경우에는 Amazon RDS에서 언제 업데이트가 수행될지를 알려 주는 알림이 수신됩니다. 기타 업데이트는 사용 가능으로 표시되며 이러한 업데이트는 무한정으로 연기할 수 있습니다.

조치를 취하려면 DB 인스턴스를 선택하여 세부 정보를 표시한 후 Maintenance & backups(유지 관리 및 백업)을 선택하십시오. 그러면 보류 중인 유지 관리 항목이 표시됩니다.

The screenshot shows the 'Maintenance & backups' tab selected in the navigation bar. Under the 'Maintenance' heading, it displays 'Auto minor version upgrade' status as 'Enabled'. Below this, a table lists a single pending maintenance task: 'Automatic minor version upgrade to postgres 9.6.11' (Type: db-upgrade, Status: next window, Apply date: February 25th 2019, 3:28:00 am UTC-8 (local)). There are buttons for 'Apply now' and 'Apply at next maintenance window'.

유지 관리 기간에 따라 대기 중인 작업의 시작 시기가 결정되지만 이러한 작업의 전체 실행 시간이 줄어들지는 않습니다. 유지 관리 기간에 끝나기 전에 반드시 유지 관리 작업이 끝나도록 되어 있는 것은 아니고, 특정 종료 시각을 지나 계속 진행될 수 있습니다. 자세한 내용은 [Amazon RDS 유지 관리 기간 \(p. 116\)](#) 단원을 참조하십시오.

DB 인스턴스의 업데이트 적용

Amazon RDS를 사용하여 유지 관리 작업을 적용하는 시기를 선택할 수 있습니다. RDS 콘솔, AWS Command Line Interface(AWS CLI) 또는 RDS API를 사용해 Amazon RDS가 업데이트를 적용하는 시기를 결정할 수 있습니다.

콘솔

DB 인스턴스의 업데이트를 관리하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 필수 업데이트가 포함된 DB 인스턴스를 선택합니다.
4. 작업에서 다음 중 하나를 선택합니다.
 - 지금 업그레이드
 - Upgrade at next window(다음에 업그레이드)

Note

다음에 업그레이드를 선택한 후 나중에 업데이트를 연기하려면 업그레이드 연기를 선택합니다.

AWS CLI

대기 중인 업데이트를 DB 인스턴스에 적용하려면 `apply-pending-maintenance-action` AWS CLI 명령을 사용합니다.

Example

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

하나 이상의 대기 중인 업데이트가 있는 리소스 목록을 반환하려면, [describe-pending-maintenance-actions](#) AWS CLI 명령을 사용합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds describe-pending-maintenance-actions \
--resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db
```

Windows의 경우:

```
aws rds describe-pending-maintenance-actions ^
--resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db
```

[describe-pending-maintenance-actions](#) AWS CLI 명령의 --filters 파라미터를 지정하여 DB 인스턴스에 대한 리소스 목록을 반환할 수도 있습니다. --filters 명령의 형식은 Name=*filter-name*, Value=*resource-id*, ...입니다.

필터의 Name 파라미터에 대해 허용되는 값은 다음과 같습니다.

- db-instance-id – DB 인스턴스 식별자 또는 Amazon 리소스 이름(ARN) 목록을 허용합니다. 반환되는 목록에는 이러한 식별자 또는 ARN으로 식별된 DB 인스턴스에 대해 보류 중인 유지 관리 작업만 포함됩니다.
- db-cluster-id – Amazon Aurora의 DB 클러스터 식별자 또는 ARN 목록을 허용합니다. 반환되는 목록에는 이러한 식별자 또는 ARN으로 식별된 DB 클러스터에 대해 보류 중인 유지 관리 작업만 포함됩니다.

예를 들어 다음 예에서는 sample-instance1 및 sample-instance2 DB 인스턴스에 대해 보류 중인 유지 관리를 반환합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds describe-pending-maintenance-actions \
--filters Name=db-instance-id,Values=sample-instance1,sample-instance2
```

Windows의 경우:

```
aws rds describe-pending-maintenance-actions ^
--filters Name=db-instance-id,Values=sample-instance1,sample-instance2
```

RDS API

업데이트를 DB 인스턴스에 적용하려면 Amazon RDS API [ApplyPendingMaintenanceAction](#) 작업을 호출합니다.

하나 이상의 대기 중인 업데이트가 있는 리소스 목록을 반환하려면 Amazon RDS API [DescribePendingMaintenanceActions](#) 작업을 호출합니다.

다중 AZ 배포 유지

DB 인스턴스를 다중 AZ 배포로 실행하는 경우에는 Amazon RDS가 다음 단계에 따라 작업 시스템 업데이트를 적용하기 때문에 유지 관리 이벤트의 영향을 최소화할 수 있습니다.

1. 대기 목록의 유지 관리를 실행합니다.
2. 대기 목록을 기본 목록으로 승격시킵니다.
3. 이전에 기본 목록이었지만 현재는 새로운 대기 목록인 유지 관리를 실행합니다.

다중 AZ 배포에서 DB 인스턴스에 대한 데이터베이스 엔진을 수정하면 Amazon RDS가 기본 및 보조 DB 인스턴스를 모두 동시에 업그레이드합니다. 이러한 경우 업그레이드하는 동안 전체 다중 AZ 배포에 대한 데이터베이스 엔진이 중단됩니다.

다중 AZ 배포에 대한 자세한 내용은 [Amazon RDS를 위한 고가용성\(다중 AZ\) \(p. 109\)](#) 단원을 참조하십시오.

Amazon RDS 유지 관리 기간

모든 DB 인스턴스에는 시스템 변경 내용이 적용되는 주 단위 유지 관리 기간이 있습니다. 유지 관리 기간은 요청이나 필요에 따라 수정하거나 소프트웨어 패치를 적용하는 시기를 조정할 수 있는 기간입니다. 유지 관리 이벤트가 특정 주에 예정되어 있는 경우 사용자가 지정하는 30분의 유지 관리 기간 중에 해당 이벤트가 시작됩니다. 또한 대부분의 유지 관리 이벤트가 30분의 유지 관리 기간 중에 완료됩니다. 단, 대규모 유지 관리 이벤트는 완료하는 데 30분이 넘게 걸릴 수 있습니다.

지역별로 8시간 블록 시간 중에서 30분 유지 관리 시간이 임의로 선택됩니다. DB 인스턴스 생성 시 기본 유지 관리 기간을 지정하지 않으면 Amazon RDS에서 임의로 선택한 요일에 30분 유지 관리 기간을 배정합니다.

유지 관리가 적용되는 동안 RDS에서 사용자의 DB 인스턴스에 있는 리소스 중 일부를 사용합니다. 이에 따라 성능에 미미한 영향이 있을 수 있습니다. DB 인스턴스의 경우 드물지만, 유지 관리 업데이트를 완료하면서 다중 AZ 장애 조치가 필요한 경우가 있을 수 있습니다.

다음에서 기본 유지 관리 기간이 할당된 리전별 시간 블록을 확인할 수 있습니다.

리전	시간 블록
미국 서부(오레곤) 지역	06:00–14:00 UTC
미국 서부(캘리포니아 북부) 리전	06:00–14:00 UTC
미국 동부(오하이오) 리전	03:00–11:00 UTC
미국 동부(버지니아 북부) 지역	03:00–11:00 UTC
아시아 태평양(뭄바이) 리전	17:30–01:30 UTC
아시아 태평양(서울) 리전	13:00–21:00 UTC
아시아 태평양(싱가포르) 리전	14:00–22:00 UTC

리전	시간 블록
아시아 태평양(시드니) 리전	12:00–20:00 UTC
아시아 태평양(도쿄) 리전	13:00–21:00 UTC
캐나다(중부) 리전	03:00–11:00 UTC
EU(프랑크푸르트) 리전	23:00–07:00 UTC
EU(아일랜드) 지역	22:00–06:00 UTC
EU(런던) 리전	22:00–06:00 UTC
남아메리카(상파울루) 리전	00:00–08:00 UTC
AWS GovCloud(미국 서부)	06:00–14:00 UTC

기본 DB 인스턴스 유지 관리 기간 조정

유지 관리 기간은 사용률이 가장 낮은 시간에 할당되어야 하므로 수시로 수정되어야 할 수 있습니다. DB 인스턴스 클래스 변경 등의 시스템 변경 내용을 적용 중인 경우 이 시간 동안 DB 인스턴스를 사용할 수 없으므로 필요한 변경을 수행하는 데 필요한 최소 시간 동안은 작동이 중단되어야 합니다.

다음 예에서는 DB 인스턴스에 대한 기본 유지 관리 기간을 조정합니다.

이 예에서는 mydbinstance라는 DB 인스턴스가 있으며 기본 유지 관리 기간이 "Sun:05:00-Sun:06:00" UTC라고 가정하겠습니다.

콘솔

기본 유지 관리 기간을 조정하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 변경하려는 DB 인스턴스를 선택합니다.
3. [Modify]를 선택합니다. [Modify DB Instance] 페이지가 나타납니다.
4. 유지 관리 섹션에서 유지 관리 기간을 업데이트합니다.

Note

DB 인스턴스에 대한 유지 관리 기간 및 백업 기간은 겹칠 수 없습니다. 백업 기간과 겹치는 유지 관리 기간의 값을 입력하면 오류 메시지가 나타납니다.

5. [Continue]를 선택합니다.

확인 페이지에서 변경 내용을 검토합니다.

6. 유지 관리 기간에 변경 사항을 즉시 적용하려면 즉시 적용을 선택합니다.
7. DB 인스턴스 수정을 선택하여 변경 사항을 저장합니다.

그렇지 않으면 [Back]을 선택하여 변경 내용을 편집하거나 [Cancel]을 선택하여 변경 내용을 취소합니다.

AWS CLI

기본 유지 관리 기간을 조정하려면 AWS CLI `modify-db-instance` 명령을 다음 파라미터와 함께 사용합니다.

- `--db-instance-identifier`
- `--preferred-maintenance-window`

Example

다음은 유지 관리 기간을 화요일 오전 4:00-4:30(UTC)로 설정하는 코드 예제입니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

Windows의 경우:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

RDS API

기본 유지 관리 기간을 조정하려면 Amazon RDS API `ModifyDBInstance` 작업을 다음 파라미터와 함께 사용합니다.

- `DBInstanceIdentifier = mydbinstance`
- `PreferredMaintenanceWindow = Tue:04:00-Tue:04:30`

Example

다음은 유지 관리 기간을 화요일 오전 4:00-4:30(UTC)로 설정하는 코드 예제입니다.

```
https://rds.us-west-2.amazonaws.com/
?Action=ModifyDBInstance
&DBInstanceIdentifier=mydbinstance
&PreferredMaintenanceWindow=Tue:04:00-Tue:04:30
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-09-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140425/us-east-1/rds/aws4_request
&X-Amz-Date=20140425T192732Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=1dc9dd716f4855e9bdf188c70f1cf9f6251b070b68b81103b59ec70c3e7854b3
```

DB 인스턴스 엔진 버전 업그레이드

Amazon RDS는 지원되는 각 데이터베이스 엔진의 최신 버전을 제공하여 DB 인스턴스를 최신 상태로 유지합니다. 최신 버전에는 데이터베이스 엔진의 버그 수정, 보안 강화 및 기타 개선 사항이 포함될 수 있습니다. Amazon RDS가 새로운 버전의 데이터베이스 엔진을 지원하는 경우, 데이터베이스 DB 인스턴스를 업그레이드할 방법과 시기를 선택할 수 있습니다.

메이저 버전 업그레이드와 마이너 버전 업그레이드라는 두 가지 업그레이드가 있습니다. 일반적으로 메이저 엔진 버전 업그레이드로 기존 애플리케이션과 호환되지 않는 변경 사항이 도입될 수 있습니다. 이와 대조적으로 마이너 버전 업그레이드에는 기존 애플리케이션과 역호환되는 변경 사항만 포함됩니다.

버전 번호 순서는 각 데이터베이스 엔진마다 다릅니다. 예를 들어 Amazon RDS MySQL 5.7 및 8.0은 메이저 엔진 버전이고, 5.7 버전에서 8.0 버전으로의 업그레이드는 메이저 버전 업그레이드입니다. Amazon RDS MySQL 버전 5.7.22 및 5.7.23은 마이너 버전이고, 5.7.22에서 5.7.23으로의 업그레이드는 마이너 버전 업그레이드입니다.

특정 DB 엔진의 메이저 및 마이너 버전 업그레이드에 대한 자세한 내용은 다음 DB 엔진 설명서를 참조하십시오.

- [MariaDB DB 엔진 업그레이드 \(p. 455\)](#)
- [Microsoft SQL Server DB 엔진 업그레이드 \(p. 516\)](#)
- [MySQL DB 엔진 업그레이드 \(p. 607\)](#)
- [Oracle DB 엔진 업그레이드 \(p. 752\)](#)
- [Amazon RDS용 PostgreSQL DB 엔진 업그레이드 \(p. 963\)](#)

메이저 버전 업그레이드를 위해서는 AWS Management 콘솔, AWS CLI 또는 RDS API를 통해 DB 엔진 버전을 수동으로 수정해야 합니다. 마이너 버전 업그레이드를 위해서는 엔진 버전을 수동으로 수정하거나 마이너 버전 자동 업그레이드를 활성화합니다.

주제

- [엔진 버전 수동 업그레이드 \(p. 119\)](#)
- [마이너 엔진 버전 자동 업그레이드 \(p. 120\)](#)

엔진 버전 수동 업그레이드

DB 인스턴스의 엔진 버전을 수동으로 업그레이드하려면 AWS Management 콘솔, AWS CLI 또는 RDS API를 사용할 수 있습니다.

콘솔을 사용하여 DB 인스턴스의 엔진 버전 업그레이드

콘솔을 사용하여 DB 인스턴스의 엔진 버전을 업그레이드하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 업그레이드하려는 DB 인스턴스를 선택합니다.
3. [Modify]를 선택합니다. [Modify DB Instance] 페이지가 나타납니다.
4. DB 엔진 버전에서 새 버전을 선택합니다.
5. [Continue]를 수정 사항을 요약한 내용을 확인합니다.
6. 변경 사항을 즉시 적용하려면 즉시 적용을 선택합니다. 일부의 경우 이 옵션을 선택하면 중단이 발생할 수 있습니다. 자세한 내용은 [즉시 적용 파라미터 사용 \(p. 112\)](#) 단원을 참조하십시오.
7. 확인 페이지에서 변경 내용을 검토합니다. 변경 내용이 정확할 경우 [Modify DB Instance]를 선택하여 변경 내용을 저장합니다.

그렇지 않으면 [Back]을 선택하여 변경 내용을 편집하거나 [Cancel]을 선택하여 변경 내용을 취소합니다.

AWS CLI를 사용하여 DB 인스턴스의 엔진 버전 업그레이드

DB 인스턴스의 엔진 버전을 업그레이드하려면 CLI `modify-db-instance` 명령을 사용합니다. 다음 파라미터를 지정합니다.

- `--db-instance-identifier` – DB 인스턴스의 이름입니다.
- `--engine-version` – 업그레이드할 데이터베이스 엔진의 버전 번호입니다.

유효한 엔진 버전에 대한 정보를 보려면 AWS CLI [describe-db-engine-versions](#) 명령을 사용합니다.

- `--allow-major-version-upgrade` – 메이저 버전을 업그레이드합니다.
- `--no-apply-immediately` – 변경 사항이 다음 유지 관리 기간에 적용됩니다. 변경 사항을 바로 적용하려면 `--apply-immediately`를 사용합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
  --db-instance-identifier mydbinstance \
  --engine-version new_version \
  --allow-major-version-upgrade \
  --no-apply-immediately
```

Windows의 경우:

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --engine-version new_version ^
  --allow-major-version-upgrade ^
  --no-apply-immediately
```

RDS API를 사용하여 DB 인스턴스의 엔진 버전 업그레이드

DB 인스턴스의 엔진 버전을 업그레이드하려면 [ModifyDBInstance](#) 작업을 사용합니다. 다음 파라미터를 지정합니다.

- `DBInstanceIdentifier` – DB 인스턴스의 이름입니다(예: *mydbinstance*).
- `EngineVersion` – 업그레이드할 데이터베이스 엔진의 버전 번호입니다. 유효한 엔진 버전에 대한 정보를 보려면 [DescribeDBEngineVersions](#) 작업을 사용합니다.
- `AllowMajorVersionUpgrade` – 메이저 버전 업그레이드를 허용하는지 여부입니다. 그렇게 하려면 값을 `true`로 설정합니다.
- `ApplyImmediately` – 변경 사항을 즉시 적용하거나 다음 유지 관리 기간에 적용합니다. 변경 사항을 바로 적용하려면 값을 `true`로 설정합니다. 변경 사항을 다음 유지 관리 기간에 적용하려면 값을 `false`로 설정합니다.

마이너 엔진 버전 자동 업그레이드

마이너 엔진 버전은 메이저 엔진 버전 내 DB 엔진 버전의 업데이트입니다. 예를 들어 메이저 엔진 버전은 그 안에 마이너 엔진 버전 5.7.22 및 5.7.23이 있는 5.7일 수 있습니다.

Amazon RDS가 데이터베이스의 DB 엔진 버전을 자동으로 업그레이드하도록 하려면 데이터베이스에 대해 마이너 버전 업그레이드를 활성화하면 됩니다. 마이너 엔진 버전이 기본 마이너 엔진 버전으로 지정된 경우, 다음 조건을 모두 충족하는 각 데이터베이스가 마이너 엔진 버전으로 자동 업그레이드됩니다.

- 데이터베이스가 기본 마이너 엔진 버전보다 낮은 DB 엔진의 마이너 버전을 실행하고 있습니다.
- 데이터베이스에 마이너 버전 자동 업그레이드가 활성화되어 있습니다.

다음 작업을 수행할 때 DB 인스턴스에서 마이너 버전 자동 업그레이드를 활성화할지 여부를 제어할 수 있습니다.

- [DB 인스턴스 생성 \(p. 111\)](#)

- DB 인스턴스 수정 (p. 111)
- 읽기 전용 복제본 만들기 (p. 138)
- 스냅샷에서 DB 인스턴스 복원 (p. 205)
- DB 인스턴스를 특정 시간으로 복원 (p. 221)
- Amazon S3에서 DB 인스턴스 가져오기 (p. 619)(Amazon S3에 MySQL 백업의 경우)

이러한 작업을 수행할 때 다음과 같은 방법으로 DB 인스턴스에서 마이너 버전 자동 업그레이드를 활성화할지 여부를 제어할 수 있습니다.

- 콘솔을 사용하여 마이너 버전 자동 업그레이드 옵션을 설정합니다.
- AWS CLI를 사용하여 `--auto-minor-version-upgrade` | `--no-auto-minor-version-upgrade` 옵션을 설정합니다.
- RDS API를 사용하여 `AutoMinorVersionUpgrade` 파라미터를 설정합니다.

데이터베이스 중 하나에 새로운 마이너 엔진 버전 업그레이드를 사용할 수 있는 경우 Amazon RDS 이벤트 알림을 받을 수 있습니다. 알림을 받으려면 Amazon Simple Notification Service(Amazon SNS)를 통해 Amazon RDS 이벤트 알림을 구독합니다. 자세한 내용은 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오.

DB 인스턴스에 DB 엔진 버전 업그레이드 등의 유지 관리 업데이트를 사용할 수 있는지 여부를 확인하려면 콘솔, AWS CLI 또는 RDS API를 사용하면 됩니다. DB 엔진 버전을 수동으로 업그레이드하고 유지 관리 기간을 조정할 수도 있습니다. 자세한 내용은 [DB 인스턴스 유지 관리 \(p. 113\)](#) 단원을 참조하십시오.

DB 인스턴스 이름 바꾸기

AWS Management 콘솔, AWS CLI `modify-db-instance` 명령 또는 Amazon RDS API `ModifyDBInstance` 작업을 사용하여 DB 인스턴스의 이름을 바꿀 수 있습니다. DB 인스턴스 이름을 변경하면 커다란 영향을 끼칠 수 있으므로 DB 인스턴스 이름을 바꾸기 전에 다음과 같이 반드시 알아야 할 몇 가지가 있습니다.

- DB 인스턴스 이름을 변경하면 DB 인스턴스의 앤드포인트도 변경됩니다. URL에는 DB 인스턴스에 할당한 이름이 포함되어 있기 때문입니다. 트래픽은 항상 이전 URL에서 새 URL로 리디렉션해야 합니다.
- DB 인스턴스 이름을 변경하면 DB 인스턴스에서 이전에 사용된 DNS 이름은 바로 삭제되지만 캐시는 몇 분 더 남을 수도 있습니다. 이름이 바뀐 DB 인스턴스의 새로운 DNS 이름은 약 10분 후부터 적용됩니다. 이름이 바뀐 DB 인스턴스를 사용하려면 새로운 이름이 적용될 때까지 기다려야 합니다.
- 인스턴스 이름이 바뀌면 기존 DB 인스턴스 이름은 사용할 수 없습니다.
- DB 인스턴스와 연동되어 있던 읽기 전용 복제본은 이름이 바뀐 후에도 모두 인스턴스와 연동된 상태를 유지합니다. 예를 들어 프로덕션 데이터베이스 역할을 하는 DB 인스턴스에 읽기 전용 복제본이 여러 개 연동되어 있다고 가정하겠습니다. 이때 DB 인스턴스 이름을 변경한 후 프로덕션 환경에서 DB 스냅샷으로 교체하더라도 이름을 바꾼 DB 인스턴스는 읽기 전용 복제본이 그대로 연동되어 있습니다.
- DB 인스턴스 이름을 재사용하면 DB 인스턴스 이름과 연동되어 있는 측정치와 이벤트가 유지됩니다. 예를 들어 읽기 전용 복제본을 승격하여 이전 마스터 이름으로 변경하는 경우 마스터와 연동되어 있던 이벤트와 측정치가 바뀐 이름의 인스턴스로 연동됩니다.
- DB 인스턴스 태그는 이름 변경 여부에 상관없이 DB 인스턴스에 그대로 남습니다.
- DB 스냅샷은 바뀐 이름의 DB 인스턴스로 유지됩니다.

기존 DB 인스턴스 교체를 위한 이름 바꾸기

DB 인스턴스 이름을 바꾸는 가장 공통적인 이유는 읽기 전용 복제본을 승격하거나 DB 스냅샷 또는 PITR에서 데이터를 복구해야 하기 때문입니다. 데이터베이스 이름을 변경하면 애플리케이션 코드를 변경하여 DB 인스턴스를 참조하지 않아도 DB 인스턴스를 교체할 수 있습니다. 이러한 경우 방법은 다음과 같습니다.

1. 마스터 DB 인스턴스로 전송되는 트래픽을 모두 차단합니다. 여기에는 DB 인스턴스의 데이터베이스에서 수신되는 트래픽을 리디렉션하거나 그 밖에 트래픽이 DB 인스턴스의 데이터베이스에 액세스하지 못하도록 차단할 수 있는 방법도 사용됩니다.
2. 이 주제와 관련하여 나중에 다시 설명하겠지만 마스터 DB 인스턴스를 마스터를 의미하지 않는 이름으로 변경합니다.
3. DB 스냅샷에서 복구하거나 읽기 전용 복제본을 승격하여 새로운 마스터 DB 인스턴스를 생성한 다음 이 새로운 인스턴스를 이전 마스터 DB 인스턴스 이름으로 명명합니다.
4. 읽기 전용 복제본을 새로운 마스터 DB 인스턴스와 연동시킵니다.

이전 마스터 DB 인스턴스를 삭제할 때 원하지 않는 DB 스냅샷까지 삭제되는 경우 사용자 본인에게 책임이 있습니다.

읽기 전용 복제본 승격에 대한 자세한 내용은 [읽기 전용 복제본을 둑립 DB 인스턴스로 승격 \(p. 139\)](#) 단원을 참조하십시오.

AWS Management 콘솔

DB 인스턴스 이름 변경 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 이름을 바꿀 DB 인스턴스를 선택합니다.
4. [Modify]를 선택합니다.
5. 설정에서 DB 인스턴스 식별자에 새 이름을 입력합니다.
6. [Continue]를 선택합니다.
7. 변경 사항을 즉시 적용하려면 즉시 적용을 선택합니다. 일부의 경우 이 옵션을 선택하면 중단이 발생할 수 있습니다. 자세한 내용은 [즉시 적용 파라미터 사용 \(p. 112\)](#) 단원을 참조하십시오.
8. 확인 페이지에서 변경 내용을 검토합니다. 변경 내용이 정확할 경우 [Modify DB Instance]를 선택하여 변경 내용을 저장합니다.

그렇지 않으면 [Back]을 선택하여 변경 내용을 편집하거나 [Cancel]을 선택하여 변경 내용을 취소합니다.

CLI

DB 인스턴스의 이름을 바꾸려면 AWS CLI 명령 `modify-db-instance`를 사용합니다. 현재 `--db-instance-identifier` 값 및 `--new-db-instance-identifier` 파라미터를 DB 인스턴스의 새 이름과 함께 제공합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
--db-instance-identifier DBInstanceIdentifier \
--new-db-instance-identifier NewDBInstanceIdentifier
```

Windows의 경우:

```
aws rds modify-db-instance ^
--db-instance-identifier DBInstanceIdentifier ^
--new-db-instance-identifier NewDBInstanceIdentifier
```

API

DB 인스턴스의 이름을 변경하려면 Amazon RDS API 함수인 [ModifyDBInstance](#)를 다음 파라미터와 함께 호출합니다.

- `DBInstanceIdentifier` = 기존 인스턴스 이름
- `NewDBInstanceIdentifier` = 새로운 인스턴스 이름

```
https://rds.amazonaws.com/
?Action=ModifyDBInstance
&DBInstanceIdentifier=mydbinstance
&NewDBInstanceIdentifier=mynewdbinstanceidentifier
&Version=2012-01-15
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2012-01-20T22%3A06%3A23.624Z
&AWSAccessKeyId=<AWS Access Key ID>
&Signature=<Signature>
```

DB 클러스터에서 DB 인스턴스

일반적으로 유지 관리를 이유로 DB 인스턴스를 재부팅해야 할 수 있습니다. 예를 들어 특정 내용을 수정하거나 DB 인스턴스 와 연결된 DB 파라미터 그룹을 변경하는 경우 변경 내용을 적용하려면 인스턴스를 재부팅해야 합니다.

Note

DB 인스턴스에서 연결된 DB 파라미터 그룹에 대한 최신 변경 내용을 사용하고 있지 않은 경우 AWS Management 콘솔에 DB 파라미터 그룹이 재시작 보류중 상태로 표시됩니다. 재시작 보류중 파라미터 그룹 상태로 인해 다음번 유지 관리 기간 중에 자동 재부팅이 되지는 않습니다. 최신 파라미터 변경 내용을 이 DB 인스턴스에 적용하려면 해당 DB 인스턴스를 수동으로 재부팅해야 합니다. 파라미터 그룹에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

DB 인스턴스를 재부팅하면 데이터베이스 엔진 서비스가 재시작됩니다. DB 인스턴스를 재부팅하면 DB 인스턴스 상태가 rebooting으로 설정되면서 잠시 중단됩니다.

Amazon RDS 인스턴스가 다중 AZ로 구성되어 있는 경우에는 장애 조치로 재부팅을 실행할 수 있습니다. 재부팅이 완료되면 Amazon RDS 이벤트가 생성됩니다. DB 인스턴스가 다중 AZ 배포인 경우 재부팅하면 한 가용 영역(AZ)의 장애 조치를 다른 가용 영역에 강제 실행할 수 있습니다. DB 인스턴스의 장애 조치를 강제로 실행하면 Amazon RDS가 다른 가용 영역의 대기 복제본으로 자동 전환되고, DB 인스턴스가 예비 DB 인스턴스를 가리키도록 DNS 레코드를 업데이트합니다. 결과적으로 기존의 DB 인스턴스 연결을 모두 삭제한 후 재구성해야 합니다. 장애 조치를 사용한 재부팅은 DB 인스턴스 결함을 시뮬레이션하여 테스트하거나, 장애 조치 이후 원래 AZ로 작업을 복구할 때 유용한 기능입니다. 자세한 정보는 [Amazon RDS를 위한 고가용성\(다중 AZ\) \(p. 109\)](#) 단원을 참조하십시오.

Note

재부팅할 때 한 가용 영역에서 다른 가용 영역으로 장애 조치를 강제로 실시하면 가용 영역 변경 내용이 몇 분 동안 AWS Management 콘솔과 AWS CLI 및 RDS에 대한 호출에 반영되지 않을 수 있습니다.

사용 가능 상태가 아닌 경우 DB 인스턴스를 재부팅할 수 없습니다. 백업이 진행 중이거나 이전에 수정을 요청했거나 유지 관리 기간 작업 등 여러 원인으로 인해 데이터베이스를 사용할 수 없습니다.

DB 인스턴스를 재부팅하는 데 걸리는 시간은 특정 데이터베이스 엔진의 충돌 복구 프로세스에 따라 다릅니다. 따라서 재부팅 시간을 단축하려면 재부팅 프로세스에서 데이터베이스 작업을 최소화하는 것이 좋습니다. 데이터베이스 작업을 줄이면 중간 트랜잭션의 롤백 작업이 줄어듭니다.

AWS Management 콘솔

DB 인스턴스를 재부팅하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 재부팅하려는 DB 인스턴스를 선택합니다.
3. 작업에서 재부팅을 선택합니다.
[Reboot DB Instance] 페이지가 나타납니다.
4. (선택 사항) 한 AZ에서 다른 AZ로 장애 조치를 강제로 실행하려면 Reboot with failover(장애 조치로 재부팅하시겠습니까?)을 선택합니다.
5. DB 인스턴스를 재부팅하려면 [Reboot]를 선택합니다.
또는 [Cancel]을 선택합니다.

CLI

AWS CLI를 사용하여 DB 인스턴스를 재부팅하려면 `reboot-db-instance` 명령을 호출하십시오.

Example 간편한 재부팅

Linux, OS X, Unix의 경우:

```
aws rds reboot-db-instance \
--db-instance-identifier mydbinstance
```

Windows의 경우:

```
aws rds reboot-db-instance ^
--db-instance-identifier mydbinstance
```

Example 장애 조치로 재부팅

한 곳의 AZ에서 나머지 AZ로 장애 조치를 강제로 실행하려면 `--force-failover` 파라미터를 사용합니다.

Linux, OS X, Unix의 경우:

```
aws rds reboot-db-instance \
--db-instance-identifier mydbinstance \
--force-failover
```

Windows의 경우:

```
aws rds reboot-db-instance ^
--db-instance-identifier mydbinstance ^
--force-failover
```

API

Amazon RDS API를 사용하여 DB 인스턴스를 재부팅하려면 `RebootDBInstance` 작업을 호출하십시오.

Example 간편한 재부팅

```
https://rds.amazonaws.com/
```

```
?Action=RebootDBInstance
&DBInstanceIdentifier=mydbinstance
&Version=2014-10-31
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20131016/us-west-1/rds/aws4_request
&X-Amz-Date=20131016T233051Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=087a8eb41cb1ab5f99e81575f23e73757ffc6a1e42d7d2b30b9cc0be988cff97
```

Example 장애 조치로 재부팅

한 곳의 AZ에서 나머지 AZ로 장애 조치를 강제로 실행하려면 `ForceFailover` 파라미터를 `true`로 설정하십시오.

```
https://rds.amazonaws.com/
?Action=RebootDBInstance
&DBInstanceIdentifier=mydbinstance
&ForceFailover=true
&Version=2014-10-31
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20131016/us-west-1/rds/aws4_request
&X-Amz-Date=20131016T233051Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=087a8eb41cb1ab5f99e81575f23e73757ffc6a1e42d7d2b30b9cc0be988cff97
```

Amazon RDS DB 인스턴스의 일시적 중지

임시 테스트나 하루 단위의 개발 작업 등 DB 인스턴스를 간헐적으로 사용하는 경우에는 Amazon RDS DB 인스턴스를 일시적으로 중지하여 비용을 일시적으로 절약할 수 있습니다. DB 인스턴스가 중지되어 있을 때는 프로비저닝된 스토리지(프로비저닝된 IOPS 포함)와 백업 스토리지(지정한 유지 기간 내 수동 스냅샷과 자동 스냅샷 포함)에 대한 비용만 부과되며, DB 인스턴스 시간에 대해서는 비용이 부과되지 않습니다. 자세한 내용은 [결제 FAQ](#) 단원을 참조하십시오.

Note

경우에 따라 DB 인스턴스를 중지하는 데 긴 시간이 필요합니다. DB 인스턴스를 중지하고 즉시 다시 시작하려면 DB 인스턴스를 재부팅하십시오. DB 인스턴스 재부팅에 대한 자세한 내용은 [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#) 단원을 참조하십시오.

다음 엔진을 실행하는 DB 인스턴스를 중지하고 시작할 수 있습니다.

- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

DB 인스턴스 중지 및 시작은 모든 DB 인스턴스 클래스 및 모든 AWS 리전에서 지원됩니다.

다중 AZ 배포를 지원하는 데이터베이스 엔진의 경우, 단일 가용 영역으로 구성되었든 다중 AZ로 구성되었든 상관없이 DB 인스턴스를 중지하고 시작할 수 있습니다. 다중 AZ 구성에서는 Amazon RDS for SQL Server DB 인스턴스를 중지할 수 있습니다.

DB 인스턴스를 중지할 경우 DB 인스턴스가 일반적인 종료 과정을 수행하고 실행을 중지합니다. DB 인스턴스 상태가 `stopping`으로 바뀌었다가 다시 `stopped`로 바뀝니다. 모든 스토리지 볼륨이 DB 인스턴스에 연결된 상태로 유지되고 해당 데이터도 남습니다. DB 인스턴스의 RAM에 저장된 데이터는 삭제됩니다.

DB 인스턴스는 최대 7일간 중지할 수 있습니다. 7일이 지날 때까지 DB 인스턴스를 수동으로 시작하지 않으면 DB 인스턴스가 자동으로 시작됩니다.

장점

DB 인스턴스를 중지했다가 다시 시작하는 것이 DB 스냅샷을 생성하여 복원하는 것보다 빠릅니다.

DB 인스턴스를 중지하더라도 인스턴스 ID, 도메인 이름 서버(DNS) 엔드포인트, 파라미터 그룹, 보안 그룹 및 옵션 그룹은 유지됩니다. DB 인스턴스를 다시 시작할 때는 중지했을 때와 동일한 구성으로 시작됩니다. 또한 DB 인스턴스를 중지할 경우 Amazon RDS에서 Amazon Simple Storage Service(Amazon S3) 트랜잭션 로그가 보존되므로 필요한 경우 특정 시점 복원을 수행할 수 있습니다.

제한 사항

DB 인스턴스를 중지했다가 다시 시작할 때는 다음과 같은 몇 가지 제약이 따릅니다.

- 읽기 전용 복제본을 포함한 또는 읽기 전용 복제본인 DB 인스턴스는 중지할 수 없습니다.
- 다중 AZ 구성에서는 Amazon RDS for SQL Server DB 인스턴스를 중지할 수 있습니다.
- 중지된 DB 인스턴스는 수정할 수 없습니다.
- 중지된 DB 인스턴스와 연결된 옵션 그룹은 삭제할 수 없습니다.
- 중지된 DB 인스턴스에 연결된 DB 파라미터 그룹을 삭제할 수 없습니다.

옵션 및 파라미터 그룹 고려 사항

옵션 그룹에 연결된 DB 인스턴스가 있는 경우 해당 옵션 그룹에서 지속적 옵션(영구 옵션 포함)을 제거할 수 없습니다. 이 기능은 상태가 `stopping`, `stopped` 또는 `starting`인 DB 인스턴스에서도 마찬가지입니다.

중지된 DB 인스턴스와 연결된 옵션 그룹 또는 DB 파라미터 그룹을 변경할 수는 있지만 변경 내용이 다음에 DB 인스턴스를 시작할 때까지는 적용되지 않습니다. 변경 사항을 즉시 적용하도록 선택한 경우에는 DB 인스턴스를 시작할 때 변경됩니다. 그렇지 않을 경우 중지된 DB 인스턴스가 시작된 후 다음 유지 관리 기간에 변경 사항이 적용됩니다.

VPC 고려 사항

DB 인스턴스를 중지하더라도 DNS 엔드포인트는 유지됩니다. Amazon Virtual Private Cloud(Amazon VPC)에 속하지 않는 DB 인스턴스를 중지한 경우에는 Amazon RDS가 DB 인스턴스의 IP 주소를 해제합니다. 반대로 VPC에 속하는 DB 인스턴스를 중지하면 DB 인스턴스가 IP 주소를 그대로 유지합니다.

Note

DB 인스턴스에는 항상 IP 주소가 아니라 DNS 엔드포인트를 사용하여 연결해야 합니다.

AWS Management 콘솔

DB 인스턴스를 중지하려면

- AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
- 탐색 창에서 데이터베이스를 선택한 후 중지하려는 DB 인스턴스를 선택합니다.
- Actions(작업)에서 Stop(중지)를 선택합니다.
- (선택 사항) Stop DB Instance(DB 인스턴스 중지) 창에서 Create Snapshot?(스냅샷을 생성하시겠습니까?)에 Yes(예)를 선택하고 Snapshot name(스냅샷 이름)에 스냅샷 이름을 입력합니다. DB 인스턴스를 중단하기 전에 DB 인스턴스의 스냅샷을 생성하려면 예를 선택합니다.
- DB 인스턴스를 중지하려면 예, 지금 중지합니다.를 선택합니다. 또는 작업을 취소하려면 취소를 선택합니다.

CLI

AWS CLI를 사용하여 DB 인스턴스를 중지하려면 다음 파라미터와 함께 [stop-db-instance](#) 명령을 호출하십시오.

- `--db-instance-identifier` – DB 인스턴스의 이름입니다.

Example

```
stop-db-instance --db-instance-identifier mydbinstance
```

API

Amazon RDS API를 사용하여 DB 인스턴스를 중지하려면 다음 파라미터와 함께 [StopDBInstance](#) 작업을 호출하십시오.

- `DBInstanceIdentifier` – DB 인스턴스의 이름입니다.

이전에 중지된 Amazon RDS DB 인스턴스 시작

Amazon RDS DB 인스턴스는 일시적으로 중지하여 비용을 절약할 수 있습니다. 중지한 DB 인스턴스는 재시작하여 다시 사용할 수 있습니다. DB 인스턴스의 중지 및 시작에 대한 자세한 내용은 [Amazon RDS DB 인스턴스의 일시적 중지](#) (p. 125) 단원을 참조하십시오.

이전에 중지한 DB 인스턴스를 다시 시작할 때 DB 인스턴스의 ID, DNS 엔드포인트, 파라미터 그룹, 보안 그룹 및 옵션 그룹도 그대로 유지됩니다. 중지된 인스턴스를 시작하면 인스턴스 시간당 요금이 부과됩니다.

AWS Management 콘솔

DB 인스턴스를 시작하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 후 시작하려는 DB 인스턴스를 선택합니다.
3. Actions(작업)에는 Start(시작)을 선택합니다.

CLI

AWS CLI를 사용하여 DB 인스턴스를 시작하려면 다음 파라미터와 함께 [start-db-instance](#) 명령을 호출하십시오.

- `--db-instance-identifier` – DB 인스턴스의 이름입니다.

Example

```
start-db-instance --db-instance-identifier mydbinstance
```

API

Amazon RDS API를 사용하여 DB 인스턴스를 시작하려면 다음 파라미터와 함께 [StartDBInstance](#) 작업을 호출하십시오.

- **DBInstanceIdentifier** – DB 인스턴스의 이름입니다.

Example

```
https://rds.amazonaws.com/  
    ?Action=StartDBInstance  
    &DBInstanceIdentifier=mydbinstance  
    &SignatureMethod=HmacSHA256  
    &SignatureVersion=4  
    &Version=2014-10-31  
    &X-Amz-Algorithm=AWS4-HMAC-SHA256  
    &X-Amz-Credential=AKIADQKE4SARGYLE/20131016/us-west-1/rds/aws4_request  
    &X-Amz-Date=20131016T233051Z  
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date  
    &X-Amz-Signature=087a8eb41cb1ab5f99e81575f23e73757fffc6a1e42d7d2b30b9cc0be988cff97
```

관련 주제

- [DB 인스턴스 삭제 \(p. 128\)](#)
- [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#)

DB 인스턴스 삭제

DB 인스턴스를 삭제하려면 다음을 수행해야 합니다.

- 인스턴스 이름을 제공합니다.
- 인스턴스의 최종 DB 스냅샷을 생성하는 옵션을 활성화하거나 비활성화합니다.
- 자동 백업을 보관하는 옵션을 활성화하거나 비활성화합니다.

삭제 방지가 활성화되지 않은 인스턴스만 삭제할 수 있습니다. DB 인스턴스를 생성하거나 수정하는 경우 사용자가 DB 인스턴스를 삭제할 수 없도록 삭제 방지를 활성화하는 옵션이 있습니다. 삭제 방지는 AWS CLI 및 API 명령 사용 시 기본적으로 비활성화됩니다. AWS Management 콘솔을 사용하여 프로덕션 DB 인스턴스를 생성할 경우 삭제 방지가 기본적으로 활성화됩니다. 하지만 Amazon RDS는 콘솔, CLI 또는 API를 사용하여 DB 인스턴스를 삭제할 때 삭제 방지를 설정합니다. 삭제 보호가 활성화된 DB 인스턴스를 삭제하려면 먼저 인스턴스를 수정하고 삭제 보호를 비활성화합니다.

삭제하려는 DB 인스턴스에 읽기 전용 복제본이 있는 경우 읽기 전용 복제본을 승격하거나 삭제해야 합니다. 자세한 내용은 [읽기 전용 복제본을 득립 DB 인스턴스로 승격 \(p. 139\)](#) 단원을 참조하십시오.

최종 스냅샷 생성 및 자동 백업 보관

DB 인스턴스를 삭제할 때는 DB 인스턴스의 최종 스냅샷 생성 여부를 선택할 수 있습니다. 또한 DB 인스턴스가 삭제된 후도 자동 백업을 보관하도록 선택할 수 있습니다. 나중에 DB 인스턴스를 복원하려면 최종 스냅샷을 생성하거나 자동 백업을 보관합니다.

	최종 스냅샷 포함	최종 스냅샷 제외	자동 백업 보관
선택 방법	삭제된 DB 인스턴스를 나중에 복원하고 싶다면 최종 DB 스냅샷을 생성합니다.	DB 인스턴스를 바로 삭제하려면 최종 DB 스냅샷 생성 단계를 건너뛸 수 있습니다.	스냅샷을 생성하는 대신에 DB 인스턴스를 삭제할 때 자동 백업 보관을 활성화하도록 선택할 수 있습니다. 이러한 백업에는 여전히 DB 인스턴스

	최종 스냅샷 포함	최종 스냅샷 제외	자동 백업 보관
		<p>Important</p> <p>스냅샷을 생략할 경우 DB 인스턴스를 복원하려면 다음 중 하나가 필요합니다.</p> <ul style="list-style-type: none"> • DB 인스턴스의 이전 수동 스냅샷을 사용하여 해당 스냅샷의 시점으로 DB 인스턴스를 복원해야 합니다. • 자동 백업을 보관하도록 선택해야 합니다. 다음을 사용하여 보존 기간 내 특정 시점으로 복원할 수 있습니다. 	의 보존 기간이 적용되며 시스템 스냅샷과 동일한 방식으로 경과 일수가 계산됩니다.
자동 백업	자동 백업 보관을 활성화하지 않는 한, 모든 자동 백업은 삭제되고 나면 복구할 수 없습니다.	DB 인스턴스를 삭제할 때 자동 백업을 보관하도록 선택하지 않는 한, 모든 자동 백업은 삭제되고 나면 복구할 수 없습니다.	최종 스냅샷을 생성하도록 선택했는지 여부와 관계 없이 설정된 기간 동안 자동 백업이 보관됩니다. 자동 백업은 삭제한 시점에 DB 인스턴스에 설정된 보존 기간 동안 보관됩니다.
수동 스냅샷 수	이전 수동 스냅샷은 삭제되지 않습니다.	이전 수동 스냅샷은 삭제되지 않습니다.	스냅샷은 삭제되지 않습니다.

DB 인스턴스가 `creating`, `failed`, `incompatible-restore`, `incompatible-network` 상태일 때는 최종 스냅샷을 생성할 수 없습니다. DB 인스턴스 상태에 대한 자세한 내용은 [DB 인스턴스 상태 \(p. 96\)](#) 단원을 참조하십시오.

콘솔, CLI, API를 사용하여 DB 인스턴스 삭제

AWS Management 콘솔, AWS CLI 또는 RDS API를 사용하여 DB 인스턴스를 삭제할 수 있습니다.

콘솔

DB 인스턴스를 삭제하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 후 삭제하려는 DB 인스턴스를 선택합니다.
3. [Actions]에 대해 [Delete]를 선택합니다.
4. 최종 스냅샷 생성 여부에서 예 또는 아니요를 선택합니다.
5. 이전 단계에서 예를 선택한 경우, 최종 스냅샷 이름에 최종 DB 스냅샷 이름을 입력합니다.
6. 자동 백업을 보관하려면 자동 백업 보관을 선택합니다.
7. 상자에 `delete me`를 입력합니다.
8. 삭제를 선택합니다.

AWS CLI

AWS CLI를 사용하여 DB 인스턴스를 삭제하려면 다음 옵션과 함께 `delete-db-instance` 명령을 호출하십시오.

- `--db-instance-identifier`
- `--final-db-snapshot-identifier` 또는 `--skip-final-snapshot`

Example 최종 스냅샷 사용, 자동 백업 보관 안 함

Linux, OS X, Unix의 경우:

```
aws rds delete-db-instance \
    --db-instance-identifier mydbinstance \
    --final-db-snapshot-identifier mydbinstancefinalsnapshot \
    --delete-automated-backups
```

Windows의 경우:

```
aws rds delete-db-instance ^
    --db-instance-identifier mydbinstance ^
    --final-db-snapshot-identifier mydbinstancefinalsnapshot ^
    --delete-automated-backups
```

Example 자동 백업 보관, 최종 스냅샷 없음

Linux, OS X, Unix의 경우:

```
aws rds delete-db-instance \
    --db-instance-identifier mydbinstance \
    --skip-final-snapshot \
    --no-delete-automated-backups
```

Windows의 경우:

```
aws rds delete-db-instance ^
    --db-instance-identifier mydbinstance ^
    --skip-final-snapshot ^
    --no-delete-automated-backups
```

RDS API

Amazon RDS API를 사용하여 DB 인스턴스를 삭제하려면 다음 파라미터와 함께 `DeleteDBInstance` 작업을 호출하십시오.

- `DBInstanceIdentifier`
- `FinalDBSnapshotIdentifier` 또는 `SkipFinalSnapshot`

Example 최종 스냅샷 사용, 자동 백업 보관 안 함

```
https://rds.amazonaws.com/
    ?Action=DeleteDBInstance
    &DBInstanceIdentifier=mydbinstance
    &FinalDBSnapshotIdentifier=mydbinstancefinalsnapshot
    &DeleteAutomatedBackups=true
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2014-10-31
    &X-Amz-Algorithm=AWS4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20140305/us-west-1/rds/aws4_request
    &X-Amz-Date=20140305T185838Z
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
```

&X-Amz-Signature=b441901545441d3c7a48f63b5b1522c5b2b37c137500c93c45e209d4b3a064a3

Example 자동 백업 보관, 최종 스냅샷 없음

```
https://rds.amazonaws.com/
?Action=DeleteDBInstance
&DBInstanceIdentifier=mydbinstance
&SkipFinalSnapshot=true
&DeleteAutomatedBackups=false
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-10-31
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140305/us-west-1/rds/aws4_request
&X-Amz-Date=20140305T185838Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=b441901545441d3c7a48f63b5b1522c5b2b37c137500c93c45e209d4b3a064a3
```

Amazon RDS 리소스에 태그 지정

Amazon RDS 태그를 사용하여 Amazon RDS 리소스에 메타데이터를 추가할 수 있습니다. 또한 IAM 정책과 함께 이러한 태그를 사용하여 Amazon RDS에 대한 액세스를 관리하고 Amazon RDS 리소스에 적용 가능한 작업을 제어할 수 있습니다. 마지막으로 비슷하게 태그가 지정된 리소스에 대한 비용을 그룹화하여 이러한 태그로 비용을 추적할 수 있습니다.

모든 Amazon RDS 리소스에 태깅할 수 있습니다.

- DB 인스턴스
- DB 클러스터
- 읽기 전용 복제본
- DB 스냅샷
- DB 클러스터 스냅샷
- 예약 DB 인스턴스
- 이벤트 구독
- DB 옵션 그룹
- DB 파라미터 그룹
- DB 클러스터 파라미터 그룹
- DB 보안 그룹
- DB 서브넷 그룹

IAM 정책으로 태그가 지정된 리소스 액세스 관리에 대한 자세한 내용은 [인증 및 액세스 제어 \(p. 339\)](#) 단원을 참조하십시오.

Amazon RDS 리소스 태그 개요

Amazon RDS 태그는 사용자가 정의하고 Amazon RDS 리소스와 연결하는 이름-값 페어입니다. 이 이름을 키라고 합니다. 키 값을 제공하는 것은 선택 사항입니다. 태그를 사용하여 Amazon RDS 리소스에 임의의 정보를 배정할 수 있습니다. 범주 정의 등에 태그 키를 사용할 수 있으며 태그 값은 해당 범주의 항목일 수 있습니다. 예를 들어, 태그 키를 "project"로 정의하고 태그 값을 "Salix"로 정의하여 Amazon RDS 리소스가 Salix project에 배정됨을 나타냅니다. environment=test나 environment =production 등의 키를 사용하여 태그로 Amazon RDS 리소스를 테스트나 프로덕션에 사용되는 것으로 지정할 수도 있습니다. Amazon RDS 리소스 와 연결된 메타데이터를 더 쉽게 추적할 수 있게 일관성 있는 태그 키 세트를 사용하는 것이 좋습니다.

태그를 사용하여 비용 구조를 반영하도록 AWS 대금을 구성하십시오. 이렇게 하려면 가입하여 태그 키 값이 포함된 AWS 계정을 가져옵니다. 그런 다음 같은 태그 키 값을 가진 리소스에 따라 결제 정보를 구성하여 리소스 비용의 합을 볼 수 있습니다. 예를 들어, 특정 애플리케이션 이름으로 여러 리소스에 태그를 지정한 다음 결제 정보를 구성하여 여러 서비스에 걸친 해당 애플리케이션의 총 비용을 볼 수 있습니다. 자세한 내용은 About AWS Billing and Cost Management의 [Cost Allocation and Tagging](#)을 참조하십시오.

각 Amazon RDS 리소스에는 해당 Amazon RDS 리소스에 배정되는 모든 태그를 포함하는 태그 세트가 있습니다. 태그 세트는 최대 10개의 태그를 포함하거나 비어 있을 수 있습니다. Amazon RDS 리소스의 기존 태그와 동일한 키를 갖는 태그를 리소스에 추가하면 새 값이 이전 값을 덮어씁니다.

AWS에서는 태그에 의미론적 의미를 적용하지 않습니다. 태그는 엄격히 문자열로 해석됩니다. Amazon RDS에서는 사용자가 리소스를 만들 때 사용하는 설정에 따라 DB 인스턴스 또는 기타 Amazon RDS 리소스에서 태그를 설정할 수 있습니다. 예를 들어 Amazon RDS에서 DB 인스턴스가 프로덕션용인지, 아니면 테스트용인지를 나타내는 태그를 추가할 수 있습니다.

- 태그 키는 태그의 필수 이름입니다. 문자열 값은 길이가 1~128자(유니코드 문자)이며 "aws:" 또는 "rds:"로 시작할 수 없습니다. 문자열에는 유니코드 문자 집합, 숫자, 공백, '_', '.', ',', '=', '+', '-' '@'(Java regex: "^([\\p{L}\\p{Z}]\\p{N}_.:/=+\\-]*\$")만 포함할 수 있습니다.
 - 태그 값은 태그의 선택적 문자열 값입니다. 문자열 값은 길이가 1~256자(유니코드 문자)이며 "aws:"로 시작할 수 없습니다. 문자열에는 유니코드 문자 집합, 숫자, 공백, '_', '.', ',', '=', '+', '-' '@'(Java regex: "^(\\p{L}\\p{Z})\\p{N}_.:/=+\\-]*\$")만 포함할 수 있습니다.
- 값은 태그 세트에서 고유할 필요는 없으며 null일 수 있습니다. 예를 들어, project/Trinity 및 cost-center/Trinity의 태그 세트에 키-값 페어가 있을 수 있습니다.

Note

스냅샷에 태그를 추가할 수 있지만 이 그룹화는 청구서에 반영되지 않습니다.

AWS Management 콘솔, 명령줄 인터페이스 또는 Amazon RDS API를 사용하여 Amazon RDS 리소스에서 태그를 추가, 나열 및 삭제할 수 있습니다. 명령줄 인터페이스나 Amazon RDS API를 사용할 때는 작업하려는 Amazon RDS 리소스에 대한 Amazon 리소스 이름(ARN)을 제공해야 합니다. ARN 생성에 대한 자세한 내용은 [Amazon RDS의 ARN 구성 \(p. 173\)](#) 주제단원을 참조하십시오.

권한 부여 목적으로 태그가 캐시됩니다. 이 때문에 Amazon RDS 리소스의 태그에 대한 추가나 업데이트가 제공되는 데 몇 분 정도 걸릴 수 있습니다.

태그 복사

DB 인스턴스를 만들거나 복원할 경우 DB 인스턴스의 태그가 DB 인스턴스의 스냅샷으로 복사되도록 지정할 수 있습니다. 태그를 복사하면 DB 스냅샷의 메타데이터가 원본 DB 인스턴스의 메타데이터와 일치하고, DB 스냅샷의 액세스 정책 또한 원본 DB 인스턴스의 액세스 정책과 일치합니다. 태그는 기본적으로 복사되지 않습니다.

다음 작업 시 DB 스냅샷으로 태그를 복사하도록 지정할 수 있습니다.

- DB 인스턴스 생성
- DB 인스턴스 복원
- 읽기 전용 복제본 생성
- DB 스냅샷 복사

Note

`create-db-snapshot` AWS CLI 명령의 `--tag-key` 파라미터에 값을 포함시킬 경우(또는 `CreateDBSnapshot` API 작업에 한 개 이상의 태그를 제공할 경우) RDS는 원본 DB 인스턴스

의 태그를 새 DB 스냅샷으로 복사하지 않습니다. 이 기능은 원본 DB 인스턴스에 --copy-tags-to-snapshot(CopyTagsToSnapshot) 옵션이 활성화되어 있어도 적용됩니다. 이 방법을 사용할 경우 새로운 DB 인스턴스에 적용되지 않는 태그를 추가하지 않고 DB 스냅샷으로부터 DB 인스턴스의 사본을 만들 수 있습니다. AWS CLI create-db-snapshot 명령(또는 CreateDBSnapshot Amazon RDS API 작업)을 사용하여 DB 스냅샷을 생성한 경우 이 주제의 뒷 부분에서 설명한 대로 태그를 추가할 수 있습니다.

AWS Management 콘솔

Amazon RDS 리소스에 태그를 지정하는 프로세스는 모든 리소스에서 비슷합니다. 다음 절차에서는 Amazon RDS DB 인스턴스에 태그를 지정하는 방법을 보여줍니다.

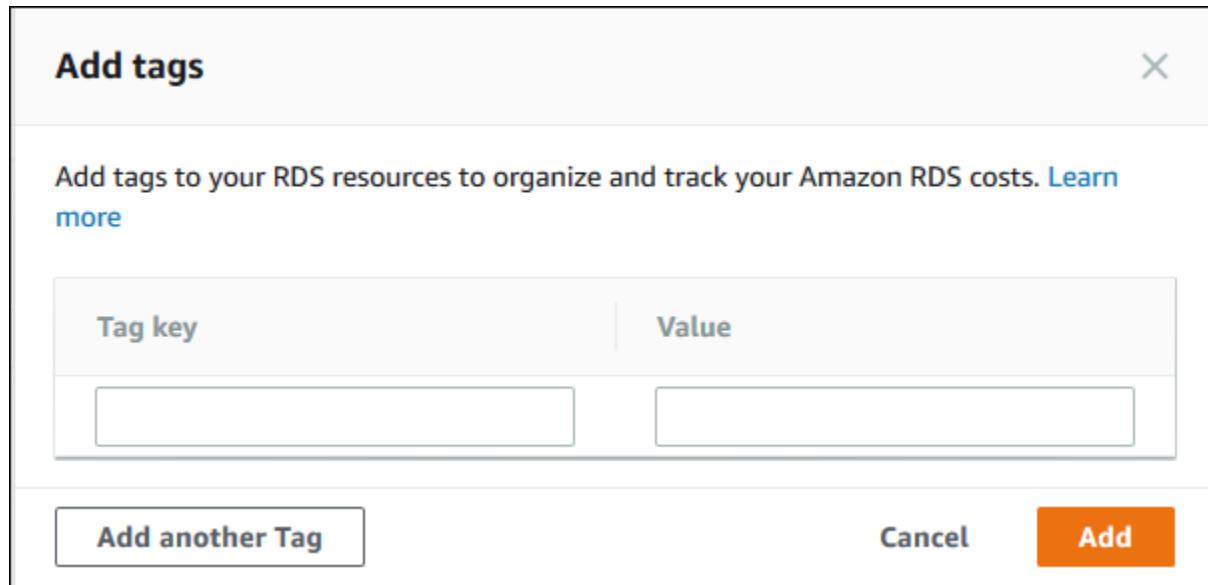
DB 인스턴스에 태그를 추가하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.

Note

DB 인스턴스 목록을 필터링하려면 데이터베이스 창의 Filter databases(데이터베이스 필터링)에 텍스트 문자열을 입력합니다. 해당 문자열을 포함하는 DB 인스턴스만 표시됩니다.

3. 세부 정보를 보기 위해 태그 지정하려는 DB 인스턴스의 이름을 선택합니다.
4. 세부 정보 섹션에서 아래에 있는 태그 섹션으로 스크롤합니다.
5. 추가를 선택합니다. 태그 추가 창이 나타납니다.



6. 태그 키와 값에 값을 입력합니다.
7. 다른 태그를 추가하려면 다른 태그 추가를 선택하고 태그 키와 값에 값을 입력합니다.
이 단계를 필요한 만큼 반복합니다.
8. 추가를 선택합니다.

DB 인스턴스에서 태그를 삭제하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.

2. 탐색 창에서 데이터베이스를 선택합니다.

Note

DB 인스턴스 목록을 필터링하려면 데이터베이스 창의 Filter databases(데이터베이스 필터링) 상자에 텍스트 문자열을 입력합니다. 해당 문자열을 포함하는 DB 인스턴스만 표시됩니다.

3. 세부 정보를 표시할 DB 인스턴스의 이름을 선택합니다.
4. 세부 정보 섹션에서 아래에 있는 태그 섹션으로 스크롤합니다.
5. 삭제하려는 태그를 선택합니다.

The screenshot shows the 'Tags (1)' section of the AWS RDS console. It has three buttons: 'Edit', 'Delete', and 'Add'. Below is a search bar labeled 'Filter tag key'. A table lists one tag: 'workload-type' with value 'other'. The 'workload-type' row is highlighted with a blue border.

6. Delete(삭제)를 선택한 다음 Delete tags(삭제 태그)창에서 Delete(삭제)를 선택합니다.

CLI

AWS CLI 사용을 통해 DB 인스턴스에 대한 태그를 추가, 나열 또는 제거할 수 있습니다.

- Amazon RDS 리소스에 하나 이상의 태그를 추가하려면 AWS CLI 명령 [add-tags-to-resource](#)를 사용합니다.
- Amazon RDS 리소스의 태그를 나열하려면 AWS CLI 명령 [list-tags-for-resource](#)를 사용합니다.
- Amazon RDS 리소스에서 하나 이상의 태그를 삭제하려면 AWS CLI 명령 [remove-tags-from-resource](#)를 사용합니다.

필수 ARN을 생성하는 방법에 대해 자세히 알아보려면 [Amazon RDS의 ARN 구성 \(p. 173\)](#) 단원을 참조하십시오.

API

Amazon RDS API를 사용하여 DB 인스턴스에 대한 태그를 추가, 나열 또는 제거할 수 있습니다.

- Amazon RDS 리소스에 태그를 추가하려면 [AddTagsToResource](#) 작업을 사용합니다.
- Amazon RDS 리소스에 배정된 태그를 나열하려면 [ListTagsForResource](#)를 사용합니다.
- Amazon RDS 리소스에서 태그를 제거하려면 [RemoveTagsFromResource](#) 작업을 사용합니다.

필수 ARN을 생성하는 방법에 대해 자세히 알아보려면 [Amazon RDS의 ARN 구성 \(p. 173\)](#) 단원을 참조하십시오.

Amazon RDS API를 사용한 XML 작업 시 다음 스키마를 사용하십시오.

```
<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Trinity</Value>
```

```

</Tag>
<Tag>
  <Key>User</Key>
  <Value>Jones</Value>
</Tag>
</TagSet>
</Tagging>

```

다음 표에는 허용되는 XML 태그와 해당 특성의 목록이 나와 있습니다. Key 및 Value 값은 대/소문자를 구분합니다. 예를 들어, project=Trinity와 PROJECT=Trinity는 서로 다른 두 개의 태그입니다.

태그 지정 요소	설명
TagSet	태그 세트에는 Amazon RDS 리소스에 배정된 모든 태그가 포함됩니다. 리소스당 하나의 태그 세트만 있을 수 있습니다. Amazon RDS API를 통해서만 TagSet로 작업합니다.
Tag	태그는 사용자가 정의하는 키-값 페어입니다. 태그 세트에 1~50개의 태그가 있을 수 있습니다.
키	키는 태그의 필수 이름입니다. 문자열 값은 길이가 1~128자(유니코드 문자)이며 "rds:" 또는 "aws:"로 시작할 수 없습니다. 문자열에는 유니코드 문자, 숫자, 공백, '_', '.', '/', '=', '+', '-'(Java regex: "^([\p{L}]\p{Z})\p{N}...:=+\ -]*\$")만 포함될 수 있습니다. 키는 태그 집합에 대해 고유해야 합니다. 예를 들어, 태그 세트에 project/Trinity 와 project/Xanadu처럼 키는 같지만 값은 다른 키-페어가 있을 수 없습니다.
값	값은 태그의 선택적 값입니다. 문자열 값은 길이가 1~256자(유니코드 문자)이며 "rds:" 또는 "aws:"로 시작할 수 없습니다. 문자열에는 유니코드 문자, 숫자, 공백, '_', '.', '/', '=', '+', '-'(Java regex: "^([\p{L}]\p{Z})\p{N}...:=+\ -]*\$")만 포함될 수 있습니다. 값은 태그 세트에서 고유할 필요는 없으며 null일 수 있습니다. 예를 들어, project/Trinity 및 cost-center/Trinity의 태그 세트에 키-값 페어가 있을 수 있습니다.

읽기 전용 복제본 작업

Amazon RDS는 MariaDB, MySQL, Oracle 및 PostgreSQL DB 엔진의 기본 복제 기능을 사용하여 원본 DB 인스턴스의 읽기 전용 복제본이라고 하는 특수한 유형의 DB 인스턴스를 만들 수 있습니다. 원본 DB 인스턴스에 적용된 변경 사항은 읽기 전용 복제본에도 동시에 적용됩니다. 애플리케이션에서 읽기 전용 복제본으로 읽기 쿼리를 라우팅하여 원본 DB 인스턴스의 로드에 대한 부하를 줄일 수 있습니다. 읽기 전용 복제본을 이용하면 읽기 중심의 데이터베이스 워크로드를 처리하기 위해 단일 DB 인스턴스의 용량 제한을 탄력적으로 늘릴 수 있습니다.

Note

다음 정보는 원본 DB 인스턴스와 동일한 AWS 리전이나 별도의 AWS 리전에 Amazon RDS 읽기 전용 복제본을 생성할 때도 똑같이 적용됩니다. 하지만 Amazon EC2 인스턴스에서 실행되거나 온프레미스인 인스턴스로 복제를 설정할 때는 적용되지 않습니다.

읽기 전용 복제본을 생성할 때는 먼저 기존 DB 인스턴스를 원본으로 지정합니다. 그러면 Amazon RDS가 원본 인스턴스의 스냅샷을 캡처한 후 이 스냅샷을 이용해 읽기 전용 인스턴스를 생성합니다. 이후부터는 원본 DB 인스턴스를 변경할 때마다 Amazon RDS가 DB 엔진에 비동기식 복제 방법을 사용하여 읽기 전용 복제본

을 업데이트합니다. 읽기 전용 복제본은 읽기 전용 연결만을 허용하는 DB 인스턴스로 작동합니다. 애플리케이션은 DB 인스턴스에 연결되는 방식과 동일하게 읽기 전용 복제본에 연결됩니다. Amazon RDS는 원본 DB 인스턴스의 데이터베이스를 모두 복제합니다.

경우에 따라 읽기 전용 복제본이 원본 DB 인스턴스와 다른 AWS 리전에 있습니다. 이 경우 Amazon RDS가 원본 DB 인스턴스와 읽기 전용 복제본 사이에 보안 통신 채널을 설정합니다. Amazon RDS는 보안 그룹 항목을 추가하는 것과 같이 보안 채널을 활성화하는 데 필요한 AWS 보안 구성은 설정합니다.

MariaDB, MySQL, Oracle 및 PostgreSQL 엔진은 읽기 전용 복제본을 지원합니다. 이 단원에서는 이러한 엔진 모두에서 읽기 전용 복제본을 사용하는 것에 대한 일반적인 정보를 제공합니다. 특정 엔진에서 읽기 전용 복제본을 사용하는 것에 대한 자세한 내용은 다음 단원을 참조하십시오.

- [MySQL 읽기 전용 복제본 작업 \(p. 644\)](#)
- [MariaDB 읽기 전용 복제본 작업 \(p. 461\)](#)
- [Amazon RDS의 Oracle 읽기 전용 복제본 작업 \(p. 768\)](#)
- [PostgreSQL 읽기 전용 복제본 작업 \(p. 967\)](#)

Amazon RDS 읽기 전용 복제본의 개요

임의의 원본 DB 인스턴스에 하나 이상의 읽기 전용 복제본을 배포하는 경우는 다음을 비롯해 다양한 시나리오에서 적용될 수 있습니다.

- 읽기 중심의 데이터베이스 워크로드를 위해 단일 DB 인스턴스의 컴퓨팅 파워 또는 I/O 용량을 확장합니다. 이 과도한 읽기 트래픽을 하나 이상의 읽기 전용 복제본으로 이동할 수 있습니다.
- 원본 DB 인스턴스를 사용할 수 없는 동안 읽기 트래픽을 처리합니다. 경우에 따라 백업 또는 예약된 유지 관리를 위한 I/O 일시 중단 등으로 인해 원본 DB 인스턴스가 I/O 요청을 처리하지 못할 수 있습니다. 이러한 경우 읽기 트래픽을 읽기 전용 복제본으로 리디렉션할 수 있습니다. 이 사용 사례의 경우 원본 DB 인스턴스를 사용할 수 없으므로 읽기 전용 복제본의 데이터가 “무효”일 수 있다는 점에 유의하십시오.
- 비즈니스 보고 또는 데이터 웨어하우징에서는 기본 복제본, 프로덕션 DB 인스턴스가 아닌 읽기 전용 복제본에 대한 비즈니스 보고 쿼리를 실행할 수 있습니다.
- 재해 복구 구현 원본 DB 인스턴스가 실패할 경우, 재해 복구 솔루션으로 읽기 전용 복제본을 듀립 인스턴스로 승격시킬 수 있습니다.

기본적으로 읽기 전용 복제본은 원본 DB 인스턴스와 동일한 스토리지 유형과 함께 생성됩니다. 하지만 다음 표의 옵션에 따라 원본 DB 인스턴스와 다른 스토리지 유형을 할당하여 읽기 전용 복제본을 생성할 수도 있습니다.

원본 DB 인스턴스 스토리지 유형	원본 DB 인스턴스 스토리지 할당	읽기 전용 복제본 스토리지 유형 옵션
PIOPS	100GiB–32TiB	PIOPS, GP2, 표준
GP2	100GiB–32TiB	PIOPS, GP2, 표준
GP2	100GiB 미만	GP2, 표준
표준	100GiB 6TiB	PIOPS, GP2, 표준
표준	100GiB 미만	GP2, 표준

Amazon RDS는 순환 복제를 지원하지 않습니다. DB 인스턴스를 기존 DB 인스턴스의 복제 소스로 구성할 수 없습니다. 기존 DB 인스턴스에서 새 읽기 전용 복제본만 생성할 수 있습니다. 예를 들어, **MyDBInstance**을 **ReadReplica1**으로 복제하는 경우, **ReadReplica1**을 **MyDBInstance**로 다시 복제하

도록 구성할 수 없습니다. MariaDB, MySQL 및 PostgreSQL의 경우 기존 읽기 전용 복제본에서 읽기 전용 복제본을 만들 수 있습니다. 예를 들어 **ReadReplica1**에서 **ReadReplica2**와 같은 새 읽기 전용 복제본을 만들 수 있습니다. Oracle의 경우 기존 읽기 전용 복제본에서 읽기 전용 복제본을 만들 수 없습니다.

다른 DB 엔진의 읽기 전용 복제본과의 차이점

Amazon RDS DB 엔진은 복제를 다르게 구현하기 때문에 다음 표와 같이 몇 가지 중요한 차이점을 알아야 합니다.

기능 또는 특성	MySQL 및 MariaDB	Oracle	PostgreSQL
어떤 복제 방식을 사용합니까?	논리적 복제	물리적 복제	물리적 복제
트랜잭션 로그는 어떻게 삭제합니까?	Amazon RDS MySQL과 MariaDB는 적용되지 않은 이진 로그를 유지합니다.	Oracle은 Amazon RDS는 원본 DB 인스턴스에서 최소 2시간의 트랜잭션 로그를 유지합니다. 로그는 <code>archivelog retention hours</code> 설정이 통과된 후 또는 두 시간 후 중에서 더 긴 시간이 경과한 후 원본에서 제거됩니다. <code>archivelog retention hours</code> 설정이 데이터베이스에 성공적으로 적용된 경우에만 설정 통과 후 로그가 읽기 전용 복제본에서 제거됩니다. <code>archivelog retention hours</code> 설정에 대한 정보는 보관된 다시 실행 로그 보존 (p. 856) 단원을 참조하십시오.	PostgreSQL에는 데이터를 읽기 전용 복제본으로 보낼 때 유지할 Write Ahead Log(WAL) 파일 수를 결정할 수 있는 파라미터인 <code>wal_keep_segments</code> 가 있습니다. 이 파라미터 값에 따라 유지할 로그 수가 결정됩니다.
복제본에 쓰기가 가능합니까?	예. MySQL 또는 MariaDB 읽기 전용 복제본은 쓰기가 가능하도록 활성화 할 수 있습니다.	아닙니다. Oracle 읽기 전용 복제본은 인쇄본이며 Oracle은 읽기 전용 복제본에 쓰기를 허용하지 않습니다. 읽기 전용 복제본을 쓰기 가능하도록 승격할 수 있습니다. 승격된 읽기 전용 복제본에는 승격 요청이 이루어진 시점까지 복제된 데이터가 있습니다.	아니요. PostgreSQL 읽기 전용 복제본은 물리적 복사본이므로 PostgreSQL은 읽기 전용 복제본에 대해 쓰기를 허용하지 않습니다.
복제본에 대해서도 백업이 가능합니까?	예. MySQL 또는 MariaDB 읽기 전용 복제본은 자동 백업을 활성화 할 수 있습니다.	아닙니다. Oracle은 Amazon RDS 읽기 전용 복제본의 수동 스냅샷을 생성하거나 자동 백업을 활성화 할 수 없습니다.	예. PostgreSQL 읽기 전용 복제본의 수동 스냅샷을 생성할 수는 있지만 자동 백업은 활성화 할 수 없습니다.
병렬 복제 기능을 사용할 수 있습니까?	예. MySQL 버전 5.6 이상 및 지원되는 모든 MariaDB 버전은 병렬 복제 스레드가 가능합니다.	예. 재실행 로그 데이터는 항상 원본 데이터베이스에서 모든 읽기 전용 복제본으로 병렬로 전송됩니다.	아니요. PostgreSQL은 복제를 처리하는 단일 프로세스가 있습니다.

읽기 전용 복제본 만들기

기존 MySQL, MariaDB, Oracle 또는 PostgreSQL DB 인스턴스에서 AWS Management 콘솔, AWS CLI 또는 AWS API를 사용하여 읽기 전용 복제본을 생성할 수 있습니다. 또한 `SourceDBInstanceIdentifier`를 지정하여 읽기 전용 복제본을 생성할 수도 있습니다. 이 코드는 복제하고자 하는 원본 DB 인스턴스의 식별자 역할을 합니다.

읽기 전용 복제본을 생성하면 Amazon RDS가 원본 DB 인스턴스의 DB 스냅샷을 캡처하고 복제를 시작합니다. 그 결과, DB 스냅샷을 캡처하는 동안 원본 DB 인스턴스에서 짧은 I/O 중단이 발생합니다. 이러한 I/O 중단은 일반적으로 1분 정도 지속됩니다. 원본 DB 인스턴스가 다중 AZ 배포인 경우에는 I/O 중단을 방지할 수 있습니다. 이 경우에는 보조 DB 인스턴스에서 스냅샷을 생성하기 때문입니다. 활성 상태의 장기 실행 트랜잭션은 읽기 전용 복제본 생성 프로세스를 늦출 수 있습니다. 읽기 전용 복제본을 생성하기 전에 장기 실행 트랜잭션이 완료되기를 기다리는 것이 좋습니다. 동일한 원본 DB 인스턴스에서 다수의 읽기 전용 복제본을 병렬 방식으로 생성하는 경우에는 Amazon RDS가 첫 번째 생성 작업을 시작하면서 한 번만 스냅샷을 캡처합니다.

읽기 전용 복제본을 생성할 때는 몇 가지 고려할 사항이 있습니다. 첫째, 백업 보존 기간을 0이 아닌 다른 값으로 설정하여 원본 DB 인스턴스의 자동 백업을 활성화해야 합니다. 이러한 요건은 다른 읽기 전용 복제본의 원본 DB 인스턴스인 읽기 전용 복제본에도 똑같이 적용됩니다. MySQL DB 인스턴스의 경우 자동 백업은 MySQL 5.6 이상을 실행하는 읽기 전용 복제본에서만 지원되며 MySQL 버전 5.5에서는 지원되지 않습니다. Amazon RDS MySQL 5.6 이상 읽기 전용 복제본에서 자동 백업을 활성화하려면 먼저 읽기 전용 복제본을 생성한 다음 자동 백업이 활성화되도록 읽기 전용 복제본을 변경해야 합니다.

콘솔

원본 MySQL, MariaDB, Oracle 또는 PostgreSQL DB 인스턴스에서 읽기 전용 복제본을 생성하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 읽기 전용 복제본의 원본으로 사용할 MySQL, MariaDB, Oracle 또는 PostgreSQL DB 인스턴스를 선택합니다.
4. 작업에서 읽기 전용 복제본 만들기를 선택합니다.
5. 사용할 인스턴스 사양을 선택합니다. 읽기 전용 복제본의 원본 DB 인스턴스와 동일한 DB 인스턴스 클래스와 스토리지 유형을 사용하는 것이 좋습니다. 복제본에 대한 장애 조치 지원을 위해 다른 가용 영역에 예비 복제본을 생성하려면 [Multi-AZ deployment]에서 [Yes]를 선택합니다. 읽기 전용 복제본을 다중 AZ DB 인스턴스로 생성하는 작업은 원본 데이터베이스가 다중 AZ DB 인스턴스인지 여부와는 독립적입니다.
6. 사용하려는 설정을 선택합니다. DB 인스턴스 식별자에 읽기 전용 복제본의 이름을 입력합니다. 필요에 따라 다른 설정을 조정합니다.
7. 사용하고자 하는 기타 설정을 선택합니다.
8. [Create read replica]를 선택합니다.

AWS CLI

원본 MySQL, MariaDB, Oracle 또는 PostgreSQL DB 인스턴스에서 읽기 전용 복제본을 생성하려면 AWS CLI 명령 `create-db-instance-read-replica`를 사용합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds create-db-instance-read-replica \
--db-instance-identifier myreadreplica \
```

```
--source-db-instance-identifier mydbinstance
```

Windows의 경우:

```
aws rds create-db-instance-read-replica ^
--db-instance-identifier myreadreplica ^
--source-db-instance-identifier mydbinstance
```

RDS API

원본 MySQL, MariaDB, Oracle 또는 PostgreSQL DB 인스턴스에서 읽기 전용 복제본을 생성하려면 Amazon RDS API 함수 [CreateDBInstanceReadReplica](#)를 호출합니다.

```
https://rds.amazonaws.com/
?Action=CreateDBInstanceReadReplica
&DBInstanceIdentifier=myreadreplica
&SourceDBInstanceIdentifier=mydbinstance
&Version=2012-01-15
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2012-01-20T22%3A06%3A23.624Z
&AWSAccessKeyId=<AWS Access Key ID>
&Signature=<Signature>
```

읽기 전용 복제본을 독립 DB 인스턴스로 승격

MySQL, MariaDB, Oracle 또는 PostgreSQL 읽기 전용 복제본을 독립 실행형 DB 인스턴스로 승격할 수 있습니다. 읽기 전용 복제본을 승격하면 DB 인스턴스가 먼저 재부팅된 후에 사용할 수 있습니다.

읽기 전용 복제본을 독립 실행형 DB 인스턴스로 승격해야 하는 몇 가지 이유가 있습니다.

- DDL 작업 실행(MySQL 및 MariaDB에 한함) – 인덱스를 생성하거나 리빌드하는 등의 DDL 작업은 시간이 걸릴 뿐만 아니라 DB 인스턴스에 상당한 성능 저하를 초래할 수 있습니다. MySQL 또는 MariaDB 읽기 전용 복제본에서 이러한 작업을 하려면 먼저 읽기 전용 복제본이 원본 DB 인스턴스와 동기화되어 있어야 합니다. 그런 다음 읽기 전용 복제본을 승격해야 애플리케이션이 승격된 인스턴스를 사용하도록 유도할 수 있습니다.
- 샤딩 – 샤딩이란 "무공유(share-nothing)" 아키텍처를 구현함으로써 기본적으로 대규모 데이터베이스를 다수의 소규모 데이터베이스로 분할하는 기술입니다. 데이터베이스 분할은 동일한 쿼리에 조인되지 않은 테이블을 다른 호스트로 분할하는 방법이 일반적입니다. 그 밖에 테이블을 여러 호스트로 복사한 후 해싱 알고리즘을 사용해 어떤 호스트를 업데이트할지 결정하는 방법도 있습니다. 분할된 각 "샤드"(소규모 데이터베이스)마다 읽기 전용 복제본을 생성한 후 "독립된" 샤드로 전환하면서 승격할 수 있습니다. 그런 다음 각 샤드마다 요건에 따라 키 범위(행을 분할한 경우)나 배포된 테이블을 얻을 수 있습니다.
- 장애 복구 실행 – 원본 DB 인스턴스에 장애가 발생할 경우, 읽기 전용 복제본을 데이터 복구 방법으로 사용할 수 있습니다. 이 방법은 동기식 복제, 장애 자동 감지 및 장애 조치를 보완합니다.

비동기식 복제의 영향이나 한계에 대해 알고 있더라도 데이터 복구를 위해 읽기 전용 복제본 승격을 사용할 수 있습니다. 이때는 먼저 읽기 전용 복제본을 생성한 다음 원본 DB 인스턴스의 장애 여부를 모니터링해야 합니다. 그 결과 장애가 발견된 경우에는 다음과 같이 실행합니다.

1. 읽기 전용 복제본을 승격합니다.
2. 데이터베이스 트래픽을 승격된 DB 인스턴스로 유도합니다.
3. 승격된 DB 인스턴스를 원본으로 하는 교체용 읽기 전용 복제본을 생성합니다.

읽기 전용 복제본을 승격하면 새 DB 인스턴스가 생성되더라도 이전 읽기 전용 복제본 원본의 백업 보존 기간, 백업 기간, 옵션 그룹 및 파라미터 그룹은 그대로 보존됩니다. 승격 프로세스는 읽기 전용 복제본의 크기에 따라 완료하는 데 몇 분에서 더 오래도 걸릴 수 있습니다. 일단 읽기 전용 복제본이 새 DB 인스턴스로 승

격되면 다른 DB 인스턴스와 다를 것이 없습니다. 예를 들어, 새 DB 인스턴스에서 읽기 전용 복제본을 생성하고 특정 시점 복원 작업을 수행할 수 있습니다. 승격된 DB 인스턴스는 더 이상 읽기 전용 복제본이 아니기 때문에 복제 대상으로는 사용할 수 없습니다. 원본 DB 인스턴스에 읽기 전용 복제본이 다수 있을 경우 읽기 전용 복제본 하나를 DB 인스턴스로 승격하더라도 나머지 복제본에는 아무런 영향도 끼치지 않습니다.

백업 기간은 이전 백업 이후 데이터베이스에 대해 이루어진 변경 횟수를 지정하는 기능입니다. 읽기 전용 복제본을 독립형 인스턴스로 승격할 계획이라면 백업을 활성화하고 승격 전에 최소 한 개 이상의 백업을 완료하는 것이 좋습니다. 또한 읽기 전용 복제본이 *backing-up* 상태인 경우에는 독립형 인스턴스로 승격할 수 없습니다. 읽기 전용 복제본에 대해 백업을 활성화한 경우, 일일 백업이 읽기 전용 복제본 승격을 방해하지 않도록 자동 백업 기간을 구성하십시오.

다음은 읽기 전용 복제본을 DB 인스턴스로 승격하기 위한 일반적인 프로세스입니다.

1. 읽기 전용 복제본의 원본 DB 인스턴스에 쓰기를 하지 못하도록 모든 트랜잭션을 중단한 후 읽기 전용 복제본의 업데이트가 모두 끝날 때까지 기다립니다. 읽기 전용 복제본의 데이터베이스 업데이트는 원본 DB 인스턴스의 업데이트가 끝난 후에 이어집니다. 이러한 복제 지연은 경우에 따라 큰 차이가 날 수도 있습니다. [Replica Lag](#) 지표를 사용하여 읽기 전용 복제본의 업데이트가 모두 완료되는 시간을 측정합니다.
2. MySQL 및 MariaDB에 한함: MySQL 또는 MariaDB 읽기 전용 복제본을 변경해야 하는 경우에는 읽기 전용 복제본의 DB 파라미터 그룹에서 `read_only` 파라미터를 0으로 설정해야 합니다. 그래야만 인덱스 생성 등 필요한 DDL 작업을 모두 읽기 전용 복제본에 실행할 수 있기 때문입니다. 읽기 전용 복제본에 실행하는 작업은 원본 DB 인스턴스의 성능에 아무런 영향도 끼치지 않습니다.
3. Amazon RDS 콘솔의 읽기 전용 복제본 승격 옵션, AWS CLI 명령 [promote-read-replica](#) 또는 [PromoteReadReplica](#) Amazon RDS API 작업 등을 사용하여 읽기 전용 복제본을 승격합니다.

Note

승격 프로세스는 완료할 때까지 몇 분 걸립니다. 읽기 전용 복제본을 승격하면 복제가 중지되고 읽기 전용 복제본이 재부팅됩니다. 재부팅이 끝나면 읽기 전용 복제본을 새 DB 인스턴스로 사용할 수 있습니다.

4. (선택 사항) 다중 AZ 배포가 되도록 새 DB 인스턴스를 수정합니다. 자세한 내용은 [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#) 및 [Amazon RDS를 위한 고가용성\(다중 AZ\) \(p. 109\)](#) 단원을 참조하십시오.

콘솔

읽기 전용 복제본을 DB 인스턴스로 승격하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. Amazon RDS 콘솔에서 데이터베이스를 선택합니다.
데이터베이스 창이 표시됩니다. 각 읽기 전용 복제본은 역할 열에 복제본이라고 표시됩니다.
3. 승격시키려는 읽기 전용 복제본을 선택합니다.
4. 작업에서 읽기 전용 복제본 승격을 선택합니다.
5. [Promote Read Replica] 페이지에서 새롭게 승격된 DB 인스턴스의 백업 보존 기간과 백업 기간을 입력합니다.
6. 원하는 대로 설정되었으면 [Continue]를 선택합니다.
7. 승인 페이지에서 [Promote Read Replica]를 선택합니다.

AWS CLI

읽기 전용 복제본을 DB 인스턴스로 승격하려면 AWS CLI [promote-read-replica](#) 명령을 사용합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds promote-read-replica \
--db-instance-identifier myreadreplica
```

Windows의 경우:

```
aws rds promote-read-replica ^
--db-instance-identifier myreadreplica
```

RDS API

읽기 전용 복제본을 DB 인스턴스로 승격하려면 [PromoteReadReplica](#)를 호출합니다.

```
https://rds.amazonaws.com/
?Action=PromoteReadReplica
&DBInstanceIdentifier=myreadreplica
&Version=2012-01-15
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2012-01-20T22%3A06%3A23.624Z
&AWSAccessKeyId=<AWS Access Key ID>
&Signature=<Signature>
```

다른 AWS 리전에서 읽기 전용 복제본 만들기

Amazon RDS를 사용하여 원본 DB 인스턴스와 다른 AWS 리전에 MariaDB, MySQL 또는 PostgreSQL 읽기 전용 복제본을 생성할 수 있습니다. 읽기 전용 복제본을 생성하면 다음 작업을 수행할 수 있습니다.

- 재해 복구 기능을 향상 시킵니다.
- 읽기 작업을 사용자에 더 가까운 AWS 리전으로 조정합니다.
- 한 AWS 리전의 데이터 센터에서 다른 AWS 리전의 데이터 센터로 마이그레이션하는 작업을 더 용이하게 합니다.

원본 인스턴스와 다른 AWS 리전에 MariaDB, MySQL 또는 PostgreSQL 읽기 전용 복제본을 생성하더라도 방법은 동일한 AWS 리전에 복제본을 생성하는 것과 비슷합니다. 여러 리전에 걸쳐 두루 읽기 전용 복제본을 생성하려면 AWS Management 콘솔을 사용하거나, [create-db-instance-read-replica](#) 명령을 실행하거나 [CreateDBInstanceReadReplica](#) API 작업을 호출합니다.

원본 DB 인스턴스와 다른 AWS 리전에서 암호화된 읽기 전용 복제본을 생성하려면 원본 DB 인스턴스를 암호화해야 합니다.

그 다음 다른 AWS 리전의 원본 MariaDB, MySQL 또는 PostgreSQL DB 인스턴스에서 읽기 전용 복제본을 생성하는 방법에 대한 정보를 찾을 수 있습니다.

콘솔

AWS Management 콘솔을 사용하여 여러 리전에 걸쳐 읽기 전용 복제본을 생성할 수 있습니다.

콘솔을 사용해 여러 리전에서 읽기 전용 복제본 만들기

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 읽기 전용 복제본의 원본으로 사용할 MariaDB, MySQL 또는 PostgreSQL DB 인스턴스를 선택한 후, 작업에서 Create read replica(읽기 전용 복제본 생성)를 선택합니다. 암호화된 읽기 전용 복제본을 생성하

려면 원본 DB 인스턴스를 암호화해야 합니다. 원본 DB 인스턴스를 암호화하는 방법에 대해 자세히 알아보려면 [Amazon RDS 리소스 암호화 \(p. 385\)](#) 단원을 참조하십시오.

4. 사용할 인스턴스 사양을 선택합니다. 읽기 전용 복제본에는 동일한 DB 인스턴스 클래스와 스토리지 유형을 사용하는 것이 좋습니다.
5. 사용하려는 기타 설정을 선택합니다.
 - DB 인스턴스 식별자에 읽기 전용 복제본의 이름을 입력합니다.
 - 네트워크 및 보안 섹션에서 Designation region(대상 리전)과 Designation DB subnet group(대상 DB 지원 그룹)에 대한 값을 선택합니다.
 - 다른 AWS 리전에서 암호화된 읽기 전용 복제본을 생성하려면 암호화 활성화를 선택한 후 마스터 키를 선택합니다. 마스터 키에서 대상 AWS 리전의 AWS Key Management Service(AWS KMS) 키 식별자를 선택합니다.
 - 사용하고자 하는 기타 설정을 선택합니다.
6. [Create read replica]를 선택합니다.

AWS CLI

다른 AWS 리전의 원본 MySQL, MariaDB 또는 PostgreSQL DB 인스턴스에서 읽기 전용 복제본을 생성하려면 명령을 사용할 수 있습니다. `create-db-instance-read-replica` 이 경우 읽기 전용 복제본이 필요한 AWS 리전에서 명령을 실행한 후 원본 DB 인스턴스에 Amazon 리소스 이름(ARN)을 지정해줍니다. `create-db-instance-read-replica` ARN은 Amazon Web Services에서 생성된 리소스를 고유하게 식별합니다.

예를 들어, 원본 DB 인스턴스가 미국 동부(버지니아 북부) 리전에 있는 경우, ARN은 다음과 유사한 모양을 띕니다.

```
arn:aws:rds:us-east-1:123456789012:db:my-mysql-instance
```

ARN에 대한 자세한 내용은 [Amazon RDS의 Amazon 리소스 이름\(ARN\)을 사용한 작업 \(p. 172\)](#) 단원을 참조하십시오.

원본 DB 인스턴스와 다른 AWS 리전에서 암호화된 읽기 전용 복제본을 생성하려면 대상 AWS 리전에서 AWS CLI `create-db-instance-read-replica` 명령을 사용할 수 있습니다. 다음 파라미터는 다른 AWS 리전에서 암호화된 읽기 전용 복제본을 생성하는 데 사용됩니다.

- `--source-region` — 암호화된 읽기 전용 복제본이 생성되는 AWS 리전. `source-region`이 지정되지 않은 경우에는 `pre-signed-url` 값을 지정해야 합니다. `pre-signed-url`은 읽기 전용 복제본이 생성되는 원본 AWS 리전에서 호출되는 `CreateDBInstanceReadReplica` 작업에 대한 서명 버전 4의 서명된 요청이 포함된 URL입니다. `pre-signed-url`에 대한 자세한 내용은 [CreateDBInstanceReadReplica](#) 단원을 참조하십시오.
- `--source-db-instance-identifier` — 생성되는 암호화된 읽기 전용 복제본의 DB 인스턴스 식별자. 이 식별자는 원본 AWS 리전용 ARN 형식으로 되어 있어야 합니다. `source-db-instance-identifier`에 지정된 AWS 리전은 `source-region`로 지정된 AWS 리전과 일치해야 합니다.
- `--db-instance-identifier` — 대상 AWS 리전에 있는 암호화된 읽기 전용 복제본의 식별자.
- `--kms-key-id` — 대상 AWS 리전에서 읽기 전용 복제본을 암호화하는 데 사용할 키의 AWS KMS 키 식별자.

다음 코드는 us-west-2 리전에서 읽기 전용 복제본을 생성합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds create-db-instance-read-replica \
--db-instance-identifier DBInstanceIdentifier \
```

```
--region us-west-2 \
--source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:my-mysql-instance
```

Windows의 경우:

```
aws rds create-db-instance-read-replica ^
--db-instance-identifier DBInstanceIdentifier ^
--region us-west-2 ^
--source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:my-mysql-instance
```

다음 코드는 원본 DB 인스턴스와 다른 AWS 리전에서 읽기 전용 복제본을 생성합니다. `create-db-instance-read-replica` 명령을 호출하는 AWS 리전이 암호화된 읽기 전용 복제본의 대상 AWS 리전입니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds create-db-instance-read-replica \
--db-instance-identifier DBInstanceIdentifier \
--region us-west-2 \
--source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:my-mysql-instance \
\
--source-region us-east-1 \
--kms-key-id my-us-east-1-key
```

Windows의 경우:

```
aws rds create-db-instance-read-replica ^
--db-instance-identifier DBInstanceIdentifier ^
--region us-west-2 ^
--source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:my-mysql-instance
^
--source-region us-east-1 ^
--kms-key-id my-us-east-1-key
```

RDS API

다른 AWS 리전의 원본 MySQL, MariaDB 또는 PostgreSQL DB 인스턴스에서 읽기 전용 복제본을 생성하려면 Amazon RDS API 함수 `CreateDBInstanceReadReplica`를 호출할 수 있습니다. 이 경우 읽기 전용 복제본이 필요한 AWS 리전에서 `CreateDBInstanceReadReplica` 명령을 호출한 후 원본 DB 인스턴스에 Amazon 리소스 이름(ARN)을 지정해줍니다. ARN은 Amazon Web Services에서 생성된 리소스를 고유하게 식별합니다.

원본 DB 인스턴스와 다른 AWS 리전에서 암호화된 읽기 전용 복제본을 생성하려면 대상 AWS 리전에서 Amazon RDS API `CreateDBInstanceReadReplica` 작업을 사용할 수 있습니다. 다른 AWS 리전에서 암호화된 읽기 전용 복제본을 생성하려면 `PreSignedURL`에 값을 지정해야 합니다. `PreSignedURL`에는 읽기 전용 복제본이 생성되는 소스 AWS 리전에서 호출하는 `CreateDBInstanceReadReplica` 작업에 대한 요청이 포함되어야 합니다. `PreSignedURL`에 대한 자세한 내용은 `CreateDBInstanceReadReplica`를 참조하십시오.

예를 들어, 원본 DB 인스턴스가 미국 동부(버지니아 북부) 리전에 있는 경우, ARN은 다음과 유사한 모양을 띕니다.

```
arn:aws:rds:us-east-1:123456789012:db:my-mysql-instance
```

ARN에 대한 자세한 내용은 [Amazon RDS의 Amazon 리소스 이름\(ARN\)을 사용한 작업 \(p. 172\)](#) 단원을 참조하십시오.

Example

```
https://us-west-2.rds.amazonaws.com/
?Action=CreateDBInstanceReadReplica
&KmsKeyId=my-us-east-1-key
&PreSignedUrl=https%253A%252F%252Frds.us-west-2.amazonaws.com%252F
%253FAction%253D CreateDBInstanceReadReplica
%2526DestinationRegion%253Dus-east-1
%2526KmsKeyId%253Dmy-us-east-1-key
%2526SourceDBIdentifier%253Darn%25253Aaws%25253Ards%25253Aus-
west-2%234567890
12%25253Adb%25253Amy-mysql-instance
%2526SignatureMethod%253DHmacSHA256
%2526SignatureVersion%253D4%2526SourceDBIdentifier%253Darn%25253Aaws
%25253Ards%25253Aus-west-2%25253A123456789012%25253Ainstance%25253Amysql-instance1-
instance-20161115
%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
&DBIdentifier=myreadreplica
&SourceDBIdentifier=arn:aws:rds:us-east-1:123456789012:db:my-mysql-instance
&Version=2012-01-15
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2012-01-20T22%3A06%3A23.624Z
&AWSAccessKeyId=<AWS Access Key ID>
&Signature=<Signature>
```

리전 간 복제 시 고려 사항

한 AWS 리전 내에서 복제할 때 고려해야 할 모든 사항이 리전 간 복제 시에도 그대로 적용됩니다. 그 밖에 리전 간 복제 시 다음과 같은 고려 사항이 추가로 적용됩니다.

- 리전 간 복제는 MariaDB, PostgreSQL(버전 9.4.7 및 9.5.2 이상) 또는 MySQL 5.6 이상의 Amazon RDS DB 인스턴스를 사용할 때만 가능합니다.
- 원본 DB 인스턴스는 다수의 리전에서 리전 간 읽기 전용 복제본을 가질 수 있습니다.
- 리전 간 Amazon RDS 읽기 전용 복제본은 다른 Amazon RDS DB 인스턴스의 읽기 전용 복제본이 아닌 원본 Amazon RDS DB 인스턴스에서만 생성할 수 있습니다.
- 복제 채널을 AWS GovCloud (US-West) 리전으로 또는 리전을 벗어나서 설정할 수 없습니다.
- 원본 인스턴스와 다른 AWS 리전에 읽기 전용 복제본을 생성할 때는 지역 시간이 증가할 수 있다는 점을 감안해야 합니다. 지역 데이터 센터 간 네트워크 채널이 긴 경우에 이러한 지역 시간이 발생합니다.
- 한 AWS 리전 내에서 똑같은 원본 DB 인스턴스를 통해 생성되는 리전 간 읽기 전용 복제본은 모두 동일한 Amazon VPC에 속하거나 VPC에서 벗어나야 합니다. 리전 간 읽기 전용 복제본의 경우 --db-subnet-group-name 파라미터를 지정하는 읽기 전용 복제본 생성 명령은 모두 DB 서브넷 그룹을 동일한 VPC에서 지정해야 합니다.
- 다른 리전의 VPC에 있는 소스 DB 인스턴스에서 VPC에 교차 리전 읽기 전용 복제본을 생성할 수 있습니다. 또한 VPC에 없는 소스 DB 인스턴스에서도 VPC에 리전 간 읽기 전용 복제본을 생성할 수 있습니다. 또한 리전 간 읽기 전용 복제본이 VPC에 속하지 않더라도 VPC에 속하는 원본 DB 인스턴스에서 생성할 수 있습니다.

- VPC에 대한 ACL(액세스 제어 목록) 항목 수 제한으로 인해 5개를 초과하는 교차 리전 읽기 전용 복제본 인스턴스를 보장할 수 없습니다.

리전 간 복제 비용

리전 간 복제를 위해 데이터를 전송할 때는 Amazon RDS 데이터 전송 요금이 발생합니다. 이러한 리전 간 복제 작업에서 요금이 발생하는 이유는 다음과 같이 원본 AWS 리전을 벗어나 데이터를 전송하기 때문입니다.

- 읽기 전용 복제본을 생성할 때는 Amazon RDS가 원본 인스턴스의 스냅샷을 캡처하여 읽기 전용 복제본 리전으로 전송합니다.
- 원본 데이터베이스에서 데이터를 변경할 때마다 Amazon RDS가 소스 AWS 리전에서 읽기 전용 복제본 리전으로 데이터를 전송합니다.

데이터 전송 요금에 대한 자세한 내용은 [Amazon RDS 요금](#)을 참조하십시오.

MySQL 및 MariaDB 인스턴스의 경우 리전 간 읽기 전용 복제본의 생성 수를 줄이면 데이터 전송 요금도 절감할 수 있습니다. 예를 들어 AWS 리전 한 곳에 원본 DB 인스턴스가 있고, 다른 AWS 리전에 읽기 전용 복제본을 생성한다고 가정하겠습니다. 이때는 원본 DB 인스턴스에서 읽기 전용 복제본을 1개만 생성합니다. 그런 다음 나머지 2개는 원본 DB 인스턴스가 아닌 첫 번째 읽기 전용 복제본에서 생성합니다.

예를 들어 한 AWS 리전에 source-instance-1이 있는 경우 다음 작업을 수행할 수 있습니다.

- 새로운 AWS 리전에 read-replica-1을 생성한 후 source-instance-1을 원본으로 지정합니다.
- read-replica-2를 read-replica-1에서 생성합니다.
- read-replica-3를 read-replica-1에서 생성합니다.

위 예에서 요금은 source-instance-1에서 read-replica-1로 데이터를 전송할 때만 발생합니다. read-replica-1에서 나머지 두 복제본으로 데이터를 전송할 때는 동일 AWS 리전이기 때문에 요금이 발생하지 않습니다. 만약 복제본 3개를 전부 source-instance-1에서 생성한다면 복제본 3개에 대한 데이터 전송 요금이 모두 발생합니다.

Amazon RDS의 리전 간 복제 방법

Amazon RDS는 다음 프로세스에 따라 리전 간 읽기 전용 복제본을 생성합니다. 이 프로세스를 완료하려면 관련된 리전과 데이터베이스의 데이터 양에 따라 몇 시간까지 걸리기도 합니다. 이 정보는 리전 간 읽기 전용 복제본을 생성하면서 프로세스의 진행 과정을 파악할 때 사용할 수 있습니다.

1. Amazon RDS가 원본 DB 인스턴스를 복제 원본으로 구성하기 시작하면서 상태를 modifying으로 설정합니다.
2. Amazon RDS가 대상 AWS 리전에서 지정된 읽기 전용 복제본 설정을 시작하면서 상태를 생성 중으로 설정합니다.
3. Amazon RDS가 소스 AWS 리전에서 소스 DB 인스턴스의 자동 DB 스냅샷을 생성합니다. DB 스냅샷 이름은 rds:<InstanceID>-<timestamp>와 같은 형식을 갖습니다. 여기에서 <InstanceID>는 원본 인스턴스의 식별자이고, <timestamp>는 복제 시작일과 시간을 나타냅니다. 예를 들어 rds:mysourceinstance-2013-11-14-09-24는 인스턴스 mysourceinstance에서 2013-11-14-09-24에 생성되었다는 것을 의미합니다. 자동 DB 스냅샷의 생성 단계에서 원본 DB 인스턴스 상태는 modifying, 읽기 전용 복제본 상태는 creating, 그리고 DB 스냅샷 상태는 creating입니다. 콘솔의 [DB Snapshots] 페이지에서 [Progress] 열을 보면 DB 스냅샷 생성이 얼마나 진행되었는지 알 수 있습니다. DB 스냅샷이 완료되면 DB 스냅샷과 원본 DB 인스턴스의 상태가 모두 available로 설정됩니다.
4. Amazon RDS가 리전 간 스냅샷 복사를 시작하면서 첫 번째 데이터 전송이 이루어집니다. 스냅샷 사본은 대상 AWS 리전에 creating 상태와 함께 자동 스냅샷으로 등록됩니다. 이름은 원본 DB 스냅샷과 동일합니다. [DB Snapshots] 페이지에서 [Progress] 열을 보면 복사가 얼마나 진행되었는지 알 수 있습니다. 복사가 완료되면 DB 스냅샷 사본 상태가 available로 설정됩니다.

5. Amazon RDS가 이제 복사된 DB 스냅샷을 사용하여 읽기 전용 복제본에 처음으로 데이터를 로드하기 시작합니다. 이 단계에서 읽기 전용 복제본이 대상 DB 인스턴스 목록에 포함되며, 이때 상태는 `creating`입니다. 로드가 완료되면 읽기 전용 복제본 상태가 `available`로 설정되고, DB 스냅샷 사본이 삭제됩니다.
6. 읽기 전용 복제본이 `available` 상태에 이르면 Amazon RDS가 읽기 전용 복제본 생성 작업을 시작한 이후 원본 인스턴스에 발생한 변경 사항을 복제하기 시작합니다. 이 단계에서 읽기 전용 복제본의 복제 지연 시간은 0보다 큽니다.

복제 지연 시간에 대한 자세한 내용은 [읽기 전용 복제본 모니터링 \(p. 146\)](#) 단원을 참조하십시오.

리전 간 복제 예

Example VPC 외부의 리전 간 읽기 전용 복제본 생성

다음은 us-east-1에 위치한 원본 DB 인스턴스에서 us-west-2에 읽기 전용 복제본을 생성하는 예입니다. 읽기 전용 복제본은 VPC 외부에 생성됩니다.

Linux, OS X, Unix의 경우:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier SimCoProd01Replica01 \
    --region us-west-2
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:SimcoProd01
```

Windows의 경우:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier SimCoProd01Replica01 ^
    --region us-west-2
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:SimcoProd01
```

Example VPC 내부의 리전 간 읽기 전용 복제본 생성

다음은 us-east-1에 위치한 원본 DB 인스턴스에서 us-west-2에 읽기 전용 복제본을 생성하는 예입니다. 읽기 전용 복제본은 지정한 DB 서브넷 그룹과 연동된 VPC 내부에 생성됩니다.

Linux, OS X, Unix의 경우:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier SimCoProd01Replica01 \
    --region us-west-2
    --db-subnet-group-name my-us-west-2-subnet
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:SimcoProd01
```

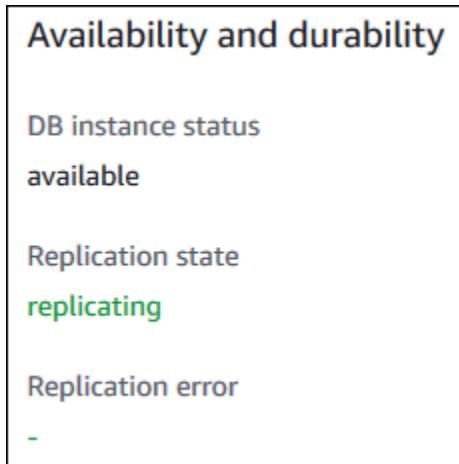
Windows의 경우:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier SimCoProd01Replica01 ^
    --region us-west-2
    --db-subnet-group-name my-us-west-2-subnet
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:SimcoProd01
```

읽기 전용 복제본 모니터링

읽기 전용 복제본의 상태를 모니터링할 수 있는 방법은 몇 가지가 있습니다. Amazon RDS 콘솔에서 읽기 전용 복제본 세부 정보의 Availability and durability(가용성 및 내구성) 섹션에 읽기 전용 복제본의 상태가 표시

됩니다. 읽기 전용 복제본의 세부 정보를 보려면 Amazon RDS 콘솔의 인스턴스 목록에서 읽기 전용 복제본의 이름을 클릭합니다.



AWS CLI `describe-db-instances` 명령 또는 Amazon RDS API `DescribeDBInstances` 작업을 사용하여 읽기 전용 복제본의 상태를 볼 수도 있습니다.

읽기 전용 복제본의 상태는 다음 중 한 가지가 될 수 있습니다.

- Replicating(복제 중)—읽기 전용 복제본이 성공적으로 복제되고 있습니다.
- Error(오류)—복제 도중 오류가 발생하였습니다. Amazon RDS 콘솔의 [Replication Error] 필드나 이벤트 로그를 검사하여 정확한 오류 원인을 찾아내야 합니다. 복제 오류의 문제 해결에 대한 자세한 내용은 [MySQL 읽기 전용 복제본의 문제 해결 \(p. 649\)](#) 단원을 참조하십시오.
- Terminated(종료)—복제가 종료되었습니다. 수동으로 또는 복제 오류로 인해 연속하여 30일 이상 복제가 중지된 경우 이 오류가 발생합니다. 이 경우 Amazon RDS는 원본 DB 인스턴스에 대한 스토리지 요건 증가와 장애 조치 시간 장기화를 방지하기 위해 Amazon RDS가 원본 DB 인스턴스와 모든 읽기 전용 복제본 사이의 복제를 종료합니다.

복제가 중단되면 대용량의 오류 메시지를 로그에 써야 하기 때문에 로그 크기와 수가 증가하면서 스토리지에 영향을 끼칠 수 있습니다. 또한 Amazon RDS가 복구 단계에서 다수의 로그를 유지 및 처리하는 데 필요한 시간 때문에 장애 복구에 영향을 끼칠 수도 있습니다.

- Stopped(중지됨) (MySQL 또는 MariaDB만 해당)—고객 요청으로 복제가 중단되었습니다.
- replication stop point set(복제 중지 지점이 설정됨) (MySQL만 해당)—고객이 시작한 중지 지점이 [mysql.rds_start_replication_until \(p. 689\)](#) 저장 프로시저를 사용하여 설정되었으며 복제가 진행 중입니다.
- replication stop point reached(복제 중지 지점에 도달함) (MySQL만 해당)—고객이 시작한 중지 지점이 [mysql.rds_start_replication_until \(p. 689\)](#) 저장 프로시저를 사용하여 설정되었으며 중지 지점에 도달했기 때문에 복제가 중지됩니다.

Amazon CloudWatch에서 Amazon RDS ReplicaLag 지표를 보고 복제 지연을 모니터링할 수 있습니다. MySQL과 MariaDB에서는 ReplicaLag 지표가 Seconds_Behind_Master 명령의 SHOW SLAVE STATUS 필드 값을 보고합니다. 그리고, PostgreSQL에서는 ReplicaLag 지표가 SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS slave_lag의 값을 보고합니다.

이렇게 MySQL 및 MariaDB에서 복제 지연이 발생하는 공통 원인은 다음과 같습니다.

- 네트워크 중단.
- 읽기 전용 복제본 관련 인덱스를 할당하여 테이블에 쓰기 작업 중일 때. 읽기 전용 복제본에서 `read_only` 파라미터가 0으로 설정되어 있지 않으면 복제가 중단될 수 있습니다.

- MyISAM과 같은 비트랜잭션 스토리지 엔진 사용. 복제는 MySQL용 InnoDB 스토리지 엔진 및 MariaDB용 XtraDB 스토리지 엔진에 대해서만 지원됩니다.

ReplicaLag 측정치가 0에 도달하면 복제본이 원본 DB 인스턴스를 따라잡은 것입니다. ReplicaLag 지표가 -1을 반환하는 경우 복제가 현재 활성이 아닙니다. ReplicaLag = -1은 Seconds_Behind_Master = NULL과 동등합니다.

Oracle의 경우, ReplicaLag 지표는 Apply Lag 값과 현재 시간과 적용 지역의 DATUM_TIME 값 차이를 합한 값입니다. DATUM_TIME 값은 읽기 전용 복제본이 마지막으로 원본 DB 인스턴스에서 데이터를 수신한 시간입니다. 자세한 내용은 Oracle 설명서에서 [V\\$DATAGUARD_STATS](#)를 참조하십시오.

PostgreSQL(버전 9.4.7 및 9.5.2 이상)은 물리적인 복제 슬롯을 사용하여 원본 인스턴스에서 Write Ahead Log(WAL) 보존을 관리합니다. Amazon RDS가 리전 간 읽기 전용 복제본 인스턴스마다 물리적 복제 슬롯을 생성하여 인스턴스와 연동시킵니다. 2개의 Amazon CloudWatch 지표인 oldest Replication Slot Lag와 Transaction Logs Disk Usage는 수신되는 WAL 데이터와 관련하여 가장 지체된 복제본이 얼마나 오래되었는지, 그리고 현재 WAL 데이터에 얼마나 많은 스토리지가 사용되고 있는지 나타냅니다. Transaction Logs Disk Usage 값은 리전 간 읽기 전용 복제본이 많이 지체될 수록 크게 증가합니다.

CloudWatch를 통한 DB 인스턴스 모니터링에 대한 자세한 정보는 [Amazon CloudWatch를 사용하여 모니터링 \(p. 230\)](#) 단원을 참조하십시오.

옵션 그룹 작업

DB 엔진 중에는 데이터베이스 엔진을 더욱 쉽게 관리할 뿐만 아니라 데이터베이스 보안을 더욱 강화한 추가 기능을 가진 엔진도 있습니다. Amazon RDS는 옵션 그룹을 사용하여 이러한 기능을 활성화하고 구성합니다. 옵션 그룹은 Amazon RDS DB 인스턴스에서 사용 가능한 옵션이라는 기능을 지정할 수 있습니다. 옵션은 여러 설정을 통해 옵션의 활용 방식을 지정합니다. 이후 DB 인스턴스와 옵션 그룹을 연동시키면 지정한 옵션과 옵션 설정이 연동된 DB 인스턴스에서 활성화됩니다.

Amazon RDS는 다음 데이터베이스 엔진 옵션을 지원합니다.

데이터베이스 엔진	관련 문서
MariaDB	MariaDB 데이터베이스 엔진을 위한 옵션 (p. 467)
Microsoft SQL Server	Microsoft SQL Server 데이터베이스 엔진의 옵션 (p. 544)
MySQL	MySQL DB 인스턴스 옵션 (p. 663)
Oracle	Oracle DB 인스턴스 옵션 (p. 772)

옵션 그룹 개요

Amazon RDS는 새로운 DB 인스턴스마다 비어있는 기본 옵션 그룹을 제공합니다. 이 기본 옵션 그룹은 수정할 수 없지만 생성되는 새로운 옵션 그룹은 기본 옵션 그룹에서 그 설정을 도출합니다. 옵션을 DB 인스턴스에 적용하려면 다음을 수행해야 합니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 하나 이상의 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동시킵니다.

DB 인스턴스와 DB 스냅샷은 모두 옵션 그룹과 연동시킬 수 있습니다. 어떤 경우에는 DB 스냅샷에서 복원하거나 DB 인스턴스에 대해 특정 시점으로의 복원을 수행할 수 있습니다. 이 경우 DB 스냅샷 또는 DB 인스턴스에 연결된 옵션 그룹은 기본적으로 복원된 DB 인스턴스와 연결됩니다. 다른 옵션 그룹을 복구된 DB 인스

터스와 연동 시킬 수 있습니다. 다만 새로운 옵션 그룹에는 원래 옵션 그룹에 포함되었던 지속적 또는 영구적 옵션이 있어야 합니다. 지속적이거나 영구적인 옵션은 다음과 같이 설명됩니다.

DB 인스턴스에서 옵션을 실행하려면 추가 메모리가 필요합니다. 따라서 DB 인스턴스의 현재 용도에 따라 옵션 사용을 위해 대용량 인스턴스를 시작해야 하는 경우가 있을 수 있습니다. 예를 들어 Oracle Enterprise Manager Database Control은 약 300MB의 RAM을 사용합니다. 스몰 DB 인스턴스에 대해 이 옵션을 활성화 할 경우 성능 문제 또는 메모리 부족 오류가 발생할 수 있습니다.

지속적이거나 영구적인 옵션

두 가지 유형인 지속적이거나 영구적인 옵션은 옵션 그룹에 추가할 때 세심한 주의가 필요합니다.

DB 인스턴스가 이 옵션 그룹과 연결되어 있는 동안에는 지속적인 옵션을 해당 옵션 그룹에서 제거할 수 없습니다. 지속적인 옵션의 한 가지 예는 Microsoft SQL Server Transparent Data Encryption(TDE)을 위한 TDE 옵션입니다. 옵션 그룹에서 지속적인 옵션을 제거하려면 먼저 모든 DB 인스턴스를 옵션 그룹에서 연동 해제해야 합니다. 어떤 경우에는 DB 스냅샷에서 복원 또는 특정 시점으로의 복원을 수행할 수 있습니다. 이러한 경우 해당 DB 스냅샷과 연결되어 있는 옵션 그룹에 지속적인 옵션이 포함된 경우에는 복원된 DB 인스턴스만 이 옵션 그룹과 연결할 수 있습니다.

반면 Oracle Advanced Security TDE를 위한 TDE 옵션처럼 영구적인 옵션은 옵션 그룹에서 절대로 제거할 수 없습니다. 영구적인 옵션을 사용 중인 DB 인스턴스의 옵션 그룹은 변경할 수 있습니다. 그러나 DB 인스턴스에 연결된 옵션 그룹에는 동일한 영구 옵션이 포함되어야 합니다. 어떤 경우에는 DB 스냅샷에서 복원 또는 특정 시점으로의 복원을 수행할 수 있습니다. 이러한 경우 해당 DB 스냅샷과 연결되어 있는 옵션 그룹에 영구적인 옵션이 포함된 경우에는 옵션 그룹이 있는 복원된 DB 인스턴스만 이 영구 옵션과 연결할 수 있습니다.

VPC 및 플랫폼 고려 사항

옵션 그룹이 DB 인스턴스에 할당되면 이 옵션 그룹은 DB 인스턴스가 있는 플랫폼에 연결됩니다. 이 플랫폼은 Amazon VPC 서비스가 지원하는 VPC이거나 Amazon EC2 서비스가 지원하는 EC2-Classic(비 VPC)일 수 있습니다. 이 두 가지 플랫폼에 대한 세부 정보는 [Amazon EC2 및 Amazon Virtual Private Cloud](#)를 참조하십시오.

DB 인스턴스가 VPC에 속한 경우 해당 인스턴스와 연결된 옵션 그룹은 바로 이 VPC에도 연결됩니다. 즉 다른 VPC나 다른 플랫폼으로 인스턴스를 복원하려고 해도 DB 인스턴스에 배정된 옵션 그룹은 사용할 수 없습니다. DB 인스턴스를 다른 VPC 또는 다른 플랫폼으로 복원하는 경우 다음 중 하나를 수행할 수 있습니다.

- 기본 옵션 그룹을 해당 DB 인스턴스에 배정.
- 연결된 옵션 그룹을 해당 VPC 또는 플랫폼에 배정.
- 새 옵션 그룹을 생성하여 해당 DB 인스턴스에 배정.

Oracle TDE와 같은 지속적 또는 영구적 옵션의 경우 DB 인스턴스를 다른 VPC로 복구할 때 지속적 또는 영구적 옵션을 포함하는 새 옵션 그룹을 생성해야 합니다.

옵션 특성은 옵션 설정에 따라 달라집니다. 예를 들어 Oracle 고급 보안 옵션인 `NATIVE_NETWORK_ENCRYPTION`은 DB 인스턴스로 전송 및 수신되는 네트워크 트래픽에 암호화 알고리즘을 지정할 수 있는 설정이 있습니다. 옵션 설정 중에는 Amazon RDS 사용에 최적화하여 변경할 수 없는 설정도 있습니다.

함께 사용할 수 없는 옵션

일부 옵션은 함께 사용할 수 없습니다. 한 옵션 또는 다른 옵션을 사용할 수 있지만 두 옵션을 동시에 사용할 수 없습니다. 함께 사용할 수 없는 옵션은 다음과 같습니다.

- Oracle Enterprise Manager Database Express (p. 791) 및 Oracle Management Agent for Enterprise Manager Cloud Control (p. 793).
- Oracle 기본 네트워크 암호화 (p. 807) 및 Oracle 보안 소켓 Layer (p. 810).

옵션 그룹 생성

기본 옵션 그룹에서 설정을 도출하는 새로운 옵션 그룹을 생성한 다음 하나 이상의 옵션을 새 옵션 그룹에 추가할 수 있습니다. 또는 이미 기존 옵션 그룹이 있는 경우에는 기존 옵션 그룹과 모든 옵션을 새 옵션 그룹에 복사할 수 있습니다. 자세한 정보는 [옵션 그룹 복사 \(p. 151\)](#) 단원을 참조하십시오.

생성된 새 옵션 그룹에는 옵션이 없습니다. 옵션을 옵션 그룹에 추가하는 방법을 배우려면 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#) 단원을 참조하십시오. 원하는 옵션을 추가했다면 이제 옵션 그룹을 DB 인스턴스에 연동시켜 옵션을 DB 인스턴스에서 사용할 수 있습니다. 옵션 그룹과 DB 인스턴스의 연동에 대한 자세한 정보는 [옵션 그룹 작업 \(p. 148\)](#)에 나열된 구체적인 해당 엔진 설명서단원을 참조하십시오.

콘솔

옵션 그룹을 생성하는 한 가지 방법은 AWS Management 콘솔을 사용하는 것입니다.

콘솔을 이용하여 새 옵션 그룹을 생성하려면,

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 옵션 그룹을 선택합니다.
3. Create group을 선택합니다.
4. 보안 그룹 생성 창에서 다음과 같이 합니다.
 - a. Name에 AWS 계정 내에서 쉽게 식별할 수 있는 옵션 그룹 이름을 입력합니다. 이름은 글자, 숫자 및 하이픈만 사용 가능합니다.
 - b. Description에 옵션 그룹에 대한 간략한 설명을 입력합니다. 이 설명은 표시 용도로만 사용됩니다.
 - c. Engine에서 원하는 DB 엔진을 선택합니다.
 - d. 메이저 엔진 버전에서 DB 엔진의 원하는 메이저 버전을 선택합니다.
5. 계속하려면 생성을 선택합니다. 작업을 취소하려면 Cancel을 선택합니다.

AWS CLI

옵션 그룹을 생성하려면 AWS CLI `create-option-group` 명령을 다음 필수 파라미터와 함께 사용합니다.

- `--option-group-name`
- `--engine-name`
- `--major-engine-version`
- `--option-group-description`

Example

다음은 `testoptiongroup`이라는 이름의 옵션 그룹을 생성하는 예제입니다. 이 옵션 그룹은 Oracle Enterprise Edition DB 엔진과 연동됩니다. 설명은 인용 부호로 둑여 있습니다.

Linux, OS X, Unix의 경우:

```
aws rds create-option-group \
--option-group-name testoptiongroup \
--engine-name oracle-ee \
--major-engine-version 12.1 \
--option-group-description "Test option group"
```

Windows의 경우:

```
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

옵션 그룹을 생성하려면 Amazon RDS API [CreateOptionGroup](#) 작업을 호출합니다. 다음 파라미터를 포함합니다.

- OptionGroupName
- EngineName
- MajorEngineVersion
- OptionGroupDescription

옵션 그룹 복사

AWS CLI 또는 Amazon RDS API를 사용하여 옵션 그룹 사본을 만들 수 있습니다. 기존 옵션 그룹이 있지만 해당 그룹의 사용자 지정 파라미터 및 값의 대부분을 새 옵션 그룹에 포함하려는 경우 옵션 그룹을 복사하면 편리합니다. 프로덕션 환경에서 사용하는 옵션 그룹 사본을 만든 다음 사본 설정을 변경하여 다른 옵션 설정을 테스트할 수도 있습니다.

AWS CLI

옵션 그룹을 복사하려면 AWS CLI [copy-option-group](#) 명령을 사용합니다. 다음 필수 파라미터를 포함합니다.

- --source-option-group-identifier
- --target-option-group-identifier
- --target-option-group-description

Example

다음은 new-local-option-group이라는 이름의 옵션 그룹을 생성하는 예제입니다. 이 옵션 그룹은 옵션 그룹 my-remote-option-group의 로컬 사본입니다.

Linux, OS X, Unix의 경우:

```
aws rds copy-option-group \
--source-option-group-identifier arn:aws:rds:us-west-2:123456789012:og:my-remote-
option-group \
--target-option-group-identifier new-local-option-group \
--target-option-group-description "Option group 2"
```

Windows의 경우:

```
aws rds copy-option-group ^
--source-option-group-identifier arn:aws:rds:us-west-2:123456789012:og:my-remote-
option-group ^
--target-option-group-identifier new-local-option-group ^
--target-option-group-description "Option group 2"
```

RDS API

옵션 그룹을 복사하려면 Amazon RDS API [CopyOptionGroup](#) 작업을 호출합니다. 다음 필수 파라미터를 포함합니다.

- `SourceOptionGroupIdentifier`
- `TargetOptionGroupIdentifier`
- `TargetOptionGroupDescription`

옵션을 옵션 그룹에 추가하기

옵션을 기존 옵션 그룹에 추가할 수 있습니다. 원하는 옵션을 추가했다면 이제 옵션 그룹을 DB 인스턴스에 연동시켜 옵션을 DB 인스턴스에서 사용할 수 있습니다. 옵션 그룹과 DB 인스턴스의 연동에 대한 자세한 정보는 [옵션 그룹 작업 \(p. 148\)](#)에 나열된 구체적인 해당 DB 엔진 설명서를 참조하십시오.

다음 두 가지 경우에 옵션 그룹 변경 사항이 즉시 적용되어야 합니다.

- OEM 옵션처럼 포트 값을 추가하거나 업데이트하는 옵션을 추가하는 경우.
- 포트 값이 포함된 옵션이 있는 옵션 그룹을 추가하거나 삭제하는 경우.

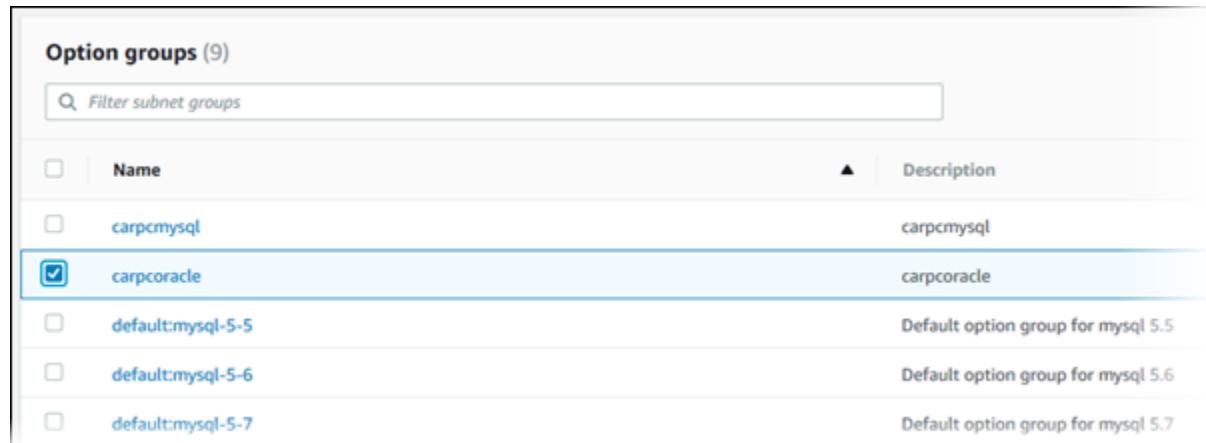
이러한 경우 콘솔에서 즉시 적용 옵션을 선택합니다. 또는 AWS CLI를 사용할 때 `--apply-immediately` 옵션을 포함하거나 Amazon RDS API를 사용할 때 `ApplyImmediately` 파라미터를 `true`로 설정할 수 있습니다. 포트 값이 포함되지 않은 옵션도 즉시 적용하거나, 혹은 DB 인스턴스의 다음 유지 관리 기간에 적용할 수 있습니다.

콘솔

AWS Management 콘솔을 사용하여 옵션을 옵션 그룹에 추가할 수 있습니다.

콘솔을 사용하여 옵션을 옵션 그룹에 추가하려면,

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 옵션 그룹을 선택합니다.
3. 변경하려는 옵션 그룹을 선택한 후 옵션 추가를 선택합니다.



Option groups (9)		
<input type="text"/> Filter subnet groups		
<input type="checkbox"/>	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. 옵션 추가 창에서 다음과 같이 합니다.

- a. 추가할 옵션을 선택합니다. 선택하는 옵션에 따라 값을 추가로 입력해야 하는 경우도 있습니다. 예를 들어 OEM 옵션을 선택하면 포트 값을 입력하고 DB 보안 그룹을 지정해야 합니다.

- b. 옵션을 추가하는 즉시 연동된 모든 DB 인스턴스에서 옵션을 활성화하려면 Apply Immediately에서 Yes를 선택합니다. No(기본 설정)를 선택하면 다음 유지 관리 기간에 연동된 모든 DB 인스턴스에서 옵션이 활성화됩니다.

Add Option

Option details

Option group name
carporacle

Option
Name of Option you want to add to this group
OEM

Port
The port number, if applicable, to use when connecting to the Option
1158

Security Groups
A list of VPC or DB Security Groups for which this Option is enabled
Choose security groups
default X

Apply Immediately info
 Yes
 No

Add Option

5. 원하는 대로 설정이 되었으면 Add Option을 선택합니다.

AWS CLI

옵션을 옵션 그룹에 추가하려면 추가할 옵션과 함께 AWS CLI [add-option-to-option-group](#) 명령을 실행합니다. 새로운 옵션을 연동된 모든 DB 인스턴스에서 즉시 활성화하려면 `--apply-immediately` 파라미터를 추가합니다. 기본적으로 이 옵션은 다음 유지 관리 기간에 연동된 모든 DB 인스턴스에서 활성화됩니다. 다음 필수 파라미터를 포함합니다.

- `--option-group-name`

Example

다음은 `testoptiongroup`이라는 이름의 옵션 그룹에 Oracle Enterprise Manager Database Control(OEM) 옵션을 추가한 후 즉시 활성화하는 예제입니다. 기본 보안 그룹을 사용하더라도 반드시 해당 보안 그룹을 지정해야 합니다.

Linux, OS X, Unix의 경우:

```
aws rds add-option-to-option-group \
--option-group-name testoptiongroup \
--options OptionName=OEM,Port=5500,DBSecurityGroupMemberships=default \
--apply-immediately
```

Windows의 경우:

```
aws rds add-option-to-option-group ^
--option-group-name testoptiongroup ^
--options OptionName=OEM,Port=5500,DBSecurityGroupMemberships=default ^
--apply-immediately
```

명령은 다음과 유사하게 출력됩니다.

```
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
DBSECURITYGROUPMEMBERSHIPS  default authorized
```

Example

다음 예에서는 Oracle OEM 옵션을 옵션 그룹에 추가합니다. 또한 사용자 지정 포트를 지정하고 이 포트에 사용할 한 쌍의 Amazon EC2 VPC 보안 그룹을 지정합니다.

Linux, OS X, Unix의 경우:

```
aws rds add-option-to-option-group \
--option-group-name testoptiongroup \
--options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2" \
--apply-immediately
```

Windows의 경우:

```
aws rds add-option-to-option-group ^
--option-group-name testoptiongroup ^
--options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2" ^
--apply-immediately
```

명령은 다음과 유사하게 출력됩니다.

```
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
VPCSECURITYGROUPMEMBERSHIPS  active  sg-test1
VPCSECURITYGROUPMEMBERSHIPS  active  sg-test2
```

Example

다음은 Oracle 옵션 NATIVE_NETWORK_ENCRYPTION을 옵션 그룹에 추가한 후 옵션 설정을 지정하는 예제입니다. 지정할 옵션 설정이 없는 경우에는 기본 값을 사용합니다.

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

명령은 다음과 유사하게 출력됩니다.

```
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

Amazon RDS API를 사용하여 옵션을 옵션 그룹에 추가하려면 추가할 옵션과 함께 [ModifyOptionGroup](#) 작업을 호출합니다. 새로운 옵션을 연동된 모든 DB 인스턴스에서 즉시 활성화하려면 `ApplyImmediately` 파라미터를 추가하고 `true`로 설정합니다. 기본적으로 이 옵션은 다음 유지 관리 기간에 연동된 모든 DB 인스턴스에서 활성화됩니다. 다음 필수 파라미터를 포함합니다.

- `OptionGroupName`

옵션 그룹의 옵션 및 옵션 설정 표시하기

옵션 그룹의 옵션과 옵션 설정을 모두 표시할 수 있습니다.

콘솔

AWS Management 콘솔을 사용하여 옵션 그룹의 옵션과 옵션 설정을 모두 표시할 수 있습니다.

옵션 그룹의 옵션과 옵션 설정을 표시하는 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 옵션 그룹을 선택합니다. 테이블의 Options 열에 옵션 그룹에 포함된 옵션과 옵션 설정이 표시됩니다.

AWS CLI

옵션 그룹의 옵션과 옵션 설정을 표시하려면 AWS CLI `describe-option-groups` 명령을 사용합니다. 표시할 옵션과 설정이 포함된 옵션 그룹 이름을 지정합니다. 옵션 그룹 이름을 지정하지 않으면 모든 옵션 그룹이 표시됩니다.

Example

다음은 모든 옵션 그룹에 포함된 옵션과 옵션 설정을 표시하는 예제입니다.

```
aws rds describe-option-groups
```

Example

다음은 `testoptiongroup`이라는 이름의 옵션 그룹에 포함된 옵션과 옵션 설정을 표시하는 예제입니다.

```
aws rds describe-option-groups --option-group-name testoptiongroup
```

RDS API

옵션 그룹의 옵션과 옵션 설정을 표시하려면 Amazon RDS API `DescribeOptionGroups` 작업을 사용합니다. 표시할 옵션과 설정이 포함된 옵션 그룹 이름을 지정합니다. 옵션 그룹 이름을 지정하지 않으면 모든 옵션 그룹이 표시됩니다.

옵션 설정 변경

옵션 설정을 변경할 수 있는 옵션은 추가한 이후에도 언제든지 설정을 변경할 수 있습니다. 옵션 그룹의 옵션 또는 옵션 설정을 변경할 경우에는 해당 옵션 그룹과 연동되어 있는 모든 DB 인스턴스에 변경 사항이 적용됩니다. 다양한 옵션에서 이용할 수 있는 설정에 대한 자세한 정보는 [옵션 그룹 작업 \(p. 148\)](#)에 나열된 구체적인 해당 엔진 설명서를 참조하십시오.

다음 두 가지 경우에 옵션 그룹 변경 사항이 즉시 적용되어야 합니다.

- OEM 옵션처럼 포트 값을 추가하거나 업데이트하는 옵션을 추가하는 경우.
- 포트 값이 포함된 옵션이 있는 옵션 그룹을 추가하거나 삭제하는 경우.

이러한 경우 콘솔에서 즉시 적용 옵션을 선택합니다. 또는 AWS CLI를 사용할 때 `--apply-immediately` 옵션을 포함하거나 RDS API를 사용할 때 `ApplyImmediately` 파라미터를 `true`로 설정할 수 있습니다. 포트 값이 포함되지 않은 옵션도 즉시 적용하거나, 혹은 DB 인스턴스의 다음 유지 관리 기간에 적용할 수 있습니다.

콘솔

AWS Management 콘솔을 사용하여 옵션 설정을 변경할 수 있습니다.

콘솔을 사용하여 옵션 설정을 변경하려면,

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 옵션 그룹을 선택합니다.
3. 옵션을 수정하려는 옵션 그룹을 선택한 후 옵션 수정을 선택합니다.
4. 옵션 수정 창의 설치된 옵션에서 설정을 수정하려는 옵션을 선택합니다. 원하는 설정을 변경합니다.
5. 옵션을 추가하는 즉시 활성화하려면 Apply Immediately에서 Yes를 선택합니다. No(기본 설정)를 선택하면 다음 유지 관리 기간에 연동된 모든 DB 인스턴스에서 옵션이 활성화됩니다.
6. 원하는 대로 설정이 되었으면 Modify Option을 선택합니다.

AWS CLI

옵션 설정을 수정하려면 변경할 옵션 그룹 및 옵션과 함께 AWS CLI `add-option-to-option-group` 명령을 사용합니다. 기본적으로 이 옵션은 다음 유지 관리 기간에 연동된 모든 DB 인스턴스에서 활성화됩니다. 변경 사항을 연동된 모든 DB 인스턴스에 즉시 적용하려면 `--apply-immediately` 파라미터를 포함합니다. 옵션 설정을 변경하려면 `--settings` 인수를 사용합니다.

Example

다음은 `testoptiongroup`이라는 이름의 옵션 그룹에서 Oracle Enterprise Manager Database Control(OEM)이 사용하는 포트 설정을 변경한 다음 이 변경 사항을 즉시 적용하는 예제입니다.

Linux, OS X, Unix의 경우:

```
aws rds add-option-to-option-group \
--option-group-name testoptiongroup \
--options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default \
--apply-immediately
```

Windows의 경우:

```
aws rds add-option-to-option-group ^
--option-group-name testoptiongroup ^
--options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default ^
--apply-immediately
```

명령은 다음과 유사하게 출력됩니다.

```
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

다음은 Oracle 옵션 NATIVE_NETWORK_ENCRYPTION의 설정을 변경하는 예제입니다.

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

명령은 다음과 유사하게 출력됩니다.

```
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

옵션 설정을 수정하려면 변경할 옵션 그룹 및 옵션과 함께 Amazon RDS API [ModifyOptionGroup](#) 명령을 사용합니다. 기본적으로 이 옵션은 다음 유지 관리 기간에 연동된 모든 DB 인스턴스에서 활성화됩니다. 변경 사항을 연동된 모든 DB 인스턴스에 즉시 적용하려면 `ApplyImmediately` 파라미터를 포함시키고 `true`로 설정합니다.

옵션을 옵션 그룹에서 제거하기

일부 옵션은 옵션 그룹에서 제거할 수 있지만 그렇지 않은 옵션도 있습니다. 지속적인 옵션은 DB 인스턴스와 해당 옵션 그룹의 연동을 해제해야만 옵션 그룹에서 제거가 가능합니다. 영구적인 옵션은 옵션 그룹에서 절대로 제거할 수 없습니다. 제거할 수 있는 옵션에 대한 자세한 정보는 [옵션 그룹 작업 \(p. 148\)](#)에 나열된 구체적인 해당 엔진 설명서를 참조하십시오.

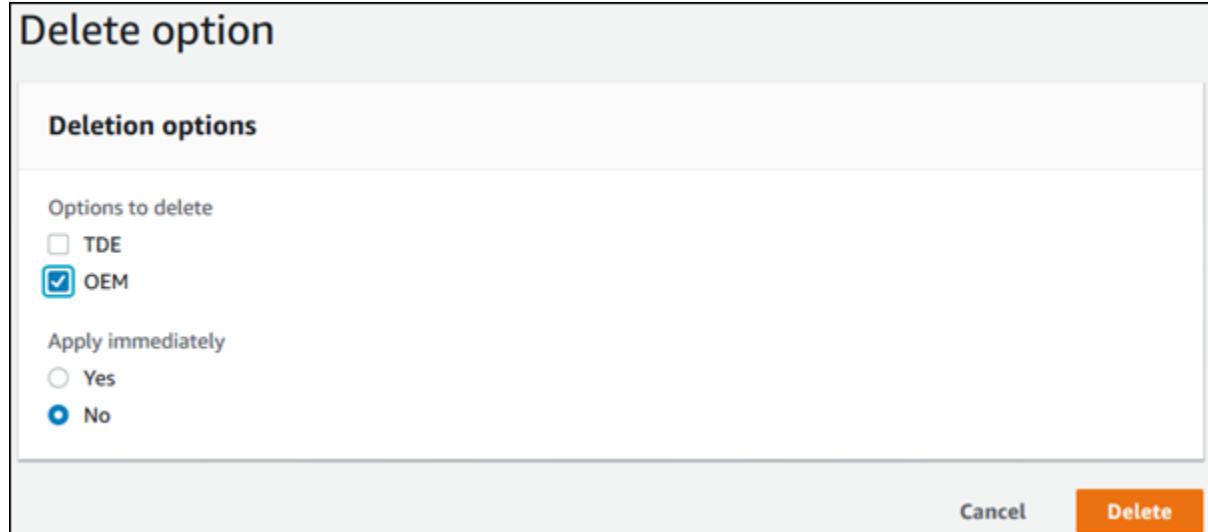
옵션 그룹에서 모든 옵션을 제거하더라도 Amazon RDS가 옵션 그룹을 삭제하지는 않습니다. 비어 있는 옵션 그룹과 연동되어 있는 DB 인스턴스도 활성화된 옵션이 없을 뿐 연동 상태는 유지합니다. 또는 DB 인스턴스를 기본(비어있는) 옵션 그룹과 연동시키면 DB 인스턴스에서 모든 옵션을 제거할 수 있습니다.

콘솔

AWS Management 콘솔을 사용하여 옵션 그룹에서 옵션을 제거할 수 있습니다.

console을 사용하여 옵션 그룹에서 옵션을 제거하려면,

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 옵션 그룹을 선택합니다.
3. 옵션을 제거하려는 옵션 그룹을 선택한 후 옵션 삭제를 선택합니다.
4. 옵션 삭제 창에서 다음과 같이 합니다.
 - 삭제하려는 옵션 확인란을 선택합니다.
 - 삭제 후 바로 적용하려면 즉시 적용에서 예를 선택합니다. No(기본 설정)를 선택하면 다음 유지 관리 기간에 연동된 모든 DB 인스턴스에서 옵션이 삭제됩니다.



5. 원하는 대로 설정이 되었으면 Yes, Delete를 선택합니다.

AWS CLI

옵션을 옵션 그룹에서 제거하려면 삭제할 옵션과 함께 AWS CLI `remove-option-from-option-group` 명령을 사용합니다. 기본적으로 이 옵션은 다음 유지 관리 기간에 연동된 모든 DB 인스턴스에서 제거됩니다. 변경 사항을 즉시 적용하려면 `--apply-immediately` 파라미터를 추가합니다.

Example

다음은 `testoptiongroup`이라는 이름의 옵션 그룹에서 Oracle Enterprise Manager Database Control(OEM) 옵션을 제거한 후 변경 사항을 즉시 적용하는 예제입니다.

Linux, OS X, Unix의 경우:

```
aws rds remove-option-from-option-group \
```

```
--option-group-name testoptiongroup \
--options OEM \
--apply-immediately
```

Windows의 경우:

```
aws rds remove-option-from-option-group ^
--option-group-name testoptiongroup ^
--options OEM ^
--apply-immediately
```

명령은 다음과 유사하게 출력됩니다.

```
OPTIONGROUP      testoptiongroup oracle-ee    12.1      Test option group
```

RDS API

옵션을 옵션 그룹에서 제거하려면 Amazon RDS API [ModifyOptionGroup](#) 작업을 사용합니다. 기본적으로 이 옵션은 다음 유지 관리 기간에 연동된 모든 DB 인스턴스에서 제거됩니다. 변경 사항을 즉시 적용하려면 `ApplyImmediately` 파라미터를 추가하고 `true`로 설정합니다.

다음 파라미터를 포함합니다.

- `OptionGroupName`
- `OptionsToRemove.OptionName`

옵션 그룹 삭제

Amazon RDS 리소스와 연결되지 않은 옵션 그룹은 삭제할 수 있습니다. 옵션 그룹은 DB 인스턴스, 수동 DB 스냅샷 또는 자동화된 DB 스냅샷에 연결할 수 있습니다.

Amazon RDS 리소스와 연결된 옵션 그룹을 삭제하려고 하면 다음과 유사한 오류가 반환됩니다.

```
An error occurred (InvalidOptionGroupStateException) when calling the DeleteOptionGroup operation: The option group 'optionGroupName' cannot be deleted because it is in use.
```

옵션 그룹과 연결된 Amazon RDS 리소스를 찾으려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 옵션 그룹을 선택합니다.
3. 세부 정보를 표시할 옵션 그룹의 이름을 선택합니다.
4. 연결된 Amazon RDS 리소스의 연결된 인스턴스 및 스냅샷 섹션을 확인하십시오.

DB 인스턴스가 옵션 그룹과 연결되어 있으면 DB 인스턴스가 다른 옵션 그룹을 사용하도록 수정하십시오. 자세한 내용은 [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#) 단원을 참조하십시오.

수동 DB 스냅샷이 옵션 그룹과 연결되어 있으면 AWS CLI [modify-db-snapshot](#) 명령을 사용해 DB 스냅샷이 다른 옵션 그룹을 사용하도록 수정하십시오.

Note

자동화된 DB 스냅샷의 옵션 그룹은 수정할 수 없습니다.

콘솔

옵션 그룹을 삭제하는 한 가지 방법은 AWS Management 콘솔을 사용하는 것입니다.

콘솔을 사용해 옵션 그룹을 삭제하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 옵션 그룹을 선택합니다.
3. 옵션 그룹을 선택합니다.
4. 그룹 삭제를 선택합니다.
5. 확인 페이지에서 삭제를 선택하여 옵션 그룹 삭제를 완료하거나 취소를 선택하여 삭제를 취소합니다.

AWS CLI

옵션 그룹을 삭제하려면 AWS CLI `delete-option-group` 명령을 다음 필수 파라미터와 함께 사용합니다.

- `--option-group-name`

Example

다음 예제에서는 이름이 `testoptiongroup`인 옵션 그룹을 삭제합니다.

Linux, OS X, Unix의 경우:

```
aws rds delete-option-group \
--option-group-name testoptiongroup
```

Windows의 경우:

```
aws rds delete-option-group ^
--option-group-name testoptiongroup
```

RDS API

옵션 그룹을 삭제하려면 Amazon RDS API `DeleteOptionGroup` 작업을 호출합니다. 다음 파라미터를 포함합니다.

- `OptionGroupName`

DB 파라미터 그룹 작업

DB 인스턴스를 파라미터 그룹과 연결하여 DB 엔진 구성을 관리합니다. Amazon RDS는 새로 생성된 DB 인스턴스에 적용되는 기본 설정으로 파라미터 그룹을 정의합니다. 맞춤형 설정으로 자신만의 파라미터 그룹을 정의할 수 있습니다. 그런 다음 DB 인스턴스를 수정해 이 파라미터 그룹을 사용하게 할 수 있습니다.

DB 파라미터 그룹은 하나 이상의 DB 인스턴스에 적용되는 엔진 구성 값의 컨테이너 역할을 합니다.

DB 파라미터 그룹을 지정하지 않고 DB 인스턴스를 만드는 경우 DB 인스턴스에서는 기본 DB 파라미터 그룹을 사용합니다. 각 기본 DB 파라미터 그룹에는 인스턴스의 엔진, 컴퓨팅 클래스 및 할당된 스토리지에 따른 데이터베이스 엔진 기본값과 Amazon RDS 시스템 기본값이 들어 있습니다. 기본 DB 파라미터 그룹의 파라미터 설정은 수정할 수 없습니다. 그 대신에 자신만의 파라미터 설정을 선택하는 경우 자신만의 파라미터 그룹을 생성합니다. 고객님이 생성하는 파라미터 그룹에서 모든 DB 엔진 파라미터를 변경할 수 있는 것은 아닙니다.

자신만의 파라미터 그룹을 사용하고 싶다면 새 파라미터 그룹을 생성하고 사용하려는 파라미터를 수정하십시오. 그런 다음 DB 인스턴스를 수정하여 새로운 파라미터 그룹을 사용하십시오. DB 파라미터 그룹 내의 파라미터를 업데이트하는 경우 변경 사항은 이 파라미터 그룹과 연결된 모든 DB 인스턴스에 적용됩니다.

AWS CLI [rds-copy-db-parameter-group](#) 명령으로 기존 DB 파라미터 그룹을 복사할 수 있습니다. 기존 DB 파라미터 그룹의 사용자 지정 파라미터 및 값 대부분을 새로운 DB 파라미터 그룹에 포함할 때는 파라미터 그룹을 복사하는 것이 편리할 수 있습니다.

다음은 DB 파라미터 그룹의 파라미터를 사용한 작업에 관한 몇 가지 주요 사항입니다.

- 동적 파라미터를 변경하고 DB 파라미터 그룹을 저장하면 즉시 적용 설정에 관계없이 변경 내용이 바로 적용됩니다. 고정 파라미터를 변경하고 DB 파라미터 그룹을 저장한 후 DB 인스턴스를 수동으로 재부팅하면 파라미터 변경 내용이 적용됩니다. RDS 콘솔을 사용하거나 명시적으로 `RebootDbInstance` API 작업을 호출하여 DB 인스턴스를 재부팅할 수 있습니다(DB 인스턴스가 다른 AZ 배포에 있는 경우 장애 조치 없음). 고정 파라미터 변경 후 연결된 DB 인스턴스를 재부팅하도록 하면 `ModifyDBInstance`를 호출하여 DB 인스턴스 클래스를 변경하거나 스토리지를 조정하는 경우와 같이 잘못된 파라미터 구성이 API 호출에 영향을 주는 위험을 완화할 수 있습니다.

DB 인스턴스에서 연결된 DB 파라미터 그룹에 대한 최신 변경 내용을 사용하고 있지 않은 경우 AWS Management 콘솔에 DB 파라미터 그룹이 재시작 보류 중 상태로 표시됩니다. 재시작 보류 중 파라미터 그룹 상태로 인해 다음 번 유지 관리 기간 중에 자동 재부팅이 되지는 않습니다. 최신 파라미터 변경 내용을 이 DB 인스턴스에 적용하려면 해당 DB 인스턴스를 수동으로 재부팅해야 합니다.

- DB 인스턴스와 연결된 DB 파라미터 그룹을 변경하면 DB 인스턴스에서 새 DB 파라미터 그룹을 사용하기 전에 인스턴스를 수동으로 재부팅해야 합니다.
- DB 파라미터에 대해 수식, 변수, 함수 및 연산자로 구성된 정수 또는 정수 식으로 지정할 수 있습니다. 함수에 수학 로그식을 넣을 수 있습니다. 자세한 내용은 [DB 파라미터 값 \(p. 170\)](#) 단원을 참조하십시오.
- DB 인스턴스를 만들기 전 및 DB 인스턴스에 데이터베이스를 만들기 전에, 파라미터 그룹에 있는 데이터베이스의 문자 세트 또는 콜레이션과 관련된 파라미터를 모두 설정해야 합니다. 이렇게 하면 DB 인스턴스의 기본 데이터베이스와 새 데이터베이스가 지정한 문자 세트 및 데이터 정렬 값을 사용하게 됩니다. DB 인스턴스의 문자 세트 또는 데이터 정렬 파라미터를 변경해도 기존 데이터베이스에는 변경된 파라미터가 적용되지 않습니다.

다음과 같이 `ALTER DATABASE` 명령을 사용하여 기존 데이터베이스의 문자 세트 또는 데이터 정렬 값을 변경할 수 있습니다.

```
ALTER DATABASE database_name CHARACTER SET character_set_name COLLATE collation;
```

- DB 파라미터 그룹에 파라미터를 잘못 설정하면 성능 저하나 시스템 불안정 등의 의도하지 않은 부작용이 있을 수 있습니다. 데이터베이스 파라미터를 수정할 때 항상 주의를 기울이고 DB 파라미터 그룹을 수정하기 전에 데이터를 백업하십시오. 파라미터 그룹 변경 내용을 프로덕션 DB 인스턴스에 적용하기 전에 테스트 DB 인스턴스에 적용해 봐야 합니다.

주제

- [DB 파라미터 그룹 생성 \(p. 163\)](#)
- [DB 파라미터 그룹의 파라미터 수정 \(p. 164\)](#)
- [DB 파라미터 그룹 복사 \(p. 166\)](#)

- DB 파라미터 그룹 나열 (p. 168)
- DB 파라미터 그룹의 파라미터 값 보기 (p. 169)
- DB 파라미터 그룹 비교 (p. 170)
- DB 파라미터 값 (p. 170)

DB 파라미터 그룹 생성

AWS Management 콘솔, AWS CLI 또는 RDS API를 사용하여 새 DB 파라미터 그룹을 생성할 수 있습니다.

콘솔

DB 파라미터 그룹을 생성하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 파라미터 그룹을 선택합니다.
3. [Create parameter group]을 선택합니다.
파라미터 그룹 생성 창이 나타납니다.
4. 파라미터 그룹 패밀리 목록에서 DB 파라미터 그룹 패밀리를 선택합니다.
5. 유형 목록에서 DB 파라미터 그룹을 선택합니다.
6. 그룹 이름 상자에 새 DB 파라미터 그룹의 이름을 입력합니다.
7. 설명 상자에 새 DB 파라미터 그룹에 대한 설명을 입력합니다.
8. 생성을 선택합니다.

AWS CLI

DB 파라미터 그룹을 생성하려면 AWS CLI `create-db-parameter-group` 명령을 사용합니다. 다음 예에서는 "My new parameter group"이라는 설명과 함께 mydbparametergroup이라는 MySQL 버전 5.6용 DB 파라미터 그룹을 생성합니다.

다음 필수 파라미터를 포함합니다.

- `--db-parameter-group-name`
- `--db-parameter-group-family`
- `--description`

사용 가능한 모든 DB 파라미터 그룹 패밀리를 나열하려면 다음 명령을 사용합니다.

```
aws rds describe-db-engine-versions --query "DBEngineVersions[].DBParameterGroupFamily"
```

Note

출력에 중복이 있습니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds create-db-parameter-group \
--db-parameter-group-name mydbparametergroup \
--db-parameter-group-family MySQL5.6 \
--description "My new parameter group"
```

Windows의 경우:

```
aws rds create-db-parameter-group ^
--db-parameter-group-name mydbparametergroup ^
--db-parameter-group-family MySQL5.6 ^
--description "My new parameter group"
```

다음과 비슷한 출력이 생성됩니다.

```
DBPARAMETERGROUP mydbparametergroup mysql5.6 My new parameter group
```

RDS API

DB 파라미터 그룹을 생성하려면 RDS API [CreateDBParameterGroup](#) 작업을 사용합니다.

다음 필수 파라미터를 포함합니다.

- DBParameterGroupName
- DBParameterGroupFamily
- Description

DB 파라미터 그룹의 파라미터 수정

고객이 생성한 DB 파라미터 그룹의 파라미터 값은 수정할 수 있지만, 기본 DB 파라미터 그룹의 파라미터 값은 변경할 수 없습니다. 고객이 생성한 DB 파라미터 그룹의 파라미터를 변경하면 DB 파라미터 그룹과 연결된 모든 DB 인스턴스에 해당 변경 내용이 적용됩니다.

파라미터 값을 변경할 경우 변경 내용이 적용되는 시기는 파라미터 유형에 의해 결정됩니다. 동적 파라미터의 변경 내용은 즉시 적용됩니다. 고정 파라미터를 변경할 경우 DB 파라미터 그룹과 연결된 DB 인스턴스를 재부팅해야 해당 변경 내용이 적용됩니다. 파라미터 유형을 확인하려면 [DB 파라미터 그룹 나열 \(p. 168\)](#) 섹션의 절차 중 하나를 사용하여 파라미터 그룹의 파라미터를 나열합니다.

RDS 콘솔에는 구성 탭에서 DB 인스턴스와 연결된 DB 파라미터 그룹의 상태가 표시됩니다. 예를 들어 DB 인스턴스에서 연결된 DB 파라미터 그룹에 대한 최신 변경 내용을 사용하고 있지 않은 경우 RDS 콘솔에 DB 파라미터 그룹이 재시작 보류중 상태로 표시됩니다. 최신 파라미터 변경 내용을 이 DB 인스턴스에 적용하려면 해당 DB 인스턴스를 수동으로 재부팅해야 합니다.

Details

Configurations

ARN
arn:aws:rds: [REDACTED]:db:orcl

Engine
Oracle Enterprise Edition 12.1.0.2.v8

License Model
Bring Your Own License

Created Time
Fri Aug 18 13:39:06 GMT-700 2017

DB Name
ORCL

Username
[REDACTED]

Character Set
AL32UTF8

Option Group
default:oracle-ee-12-1

Parameter group
oraclegp (pending-reboot)

Copy tags to snapshots
No

콘솔

DB 파라미터 그룹을 수정하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 파라미터 그룹을 선택합니다.
3. 목록에서 수정할 파라미터 그룹을 선택합니다.
4. 파라미터 그룹 작업에서 편집을 선택합니다.
5. 수정할 파라미터의 값을 변경합니다. 대화 상자 오른쪽 위의 화살표 키를 사용하여 파라미터를 스크롤할 수 있습니다.
기본 파라미터 그룹의 값은 변경할 수 없습니다.
6. [Save changes]를 선택합니다.

CLI

DB 파라미터 그룹을 수정하려면 AWS CLI [modify-db-parameter-group](#) 명령을 다음 필수 파라미터와 함께 사용합니다.

- `--db-parameter-group-name`
- `--parameters`

다음 예에서는 `mydbparametergroup`이라는 DB 파라미터 그룹에서 `max_connections` 및 `max_allowed_packet` 값을 수정합니다.

Note

Amazon RDS는 단일 파라미터에 대해 쉼표로 구분된 여러 파라미터 값 전달을 지원하지 않습니다.

Example

Linux, OS X, Unix의 경우:

```
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"
```

Windows의 경우:

```
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"
```

다음과 같은 출력이 생성됩니다.

```
DBPARAMETERGROUP mydbparametergroup
```

RDS API

DB 파라미터 그룹을 수정하려면 RDS API [ModifyDBParameterGroup](#) 명령을 다음 필수 파라미터와 함께 사용합니다.

- `DBParameterGroupName`
- `Parameters`

DB 파라미터 그룹 복사

생성하는 사용자 지정 DB 파라미터 그룹을 복사할 수 있습니다. DB 파라미터 그룹을 이미 생성했으며 해당 그룹의 사용자 지정 파라미터와 값의 대부분을 새 DB 파라미터 그룹에 포함하려는 경우 파라미터 그룹을 복사하면 편리합니다. AWS CLI [copy-db-parameter-group](#) 명령 또는 RDS API [CopyDBParameterGroup](#) 작업을 사용하여 DB 파라미터 그룹을 복사할 수 있습니다.

DB 파라미터 그룹을 복사한 후 5분 이상 기다렸다가 해당 DB 파라미터 그룹을 기본 파라미터 그룹으로 사용하는 첫 번째 DB 인스턴스를 생성하십시오. 이렇게 하면 파라미터 그룹이 사용되기 전에 Amazon RDS에서 복사 작업을 완전히 마칠 수 있습니다. 이는 DB 인스턴스의 기본 데이터베이스를 생성할 때 필수적인 파라미터에 특히 중요합니다. 한 가지 예는 `character_set_database` 파라미터로 정의되는 기본 데이터베이스

에 대한 문자 집합입니다. [Amazon RDS 콘솔](#)의 파라미터 그룹 옵션이나 [describe-db-parameters](#) 명령을 사용하여 DB 파라미터 그룹이 생성되었는지 확인하십시오.

Note

기본 파라미터 그룹은 복사할 수 없습니다. 하지만 기본 파라미터 그룹을 바탕으로 하는 새로운 파라미터 그룹을 만들 수 있습니다.

콘솔

DB 파라미터 그룹을 복사하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 파라미터 그룹을 선택합니다.
3. 목록에서 복사할 사용자 지정 파라미터 그룹을 선택합니다.
4. 파라미터 그룹 작업에서 복사를 선택합니다.
5. 새로운 DB 파라미터 그룹 식별자에 새로운 파라미터 그룹의 이름을 입력합니다.
6. 설명에 새로운 파라미터 그룹에 대한 설명을 입력합니다.
7. [Copy]를 선택합니다.

CLI

DB 파라미터 그룹을 복사하려면 AWS CLI [copy-db-parameter-group](#) 명령을 다음 필수 파라미터와 함께 사용합니다.

- `--source-db-parameter-group-identifier`
- `--target-db-parameter-group-identifier`
- `--target-db-parameter-group-description`

다음 예에서는 DB 파라미터 그룹 `mygroup1`을 복사하여 `mygroup2`라는 새 DB 파라미터 그룹을 생성합니다.

Example

Linux, OS X, Unix의 경우:

```
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"
```

Windows의 경우:

```
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

DB 파라미터 그룹을 복사하려면 RDS API [CopyDBParameterGroup](#) 작업을 다음 필수 파라미터와 함께 사용합니다.

- `SourceDBParameterGroupIdentifier`

- TargetDBParameterGroupIdentifier
- TargetDBParameterGroupDescription

DB 파라미터 그룹 나열

AWS 계정에 대해 생성한 DB 파라미터 그룹을 나열할 수 있습니다.

Note

특정 DB 엔진과 버전에 대한 DB 인스턴스를 생성할 때 기존 파라미터 템플릿에서 기본 파라미터 그룹이 자동으로 생성됩니다. 이 기본 파라미터 그룹은 기본 파라미터 설정을 포함하며 수정할 수 없습니다. 사용자 지정 파라미터 그룹을 생성할 때 파라미터 설정을 수정할 수 있습니다.

콘솔

AWS 계정에 대한 모든 DB 파라미터 그룹을 나열하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 파라미터 그룹을 선택합니다.

DB 파라미터 그룹이 목록에 나타납니다.

CLI

AWS 계정에 사용할 수 있는 모든 DB 파라미터 그룹을 나열하려면 AWS CLI `describe-db-parameter-groups` 명령을 사용합니다.

Example

다음 예에서는 AWS 계정에 사용할 수 있는 모든 DB 파라미터 그룹을 나열합니다.

```
aws rds describe-db-parameter-groups
```

다음과 같은 응답이 반환됩니다.

```
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
```

다음은 `mydbparamgroup1` 파라미터 그룹을 설명하는 예제입니다.

Linux, OS X, Unix의 경우:

```
aws rds describe-db-parameter-groups \
--db-parameter-group-name mydbparamgroup1
```

Windows의 경우:

```
aws rds describe-db-parameter-groups ^
--db-parameter-group-name mydbparamgroup1
```

다음과 같은 응답이 반환됩니다.

```
DBPARAMETERGROUP mydbparametergroup1  mysql5.5  My new parameter group
```

RDS API

AWS 계정에 사용할 수 있는 모든 DB 파라미터 그룹을 나열하려면 RDS API [DescribeDBParameterGroups](#) 작업을 사용합니다.

DB 파라미터 그룹의 파라미터 값 보기

DB 파라미터 그룹의 모든 파라미터와 해당 값 목록을 가져올 수 있습니다.

콘솔

DB 파라미터 그룹의 파라미터 값을 보려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 파라미터 그룹을 선택합니다.
DB 파라미터 그룹이 목록에 나타납니다.
3. 파라미터 그룹의 이름을 선택하여 파라미터 목록을 봅니다.

CLI

DB 파라미터 그룹의 파라미터 값을 보려면 AWS CLI [describe-db-parameters](#) 명령을 다음 필수 파라미터와 함께 사용합니다.

- `--db-parameter-group-name`

Example

다음 예에서는 mydbparametergroup이라는 DB 파라미터 그룹에 대한 파라미터와 파라미터 값을 나열합니다.

```
aws rds describe-db-parameters --db-parameter-group-name mydbparametergroup
```

다음과 같은 응답이 반환됩니다.

DBPARAMETER	Parameter Name	Parameter Value	Source	Data Type	Apply
DBPARAMETER	Type	Is Modifiable			
DBPARAMETER	allow-suspicious-udfs	false	engine-default	boolean	static
DBPARAMETER	auto_increment_increment	true	engine-default	integer	dynamic
DBPARAMETER	auto_increment_offset	true	engine-default	integer	dynamic
DBPARAMETER	binlog_cache_size	32768	system	integer	dynamic
DBPARAMETER	socket	/tmp/mysql.sock	system	string	static

RDS API

DB 파라미터 그룹의 파라미터 값을 보려면 RDS API [DescribeDBParameters](#) 명령을 다음 필수 파라미터와 함께 사용하십시오.

- `DBParameterGroupName`

DB 파라미터 그룹 비교

AWS Management 콘솔을 사용하여 동일한 DB 엔진 및 버전의 두 파라미터 그룹간 차이를 확인할 수 있습니다.

두 파라미터 그룹을 비교하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 파라미터 그룹을 선택합니다.
3. 목록에서 비교하려는 두 파라미터 그룹을 선택합니다.
4. 파라미터 그룹 작업에서 비교를 선택합니다.

DB 파라미터 값

DB 파라미터의 값으로 다음 중 어느 것이든 지정할 수 있습니다.

- 정수 상수
- DB 파라미터 수식
- DB 파라미터 함수
- 문자열 상수
- `value={log(DBInstanceClassMemory/8187281418)*1000}`과 같은 로그식(로그 함수는 log base 2로 표시됨)

DB 파라미터 수식

DB 파라미터 수식은 정수 값 또는 부울 값으로 확인되는 식이며 중괄호({})로 둘입니다. DB 파라미터 함수에 대한 인수나 DB 파라미터 값에 대해 수식을 지정할 수 있습니다.

구문

```
{FormulaVariable}  
{FormulaVariable*Integer}  
{FormulaVariable*Integer/Integer}  
{FormulaVariable/Integer}
```

DB 파라미터 수식 변수

각 수식 변수는 정수 또는 부울 값을 반환합니다. 변수 이름은 대소문자를 구분합니다.

AllocatedStorage

데이터 볼륨의 크기를 바이트 단위로 반환합니다.

DBInstanceClassMemory

현재 DB 인스턴스와 연결된 DB 인스턴스에 할당된 메모리에서 인스턴스를 관리하는 Amazon RDS 프로세스에 사용되는 메모리를 뱀 바이트 수를 반환합니다.

EndPointPort

DB 인스턴스에 연결하는 데 사용되는 포트의 번호를 반환합니다.

DBInstanceClassHugePagesDefault

부울 값 반환 현재, Oracle 엔진에 대해서만 지원됩니다.

자세한 내용은 [Oracle DB 인스턴스에 방대한 페이지 사용 \(p. 721\)](#) 단원을 참조하십시오.

DB 파라미터 수식 연산자

DB 파라미터 수식은 나눗셈과 곱셈의 두 가지 연산자를 지원합니다.

나눗셈 연산자: /

나눗수를 나눗수로 나누어 정수 봄을 반환합니다. 봄의 소수는 잘리며 반올림되지 않습니다.

구문

```
dividend / divisor
```

나눗수 및 나눗수 인수는 정수 식이어야 합니다.

곱셈 연산자: *

표현식을 곱하여 해당 표현식의 곱을 반환합니다. 표현식 소수는 잘리며 반올림되지 않습니다.

구문

```
expression * expression
```

두 식 모두 정수여야 합니다.

DB 파라미터 함수

정수나 수식으로 파라미터 인수를 지정할 수 있습니다. 함수마다 인수가 하나 이상 있어야 합니다. 쉼표로 구분된 목록으로 여러 인수를 지정할 수 있습니다. 목록에 argument1, argument3과 같은 빈 멤버를 넣을 수 없습니다. 함수 이름은 대소문자를 구분하지 않습니다.

Note

현재는 AWS CLI에서 DB 파라미터 함수가 지원되지 않습니다.

IF()

인수를 반환합니다.

현재, Oracle 엔진에 대해서만 지원하고 지원하는 첫 인수는 {DBInstanceClassHugePagesDefault}입니다. 자세한 내용은 [Oracle DB 인스턴스에 방대한 페이지 사용 \(p. 721\)](#) 단원을 참조하십시오.

구문

```
IF(argument1, argument2, argument3)
```

첫 번째 인수가 true이면 두 번째 인수를 반환합니다. 그렇지 않으면 세 번째 인수를 반환합니다.

GREATEST()

정수 또는 파라미터 수식 목록에서 가장 큰 값을 반환합니다.

구문

```
GREATEST(argument1, argument2,...argumentn)
```

정수를 반환합니다.

LEAST()

정수 또는 파라미터 수식 목록에서 가장 작은 값을 반환합니다.

구문

```
LEAST(argument1, argument2,...argumentn)
```

정수를 반환합니다.

SUM()

지정된 정수나 파라미터 수식의 값을 더합니다.

구문

```
SUM(argument1, argument2,...argumentn)
```

정수를 반환합니다.

DB 파라미터 값 예

다음 예에서는 DB 파라미터에 대한 값에 수식과 함수를 사용하는 방법을 보여 줍니다.

Warning

DB 파라미터 그룹에 파라미터를 잘못 설정하면 성능 저하나 시스템 불안정 등의 의도하지 않은 부작용이 있을 수 있습니다. 데이터베이스 파라미터를 수정할 때 항상 주의하고 DB 파라미터 그룹을 수정하기 전에 데이터를 백업하십시오. 파라미터 그룹 변경 내용을 프로덕션 DB 인스턴스에 적용하기 전에 특정 시점으로 복원을 사용하여 생성한 테스트 DB 인스턴스에 적용해 봄아 합니다.

Oracle 프로세스 파라미터에 GREATEST 함수를 지정하여 DBInstanceClassMemory를 9868951로 나눈 값과 80 중 더 큰 수로 사용자 프로세스 수를 설정할 수 있습니다.

```
GREATEST({DBInstanceClassMemory/9868951},80)
```

MySQL max_binlog_cache_size 파라미터 값에 LEAST() 함수를 지정하여 트랜잭션이 MySQL 인스턴스에서 사용할 수 있는 최대 캐시 값을 1MB와 DBInstanceClass/256 중 더 작은 값으로 설정할 수 있습니다.

```
LEAST({DBInstanceClassMemory/256},10485760)
```

Amazon RDS의 Amazon 리소스 이름(ARN)을 사용한 작업

Amazon Web Services에 생성되는 리소스는 각기 고유한 Amazon 리소스 이름(ARN)으로 식별됩니다. 특정 Amazon RDS 작업에서는 ARN을 지정하여 Amazon RDS 리소스를 고유한 이름으로 식별해야 합니다. 예를

들어, RDS DB 인스턴스 읽기 전용 복제본을 생성할 때에도 해당 소스 DB 인스턴스에 ARN을 부여해야 합니다.

Amazon RDS의 ARN 구성

Amazon Web Services에 생성되는 리소스는 각기 고유한 Amazon 리소스 이름(ARN)으로 식별됩니다. 다음 구문을 사용하여 Amazon RDS 리소스에 대한 ARN을 생성할 수 있습니다.

`arn:aws:rds:<region>:<account number>:<resourcetype>:<name>`

Region Name	Region	Endpoint	Protocol	
미국 동부 (오하이오)	us-east-2	rds.us-east-2.amazonaws.com	HTTPS	
미국 동부 (버지니아 북부)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS	
미국 서부 (캘리포니 아 북부)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS	
미국 서부 (오레곤)	us-west-2	rds.us-west-2.amazonaws.com	HTTPS	
아시아 태 평양(홍콩)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS	
아시아 태 평양(뭄바 이)	ap-south-1	rds.ap-south-1.amazonaws.com	HTTPS	
아시아 태 평양(오사 카-로컬)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com	HTTPS	
아시아 태 평양(서울)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com	HTTPS	
아시아 태 평양(싱가 포르)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com	HTTPS	
아시아 태 평양(시드 니)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com	HTTPS	
아시아 태 평양(도쿄)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS	
캐나다(중 부)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS	
중국(베이 징)	cn-north-1	rds.cn-north-1.amazonaws.com.cn	HTTPS	

Region Name	Region	Endpoint	Protocol	
중국(닝샤)	cn-northwest-1	rds.cn-northwest-1.amazonaws.com.cn	HTTPS	
EU(프랑크푸르트)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS	
EU(아일랜드)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS	
EU(런던)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS	
EU(파리)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS	
EU(스톡홀름)	eu-north-1	rds.eu-north-1.amazonaws.com	HTTPS	
남아메리카(상파울루)	sa-east-1	rds.sa-east-1.amazonaws.com	HTTPS	
AWS GovCloud(미국 동부)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com	HTTPS	
AWS GovCloud(US)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com	HTTPS	

다음 표에는 특정 Amazon RDS 리소스 유형에 대한 ARN 생성 시 사용해야 하는 형식이 나와 있습니다.

리소스 유형	ARN 형식
DB 인스턴스	arn:aws:rds:<region>:<account>:db:<name> 예: <div style="border: 1px solid black; padding: 2px;"><code>arn:aws:rds:us-east-2:123456789012:db:my-mysql-instance-1</code></div>
DB 클러스터	arn:aws:rds:<region>:<account>:cluster:<name> 예: <div style="border: 1px solid black; padding: 2px;"><code>arn:aws:rds:us-east-2:123456789012:cluster:my-aurora-cluster-1</code></div>
이벤트 구독	arn:aws:rds:<region>:<account>:es:<name> 예: <div style="border: 1px solid black; padding: 2px;"><code>arn:aws:rds:us-east-2:123456789012:es:my-subscription</code></div>
DB 옵션 그룹	arn:aws:rds:<region>:<account>:og:<name> 예:

리소스 유형	ARN 형식
	arn:aws:rds: us-east-2 : 123456789012 :og: my-og
DB 파라미터 그룹	arn:aws:rds:< region >:< account >:pg:< name > 예: arn:aws:rds: us-east-2 : 123456789012 :pg: my-param-enable-logs
DB 클러스터 파라미터 그룹	arn:aws:rds:< region >:< account >:cluster-pg:< name > 예: arn:aws:rds: us-east-2 : 123456789012 :cluster-pg: my-cluster-param-timezone
예약 DB 인스턴스	arn:aws:rds:< region >:< account >:ri:< name > 예: arn:aws:rds: us-east-2 : 123456789012 :ri: my-reserved-postgresql
DB 보안 그룹	arn:aws:rds:< region >:< account >:secgrp:< name > 예: arn:aws:rds: us-east-2 : 123456789012 :secgrp: my-public
DB 스냅샷	arn:aws:rds:< region >:< account >:snapshot:< name > 예: arn:aws:rds: us-east-2 : 123456789012 :snapshot: my-mysql-snap-20130507
DB 클러스터 스냅샷	arn:aws:rds:< region >:< account >:cluster-snapshot:< name > 예: arn:aws:rds: us-east-2 : 123456789012 :cluster-snapshot: my-aurora-snap-20160809
DB 서브넷 그룹	arn:aws:rds:< region >:< account >:subgrp:< name > 예: arn:aws:rds: us-east-2 : 123456789012 :subgrp: my-subnet-10

기존 ARN 가져오기

AWS Management 콘솔, AWS Command Line Interface(AWS CLI) 또는 RDS API를 사용해 RDS 리소스에 대한 ARN을 가져올 수 있습니다.

AWS Management 콘솔

AWS Management 콘솔에서 ARN을 확인하려면 ARN을 보려는 리소스를 탐색하거나, 해당 리소스의 세부 정보를 확인합니다. 예를 들어, 다음과 같이 DB 인스턴스 세부 정보의 구성 탭에서 DB 인스턴스의 ARN을 가져올 수 있습니다.

The screenshot shows the 'Details' page for an RDS instance. The ARN field, which contains the value 'arn:aws:rds:...', is highlighted with a large red oval. Other visible fields include Engine (Oracle Enterprise Edition 12.1.0.2.v8), License Model (Bring Your Own License), Created Time (Fri Aug 18 13:39:06 GMT-700 2017), DB Name (ORCL), Username (redacted), and Character Set (AL32UTF8).

AWS CLI

AWS CLI에서 특정 RDS 리소스에 대한 ARN을 가져오려면 해당 리소스에 `describe` 명령을 사용합니다. 다음 표에서는 각 ARN 명령, 그리고 ARN을 가져오기 위해 명령과 함께 사용하는 ARN 속성을 보여 줍니다.

AWS CLI 명령	ARN 속성
describe-event-subscriptions	EventSubscriptionArn
describe-certificates	CertificateArn
describe-db-parameter-groups	DBParameterGroupArn
describe-db-cluster-parameter-groups	DBClusterParameterGroupArn
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
describe-db-clusters	DBClusterArn
describe-db-cluster-snapshots	DBClusterSnapshotArn

예를 들어, 다음 AWS CLI 명령은 DB 인스턴스에 대한 ARN을 가져옵니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds describe-db-instances \
--db-instance-identifier DBInstanceIdentifier \
--region us-west-2
```

Windows의 경우:

```
aws rds describe-db-instances ^
--db-instance-identifier DBInstanceIdentifier ^
--region us-west-2
```

API

특정 RDS 리소스에 대한 ARN을 가져오려면, 다음과 같은 RDS API 작업을 호출하고 ARN 속성을 사용할 수 있습니다.

RDS API 작업	ARN 속성
DescribeEventSubscriptions	EventSubscriptionArn
DescribeCertificates	CertificateArn
DescribeDBParameterGroups	DBParameterGroupArn
DescribeDBClusterParameterGroups	DBClusterParameterGroupArn

RDS API 작업	ARN 속성
DescribeDBInstances	DBInstanceArn
DescribeDBSecurityGroups	DBSecurityGroupArn
DescribeDBSnapshots	DBSnapshotArn
DescribeEvents	SourceArn
DescribeReservedDBInstances	ReservedDBInstanceArn
DescribeDBSubnetGroups	DBSubnetGroupArn
DescribeOptionGroups	OptionGroupArn
DescribeDBClusters	DBClusterArn
DescribeDBClusterSnapshots	DBClusterSnapshotArn

Amazon RDS용 스토리지 DB 인스턴스 작업

Amazon RDS에 데이터를 저장하는 방식을 지정하려면 DB 인스턴스를 만들거나 수정할 때 스토리지 유형을 선택하고 스토리지 크기를 입력합니다. 추후에는 DB 인스턴스를 수정하여 스토리지의 양을 늘리거나 스토리지의 유형을 변경할 수 있습니다. 워크로드를 사용할 때 어떤 스토리지 유형을 사용할지에 대한 자세한 내용은 [Amazon RDS 스토리지 유형 \(p. 101\)](#) 단원을 참조하십시오.

주제

- [DB 인스턴스 스토리지 용량 증가 \(p. 178\)](#)
- [스토리지 유형 변경 \(p. 180\)](#)
- [프로비저닝된 IOPS SSD 스토리지 설정을 수정할 수 있습니다. \(p. 181\)](#)

DB 인스턴스 스토리지 용량 증가

추가 데이터에 대한 공간이 필요할 경우, 기존 DB 인스턴스의 스토리지를 확장할 수 있습니다. Amazon RDS Management Console, Amazon RDS API, 또는 AWS Command Line Interface(AWS CLI)를 사용하여 이 작업을 수행할 수 있습니다. 범용 SSD 또는 프로비저닝된 IOPS SSD 스토리지를 사용할 경우, 스토리지를 최대 16TiB까지 증가시킬 수 있습니다. SQL Server 데이터베이스 인스턴스용 Amazon RDS 스토리지를 확장하여 범용 SSD 또는 프로비저닝된 IOPS SSD 스토리지 유형에만 지원합니다.

필요한 경우 대응할 수 있도록 Amazon CloudWatch 경보를 생성하여 DB 인스턴스의 무료 스토리지 양을 모니터링하는 것이 좋습니다. CloudWatch 경보 설정에 대한 자세한 내용은 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오.

대부분의 경우, 스토리지를 확장할 때 어떠한 중단도 필요하지 않고 서버 성능을 저하하지 않습니다. DB 인스턴스에 대한 스토리지 크기를 수정하면 DB 인스턴스의 상태가 Storage-optimization으로 됩니다. 스토리지 수정 후 DB 인스턴스가 완전히 작동합니다. 하지만 6시간 동안 또는 DB 인스턴스 상태가 storage-optimization인 동안 중 더 긴 시간 동안 더 이상 스토리지를 수정할 수 없습니다.

SQL 서버 DB 인스턴스가 있고 2017년 11월부터 스토리지 구성을 수정하지 않았다면 할당된 스토리지를 증가시키기 위해 DB 인스턴스를 수정하고자 한다면 몇 분 동안의 짧게 중단될 수 있습니다. 중단된 후 DB 인스턴스는 온라인 상태이나 Storage-optimization 상태에 있습니다. 스토리지 최적화 중에 성능이 저하될 수 있습니다.

Note

할당된 후에 DB 인스턴스의 스토리지 양을 줄일 수 없습니다.

AWS Management 콘솔

DB 인스턴스 스토리지 증가

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 수정하려는 DB 인스턴스를 선택합니다.
4. [Modify]를 선택합니다.
5. 할당된 스토리지에 새로운 값을 입력합니다. 현재 값보다 커야 합니다.

Storage type

General Purpose (SSD)

Allocated storage

16384 GiB

This instance supports multiple storage ranges between 20 and 16384 GiB. [See all](#)

⚠ Scaling your instance storage can:

- Deplete the initial General Purpose (SSD) I/O credits, leading to longer conversion times. [Learn more](#)
- Impact instance performance until operation completes. [Learn more](#)

Note

할당된 스토리지를 늘릴 경우, 10% 이상 늘려야 합니다. 10% 미만으로 늘리면 오류가 발생합니다.

6. [Continue]를 선택하여 다음 화면으로 이동합니다.
7. 새 스토리지 유형을 사용하도록 DB 인스턴스의 변환을 즉시 시작하려면 Scheduling of modifications(수정 사항 예약) 섹션에서 즉시 적용을 선택합니다. 또는 [Apply during the next scheduled maintenance window]를 선택할 수도 있습니다.
8. 원하는 대로 설정이 되었으면 [Modify DB instance]를 선택합니다.

CLI

DB 인스턴스 스토리지를 늘리려면 AWS CLI `modify-db-instance` 명령을 사용합니다. 다음 파라미터를 설정합니다.

- `--allocated-storage` - DB 인스턴스용으로 할당할 스토리지의 양(GiB)
- `--apply-immediately` --apply-immediately를 사용하여 즉시 변환을 시작하거나 --no-apply-immediately(기본값)를 사용하여 다음 유지 관리 기간 중에 변환을 적용합니다. 변환이 적용될 때 즉시 종단됩니다. 스토리지에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 스토리지 \(p. 101\)](#) 단원을 참조하십시오.

API

DB 인스턴스 스토리지를 늘리려면 Amazon RDS API `ModifyDBInstance` 작업을 사용합니다. 다음 파라미터를 설정합니다.

- `AllocatedStorage` - DB 인스턴스용으로 할당할 스토리지의 양(GiB)
- `ApplyImmediately` - 즉시 변환을 시작하려면 이 옵션을 `True`로 설정합니다. 이 옵션이 `False`(기본값)로 설정된 경우에는 다음 유지 관리 기간 중에 변환이 확장됩니다. 변환이 적용될 때 즉시 종단됩니다.

스토리지에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 스토리지 \(p. 101\)](#) 단원을 참조하십시오.

스토리지 유형 변경

DB 인스턴스용 스토리지 유형은 AWS Management 콘솔, Amazon RDS API 또는 AWS Command Line Interface(AWS CLI)를 통해 변경할 수 있습니다.

스토리지 유형을 다른 스토리지 유형으로 변환하는 경우, 변환 프로세스가 시작될 때 중단이 발생할 수 있습니다. 이 경우 중단을 일으킬 수 있는 요소에 대해 알아보려면 [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#) 단원을 참조하십시오. 마이그레이션 지속 시간은 데이터베이스 로드, 스토리지 크기, 스토리지 유형 및 (있는 경우) 프로비저닝된 IOPS의 양과 같은 여러 가지 요인에 따라 다릅니다. 마이그레이션 시간은 일반적으로 몇 분 정도입니다. 마이그레이션 중에 DB 인스턴스를 사용할 수 있습니다.

그러나, 마그네틱 스토리지로 또는 해당 스토리지에서 마이그레이션할 경우에는 일반적으로 마이그레이션 시간이 더 걸리며, 경우에 따라 며칠이 걸릴 수도 있습니다. 마그네틱 스토리지로 또는 해당 스토리지에서 마이그레이션 중에 DB 인스턴스를 사용할 수 있지만, 성능 저하가 발생할 수 있습니다.

프로비저닝된 IOPS SSD 또는 마그네틱 스토리지에서 범용 SSD 스토리지로 스토리지 전환할 때는 범용 SSD 스토리지에 할당된 I/O 크레딧이 잠재적으로 고갈될 수도 있습니다. 특히 더 작은 볼륨의 경우도 해당 됩니다. 볼륨에 따라 초기 I/O 버스트 크레딧이 고갈된 후, 나머지 데이터는 할당된 범용 SSD 스토리지에서 GiB당 3IOPS의 기본 성능 속도로 전환됩니다. 이로 인해 전환 시간이 매우 길어질 수 있습니다.

AWS Management 콘솔

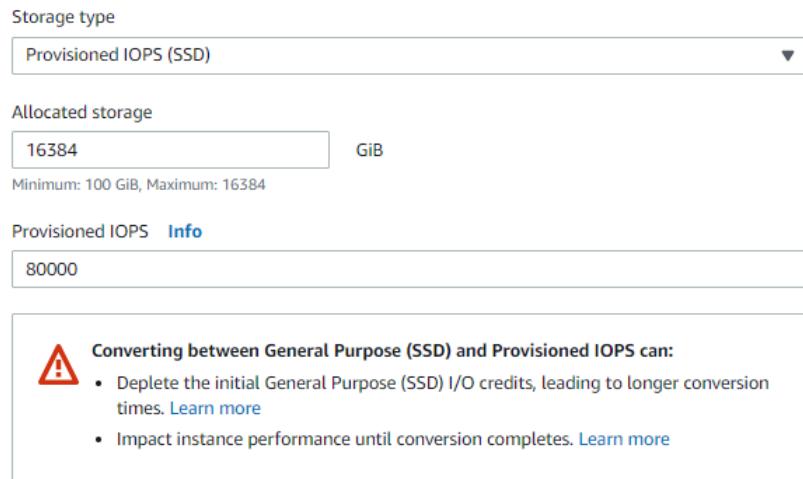
DB 인스턴스 스토리지 유형 변경

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.

Note

DB 인스턴스의 목록을 필터링하려면 Filter databases(데이터베이스 필터링)에 Amazon RDS 가 결과를 필터하는 데 사용할 텍스트 문자열을 입력합니다. 이름이 해당 문자열을 포함하는 DB 인스턴스만 표시됩니다.

3. 수정하려는 DB 인스턴스를 선택합니다.
4. [Modify]를 선택합니다.
5. Modify DB Instance page(DB 인스턴스 페이지 수정)에서 Storage type(스토리지 유형) 목록에서 스토리지 유형을 선택합니다. 프로비저닝된 IOPS SSD 스토리지 유형을 사용하도록 DB 인스턴스를 수정하려는 경우에는 프로비저닝된 IOPS 값도 입력합니다.



6. [Continue]를 선택합니다.
7. 변경 내용을 DB 인스턴스에 즉시 적용하려면 Scheduling of modifications(수정 사항 예약) 섹션에서 즉시 적용 확인란을 선택합니다. 또는 [Apply during the next scheduled maintenance window]를 선택할 수도 있습니다.
스토리지 유형을 변경할 경우 즉시 중단됩니다. 스토리지에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 스토리지 \(p. 101\)](#) 단원을 참조하십시오.
8. 변경될 파라미터를 검토하고 [Modify DB instance]를 선택하여 수정을 완료합니다.

CLI

DB 인스턴스 스토리지 유형을 변경하려면 AWS CLI `modify-db-instance` 명령을 사용합니다. 다음 파라미터를 설정합니다.

- `--storage-type` - 프로비저닝된 IOPS의 경우 `io1`로 설정합니다.
- `--apply-immediately` ---`apply-immediately`를 사용하여 즉시 변환을 시작합니다. `--no-apply-immediately`(기본값)를 사용하여 다음 유지 관리 기간 중에 변환을 적용합니다.

API

DB 인스턴스 스토리지 유형을 변경하려면 Amazon RDS API `ModifyDBInstance` 작업을 사용합니다. 다음 파라미터를 설정합니다.

- `StorageType` - 프로비저닝된 IOPS의 경우 `io1`로 설정합니다.
- `ApplyImmediately` - 즉시 변환을 시작하려면 이 옵션을 `True`로 설정합니다. 이 옵션이 `False`(기본값)로 설정된 경우에는 다음 유지 관리 기간 중에 변환이 적용됩니다.

프로비저닝된 IOPS SSD 스토리지 설정을 수정할 수 있습니다.

AWS Management Console, Amazon RDS API, 또는 AWS CLI를 사용하여 프로비저닝된 IOPS SSD 스토리지를 사용하는 DB 인스턴스의 설정을 수정할 수 있습니다. 필요한 스토리지 유형, 할당된 스토리지 및 프로비저닝된 IOPS의 양을 지정해야 합니다. 데이터베이스 엔진 및 인스턴스 유형에 따라 프로비저닝된 IOPS 및 스토리지 용량을 선택하십시오. 자세한 내용은 [프로비저닝된 IOPS SSD 스토리지 \(p. 103\)](#) 단원을 참조하십시오.

인스턴스에 따라 프로비저닝된 IOPS 양을 줄이더라도 할당된 범용 SSD 또는 마그네틱 스토리지 양을 줄일 수 없습니다.

AWS Management 콘솔

DB 인스턴스에 대한 프로비저닝된 IOPS 설정을 변경하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.

Note

DB 인스턴스의 목록을 필터링하려면 Filter databases(데이터베이스 필터링)에 Amazon RDS가 결과를 필터하는 데 사용할 텍스트 문자열을 입력합니다. 이름이 해당 문자열을 포함하는 DB 인스턴스만 표시됩니다.

3. 수정하려는 프로비저닝된 IOPS가 있는 DB 인스턴스를 선택합니다.
4. [Modify]를 선택합니다.
5. Modify DB Instance page(DB 인스턴스 수정 페이지)에서 Storage type(스토리지 유형)에 따른 프로비저닝된 IOPS를 선택한 후 프로비저닝된 IOPS 값을 입력합니다.

The screenshot shows the 'Storage type' dropdown set to 'Provisioned IOPS (SSD)'. Below it, the 'Allocated storage' field is set to 16384 GiB. A note indicates the minimum is 100 GiB and the maximum is 16384. The 'Provisioned IOPS' field is set to 80000. There are 'Info' and 'Edit' buttons next to the IOPS field.

[Allocated storage] 또는 [Provisioned IOPS]에 대해 지정하는 값이 다른 파라미터에 필요한 값의 범위를 나타내는 경고 메시지가 표시됩니다. 이 메시지는 다른 파라미터에 대하여 요구되는 값의 범위를 보여줍니다.

6. [Continue]를 선택합니다.
7. 변경 내용을 DB 인스턴스에 즉시 적용하려면 Scheduling of modifications(수정 사항 예약) 섹션에서 즉시 적용 확인란을 선택합니다. 또는 [Apply during the next scheduled maintenance window]를 선택할 수도 있습니다.

스토리지 유형을 변경할 경우 즉시 종단됩니다. 스토리지에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 스토리지 \(p. 101\)](#) 단원을 참조하십시오.

8. 변경될 파라미터를 검토하고 [Modify DB instance]를 선택하여 수정을 완료합니다.

할당된 스토리지나 프로비저닝된 IOPS에 대한 새 값은 [Status] 열에 나타납니다.

CLI

DB 인스턴스의 프로비저닝된 IOPS 설정을 변경하려면 AWS CLI `modify-db-instance` 명령을 사용합니다. 다음 파라미터를 설정합니다.

- `--storage-type` - 프로비저닝된 IOPS의 경우 `io1`로 설정합니다.
- `--allocated-storage` - DB 인스턴스용으로 할당할 스토리지의 양(GiB)
- `--iops` - DB 인스턴스에 대해 새로 설정하는 프로비저닝된 IOPS의 크기이며, 초당 I/O 작업 수로 표현됩니다.

- `--apply-immediately` `--apply-immediately`를 사용하여 즉시 변환을 시작합니다. `--no-apply-immediately`(기본값)를 사용하여 다음 유지 관리 기간 중에 변환을 적용합니다.

API

DB 인스턴스에 대한 프로비저닝된 IOPS 설정을 변경하려면 Amazon RDS API [ModifyDBInstance](#) 작업을 사용합니다. 다음 파라미터를 설정합니다.

- `StorageType` - 프로비저닝된 IOPS의 경우 `io1`로 설정합니다.
- `AllocatedStorage` - DB 인스턴스용으로 할당할 스토리지의 양(GiB)
- `Iops` - DB 인스턴스에 대해 새로 설정하는 IOPS 속도이며, 초당 I/O 작업 수로 표현됩니다.
- `ApplyImmediately` - 즉시 변환을 시작하려면 이 옵션을 `True`로 설정합니다. 이 옵션이 `False`(기본값)로 설정된 경우에는 다음 유지 관리 기간 중에 수정이 적용됩니다.

Amazon RDS에 대한 DB 인스턴스 결제

Amazon RDS 인스턴스는 다음 구성 요소를 기준으로 청구됩니다.

- DB 인스턴스 시간(시간별) – DB 인스턴스의 DB 인스턴스 클래스를 기준으로 합니다(예: db.t2.small 또는 db.m4.large). 요금은 시간 단위로 고시되지만, 청구서는 초 단위로 계산되고 시간을 10진수 형식으로 표시합니다. RDS 사용량은 1초 단위로 청구되며 최소 청구 시간은 10분입니다. 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.
- 스토리지(월별 GiB별) – DB 인스턴스에 프로비저닝한 스토리지 용량입니다. 해당 월에 프로비저닝된 스토리지 용량을 변경하는 경우 이에 따라 청구서 금액도 조정됩니다. 자세한 정보는 [Amazon RDS DB 인스턴스 스토리지 \(p. 101\)](#) 단원을 참조하십시오.
- I/O 요청(월별 1백만 요청별) – 결제 주기에 요청한 총 스토리지 I/O 요청 수입니다. Amazon RDS 마그네틱 스토리지만 해당됩니다.
- 프로비저닝된 IOPS(월별 IOPS별) – 사용된 IOPS에 상관없이 프로비저닝된 IOPS 비율입니다. Amazon RDS 프로비저닝된 IOPS(SSD) 스토리지만 해당됩니다. EBS 볼륨에 대해 프로비저닝된 스토리지는 1초 단위로 청구되며 최소 청구 시간은 10분입니다.
- 백업 스토리지(월별 GiB별) – 백업 스토리지는 자동화된 데이터베이스 백업 및 생성한 활성 데이터베이스 스냅샷과 연결된 스토리지입니다. 백업 보존 기간을 연장하거나 추가 데이터베이스 스냅샷을 찍으면 데이터베이스가 사용하는 백업 스토리지가 증가합니다. 초 단위 결제는 백업 스토리지(GB-월 단위로 측정됨)에는 적용되지 않습니다.

자세한 내용은 [Amazon RDS DB 인스턴스 백업 및 복원 \(p. 196\)](#) 단원을 참조하십시오.

- 데이터 전송(GB별) – DB 인스턴스와 인터넷 및 기타 AWS 리전 간의 양방향 데이터 전송입니다.

Amazon RDS는 사용자가 요구 사항에 따라 비용을 최적화할 수 있도록 다음과 같은 구입 옵션을 제공합니다.

- 온디맨드 인스턴스 – 사용하는 DB 인스턴스 시간에 대해 시간별로 지불합니다. 요금은 시간 단위로 고시되지만, 청구서는 초 단위로 계산되고 시간을 10진수 형식으로 표시합니다. RDS 사용량은 이제 1초 단위로 청구되며 최소 청구 시간은 10분입니다.
- 예약 인스턴스 – 1년 또는 3년 기간으로 DB 인스턴스를 예약하고 온디맨드 DB 인스턴스 요금에 비해 상당한 할인을 받습니다. 예약 인스턴스 사용 시 여러 인스턴스를 1시간 내에 시작, 삭제, 사용 또는 종료하고 모든 인스턴스에 대해 예약 인스턴스 혜택을 적용받을 수 있습니다.

Amazon RDS 요금에 대한 자세한 정보는 [Amazon RDS 제품 페이지](#)를 참조하십시오.

주제

- [온 디맨드 DB 인스턴스 \(p. 184\)](#)
- [예약 DB 인스턴스 \(p. 184\)](#)

온 디맨드 DB 인스턴스

Amazon RDS 온디맨드 DB 인스턴스는 DB 인스턴스 클래스를 기반으로 청구됩니다(예: db.t2.small 또는 db.m4.large). DB 인스턴스 시간을 일부만 사용해도 전체 시간으로 결제됩니다. Amazon RDS 요금에 대한 자세한 정보는 [Amazon RDS 제품 페이지](#)를 참조하십시오.

DB 인스턴스가 사용 가능하면 즉시 DB 인스턴스에 대한 청구가 시작됩니다. 요금은 시간 단위로 고시되지만, 청구서는 초 단위로 계산되고 시간을 10진수 형식으로 표시합니다. Amazon RDS 사용량은 1초 단위로 청구되며 최소 청구 시간은 10분입니다. 컴퓨팅 또는 스토리지 용량 조정과 같은 청구 대상 구성 변경의 경우 최소 시간인 10분에 대해 요금이 부과됩니다. DB 인스턴스를 삭제하거나 DB 인스턴스에 장애가 발생하여 DB 인스턴스가 종료될 때까지 청구가 계속됩니다.

DB 인스턴스 요금이 더 이상 부과되지 않도록 하려면 추가 DB 인스턴스 시간에 대해 청구되지 않도록 인스턴스를 종지하거나 삭제해야 합니다. 청구되는 DB 인스턴스 상태에 대한 자세한 정보는 [DB 인스턴스 상태 \(p. 96\)](#) 단원을 참조하십시오.

중지된 DB 인스턴스

DB 인스턴스가 중지되는 동안 프로비저닝된 IOPS를 포함하여 프로비저닝된 스토리지에 대해 요금이 부과됩니다. 지정된 보존 기간 내의 수동 스냅샷 및 자동 백업용 스토리지를 포함하여 백업 스토리지에 대한 요금도 부과됩니다. DB 인스턴스 시간에 대해서는 요금이 부과되지 않습니다.

다중 AZ DB 인스턴스

DB 인스턴스가 다중 AZ 배포가 되도록 지정하면 Amazon RDS 요금 페이지에 게시된 다중 AZ 요금에 따라 청구됩니다.

예약 DB 인스턴스

예약 DB 인스턴스를 사용하면 1년 또는 3년 단위로 DB 인스턴스를 예약할 수 있습니다. 예약 DB 인스턴스는 온디맨드 DB 인스턴스 요금과 비교하여 대폭 할인된 요금을 제공합니다. 예약 DB 인스턴스는 물리적 인스턴스가 아니고 오히려 계정에서 온디맨드 DB 인스턴스를 사용할 때 적용되는 결제 할인에 가깝습니다. 예약 DB 인스턴스의 할인 요금은 인스턴스 유형 및 AWS 리전에 따라 결정됩니다.

DB 인스턴스를 예약하기 위한 프로세스는 다음과 같습니다. 먼저 구매할 수 있는 DB 인스턴스 예약 상품에 대한 정보를 확인합니다. 그런 다음 DB 인스턴스 예약 상품을 구매하고 마지막으로 기존에 예약되어 있는 DB 인스턴스에 대한 정보를 확인합니다.

예약 DB 인스턴스 개요

Amazon RDS에서 예약 DB 인스턴스를 구매할 때는 예약 DB 인스턴스의 기간 동안 특정 DB 인스턴스 유형에 대해 할인 요금을 이용하는 약정을 구매하는 것입니다. Amazon RDS 예약 DB 인스턴스를 사용하려면 온디맨드 인스턴스와 똑같은 방법으로 새로운 DB 인스턴스를 생성해야 합니다. 새롭게 생성한 DB 인스턴스는 예약 DB 인스턴스의 사양과 일치해야 합니다. 새로운 DB 인스턴스의 사양이 계정의 기존 DB 예약 인스턴스와 일치하면 예약 DB 인스턴스에 제공되는 할인 요금이 청구됩니다. 그렇지 않으면 DB 인스턴스에 대해 온디맨드 요금이 청구됩니다.

요금을 포함하여 예약 DB 인스턴스에 대한 자세한 내용은 [Amazon RDS 예약 인스턴스](#)를 참조하십시오.

제공 유형

예약 DB 인스턴스는 세 가지 유형(No Upfront, Partial Upfront 및 All Upfront)으로 제공되며 예상되는 사용률에 따라 Amazon RDS 비용을 최적화할 수 있습니다.

선수금 없음

선결제 없이 예약 DB 인스턴스에 액세스할 수 있는 옵션입니다. 비선결제 예약 DB 인스턴스는 사용 기간 동안 사용량에 상관없이 할인된 시간당 요금이 청구되며, 선결제가 필요하지 않습니다. 이 옵션은 1년 예약만 가능합니다.

부분 선결제

예약 DB 인스턴스 사용비의 일부를 먼저 결제해야 하는 옵션입니다. 결제하지 않은 시간에 대해서는 사용 기간 동안 사용량에 상관없이 할인된 시간당 요금이 청구됩니다. 이 옵션은 이전 Heavy 사용률 옵션을 대신합니다.

전체 선결제

약관이 시작되는 시점에서 모든 금액을 결제하고 사용 기간 동안 추가 비용 없이 무제한으로 사용할 수 있습니다.

통합 결제를 사용하는 경우, 결제의 편의를 위해 조직 내 모든 계정은 하나의 계정으로 취급됩니다. 즉 조직 내 모든 계정은 다른 계정에서 구입한 예약 DB 인스턴스에 대해 시간당 비용 혜택을 받을 수 있습니다. 통합 결제에 대한 자세한 내용은 AWS 결제 및 비용 관리 사용 설명서에서 [Amazon RDS 예약 DB 인스턴스](#)를 참조하십시오.

유연한 크기의 예약 DB 인스턴스

예약 DB 인스턴스를 구매할 때 지정해야 하는 것 중 하나가 인스턴스 클래스(db.m4.large 등)입니다. 인스턴스 클래스에 대한 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.

이미 DB 인스턴스가 있지만 용량을 확장해야 하는 경우에는 예약 DB 인스턴스가 확장된 DB 인스턴스에 자동으로 적용됩니다. 다시 말해서 예약 DB 인스턴스는 모든 DB 인스턴스 클래스 크기에 자동으로 적용됩니다. 동일한 AWS 리전 및 데이터베이스 엔진에서 유연한 크기의 예약 DB 인스턴스를 DB 인스턴스에 사용할 수 있습니다. 유연한 크기의 예약 DB 인스턴스는 해당 인스턴스 클래스 유형에서만 확장할 수 있습니다. 예를 들어 db.m4.large의 예약 DB 인스턴스는 db.m4.large에 적용할 수 있지만, db.m5.large에는 적용할 수 없습니다. db.m4와 db.m5는 다른 인스턴스 클래스 유형이기 때문입니다. 이러한 예약 DB 인스턴스의 이점은 다중 AZ와 단일 AZ 구성 모두에게 적용됩니다.

유연한 크기의 예약 DB 인스턴스는 다음 데이터베이스 엔진에서 제공됩니다.

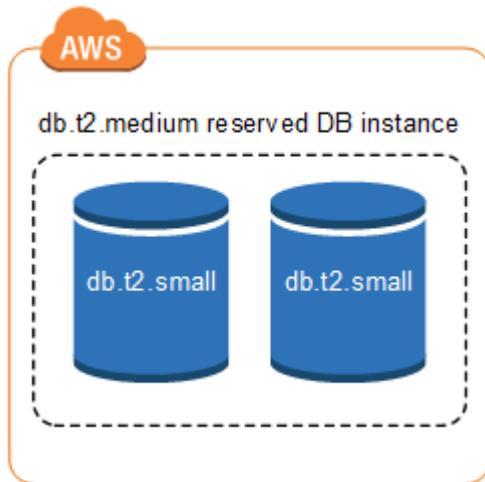
- MariaDB
- MySQL
- Oracle, Bring Your Own License
- PostgreSQL

예약 DB 인스턴스의 크기에 따른 사용량은 정규화 유닛을 사용하여 비교할 수 있습니다. 예를 들어 db.m3.large DB 인스턴스 2개일 때 사용량의 정규화 유닛 1개는 db.m3.small 1개일 때 사용량의 정규화 유닛 8개와 같습니다. 다음 표는 DB 인스턴스 크기에 따른 정규화 유닛의 수를 나타낸 것입니다.

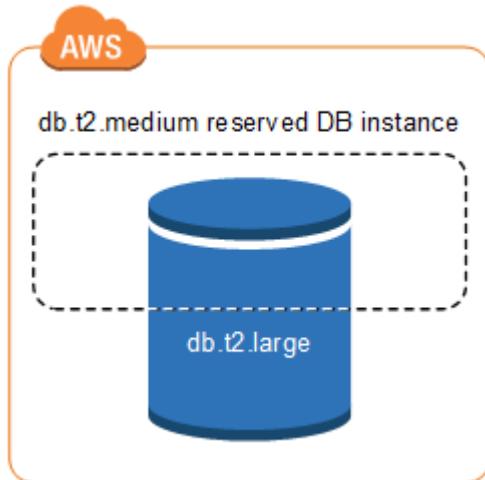
인스턴스 크기	단일 AZ 정규화 유닛	다중 AZ 정규화 유닛
micro	0.5	1
small	1	2
medium	2	4
large	4	8
xlarge	8	16
2xlarge	16	32

인스턴스 크기	단일 AZ 정규화 유닛	다중 AZ 정규화 유닛
4xlarge	32	64
8xlarge	64	128
10xlarge	80	160
16xlarge	128	256

예약 DB 인스턴스로 db.t2.medium을 1개 구매하고, 동일한 AWS 리전의 계정에서 db.t2.small DB 인스턴스를 2개 실행하는 경우를 예로 들어 보겠습니다. 이 경우 결제 혜택은 두 인스턴스에 100% 적용됩니다.



또는 동일한 AWS 리전의 계정에서 실행 중인 db.t2.large 인스턴스 1개가 있는 경우 결제 혜택은 DB 인스턴스 사용량의 50%에 적용됩니다.



예약 DB 인스턴스 결제의 예

예약 DB 인스턴스의 요금에는 스토리지, 백업, IO와 관련된 정기 비용이 포함되지 않습니다. 다음 예는 예약 DB 인스턴스의 월 총 요금을 보여 줍니다.

- 미국 동부(버지니아 북부)의 Amazon RDS MySQL 예약 단일 AZ db.r4.large DB 인스턴스 클래스, 선결제 없음 옵션 포함, 인스턴스 요금 0.12 USD(월 90 USD)

- 월 기준 GiB당 0.115 USD(월 45.60 USD)의 400GiB of 범용 SSD(gp2) 스토리지
- 0.095 USD(월 19 USD)의 600GiB 백업 스토리지(400GiB 무료)

예약 DB 인스턴스에 이러한 옵션을 모두 추가할 경우(90 USD + 45.60 USD + 19 USD), 월 총 요금은 154.60 USD입니다.

예약 DB 인스턴스 대신 온디맨드 DB 인스턴스를 사용하기로 선택한 경우, 미국 동부(버지니아 북부)의 Amazon RDS MySQL 단일 AZ db.r4.large DB 인스턴스 클래스 요금은 시간당 0.1386 USD(월 101.18 USD)입니다. 따라서 온디맨드 DB 인스턴스의 경우, 이러한 옵션을 모두 추가하면(101.18 USD + 45.60 USD + 19 USD), 월 총 요금은 165.78 USD입니다.

Note

이 예의 요금은 예제 요금이며 실제 요금과 다를 수 있습니다.

Amazon RDS 요금에 대한 자세한 내용은 [Amazon RDS 제품 페이지](#)를 참조하십시오.

예약 DB 인스턴스 삭제

예약 DB 인스턴스에 대한 약정 기간은 1년 또는 3년입니다. 예약 DB 인스턴스는 취소할 수 없습니다. 하지만 예약 DB 인스턴스 할인이 적용되는 DB 인스턴스를 삭제할 수는 있습니다. 예약 DB 인스턴스 할인이 적용되는 DB 인스턴스의 삭제 프로세스는 다른 DB 인스턴스를 삭제할 때와 동일합니다.

예약 DB 인스턴스에 대해 선결제를 하면 사용할 리소스가 예약됩니다. 이러한 리소스는 예약되므로 사용 여부에 관계없이 리소스에 대한 요금이 청구됩니다.

예약 DB 인스턴스 할인이 적용되는 DB 인스턴스를 삭제할 경우에는 다르지만 서로 사양이 호환되는 DB 인스턴스를 시작할 수 있습니다. 이 경우 예약 기간(1년 또는 3년)에 요금 할인을 계속 받을 수 있습니다.

콘솔

예약 DB 인스턴스에 대한 작업은 AWS Management 콘솔에서 다음 절차에 따라 진행할 수 있습니다.

사용 가능한 예약 DB 인스턴스 상품에 대한 요금과 정보를 가져오려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택합니다.
3. [Purchase Reserved DB Instance]를 선택합니다.
4. 제품 설명에서 DB 엔진과 라이선스 유형을 선택합니다.
5. DB 인스턴스 클래스에서 DB 인스턴스 클래스를 선택합니다.
6. 다중 AZ 배포에서 다중 AZ 배포를 사용할지 여부를 선택합니다.
7. [Term]에서 DB 인스턴스를 예약할 기간을 선택합니다.
8. 제공 유형에서 해당 제공 유형을 선택합니다.

상품 유형을 선택하면 요금 정보가 표시됩니다.

Important

취소를 선택하면 예약 DB 인스턴스를 구입하지 않으며 요금이 발생하지 않습니다.

구매할 수 있는 DB 인스턴스 예약 상품에 대한 정보를 확인하였으면 이제 정보를 사용하여 다음 절차에 따라 상품을 구매할 수 있습니다.

예약 DB 인스턴스를 구입하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.

2. 탐색 창에서 예약 인스턴스를 선택합니다.
3. [Purchase Reserved DB Instance]를 선택합니다.
4. 제품 설명에서 DB 엔진과 라이선스 유형을 선택합니다.
5. DB 인스턴스 클래스에서 DB 인스턴스 클래스를 선택합니다.
6. 다중 AZ 배포에서 다중 AZ 배포를 사용할지 여부를 선택합니다.
7. [Term]에서 DB 인스턴스를 예약할 기간을 선택합니다.
8. 제공 유형에서 해당 제공 유형을 선택합니다.

상품 유형을 선택하면 아래와 같이 요금 정보가 표시됩니다.

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

aurora-mysql

DB instance class

db.r4.4xlarge — 16 vCPU, 122 GiB RAM

Multi AZ deployment

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

Yes

No

Term

1 year

Offering type

All Upfront

Reserved Id (optional)

Optional tag to track your reservation

Number of DB instances

1

Pricing details

One-time payment (per instance)

Usage charges*

USD (hourly)

*Additional taxes may apply

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

Normalized units per hour [info](#)

32

Charges for your usage will appear on your monthly bill.

[Cancel](#)

[Continue](#)

9. (선택 사항) - 예약 DB 인스턴스를 조회할 수 있도록 구매하는 예약 인스턴스에 자체 식별자를 할당할 수 있습니다. [Reserved Id]에 자신이 예약한 DB 인스턴스 식별자를 입력하면 됩니다.
10. [Continue]를 선택합니다.

[Purchase Reserved DB Instance] 대화 상자가 나타나면서 아래와 같이 선택한 예약 DB 인스턴스 속성에 대한 요약 내용과 지불 기한이 각각 표시됩니다.

Purchase Reserved DB Instances

Summary of Purchase

You are about to purchase a Reserved DB Instance with the following information.

Region	US East (N. Virginia)
Product Description	aurora-mysql
DB Instance Class	db.r4.4xlarge
Offering Type	All Upfront
Multi AZ Deployment	No
Term	1 year
Reserved DB Instance	default
Quantity	1
Price Per Instance	[REDACTED]
Total Payment Due Now	[REDACTED]

⚠️ 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?

[Cancel](#) [Back](#) [Purchase](#)

11. 확인 페이지에서 자신이 예약한 DB 인스턴스를 확인합니다. 정보가 정확하면 [Purchase]를 선택하여 예약한 DB 인스턴스를 구매합니다.

또는 [Back]을 선택하여 예약한 DB 인스턴스를 편집합니다.

예약한 DB 인스턴스를 구매한 후에는 다음 절차에 따라 예약한 DB 인스턴스에 대한 정보를 가져올 수 있습니다.

AWS 계정에 대한 예약 DB 인스턴스 관련 정보를 가져오려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 예약 인스턴스를 선택합니다.

현재 계정에서 예약한 DB 인스턴스가 나타납니다. 특정 예약 DB 인스턴스의 세부 정보를 보려면 목록에서 해당 인스턴스를 선택합니다. 그러면 콘솔 아래쪽의 세부 정보 창에 인스턴스에 대한 세부 정보가 표시됩니다.

AWS CLI

예약 DB 인스턴스에 대한 작업은 다음 예제와 같이 AWS CLI를 사용하여 진행할 수 있습니다.

Example 사용 가능한 예약 DB 인스턴스 상품 보기

구매 가능한 DB 인스턴스 상품에 대한 정보를 가져오려면 AWS CLI 명령 `describe-reserved-db-instances-offerings`를 호출합니다.

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

이 호출은 다음과 비슷한 출력을 반환합니다.

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	mysql	Partial Upfront		
OFFERING	649fd0c8-cf6d-47a0-bfa6-060f8e75e95f	db.m1.small	n	1y	227.50
USD	0.046	mysql	Partial Upfront		
OFFERING	123456cd-ab1c-47a0-bfa6-12345667232f	db.m1.small	n	1y	162.00
USD	0.00	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	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	mysql	No Upfront		

구매할 수 있는 DB 인스턴스 예약 상품에 대한 정보를 확인하였으면 이제 정보를 사용하여 다음 예제와 같이 상품을 구매할 수 있습니다.

Example 예약 DB 인스턴스 구입

예약 DB 인스턴스를 구매하려면 다음 파라미터와 함께 AWS CLI 명령 `purchase-reserved-db-instances-offering`을 사용합니다.

- **--reserved-db-instances-offering-id** - 구매하려는 상품의 ID입니다. 위의 예제를 참조하여 상품 ID를 가져옵니다.
- **--reserved-db-instance-id** - 구매하는 예약 DB 인스턴스에 자체 식별자를 할당하여 관리할 수 있습니다.

다음은 ID가 **649fd0c8-cf6d-47a0-bfa6-060f8e75e95f**인 DB 인스턴스 예약 상품을 구매하고 식별자로 **MyReservation**을 할당하는 예제입니다.

Linux, OS X, Unix의 경우:

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

Windows의 경우:

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

이 명령은 다음과 비슷한 출력을 반환합니다.

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

예약한 DB 인스턴스를 구매한 후에는 다음 예제와 같이 예약한 DB 인스턴스에 대한 정보를 가져올 수 있습니다.

Example 예약 DB 인스턴스 보기

자신의 AWS 계정에서 예약 DB 인스턴스에 대한 정보를 가져오려면 AWS CLI 명령 [describe-reserved-db-instances](#)를 호출합니다.

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

이 명령은 다음과 비슷한 출력을 반환합니다.

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

예약 DB 인스턴스에 대한 작업은 다음 예제와 같이 RDS API를 사용하여 진행할 수 있습니다.

Example 사용 가능한 예약 DB 인스턴스 상품 보기

구매할 수 있는 DB 인스턴스 예약 상품에 대한 정보를 가져오려면 Amazon RDS API 함수 [DescribeReservedDBInstancesOfferings](#)를 호출합니다.

```
https://rds.us-east-1.amazonaws.com/
?Action=DescribeReservedDBInstancesOfferings
&SignatureMethod=HmacSHA256
&SignatureVersion=4
```

```
&Version=2014-09-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140411/us-east-1/rds/aws4_request
&X-Amz-Date=20140411T203327Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=545f04acffeb4b80d2e778526b1c9da79d0b3097151c24f28e83e851d65422e2
```

이 호출은 다음과 비슷한 출력을 반환합니다.

```
<DescribeReservedDBInstancesOfferingsResponse xmlns="http://rds.amazonaws.com/doc/2014-10-31/">
  <DescribeReservedDBInstancesOfferingsResult>
    <ReservedDBInstancesOfferings>
      <ReservedDBInstancesOffering>
        <Duration>31536000</Duration>
        <OfferingType>Partial Upfront</OfferingType>
        <CurrencyCode>USD</CurrencyCode>
        <RecurringCharges/>
        <FixedPrice>1820.0</FixedPrice>
        <ProductDescription>mysql</ProductDescription>
        <UsagePrice>0.368</UsagePrice>
        <MultiAZ>true</MultiAZ>
        <ReservedDBInstancesOfferingId>438012d3-4052-4cc7-b2e3-8d3372e0e706</
      ReservedDBInstancesOfferingId>
        <DBInstanceClass>db.m1.large</DBInstanceClass>
      </ReservedDBInstancesOffering>
      <ReservedDBInstancesOffering>
        <Duration>31536000</Duration>
        <OfferingType>Partial Upfront</OfferingType>
        <CurrencyCode>USD</CurrencyCode>
        <RecurringCharges/>
        <FixedPrice>227.5</FixedPrice>
        <ProductDescription>mysql</ProductDescription>
        <UsagePrice>0.046</UsagePrice>
        <MultiAZ>false</MultiAZ>
        <ReservedDBInstancesOfferingId>649fd0c8-cf6d-47a0-bfa6-060f8e75e95f</
      ReservedDBInstancesOfferingId>
        <DBInstanceClass>db.m1.small</DBInstanceClass>
      </ReservedDBInstancesOffering>
    </ReservedDBInstancesOfferings>
  </DescribeReservedDBInstancesOfferingsResult>
  <ResponseMetadata>
    <RequestId>5e4ec40b-2978-11e1-9e6d-771388d6ed6b</RequestId>
  </ResponseMetadata>
</DescribeReservedDBInstancesOfferingsResponse>
```

구매할 수 있는 DB 인스턴스 예약 상품에 대한 정보를 확인하였으면 이제 정보를 사용하여 다음 예제와 같이 상품을 구매할 수 있습니다.

Example 예약 DB 인스턴스 구입

예약 DB 인스턴스를 구매하려면 Amazon RDS API 작업 [PurchaseReservedDBInstancesOffering](#)을 다음 파라미터와 함께 호출합니다.

- --reserved-db-instances-offering-id - 구매하려는 상품의 ID입니다. 위의 예제를 참조하여 상품 ID를 가져옵니다.
- --reserved-db-instance-id - 구매하는 예약 DB 인스턴스에 자체 식별자를 할당하여 관리할 수 있습니다.

다음은 ID가 [649fd0c8-cf6d-47a0-bfa6-060f8e75e95f](#)인 DB 인스턴스 예약 상품을 구매하고 식별자로 [MyReservation](#)을 할당하는 예제입니다.

```
https://rds.us-east-1.amazonaws.com/
?Action=PurchaseReservedDBInstancesOffering
&ReservedDBInstanceId=MyReservation
&ReservedDBInstancesOfferingId=438012d3-4052-4cc7-b2e3-8d3372e0e706
&DBInstanceCount=10
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-09-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140415/us-east-1/rds/aws4_request
&X-Amz-Date=20140415T232655Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=c2ac761e8c8f54a8c0727f5a87ad0a766fbb0024510b9aa34ea6d1f7df52fb11
```

이 호출은 다음과 비슷한 출력을 반환합니다.

```
<PurchaseReservedDBInstancesOfferingResponse xmlns="http://rds.amazonaws.com/
doc/2014-10-31/">
  <PurchaseReservedDBInstancesOfferingResult>
    <ReservedDBInstance>
      <OfferingType>Partial Upfront</OfferingType>
      <CurrencyCode>USD</CurrencyCode>
      <RecurringCharges/>
      <ProductDescription>mysql</ProductDescription>
      <ReservedDBInstancesOfferingId>649fd0c8-cf6d-47a0-bfa6-060f8e75e95f</
      ReservedDBInstancesOfferingId>
      <MultiAZ>true</MultiAZ>
      <State>payment-pending</State>
      <ReservedDBInstanceId>MyReservation</ReservedDBInstanceId>
      <DBInstanceCount>10</DBInstanceCount>
      <StartTime>2011-12-18T23:24:56.577Z</StartTime>
      <Duration>31536000</Duration>
      <FixedPrice>123.0</FixedPrice>
      <UsagePrice>0.123</UsagePrice>
      <DBInstanceClass>db.m1.small</DBInstanceClass>
    </ReservedDBInstance>
  </PurchaseReservedDBInstancesOfferingResult>
  <ResponseMetadata>
    <RequestId>7f099901-29cf-11e1-bd06-6fe008f046c3</RequestId>
  </ResponseMetadata>
</PurchaseReservedDBInstancesOfferingResponse>
```

예약한 DB 인스턴스를 구매한 후에는 다음 예제와 같이 예약한 DB 인스턴스에 대한 정보를 가져올 수 있습니다.

Example 예약 DB 인스턴스 보기

자신의 AWS 계정에 대해 예약된 DB 인스턴스에 대한 자세한 내용을 보려면 Amazon RDS API 작업 [DescribeReservedDBInstances](#)를 호출합니다.

```
https://rds.us-west-2.amazonaws.com/
?Action=DescribeReservedDBInstances
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-09-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140420/us-west-2/rds/aws4_request
&X-Amz-Date=20140420T162211Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=3312d17a4c43bcd209bc22a0778dd23e73f8434254abbd7ac53b89ade3dae88e
```

이 API는 다음과 비슷한 출력을 반환합니다.

```
<DescribeReservedDBInstancesResponse xmlns="http://rds.amazonaws.com/doc/2014-10-31/">
<DescribeReservedDBInstancesResult>
  <ReservedDBInstances>
    <ReservedDBInstance>
      <OfferingType>Partial Upfront</OfferingType>
      <CurrencyCode>USD</CurrencyCode>
      <RecurringCharges/>
      <ProductDescription>mysql</ProductDescription>
      <ReservedDBInstancesOfferingId>649fd0c8-cf6d-47a0-bfa6-060f8e75e95f</
      ReservedDBInstancesOfferingId>
      <MultiAZ>false</MultiAZ>
      <State>payment-failed</State>
      <ReservedDBInstanceId>MyReservation</ReservedDBInstanceId>
      <DBInstanceCount>1</DBInstanceCount>
      <StartTime>2010-12-15T00:25:14.131Z</StartTime>
      <Duration>31536000</Duration>
      <FixedPrice>227.5</FixedPrice>
      <UsagePrice>0.046</UsagePrice>
      <DBInstanceClass>db.m1.small</DBInstanceClass>
    </ReservedDBInstance>
    <ReservedDBInstance>
      <OfferingType>Partial Upfront</OfferingType>
      <CurrencyCode>USD</CurrencyCode>
      <RecurringCharges/>
      <ProductDescription>mysql</ProductDescription>
      <ReservedDBInstancesOfferingId>649fd0c8-cf6d-47a0-bfa6-060f8e75e95f</
      ReservedDBInstancesOfferingId>
      <MultiAZ>false</MultiAZ>
      <State>payment-failed</State>
      <ReservedDBInstanceId>MyReservation</ReservedDBInstanceId>
      <DBInstanceCount>1</DBInstanceCount>
      <StartTime>2010-12-15T01:07:22.275Z</StartTime>
      <Duration>31536000</Duration>
      <FixedPrice>227.5</FixedPrice>
      <UsagePrice>0.046</UsagePrice>
      <DBInstanceClass>db.m1.small</DBInstanceClass>
    </ReservedDBInstance>
  </ReservedDBInstances>
</DescribeReservedDBInstancesResult>
<ResponseMetadata>
  <RequestId>23400d50-2978-11e1-9e6d-771388d6ed6b</RequestId>
</ResponseMetadata>
</DescribeReservedDBInstancesResponse>
```

Amazon RDS DB 인스턴스 백업 및 복원

이 섹션에서는 DB 인스턴스를 백업 및 복구하는 방법에 대해 설명합니다.

주제

- [백업 작업 \(p. 196\)](#)
- [DB 스냅샷 생성 \(p. 203\)](#)
- [DB 스냅샷에서 복원 \(p. 205\)](#)
- [스냅샷 복사 \(p. 207\)](#)
- [DB 스냅샷 공유 \(p. 215\)](#)
- [DB 인스턴스를 지정된 시간으로 복원 \(p. 221\)](#)
- [스냅샷 삭제 \(p. 222\)](#)
- [자습: DB 스냅샷에서 DB 인스턴스 복원 \(p. 223\)](#)

백업 작업

Amazon RDS는 DB 인스턴스의 자동 백업을 생성하여 저장합니다. Amazon RDS는 개별 데이터베이스가 아닌 전체 DB 인스턴스를 백업하여 DB 인스턴스의 스토리지 볼륨 스냅샷을 생성합니다.

Amazon RDS는 DB 인스턴스 백업 기간 동안 DB 인스턴스의 자동 백업을 생성합니다. Amazon RDS는 사용자가 지정한 백업 보존 기간에 따라 DB 인스턴스의 자동 백업을 저장합니다. 필요할 경우 백업 보존 기간 중 어느 특정 시점으로든 데이터베이스를 복구할 수 있습니다.

자동 백업은 이 규칙을 따릅니다.

- 자동 백업이 실행되려면 DB 인스턴스가 ACTIVE 상태여야 합니다. DB 인스턴스가 ACTIVE 외의 상태인 경우, 예컨대 STORAGE_FULL 상태인 경우 자동 백업이 실행되지 않습니다.
- 동일 리전에서 동일 DB 인스턴스에 복제본을 실행하는 동안 자동 백업과 자동 스냅샷은 실행되지 않습니다.

또한 수동으로 DB 스냅샷을 생성하여 DB 인스턴스를 백업할 수도 있습니다. DB 스냅샷 생성에 대한 자세한 내용은 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.

DB 인스턴스의 첫 번째 스냅샷에는 전체 DB 인스턴스에 대한 데이터가 포함됩니다. 동일한 DB 인스턴스의 후속 스냅샷은 종분식이며, 마지막 스냅샷 이후 변경된 데이터만 저장됩니다.

자동 및 수동 DB 스냅샷을 모두 복사하고 수동 DB 스냅샷을 공유할 수 있습니다. DB 스냅샷 복사에 대한 자세한 내용은 [스냅샷 복사 \(p. 207\)](#) 단원을 참조하십시오. DB 스냅샷 공유에 대한 자세한 내용은 [DB 스냅샷 공유 \(p. 215\)](#) 단원을 참조하십시오.

백업 스토리지

각 리전의 Amazon RDS 백업 스토리지는 해당 리전의 자동 백업 및 수동 DB 스냅샷으로 구성됩니다. 백업 스토리지는 해당 리전의 모든 인스턴스에 대한 데이터베이스 스토리지 합계와 동일합니다. DB 스냅샷을 다른 리전으로 이동하면 대상 리전의 백업 스토리지가 증가합니다.

백업 스토리지 비용에 대한 자세한 내용은 [Amazon RDS 요금](#)을 참조하십시오.

DB 인스턴스를 삭제할 때 자동 백업을 유지하기로 선택한 경우, 자동 백업이 최대 보존 기간 동안 저장됩니다. DB 인스턴스를 삭제할 때 자동 백업 보존을 선택하지 않은 경우, 모든 자동 백업이 DB 인스턴스와 함께 삭제됩니다. 자동 백업은 삭제된 후에는 복구할 수 없습니다. Amazon RDS가 DB 인스턴스를 삭제하기 전에 최종 DB 스냅샷을 생성하도록 선택한 경우 이 스냅샷을 사용하여 DB 인스턴스를 복구할 수 있습니다. 또는 이전에 생성한 수동 스냅샷을 사용할 수 있습니다. 수동 스냅샷은 삭제되지 않습니다. 리전당 최대 100개의 수동 스냅샷을 보유할 수 있습니다.

Note

AWS Backup이 관리하는 백업은 수동 스냅샷 제한을 위한 수동 스냅샷으로 간주됩니다. AWS Backup에 대한 자세한 내용은 [AWS 백업 개발자 안내서](#)를 참조하십시오.

백업 기간

자동 백업은 기본 백업 기간 동안 매일 실행됩니다. 백업 시간이 백업 기간에 할당된 시간보다 오래 걸릴 경우 백업은 백업 기간이 종료한 후에도 완료 시까지 계속 실행됩니다. 백업 기간은 해당 DB 인스턴스에 대한 주간 유지 보수 기간과 겹칠 수 없습니다.

자동 백업 기간 중에 백업 프로세스가 시작될 때 스토리지 I/O가 일시적으로 중단될 수 있습니다(일반적으로 몇 초). 다중 AZ 배포에 대한 백업 시 지역 시간이 몇 분으로 증가할 수도 있습니다. MariaDB, MySQL, Oracle 및 PostgreSQL의 경우, 다중 AZ 배포에 대한 백업 시 기본 AZ에서는 I/O 작업이 중단되지 않습니다. 백업이 예비 복제본으로부터 수행되기 때문입니다. SQL Server의 경우, 다중 AZ 배포에 대한 백업 도중 I/O 작업이 일시적으로 중단됩니다.

DB 인스턴스를 생성할 때 원하는 백업 기간을 지정하지 않을 경우에는 Amazon RDS가 기본 30분 백업 기간을 할당합니다. 이 기간은 각 AWS 리전에 대해 8시간 블록 시간 중에서 임의로 선택됩니다. 다음 표는 기본 백업 기간이 할당된 각 지역별 시간 블록 목록입니다.

리전	시간 블록
미국 서부(오레곤) 지역	06:00–14:00 UTC
미국 서부(캘리포니아 북부) 리전	06:00–14:00 UTC
미국 동부(오하이오) 리전	03:00–11:00 UTC
미국 동부(버지니아 북부) 지역	03:00–11:00 UTC
아시아 태평양(괌바이) 리전	16:30–00:30 UTC
아시아 태평양(서울) 리전	13:00–21:00 UTC
아시아 태평양(싱가포르) 리전	14:00–22:00 UTC
아시아 태평양(시드니) 리전	12:00–20:00 UTC
아시아 태평양(도쿄) 리전	13:00–21:00 UTC
캐나다(중부) 리전	06:29–14:29 UTC
EU(프랑크푸르트) 리전	20:00–04:00 UTC
EU(아일랜드) 지역	22:00–06:00 UTC
EU(런던) 리전	06:00–14:00 UTC

리전	시간 블록
남아메리카(상파울루) 리전	23:00–07:00 UTC
AWS GovCloud(미국 서부)	03:00–11:00 UTC

백업 보존 기간

또한 DB 인스턴스를 생성하면서 백업 보존 기간도 설정할 수 있습니다. 백업 보존 기간을 설정하지 않으면 Amazon RDS API 또는 AWS CLI를 사용하여 DB 인스턴스를 생성하는 경우 기본 백업 보존 기간은 1일입니다. 콘솔을 사용하여 DB 인스턴스를 생성하는 경우 기본 백업 보존 기간은 7일입니다. DB 인스턴스를 생성한 후 백업 보존 기간을 수정할 수 있습니다. 백업 보존 기간은 1일에서 35일 사이로 설정할 수 있습니다. 백업 보존 기간을 0으로 설정하면 자동 백업이 비활성화됩니다. 수동 스냅샷 한도(리전당 100개)는 자동 백업에 적용되지 않습니다.

Important

백업 보존 기간을 0에서 0이 아닌 값으로 또는 0이 아닌 값에서 0으로 변경할 경우 인스턴스가 중단 됩니다.

자동 백업 비활성화

대량의 데이터를 로드하는 등 특정 상황에서는 자동 백업을 일시적으로 비활성화해야 하는 경우가 있습니다.

Important

특정 시점으로 복구가 어렵기 때문에 자동 복구의 비활성화는 최대한 자제하는 것이 좋습니다. DB 인스턴스 자동 백업을 비활성화하면 해당 인스턴스에 대한 기존 자동 백업이 모두 삭제됩니다. 자동 백업을 비활성화한 후 다시 활성화하면 자동 백업을 다시 활성화한 시점부터만 복구할 수 있습니다.

이 예시에서는 백업 보존 파라미터를 0으로 설정하여 DB 인스턴스 mydbinstance의 자동 백업을 비활성화하겠습니다.

콘솔

자동 백업을 즉시 비활성화하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 변경하려는 DB 인스턴스를 선택합니다.
3. [Modify]를 선택합니다. [Modify DB Instance] 페이지가 나타납니다.
4. 백업 보존 기간으로 0일을 선택합니다.
5. [Continue]를 선택합니다.
6. Apply Immediately(즉시 적용)를 선택합니다.
7. 확인 페이지에서 [Modify DB Instance]를 선택하여 변경 내용을 저장하고 자동 백업을 비활성화합니다.

AWS CLI

자동 백업을 즉시 비활성화하려면, `modify-db-instance` 명령을 사용하고 `--apply-immediately`를 사용하여 백업 보존 기간을 0으로 설정합니다.

Example

다음은 자동 백업을 즉시 비활성화하는 예제입니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \  
  --db-instance-identifier mydbinstance \  
  --backup-retention-period 0 \  
  --apply-immediately
```

Windows의 경우:

```
aws rds modify-db-instance ^  
  --db-instance-identifier mydbinstance ^  
  --backup-retention-period 0 ^  
  --apply-immediately
```

수정 사항이 적용되는 시점을 알아보려면 백업 보존 기간 값이 0이 되고, mydbinstance가 이용 가능한 상태가 될 때까지 DB 인스턴스에 대해 `describe-db-instances`를 호출합니다.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

RDS API

자동 백업을 즉시 비활성화하려면 다음 파라미터를 사용하여 [ModifyDBInstance](#) 작업을 호출합니다.

- `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>
```

자동 백업 활성화

DB 인스턴스에 자동 백업이 활성화되어 있지 않더라도 언제든지 활성화할 수 있습니다. 백업 보존 기간을 0이 아닌 양수 값으로 설정하여 자동 백업을 활성화합니다. 자동 백업을 활성화하면 RDS 인스턴스 및 데이터베이스가 오프라인으로 설정되고 백업이 즉시 생성됩니다.

이 예시에서는 백업 보존 기간을 0이 아닌 다른 양수 값(여기서는 3)으로 설정하여 DB 인스턴스 mydbinstance의 자동 백업을 활성화하겠습니다.

콘솔

자동 백업을 즉시 활성화하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 변경하려는 DB 인스턴스를 선택합니다.
3. [Modify]를 선택합니다. [Modify DB Instance] 페이지가 나타납니다.

4. 백업 보존 기간으로 0이 아닌 양수 값(예: 3일)을 선택합니다.
5. [Continue]를 선택합니다.
6. Apply Immediately(즉시 적용)를 선택합니다.
7. 확인 페이지에서 [Modify DB Instance]를 선택하여 변경 내용을 저장하고 자동 백업을 활성화합니다.

AWS CLI

자동 백업을 즉시 활성화하려면, AWS CLI `modify-db-instance` 명령을 사용합니다.

이 예에서는 백업 보존 기간을 3일로 설정하여 자동 백업을 활성화합니다.

다음 파라미터를 포함합니다.

- `--db-instance-identifier`
- `--backup-retention-period`
- `--apply-immediately` 또는 `--no-apply-immediately`

Example

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
  --db-instance-identifier mydbinstance \
  --backup-retention-period 3 \
  --apply-immediately
```

Windows의 경우:

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --backup-retention-period 3 ^
  --apply-immediately
```

RDS API

자동 백업을 즉시 활성화하려면, RDS API `ModifyDBInstance` 작업을 사용합니다.

이 예에서는 백업 보존 기간을 3일로 설정하여 자동 백업을 활성화합니다.

다음 파라미터를 포함합니다.

- `DBInstanceIdentifier`
- `BackupRetentionPeriod`
- `ApplyImmediately = true`

Example

```
https://rds.amazonaws.com/
?Action=ModifyDBInstance
&DBInstanceIdentifier=mydbinstance
&BackupRetentionPeriod=3
&ApplyImmediately=true
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2009-10-14T17%3A48%3A21.746Z
```

```
&AWSAccessKeyId=<AWS Access Key ID>
&Signature=<Signature>
```

자동 백업 보존

DB 인스턴스를 삭제할 때 자동 백업을 보존할 수 있습니다.

보존된 자동 백업에는 DB 인스턴스의 시스템 스냅샷 및 트랜잭션 로그가 포함되어 있습니다. 여기에는 할당된 스토리지 및 DB 인스턴스 클래스와 같은 DB 인스턴스 속성이 포함되는데, 이러한 속성은 활성 인스턴스로 복원하는 데 필요합니다.

MySQL, MariaDB, PostgreSQL, Oracle 및 Microsoft SQL Server 엔진을 실행 중인 RDS 인스턴스에 대한 자동 백업을 보존할 수 있습니다.

AWS Management 콘솔, RDS API 및 AWS CLI를 사용하여 보존된 자동 백업을 복원하거나 제거할 수 있습니다.

보존 기간

보존된 자동 백업의 시스템 스냅샷 및 트랜잭션 로그는 원본 DB 인스턴스에 대해 만료될 때와 동일한 방식으로 만료됩니다. 이 인스턴스에 대해 생성된 새 스냅샷이나 로그가 없으므로 보존 자동 백업은 결국 완전히 만료됩니다. 사실상 보존된 자동 백업은 삭제 시 원본 인스턴스에 지정되었던 보존 기간에 대한 설정에 따라 마지막 시스템 스냅샷이 수행될 동안 유지됩니다. 보존된 자동 백업은 마지막 시스템 스냅샷이 만료된 후 시스템에 의해 제거됩니다.

DB 인스턴스를 삭제할 때와 동일한 방식으로 보존된 자동 백업을 제거할 수 있습니다. 콘솔이나 RDS API 작업 [DeleteDBInstanceAutomatedBackup](#)을 사용하여 보존된 자동 백업을 제거할 수 있습니다.

마지막 스냅샷은 보존된 자동 백업과 무관합니다. 보존된 자동 백업도 결국 만료되므로 자동 백업을 보존하더라도 가급적이면 최종 스냅샷을 생성하는 것이 좋습니다. 최종 스냅샷은 만료되지 않습니다.

복원

보존된 자동 백업을 보려면 자동 백업 페이지로 전환합니다. 콘솔의 데이터베이스 스냅샷 페이지에서 보존된 자동 백업과 연결된 개별 스냅샷을 볼 수 있습니다. 또는 보존된 자동 백업과 연결된 개별 스냅샷을 설명할 수 있습니다. 그런 다음 이러한 스냅샷 중 하나에서 DB 인스턴스를 직접 복원할 수 있습니다.

복원된 DB 인스턴스는 기본 파라미터 및 옵션 그룹과 자동으로 연결됩니다. 하지만 복원 중에 사용자 지정 파라미터 그룹 및 옵션 그룹을 지정하여 적용할 수 있습니다.

이 예제에서는 보존된 자동 백업을 사용하여 DB 인스턴스를 특정 시점으로 복원합니다. 먼저 보존된 자동 백업을 설명하여 복원할 자동 백업을 표시할 수 있습니다.

RDS API를 사용하여 보존된 자동 백업을 설명하려면 다음 파라미터 중 하나를 사용하여 [DescribeDBInstanceAutomatedBackups](#) 작업을 호출합니다.

- `DBInstanceIdentifier`
- `DbiResourceId`

```
aws rds describe-db-instance-automated-backups --db-instance-
identifier DBInstanceIdentifier
OR
aws rds describe-db-instance-automated-backups --dbi-resource-idDbiResourceId
```

그런 다음 RDS API를 사용하여 보존 자동 백업을 특정 시점으로 복원하려면 다음 파라미터를 사용하여 [RestoreDBInstanceToPointInTime](#) 작업을 호출합니다.

- `SourceDbiResourceId`
- `TargetDBInstanceIdentifier`

```
aws rds restore-db-instance-to-point-in-time --source-dbi-resource-id SourceDbiResourceId
--target-db-instance-identifier TargetDBInstanceIdentifier --use-latest-restorable-time
```

보존 비용

보존된 자동 백업의 비용은 연결된 시스템 스냅샷의 총 스토리지 비용입니다. 트랜잭션 로그나 인스턴스 메타데이터에 대해 추가 요금이 부과되지 않습니다. 백업에 대한 기타 모든 요금 규칙이 복원 가능한 인스턴스에 적용됩니다.

예를 들어 실행 중인 인스턴스에 할당된 총 스토리지가 100GB라고 가정하겠습니다. 또한 수동 스냅샷 50GB와 보존된 자동 백업과 연결된 시스템 스냅샷 75GB가 있다고 가정하겠습니다. 이 경우 백업 스토리지의 추가 25GB에 대해서만 요금이 부과됩니다. 즉, $(50\text{GB} + 75\text{GB}) - 100\text{GB} = 25\text{GB}$ 입니다.

제한 및 권장 사항

다음 제한은 보존된 자동 백업에 적용됩니다.

- 한 리전에서 보존된 자동 백업의 최대 개수는 20개입니다. 이 개수는 DB 인스턴스 제한에 포함되지 않습니다. 실행 중인 DB 인스턴스 20개와 추가로 보존된 자동 백업 20개를 동시에 보유할 수 있습니다.
- 보존된 자동 백업에는 파라미터나 옵션 그룹에 대한 정보가 포함되지 않습니다.
- 삭제된 인스턴스를 삭제할 때의 보존 기간 내 특정 시점으로 복원할 수 있습니다.
- 보존된 자동 백업은 원본 인스턴스를 삭제할 때 존재했던 시스템 백업, 트랜잭션 로그 및 DB 인스턴스 속성으로 구성되므로 수정할 수 없습니다.

지원되지 않는 MySQL 스토리지 엔진에 대한 자동 백업

MySQL DB 엔진의 경우, 자동 백업은 InnoDB 스토리지 엔진에만 지원됩니다. MyISAM 등의 다른 MySQL 스토리지 엔진에서 이러한 기능을 사용하는 경우 백업 복원 시에 작동이 불안정할 수 있습니다. 특히 MyISAM과 같은 스토리지 엔진은 안정적인 충돌 복구를 지원하지 않으므로 충돌 시 테이블이 손상될 수 있습니다. 이 같은 이유로 InnoDB 스토리지 엔진을 사용할 것을 권장합니다.

- 기존 MyISAM 테이블을 InnoDB 테이블로 변환하려면 `ALTER TABLE` 명령을 사용하면 됩니다(예: `ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;`).
- MyISAM을 사용하는 경우 `REPAIR` 명령을 사용하여 충돌 후 손상된 테이블을 수동으로 복구할 수 있습니다. 자세한 내용은 MySQL 설명서에서 [REPAIR TABLE 구문](#)을 참조하십시오. 그러나 MySQL 문서에 기재된 바와 같이 데이터를 전부 복원하지 못할 수도 있습니다.
- MyISAM 테이블을 복원하기 전에 테이블의 스냅샷을 생성하려면 다음 단계를 따릅니다.
 1. MyISAM 테이블의 모든 활동을 중지합니다(모든 세션 닫기).

`SHOW FULL PROCESSLIST` 명령에서 반환되는 각 프로세스에 대해 `mysql.rds_kill` 명령을 호출하여 모든 세션을 닫을 수 있습니다.

2. 각 MyISAM 테이블에 잠금 및 플러시를 수행합니다. 예를 들어, 다음 명령은 `myisam_table1` 및 `myisam_table2`라는 두 테이블을 잠그고 플러시합니다.

```
mysql> FLUSH TABLES myisam_table1, myisam_table2 WITH READ LOCK;
```

3. DB 인스턴스의 DB 스냅샷을 만듭니다. 스냅샷을 완료하면 MyISAM 테이블에서 잠금을 해제하고 활동을 다시 시작합니다. 다음 명령을 사용하여 테이블의 잠금을 해제할 수 있습니다.

```
mysql> UNLOCK TABLES;
```

이러한 단계를 통해 MyISAM에서 메모리에 저장된 데이터를 디스크로 강제 플러시하여 DB 스냅샷 복원 시 깨끗한 상태에서 시작할 수 있습니다. DB 스냅샷을 만드는 방법에 대한 자세한 내용은 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.

지원되지 않는 MariaDB 스토리지 엔진에 대한 자동 백업

MariaDB DB 엔진의 경우, 자동화된 백업은 InnoDB 스토리지 엔진(버전 10.2 이상) 및 XtraDB 스토리지 엔진(버전 10.0과 10.1)에서만 지원됩니다. Aria 등의 다른 MariaDB 스토리지 엔진에서 이러한 기능을 사용하는 경우 백업 복원 시에 작동이 불안정할 수 있습니다. Aria가 충돌 안정성을 개선한 MyISAM 대체 스토리지 엔진이지만, 충돌 이벤트가 발생하는 경우 여전히 테이블이 손상될 수 있습니다. 이 같은 이유로 XtraDB 스토리지 엔진을 사용할 것을 권장합니다.

- 기존 Aria 테이블을 InnoDB 테이블로 변환하려면 `ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;`
- 기존 Aria 테이블을 XtraDB 테이블로 변환하려면 `ALTER TABLE table_name ENGINE=xtradb, ALGORITHM=COPY;`
- Aria를 사용하는 경우 `REPAIR TABLE` 명령을 사용하여 충돌 후 손상된 테이블을 수동으로 복구할 수 있습니다. 자세한 내용은 <http://mariadb.com/kb/en/mariadb/repair-table/> 단원을 참조하십시오.
- Aria 테이블을 복원하기 전에 테이블의 스냅샷을 생성하려면 다음 단계를 따릅니다.
 - Aria 테이블의 모든 활동을 중지합니다(모든 세션 닫기).
 - 각 Aria 테이블에 잠금 및 플러시를 수행합니다.
 - DB 인스턴스의 DB 스냅샷을 만듭니다. 스냅샷을 완료하면 Aria 테이블에서 잠금을 해제하고 활동을 다시 시작합니다. 이러한 단계를 통해 Aria에서 메모리에 저장된 데이터를 디스크로 강제 플러시하여 DB 스냅샷 복원 시 깨끗한 상태에서 시작할 수 있습니다.

DB 스냅샷 생성

Amazon RDS는 개별 데이터베이스가 아닌 전체 DB 인스턴스를 백업하여 DB 인스턴스의 스토리지 볼륨 스냅샷을 생성합니다. 단일 AZ DB 인스턴스에서 이 DB 스냅샷을 생성하면 잠시 I/O가 중단되는데, 해당 DB 인스턴스의 크기 및 클래스에 따라 대체로 몇 초에서 몇 분 정도 지속됩니다. 다중 AZ DB 인스턴스는 예비 인스턴스에서 백업이 이루어지기 때문에 이 I/O 중단의 영향을 받지 않습니다.

DB 스냅샷을 생성할 때는 백업할 DB 인스턴스를 구분한 다음 나중에 복구할 수 있도록 DB 스냅샷을 명명해야 합니다. 스냅샷을 생성하는 데 걸리는 시간은 데이터베이스 크기에 따라 다릅니다. 스냅샷에는 전체 스토리지 볼륨이 포함되기 때문에 임시 파일 같은 파일들의 크기가 스냅샷을 생성하는 데 걸리는 시간에 영향을 미치기도 합니다.

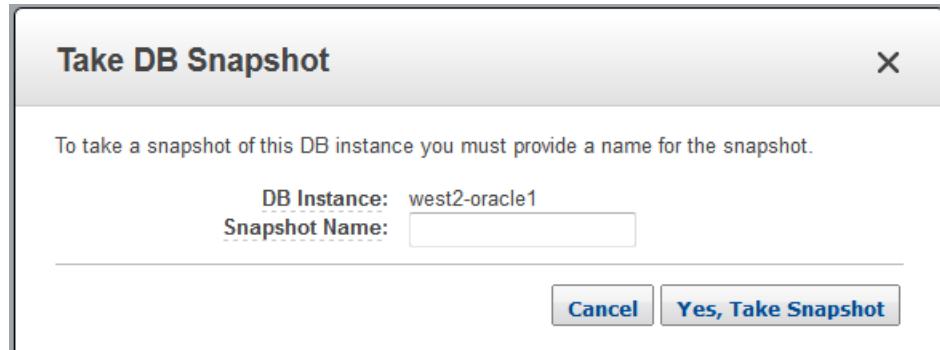
AWS Management 콘솔, AWS CLI 또는 RDS API를 사용하여 DB 스냅샷을 생성할 수 있습니다.

AWS Management 콘솔

DB 스냅샷을 생성하려면

- AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.

2. 탐색 창에서 데이터베이스를 선택합니다.
3. DB 인스턴스 목록에서 스냅샷을 생성하려는 DB 인스턴스를 선택합니다.
4. 작업에서 스냅샷 만들기를 선택합니다.
[Take DB Snapshot] 창이 나타납니다.
5. 스냅샷 이름 상자에 스냅샷의 이름을 입력합니다.



6. Take Snapshot(스냅샷 만들기)을 선택합니다.

CLI

AWS CLI를 사용하여 DB 스냅샷을 생성할 때는 백업할 DB 인스턴스를 식별한 후 나중에 복구할 수 있도록 DB 스냅샷에 이름을 지정해야 합니다. 이를 위해서는 AWS CLI [create-db-snapshot](#) 명령을 다음 파라미터와 함께 사용하면 됩니다.

- `--db-instance-identifier`
- `--db-snapshot-identifier`

이번 예에서는 `mydbinstance`라는 이름의 DB 인스턴스에서 `mydbsnapshot`라는 이름의 DB 스냅샷을 생성합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds create-db-snapshot /  
  --db-instance-identifier mydbinstance /  
  --db-snapshot-identifier mydbsnapshot
```

Windows의 경우:

```
aws rds create-db-snapshot ^  
  --db-instance-identifier mydbinstance ^  
  --db-snapshot-identifier mydbsnapshot
```

API

Amazon RDS API를 사용하여 DB 스냅샷을 생성할 때는 백업할 DB 인스턴스를 식별한 후 나중에 복구할 수 있도록 DB 스냅샷에 이름을 지정해야 합니다. 이를 위해서는 Amazon RDS API [CreateDBSnapshot](#) 명령을 다음 파라미터와 함께 사용하면 됩니다.

- `DBInstanceIdentifier`
- `DBSnapshotIdentifier`

DB 스냅샷에서 복원

Amazon RDS는 개별 데이터베이스가 아닌 전체 DB 인스턴스를 백업하여 DB 인스턴스의 스토리지 볼륨 스냅샷을 생성합니다. 이 DB 스냅샷에서 복원하여 DB 인스턴스를 생성할 수 있습니다. DB 인스턴스를 복원하는 경우 복원 원본으로 사용할 DB 스냅샷의 이름을 입력한 다음 복원 작업에서 생성되는 새 DB 인스턴스의 이름을 입력합니다. DB 스냅샷에서 기존 DB 인스턴스로 복원할 수 없습니다. 복원하면 새 DB 인스턴스가 생성됩니다.

DB 인스턴스를 복원하고 원본 DB 스냅샷과 다른 스토리지 유형을 사용할 수 있습니다. 이 경우 데이터를 새로운 스토리지 유형으로 마이그레이션하기 위해 추가 작업이 필요하기 때문에 복원 프로세스가 오래 걸립니다. 마그네틱(표준) 스토리지로 또는 스토리지에서 복원할 때 마이그레이션 프로세스의 속도가 가장 느립니다. 이는 마그네틱 스토리지에는 프로비저닝된 IOPS 또는 범용(SSD) 스토리지의 IOPS 기능이 없기 때문입니다.

Note

공유되고 동시에 암호화된 DB 스냅샷에서는 DB 인스턴스를 복원할 수 없습니다. 대신 DB 스냅샷의 복사본을 생성하고 해당 복사본에서 DB 인스턴스를 복원할 수 있습니다.

파라미터 그룹 고려 사항

복원된 DB 인스턴스를 바른 파라미터 그룹과 연결할 수 있도록 생성한 DB 스냅샷에 대한 파라미터 그룹을 유지하는 것이 좋습니다. DB 인스턴스를 복원할 때 파라미터 그룹을 지정할 수 있습니다.

보안 그룹 고려 사항

DB 인스턴스를 복원할 경우 기본 보안 그룹은 복원된 인스턴스와 연결됩니다. 복원이 완료되고 새 DB 인스턴스가 사용 가능하게 되는 즉시 복원의 소스 인스턴스에서 사용하는 사용자 지정 보안 그룹을 연결해야 합니다. RDS 콘솔의 `Modify` 명령, `ModifyDBInstance` Amazon RDS API 또는 AWS CLI `modify-db-instance` 명령을 사용하여 이러한 변경 내용을 적용해야 합니다.

옵션 그룹 고려 사항

DB 인스턴스를 복원하면, DB 스냅샷과 연결된 옵션 그룹은 생성 후 복원된 DB 인스턴스와 연결됩니다. 예를 들어, 복원 원본으로 사용하려는 DB 스냅샷에서 Oracle Transparent Data Encryption을 사용하는 경우 복원된 DB 인스턴스에서도 동일한 옵션 그룹을 사용합니다.

옵션 그룹을 DB 인스턴스에 배정하는 경우 이 옵션 그룹은 DB 인스턴스가 위치한 지원되는 플랫폼(VPC 또는 EC2-Classic(VPC 아님))에도 연결됩니다. DB 인스턴스가 VPC에 속한 경우 DB 인스턴스와 연결된 옵션 그룹은 이 VPC에도 연결됩니다. 즉, 다른 VPC나 다른 플랫폼으로 DB 인스턴스를 복원하려고 해도 DB 인스턴스에 배정된 옵션 그룹은 사용할 수 없습니다. DB 인스턴스를 다른 VPC나 다른 플랫폼으로 복구할 경우 기본 옵션 그룹을 인스턴스에 할당하거나, 해당 VPC 또는 플랫폼에 연결된 옵션 그룹을 할당하거나, 새 옵션 그룹을 생성하여 DB 인스턴스에 할당해야 합니다. 지속적 또는 영구적 옵션에서 DB 인스턴스를 다른 VPC로 복원할 때 지속적 또는 영구적 옵션을 포함하는 새 옵션 그룹을 만들어야 합니다.

Microsoft SQL Server 고려 사항

Microsoft SQL Server DB 스냅샷을 새 인스턴스로 복원할 때에는 항상 스냅샷과 동일한 버전으로 복원할 수 있습니다. 경우에 따라서는 DB 인스턴스의 버전을 변경할 수도 있습니다. 다음은 버전을 변경할 때 적용되는 제한 사항입니다.

- DB 스냅샷에는 새 버전에 할당되는 스토리지가 충분히 있어야 합니다.
- 다음 버전 변경만이 지원됩니다.
 - Standard Edition에서 Enterprise Edition으로 변경
 - Web Edition에서 Standard Edition 또는 Enterprise Edition으로 변경
 - Express Edition에서 Web Edition, Standard Edition 또는 Enterprise Edition으로 변경

스냅샷을 복원하여 지원되지 않는 새 버전으로 변경할 경우, 기본 백업과 복원 기능을 사용할 수 있습니다. SQL Server는 데이터베이스에서 활성화된 SQL Server 기능을 기반으로 데이터베이스가 새 버전과 호환되는지 여부를 확인합니다. 자세한 내용은 [SQL Server 데이터베이스 가져오기 및 내보내기 \(p. 520\)](#) 단원을 참조하십시오.

Oracle 고려 사항

Oracle GoldenGate를 사용하는 경우 항상 `compatible` 파라미터를 포함하는 파라미터 그룹을 유지하십시오. DB 스냅샷에서 DB 인스턴스를 복원하는 경우 `compatible` 파라미터 값 이상을 가진 파라미터 그룹을 지정해야 합니다.

DB 인스턴스를 복원하기 전에 아직 DB 스냅샷 상태일 때 DB 스냅샷을 업그레이드할 수 있습니다. 자세한 내용은 [Oracle DB 스냅샷 업그레이드 \(p. 756\)](#) 단원을 참조하십시오.

스냅샷에서 복구

AWS Management 콘솔, AWS CLI 또는 RDS API를 사용하여 DB 스냅샷에서 DB 인스턴스 복원할 수 있습니다.

AWS Management 콘솔

DB 스냅샷에서 DB 인스턴스를 복원하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 복원 원본으로 사용할 DB 스냅샷을 선택합니다.
4. 작업에서 스냅샷 복원을 선택합니다.
5. DB 인스턴스 복원 페이지의 DB 인스턴스 식별자에 복원된 DB 인스턴스의 이름을 입력합니다.
6. [Restore DB Instance]를 선택합니다.
7. DB 인스턴스의 기능을 스냅샷을 생성할 때 원본으로 사용한 DB 인스턴스의 기능으로 복원하려면, 보안 그룹을 사용하도록 DB 인스턴스를 수정해야 합니다. 다음 단계에서는 사용자의 DB 인스턴스가 VPC에 있다고 가정합니다. DB 인스턴스가 VPC에 있지 않은 경우 EC2 Management Console을 사용하여 DB 인스턴스에 필요한 보안 그룹을 찾아야 합니다.
 - a. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
 - b. 탐색 창에서 [Security Groups]를 선택합니다.
 - c. DB 인스턴스에 사용할 보안 그룹을 선택합니다. 필요에 따라 이 보안 그룹을 EC2 인스턴스에 대한 보안 그룹과 연결하는 규칙을 추가합니다. 자세한 내용은 [동일한 VPC에 있는 EC2 인스턴스가 VPC 내에 있는 DB 인스턴스에 액세스 \(p. 409\)](#) 단원을 참조하십시오.

CLI

DB 스냅샷에서 DB 인스턴스를 복원하려면 AWS CLI 명령 `restore-db-instance-from-db-snapshot`을 사용합니다.

이 예에서는 `mydbsnapshot`이라는 이전에 생성된 DB 스냅샷에서 복원합니다. 그리고 `mynewdbinstance`라는 새 DB 인스턴스로 복원해야 합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds restore-db-instance-from-db-snapshot \
--db-instance-identifier mynewdbinstance \
--db-snapshot-identifier mydbsnapshot
```

Windows의 경우:

```
aws rds restore-db-instance-from-db-snapshot ^
--db-instance-identifier mynewdbinstance ^
--db-snapshot-identifier mydbsnapshot
```

다음과 비슷한 출력이 반환됩니다.

```
DBINSTANCE mynewdbinstance db.m3.large MySQL      50          sa           creating  3 n
5.6.40 general-public-license
```

DB 인스턴스가 복원된 후 이전 DB 인스턴스와 동일한 기능을 원할 경우 DB 스냅샷을 생성하는 데 사용한 DB 인스턴스가 사용하는 보안 그룹에 복원된 DB 인스턴스를 추가해야 합니다.

API

DB 스냅샷에서 DB 인스턴스를 복원하려면 Amazon RDS API 함수 [RestoreDBInstanceFromDBSnapshot](#)을 다음 파라미터와 함께 호출합니다.

- `DBInstanceIdentifier`
- `DBSnapshotIdentifier`

스냅샷 복사

Amazon RDS에서는 자동 또는 수동 DB 스냅샷을 복사할 수 있습니다. 스냅샷을 복사한 후 사본은 수동 스냅샷입니다.

동일한 AWS 리전 내의 스냅샷을 복사할 수 있고, AWS 리전 간에 스냅샷을 복사할 수 있고, AWS 계정 간에 스냅샷을 복사할 수 있습니다.

자동화된 스냅샷을 다른 AWS 계정으로 복사하는 것은 2단계 프로세스로 되어 있습니다. 자동화된 스냅샷으로부터 수동 스냅샷을 생성한 다음 수동 스냅샷을 다른 계정으로 복사합니다.

제한 사항

다음은 스냅샷을 복사할 때 적용되는 몇몇 제한 사항입니다.

- AWS 리전 중국(베이징) 또는 중국(닝샤)로/에서는 스냅샷을 복사할 수 없습니다.
- AWS GovCloud(US-East)과 AWS GovCloud (US-West) 간에는 스냅샷을 복사할 수 있지만, AWS GovCloud (US) 리전과 다른 AWS 리전 간에는 스냅샷을 복사할 수 없습니다.
- 대상 스냅샷이 사용 가능해지기 전에 원본 스냅샷을 삭제할 경우 스냅샷 복사가 실패할 수 있습니다. 원본 스냅샷을 삭제하기 전에 대상 스냅샷의 상태가 AVAILABLE인지 확인하십시오.
- 계정당 단일 대상 리전으로 최대 다섯 개까지 스냅샷 사본 요청이 진행될 수 있습니다.
- 관련된 리전과 복사할 데이터 양에 따라 리전 간 스냅샷 복사를 완료하는 데 몇 시간이 걸릴 수 있습니다. 지정된 원본 AWS 리전에서 리전 간 DB 스냅샷 복사 요청이 많은 경우 진행 중인 복사 중 일부가 완료될 때까지 Amazon RDS에서 해당 원본 AWS 리전에 대한 새로운 리전 간 복사 요청을 대기열에 넣을 수 있습니다. 대기열에 있는 복사 요청에 대한 진행 정보는 표시되지 않습니다. 복사가 시작되면 진행 정보가 표시됩니다.

스냅샷 보존

DB 인스턴스에 대한 자동 스냅샷을 비활성화하거나 DB 인스턴스를 삭제하면 보존 기간이 끝날 때 Amazon RDS에서 자동 스냅샷이 삭제됩니다. 자동 DB 스냅샷을 더 오랜 기간 동안 유지하려면 자동 DB 스냅샷을 복사하여 수동 DB 스냅샷을 만듭니다. 그러면 사용자가 삭제할 때까지 스냅샷이 보존됩니다. 기본 스토리지 공간을 초과하는 경우 수동 스냅샷에 Amazon RDS 스토리지 비용이 적용될 수 있습니다.

백업 스토리지 비용에 대한 자세한 내용은 [Amazon RDS 요금](#)을 참조하십시오.

공유 스냅샷 복사

다른 AWS 계정이 공유하는 스냅샷을 복사할 수 있습니다. 다른 AWS 계정이 공유한 암호화된 스냅샷을 복사하려는 경우 스냅샷을 암호화하는 데 사용된 KMS 암호화 키에 대한 액세스 권한이 있어야 합니다.

스냅샷이 암호화되지 않은 경우 리전 간에 공유 DB 스냅샷을 복사할 수 있습니다. 하지만 공유 DB 스냅샷이 암호화된 경우 이를 동일한 AWS 리전에서만 복사할 수 있습니다.

암호화 처리

AWS KMS 암호화 키를 사용하여 암호화된 스냅샷을 복사할 수 있습니다. 암호화된 스냅샷을 복사할 경우 스냅샷의 사본도 암호화해야 합니다. 동일한 AWS 리전 내에 암호화된 스냅샷을 복사하는 경우, 원본 스냅샷과 동일한 KMS 암호화 키를 사용하여 사본을 암호화하거나, 다른 KMS 암호화 키를 지정할 수 있습니다. 리전 간에 암호화된 스냅샷을 복사하는 경우, KMS 키가 리전마다 다르므로 이 복사에는 원본 스냅샷에 사용한 것과 동일한 KMS 암호화 키를 사용할 수 없습니다. 대신에 대상 AWS 리전에 유효한 KMS 키를 지정해야 합니다.

암호화되지 않은 스냅샷의 사본을 암호화할 수도 있습니다. 이렇게 하면 이전의 암호화되지 않은 DB 인스턴스에 신속히 암호화를 추가할 수 있습니다. 즉, 암호화 준비가 완료되면 DB 인스턴스의 스냅샷을 만든 후 이 스냅샷의 사본을 만들고 KMS 암호화 키를 지정하여 스냅샷 사본을 암호화할 수 있습니다. 그런 다음 암호화된 스냅샷에서 암호화된 DB 인스턴스를 복원할 수 있습니다.

AWS 리전 간 스냅샷 복사

소스 스냅샷의 AWS 리전과 다른 AWS 리전에 스냅샷을 복사할 때 종분 스냅샷을 복사하는 경우에도 첫 번째 복사는 전체 스냅샷 복사입니다. 전체 스냅샷 복사에는 DB 인스턴스 복원에 필요한 모든 데이터 및 메타데이터가 포함됩니다. 첫 번째 스냅샷 복사 후 동일한 DB 인스턴스의 종분 스냅샷을 동일한 AWS 리전 내의 동일한 대상 리전에 복사할 수 있습니다.

종분 스냅샷에는 동일한 DB 인스턴스의 마지막 스냅샷 이후 변경된 데이터만 포함됩니다. 종분 스냅샷 복사는 전체 스냅샷 복사에 비해 속도가 더욱 빠르고 스토리지 비용이 낮습니다. AWS 리전 간 종분 스냅샷 복사는 비암호화 및 암호화된 스냅샷 모두에 대해 지원됩니다.

Important

여러 AWS 계정에 종분 스냅샷을 복사하는 기능은 지원되지 않습니다. 하나의 AWS 계정에서 다른 AWS 계정으로의 스냅샷 사본을 설정하면 동일 리전 내에서도 모든 사본은 전체 스냅샷입니다.

관련된 AWS 리전과 복사할 데이터 양에 따라 리전 간 스냅샷 복사를 완료하는 데 몇 시간이 걸릴 수 있습니다. 일부 경우에는 특정 소스 AWS 리전으로부터 대량의 리전 간 스냅샷 복사 요청이 있을 수 있습니다. 이 경우 진행 중인 복사 중 일부가 완료될 때까지 Amazon RDS에서 해당 소스 AWS 리전에 대한 새로운 리전 간 복사 요청을 대기열에 넣을 수 있습니다. 대기열에 있는 복사 요청에 대한 진행 정보는 표시되지 않습니다. 복사가 시작되면 진행 정보가 표시됩니다.

Note

스냅샷 복사인 소스 스냅샷을 복사할 때 스냅샷 복사에 종분 복사에 대해 필요한 메타데이터가 포함되어 있지 않기 때문에 이는 종분 복사가 아닙니다.

옵션 그룹 고려 사항

옵션 그룹은 그룹이 생성된 AWS 리전에 고유하며, 한 AWS 리전의 옵션 그룹을 다른 AWS 리전에 사용할 수 없습니다.

리전 간에 스냅샷을 복사할 경우 스냅샷에 새 옵션 그룹을 지정할 수 있습니다. 스냅샷을 복사하기 전에 새 옵션 그룹을 준비하는 것이 좋습니다. 대상 AWS 리전에서 원래 DB 인스턴스 와 동일한 설정으로 옵션 그룹을 생성합니다. 새 AWS 리전에 이미 옵션 그룹이 있는 경우 이 그룹을 사용할 수 있습니다.

스냅샷을 복사할 때 스냅샷을 저장하면서 새 옵션 그룹을 지정하지 않을 경우 DB 인스턴스 가 기본 옵션 그룹을 가져옵니다. 새 DB 인스턴스 에 원본과 같은 옵션을 지정하려면 다음을 수행해야 합니다.

1. 대상 AWS 리전에서 원래 DB 인스턴스 와 동일한 설정으로 옵션 그룹을 생성합니다. 새 AWS 리전에 이미 옵션 그룹이 있는 경우 이 그룹을 사용할 수 있습니다.
2. 대상 AWS 리전에 스냅샷을 복원한 후 새 DB 인스턴스 를 수정하여 이전 단계의 새로운 또는 기존 옵션 그룹을 추가합니다.

파라미터 그룹 고려 사항

리전 간에 스냅샷을 복사하는 경우 사본에는 원래 DB 인스턴스 에서 사용된 파라미터 그룹이 포함되지 않습니다. 스냅샷을 복원하여 새 DB 인스턴스 를 생성하면 이 DB 인스턴스 가 생성되는 AWS 리전의 기본 파라미터 그룹이 DB 인스턴스 또는 DB 클러스터에 배정됩니다. 새 DB 인스턴스 에 원본과 같은 파라미터를 지정하려면 다음을 수행해야 합니다.

1. 대상 AWS 리전에서 원래 DB 인스턴스 와 동일한 설정으로 DB 파라미터 그룹을 생성합니다. 새 AWS 리전에 이미 옵션 그룹이 있는 경우 이 그룹을 사용할 수 있습니다.
2. 대상 AWS 리전에 스냅샷을 복원한 후 새 DB 인스턴스 를 수정하여 이전 단계의 새로운 또는 기존 파라미터 그룹을 추가합니다.

DB 스냅샷 복사

DB 스냅샷을 복사하려면 이 항목의 절차를 사용합니다. 스냅샷 복사에 대한 개요는 [스냅샷 복사 \(p. 207\)](#) 단원을 참조하십시오.

각 AWS 계정에 대해 AWS 리전 간에 DB 스냅샷을 한 번에 5개까지 복사할 수 있습니다. DB 스냅샷을 다른 AWS 리전으로 복사하면 그 AWS 리전에 유지되는 수동 DB 스냅샷이 생성됩니다. 소스 AWS 리전 밖으로 DB 스냅샷을 복사하면 Amazon RDS 데이터 전송 요금이 발생합니다.

데이터 전송 요금에 대한 자세한 내용은 [Amazon RDS 요금](#)을 참조하십시오.

새 AWS 리전에 DB 스냅샷 사본이 생성된 후 DB 스냅샷 사본은 해당 AWS 리전의 다른 모든 DB 스냅샷과 똑같이 동작합니다.

AWS Management 콘솔, AWS CLI 또는 RDS API를 사용하여 DB 스냅샷을 복사할 수 있습니다.

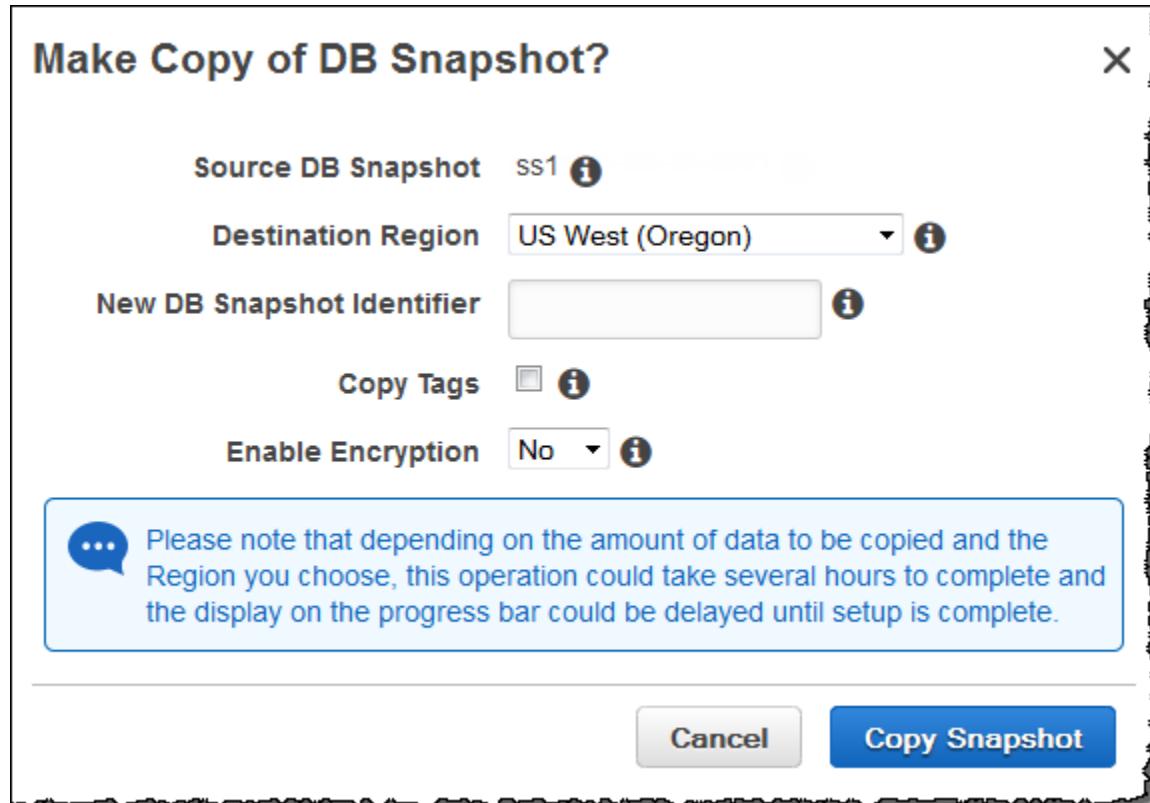
AWS Management 콘솔

이 절차에서는 AWS Management 콘솔을 사용하여 암호화된 DB 스냅샷 또는 암호화되지 않은 DB 스냅샷을 동일한 AWS 리전에서 또는 리전 간에 복사합니다.

DB 스냅샷을 복사하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔 을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 복사하려는 DB 스냅샷을 선택합니다.

4. 작업에서 스냅샷 복사를 선택합니다. [Make Copy of DB Snapshot] 페이지가 나타납니다.



5. (선택 사항) DB 스냅샷을 다른 AWS 리전에 복사하려면 [Destination Region]에 새 AWS 리전을 선택합니다.

Note

대상 AWS 리전에는 원본 AWS 리전과 동일한 가용 데이터베이스 엔진 버전이 있어야 합니다.

6. [New DB Snapshot Identifier]에 DB 스냅샷 사본의 이름을 입력합니다.
7. (선택 사항) 대상 옵션 그룹에서 새로운 옵션 그룹을 선택합니다.

한 AWS 리전에서 다른 리전으로 스냅샷을 복사하고 DB 인스턴스가 기본값이 아닌 옵션 그룹을 사용하는 경우 이 옵션을 지정합니다.

원본 DB 인스턴스가 Oracle 또는 Microsoft SQL Server용 Transparent Data Encryption을 사용하는 경우 리전 간 복사 시 이 옵션을 지정해야 합니다. 자세한 내용은 [옵션 그룹 고려 사항 \(p. 209\)](#) 단원을 참조하십시오.

8. (선택 사항) [Copy Tags]를 선택하여 스냅샷의 태그와 값을 스냅샷 사본에 복사합니다.
9. (선택 사항) [Enable Encryption]에 대해 다음 옵션 중 하나를 선택합니다.
 - DB 스냅샷이 암호화되지 않았고 사본도 암호화하지 않으려면 Disable encryption(암호화 비활성화)을 선택합니다.
 - DB 스냅샷이 암호화되지 않았지만 사본을 암호화하려면 Enable encryption(암호화 활성화)을 선택합니다. 이 경우 [Master Key]에 대해 DB 스냅샷 사본을 암호화할 때 사용할 KMS 키 식별자를 지정합니다.
 - DB 스냅샷이 암호화된 경우 Enable encryption(암호화 활성화)을 선택합니다. 이 경우 [Yes]가 이미 선택되어 있으므로 사본을 암호화해야 합니다. [Master Key]에 대해 DB 스냅샷 사본을 암호화할 때 사용할 KMS 키 식별자를 지정합니다.
10. [Copy Snapshot]을 선택합니다.

CLI

AWS CLI 명령 [copy-db-snapshot](#)을 사용하여 DB 스냅샷을 복사할 수 있습니다. 스냅샷을 새 AWS 리전으로 복사하는 경우 새 AWS 리전에서 명령을 실행하십시오.

다음 옵션을 사용하여 DB 스냅샷을 복사할 수 있습니다. 시나리오에 따라 필요하지 않은 옵션도 있습니다. 다음의 설명 및 예제를 사용하여 어느 옵션을 사용할지 결정하십시오.

- `--source-db-snapshot-identifier` – 원본 DB 스냅샷의 식별자입니다.
- 원본 스냅샷이 사본과 동일한 AWS 리전에 있는 경우 유효한 DB 스냅샷 식별자를 지정합니다. 예, `rds:mysql-instance1-snapshot-20130805`.
- 원본 스냅샷이 사본과 다른 AWS 리전에 있는 경우 유효한 DB 스냅샷 ARN을 지정합니다. 예, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
- 공유된 수동 DB 스냅샷에서 복사하는 경우에는 이 파라미터가 공유된 DB 스냅샷의 Amazon 리소스 이름(ARN)이어야 합니다.
- 암호화된 스냅샷에서 복사하는 경우 이 파라미터가 원본 AWS 리전의 ARN 형식이어야 하며 `PreSignedUrl` 파라미터의 `SourceDBSnapshotIdentifier`와 일치해야 합니다.
- `--target-db-snapshot-identifier` – 암호화된 DB 스냅샷의 새 사본의 식별자입니다.
- `--copy-tags` – 스냅샷의 태그와 값들을 스냅샷 사본에 복사하기 위한 태그 복사 옵션을 포함합니다.
- `--option-group-name` – 스냅샷 사본에 연결할 옵션 그룹입니다.

한 AWS 리전에서 다른 리전으로 스냅샷을 복사하고 DB 인스턴스가 기본값이 아닌 옵션 그룹을 사용하는 경우 이 옵션을 지정합니다.

원본 DB 인스턴스가 Oracle 또는 Microsoft SQL Server용 Transparent Data Encryption을 사용하는 경우 리전 간 복사 시 이 옵션을 지정해야 합니다. 자세한 내용은 [옵션 그룹 고려 사항 \(p. 209\)](#) 단원을 참조하십시오.

- `--kms-key-id` – 암호화된 DB 스냅샷의 AWS KMS 키 ID입니다. KMS 키 ID는 Amazon 리소스 이름(ARN), KMS 키 식별자 또는 KMS 암호화 키에 대한 KMS 키 별칭입니다.
 - AWS 계정에서 암호화된 DB 스냅샷을 복사하는 경우 이 파라미터에 값을 지정하여 새 KMS 암호화 키를 사용하여 사본을 암호화할 수 있습니다. 이 파라미터에 값을 지정하지 않을 경우 DB 스냅샷 사본이 원본 DB 스냅샷과 동일한 KMS 키를 사용하여 암호화됩니다.
 - 다른 AWS 계정에서 공유되는 암호화된 DB 스냅샷을 복사하는 경우 이 파라미터에 값을 지정해야 합니다.
 - 암호화되지 않은 스냅샷을 복사할 때 이 파라미터를 지정할 경우 사본이 암호화됩니다.
 - 암호화된 스냅샷을 다른 AWS 리전에 복사하는 경우 대상 AWS 리전에 대해 KMS 키를 지정해야 합니다. KMS 암호화 키는 해당 키를 만든 AWS 리전에 고유하며, 한 AWS 리전의 암호화 키를 다른 AWS 리전에서 사용할 수는 없습니다.
- `--source-region` – 원본 DB 스냅샷의 AWS 리전 ID입니다. 암호화된 스냅샷을 다른 AWS 리전에 복사하는 경우 이 옵션을 지정해야 합니다.

Example 암호화되지 않은 스냅샷을 동일한 리전으로 복사

다음 코드는 원본 스냅샷이 생성된 AWS 리전에서 `mydbsnapshotcopy`라는 새 이름으로 스냅샷 사본을 생성합니다. 복사가 완료되면 원래 스냅샷의 모든 태그가 스냅샷 사본에 복사됩니다.

Linux, OS X, Unix의 경우:

```
aws rds copy-db-snapshot \
  --source-db-snapshot-identifier mysql-instance1-snapshot-20130805 \
  --target-db-snapshot-identifier mydbsnapshotcopy \
  --copy-tags
```

Windows의 경우:

```
aws rds copy-db-snapshot ^
--source-db-snapshot-identifier mysql-instance1-snapshot-20130805 ^
--target-db-snapshot-identifier mydbsnapshotcopy ^
--copy-tags
```

Example 암호화되지 않은 스냅샷을 다른 리전으로 복사

다음 코드는 명령이 실행되는 AWS 리전에서 mydbsnapshotcopy라는 새 이름으로 스냅샷의 사본을 생성합니다.

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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 암호화된 스냅샷을 다른 리전으로 복사

다음 코드 예제는 암호화된 DB 스냅샷을 us-west-2 리전에서 us-east-1 리전으로 복사합니다. 명령은 us-east-1 리전에서 실행합니다.

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

API

Amazon RDS API 작업 [CopyDBSnapshot](#)을 사용하여 DB 스냅샷을 복사할 수 있습니다. 스냅샷을 새 AWS 리전으로 복사하는 경우 새 AWS 리전에서 작업을 수행하십시오.

다음 파라미터를 사용하여 DB 스냅샷을 복사할 수 있습니다. 시나리오에 따라 필요하지 않은 파라미터도 있습니다. 다음의 설명 및 예제를 사용하여 어느 파라미터를 사용할지 결정하십시오.

- **SourceDBSnapshotIdentifier** – 원본 DB 스냅샷의 식별자입니다.
 - 원본 스냅샷이 사본과 동일한 AWS 리전에 있는 경우 유효한 DB 스냅샷 식별자를 지정합니다. 예, rds:mysql-instance1-snapshot-20130805.
 - 원본 스냅샷이 사본과 다른 AWS 리전에 있는 경우 유효한 DB 스냅샷 ARN을 지정합니다. 예, arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805.
- 공유된 수동 DB 스냅샷에서 복사하는 경우에는 이 파라미터가 공유된 DB 스냅샷의 Amazon 리소스 이름(ARN)이어야 합니다.
- 암호화된 스냅샷에서 복사하는 경우 이 파라미터가 원본 AWS 리전의 ARN 형식이어야 하며 PreSignedUrl 파라미터의 SourceDBSnapshotIdentifier와 일치해야 합니다.
- **TargetDBSnapshotIdentifier** – 암호화된 DB 스냅샷의 새 사본의 식별자입니다.
- **CopyTags** – 스냅샷의 태그와 값을 스냅샷 사본에 복사하려면 이 파라미터를 **true**로 설정합니다. 기본값은 **false**입니다.
- **OptionGroupName** – 스냅샷 사본에 연결할 옵션 그룹입니다.

한 AWS 리전에서 다른 리전으로 스냅샷을 복사하고 DB 인스턴스가 기본값이 아닌 옵션 그룹을 사용하는 경우 이 파라미터를 지정합니다.

원본 DB 인스턴스가 Oracle 또는 Microsoft SQL Server용 Transparent Data Encryption을 사용하는 경우 리전 간 복사 시 이 파라미터를 지정해야 합니다. 자세한 내용은 [옵션 그룹 고려 사항 \(p. 209\)](#) 단원을 참조하십시오.

- **KmsKeyId** – 암호화된 DB 스냅샷의 AWS KMS 키 ID입니다. KMS 키 ID는 Amazon 리소스 이름(ARN), KMS 키 식별자 또는 KMS 암호화 키에 대한 KMS 키 별칭입니다.
 - AWS 계정에서 암호화된 DB 스냅샷을 복사하는 경우 이 파라미터에 값을 지정하여 새 KMS 암호화 키를 사용하여 사본을 암호화할 수 있습니다. 이 파라미터에 값을 지정하지 않을 경우 DB 스냅샷 사본이 원본 DB 스냅샷과 동일한 KMS 키를 사용하여 암호화됩니다.
 - 다른 AWS 계정에서 공유되는 암호화된 DB 스냅샷을 복사하는 경우 이 파라미터에 값을 지정해야 합니다.
 - 암호화되지 않은 스냅샷을 복사할 때 이 파라미터를 지정할 경우 사본이 암호화됩니다.
 - 암호화된 스냅샷을 다른 AWS 리전에 복사하는 경우 대상 AWS 리전에 대해 KMS 키를 지정해야 합니다. KMS 암호화 키는 해당 키를 만든 AWS 리전에 고유하며, 한 AWS 리전의 암호화 키를 다른 AWS 리전에서 사용할 수는 없습니다.
- **PreSignedUrl** – 복사할 원본 DB 스냅샷이 위치하는 원본 AWS 리전에서 CopyDBSnapshot API 작업에 대한 서명 버전 4 서명된 요청이 포함된 URL입니다.

Amazon RDS API를 사용하여 다른 AWS 리전으로부터 암호화된 DB 스냅샷을 복사하는 경우 이 파라미터를 지정해야 합니다. AWS CLI를 사용하여 다른 AWS 리전으로부터 암호화된 DB 스냅샷을 복사하는 경우 이 파라미터 대신 원본 리전을 지정할 수 있습니다.

미리 서명된 URL은 복사할 암호화된 DB 스냅샷이 위치하는 원본 AWS 리전에서 실행할 수 있는 CopyDBSnapshot API 작업에 대한 유효한 요청이어야 합니다. 미리 서명된 URL 요청은 다음 파라미터 값을 포함해야 합니다.

- **DestinationRegion** - 암호화된 DB 스냅샷이 복사될 AWS 리전입니다. 이 AWS 리전은 이 미리 서명된 URL을 포함하는, CopyDBSnapshot 작업이 호출되는 리전과 동일합니다.

예를 들어 us-west-2 리전에서 us-east-1 리전으로 암호화된 DB 스냅샷을 복사하는 경우 us-east-1 리전에서 CopyDBSnapshot 작업을 호출하고 us-west-2 리전에서 CopyDBSnapshot 작업에 대한 호출을 포함하는 미리 서명된 URL을 제공합니다. 이 예제의 경우, 미리 서명된 URL에서 DestinationRegion이 us-east-1 리전으로 설정되어야 합니다.

- **KmsKeyId** - 대상 AWS 리전에서 DB 스냅샷의 사본을 암호화하는 데 사용할 키에 대한 KMS 키 식별자. 이 식별자는 대상 AWS 리전에서 호출되는 CopyDBSnapshot 작업과 미리 서명된 URL에 포함된 작업 모두에 동일합니다.
- **SourceDBSnapshotIdentifier** - 복사할 암호화된 DB 스냅샷의 DB 스냅샷 식별자입니다. 이 식별자는 원본 AWS 리전용 Amazon 리소스 이름(ARN) 형식으로 되어 있어야 합니다. 예를 들어 us-

west-2 리전에서 암호화된 DB 스냅샷을 복사하는 경우 SourceDBSnapshotIdentifier는 다음 예와 같습니다. arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115.

서명 버전 4로 서명한 요청에 대한 자세한 내용은 다음을 참조하십시오.

- Amazon Simple Storage Service API Reference의 [요청 인증: 쿼리 파라미터 사용\(AWS 서명 버전 4\)](#)
- AWS General Reference의 [서명 버전 4 서명 프로세스](#)

Example 암호화되지 않은 스냅샷을 동일한 리전으로 복사

다음 코드는 원본 스냅샷이 생성된 AWS 리전에서 mydbsnapshotcopy라는 새 이름으로 스냅샷 사본을 생성합니다. 복사가 완료되면 원래 스냅샷의 모든 태그가 스냅샷 사본에 복사됩니다.

```
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 암호화되지 않은 스냅샷을 다른 리전으로 복사

다음 코드는 us-west-1 리전에서 mydbsnapshotcopy라는 새 이름으로 스냅샷 사본을 생성합니다.

```
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 암호화된 스냅샷을 다른 리전으로 복사

다음 코드는 us-east-1 리전에서 mydbsnapshotcopy라는 새 이름으로 스냅샷 사본을 생성합니다.

```
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
```

DB 스냅샷 공유

Amazon RDS를 사용하면 다른 방법으로 수동 DB 스냅샷을 공유할 수 있습니다.

- 암호화되었거나 암호화되지 않은 수동 DB 스냅샷을 공유하면 권한이 있는 AWS 계정에서 해당 스냅샷을 복사할 수 있습니다.
- 암호화되지 않은 수동 DB 스냅샷을 공유하면 권한이 있는 AWS 계정에서 스냅샷의 복사본을 만든 후 복원하는 대신에 해당 스냅샷에서 DB 인스턴스를 직접 복원할 수 있습니다. 그러나 공유되고 동시에 암호화된 DB 스냅샷에서는 DB 인스턴스를 복원할 수 없습니다. 대신 DB 스냅샷의 사본을 만들어 거기에서 DB 인스턴스를 복원할 수 있습니다.

Note

자동 DB 스냅샷을 공유하려면 자동 스냅샷을 복사하여 수동 DB 스냅샷을 생성한 다음 해당 복사본을 공유합니다.

스냅샷 복사에 대한 자세한 정보는 [스냅샷 복사 \(p. 207\)](#) 단원을 참조하십시오. DB 스냅샷에서 DB 인스턴스를 복원하는 방법에 대한 자세한 정보는 [DB 스냅샷에서 복원 \(p. 205\)](#) 단원을 참조하십시오.

최대 20개의 다른 AWS 계정과 수동 스냅샷을 공유할 수 있습니다. 암호화되지 않은 수동 스냅샷을 퍼블릭으로 공유할 수 있습니다. 그러면 모든 AWS 계정에서 해당 스냅샷을 사용할 수 있습니다. 스냅샷을 퍼블릭으로 공유할 경우 비공개 정보가 퍼블릭 스냅샷에 포함되지 않도록 유의하십시오.

수동 스냅샷을 다른 AWS 계정과 공유할 경우 다음과 같은 제한이 적용됩니다.

- AWS Command Line Interface(AWS CLI) 또는 Amazon RDS API를 사용하여 공유 스냅샷에서 DB 인스턴스 또는 DB 인스턴스를 복원할 때는 공유 스냅샷의 Amazon 리소스 이름(ARN)을 스냅샷 식별자로 지정해야 합니다.
- 영구적 또는 지속적 옵션이 포함된 옵션 그룹을 사용하는 DB 스냅샷은 공유할 수 없습니다.

영구적 옵션은 옵션 그룹에서 제거할 수 없습니다. 옵션 그룹이 DB 인스턴스에 할당된 경우 영구적 옵션을 포함하는 옵션 그룹을 DB 인스턴스에서 제거할 수 없습니다.

아래 표에 영구적 옵션과 지속적 옵션 및 관련된 DB 엔진이 나열되어 있습니다.

옵션 이름	지속적	영구적	DB 엔진
TDE	예	아니요	Microsoft SQL Server Enterprise Edition
TDE	예	예	Oracle Enterprise Edition
시간대	예	예	Oracle Enterprise Edition Oracle Standard Edition Oracle Standard Edition One Oracle Standard Edition Two

암호화된 스냅샷 공유

[Amazon RDS 리소스 암호화 \(p. 385\)](#)에 설명된 대로 AES-256 암호화 알고리즘을 사용하여 "저장 상태"에서 암호화된 DB 스냅샷을 공유할 수 있습니다. 이렇게 하려면 다음 단계를 따라야 합니다.

1. 스냅샷을 암호화하는 데 사용한 AWS Key Management Service(AWS KMS) 암호화 키를 해당 스냅샷에 액세스할 수 있도록 하려는 계정과 공유합니다.
KMS 키 정책에 다른 계정을 추가하여 AWS KMS 암호화 키를 다른 AWS 계정과 공유할 수 있습니다. 키 정책 업데이트에 관한 세부 정보는 AWS KMS 개발자 안내서의 [키 정책 단원](#)을 참조하십시오. 키 정책 생성의 예는 이번 주제 후반부의 [AWS KMS 암호화 키에 대한 액세스 허용 \(p. 216\)](#) 단원을 참조하십시오.
2. AWS Management 콘솔, AWS CLI 또는 Amazon RDS API를 사용하여 암호화된 스냅샷을 다른 계정과 공유합니다.

이러한 제한은 암호화된 스냅샷 공유에 적용됩니다.

- 암호화된 스냅샷을 퍼블릭으로는 공유할 수 없습니다.
- TDE(Transparent Data Encryption)를 사용하여 암호화된 Oracle 또는 Microsoft SQL Server 스냅샷은 공유할 수 없습니다.
- 스냅샷을 공유한 AWS 계정의 기본 AWS KMS 암호화 키를 사용하여 암호화된 스냅샷은 공유할 수 없습니다.

AWS KMS 암호화 키에 대한 액세스 허용

다른 AWS 계정이 사용자 계정에서 공유된 암호화된 DB 스냅샷을 복사하려면 스냅샷을 공유하는 계정에 스냅샷을 암호화한 KMS 키에 대한 액세스 권한이 있어야 합니다. 다른 AWS 계정이 AWS KMS 키에 액세스할 수 있도록 허용하려면 KMS 키에 대한 키 정책을 공유하는 AWS 계정의 ARN으로 업데이트합니다. 즉, KMS 키 정책에서 Principal을 포함하고 kms>CreateGrant 작업을 허용합니다.

AWS 계정에 사용자의 KMS 암호화 키에 대한 액세스 권한을 부여한 후에는, 암호화된 스냅샷을 복사하려면 해당 AWS 계정에서 AWS Identity and Access Management(IAM) 사용자를 만들어야 합니다(아직 없는 경우). 또한 IAM 사용자가 해당 KMS 키를 사용하여 암호화된 DB 스냅샷을 복사할 수 있도록 허용하는 IAM 정책을 이 AWS 계정에 연결해야 합니다. KMS 보안 제약으로 인해 이 계정은 IAM 사용자여야 하며 루트 AWS 계정 자격 증명일 수 없습니다.

다음 키 정책 예에서는 사용자 111122223333이 KMS 암호화 키의 소유자이고 사용자 444455556666이 키를 공유하는 계정입니다. 업데이트된 이 키 정책으로 인해 AWS 계정이 KMS 키에 액세스할 수 있습니다.

다. 사용자 444455556666에 대한 AWS 계정 자격 증명의 ARN을 정책에 대한 Principal로 포함하고 kms:CreateGrant 작업을 허용했습니다.

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

암호화된 스냅샷을 복사할 수 있도록 IAM 정책 만들기

외부 AWS 계정에 사용자의 KMS 키에 대한 액세스 권한을 부여한 후에는 해당 AWS 계정의 소유자가 이 계정에 대해 생성된 IAM 사용자가 해당 KMS 키를 사용하여 암호화된 스냅샷을 복사할 수 있게 하는 정책을 만들 수 있습니다.

다음 예에서는 AWS 계정 444455556666에 대해 IAM 사용자에게 추가할 수 있으며 IAM 사용자가 us-west-2 리전에서 KMS 키 c989c1dd-a3f2-4a5d-8d96-e793d082ab26을 사용하여 암호화된 AWS 계정 111122223333에서 공유 스냅샷을 복사할 수 있게 하는 정책을 보여 줍니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowUseOfTheKey",  
            "Effect": "Allow",  
            "Action": [  
                "kms:Encrypt",  
                "kms:Decrypt",  
                "kms:ReEncrypt*",  
                "kms:GenerateDataKey*",  
                "kms:DescribeKey"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

```
        "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
        }
    }
}
]
```

키 정책 업데이트에 관한 세부 정보는 AWS KMS 개발자 안내서의 [키 정책](#) 단원을 참조하십시오.

스냅샷 공유

AWS Management 콘솔, AWS CLI 또는 RDS API를 사용하여 DB 스냅샷을 공유할 수 있습니다.

AWS Management 콘솔

Amazon RDS 콘솔을 사용하여 최대 20개의 AWS 계정과 수동 DB 스냅샷을 공유할 수 있습니다. 콘솔을 사용하여 하나 이상의 계정에 대한 수동 스냅샷 공유를 중지할 수도 있습니다.

Amazon RDS 콘솔을 사용하여 수동 DB 스냅샷을 공유하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 공유할 수동 스냅샷을 선택합니다.
4. 작업에서 스냅샷 공유를 선택합니다.
5. DB 스냅샷 공개 여부에 대해 다음 옵션 중 하나를 선택합니다.
 - 소스가 암호화되어 있지 않은 경우 Public(퍼블릭)을 선택하여 모든 AWS 계정이 수동 DB 스냅샷에서 DB 인스턴스를 복원하도록 허용하거나, Private(프라이빗)을 선택하여 지정한 AWS 계정만 수동 DB 스냅샷에서 DB 인스턴스를 복원하도록 허용합니다.

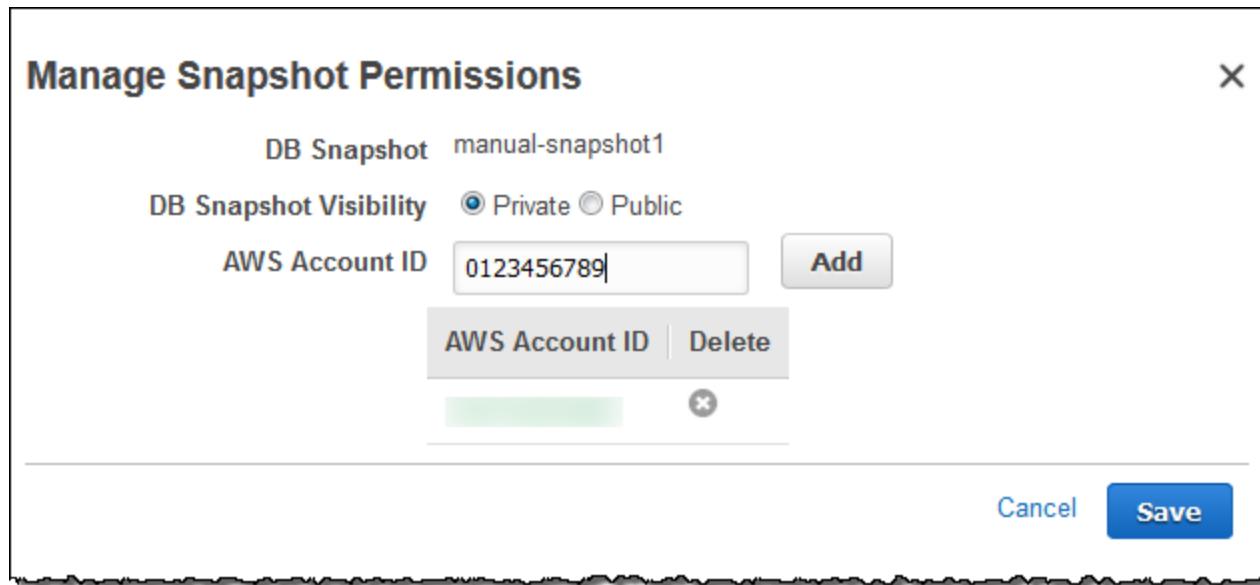
Warning

DB snapshot visibility(DB 스냅샷 가시성)를 Public(퍼블릭)으로 설정한 경우 모든 AWS 계정은 수동 DB 스냅샷에서 DB 인스턴스를 복원하고 사용자 데이터에 액세스할 수 있습니다. 프라이빗 정보가 포함된 수동 DB 스냅샷을 Public(퍼블릭)으로 공유하지 마십시오.

- 원본 DB 클러스터가 암호화되어 있는 경우 암호화된 스냅샷을 퍼블릭으로 공유할 수 없으므로 DB snapshot visibility(DB 스냅샷 가시성)는 Private(프라이빗)으로 설정됩니다.

- 수동 스냅샷에서 DB 인스턴스를 복원하도록 허용할 계정의 AWS 계정 식별자를 AWS 계정 ID에 입력한 다음 Add(추가)를 선택합니다. 이 과정을 반복하여 AWS 계정 식별자를 최대 20개까지 추가합니다.

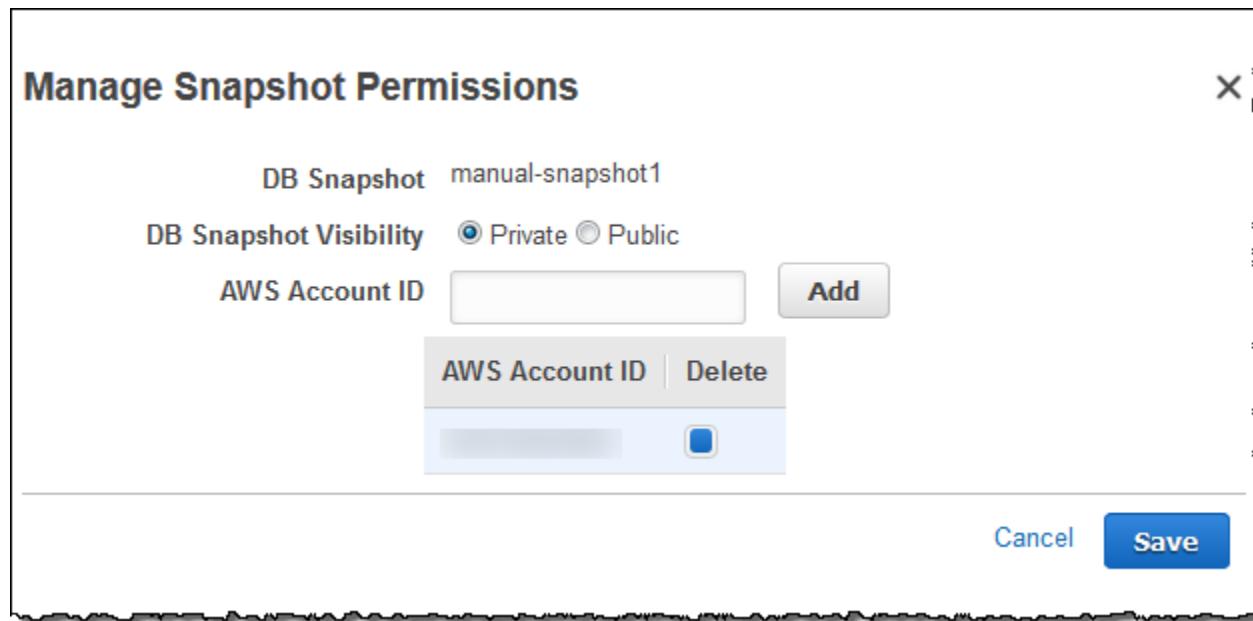
허용된 계정 목록에 AWS 계정 식별자를 추가하다가 실수하는 경우, 잘못된 AWS 계정 식별자의 오른쪽에 있는 [Delete]를 선택하여 목록에서 해당 식별자를 삭제할 수 있습니다.



- 수동 스냅샷을 복원할 수 있도록 허용할 모든 AWS 계정의 식별자를 추가한 후 [Save]를 선택하여 변경 내용을 저장합니다.

AWS 계정과 수동 DB 스냅샷 공유를 중지하려면

- AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
- 탐색 창에서 [Snapshots]를 선택합니다.
- 공유를 중지할 수동 스냅샷을 선택합니다.
- 작업을 선택한 다음 스냅샷 공유를 선택합니다.
- AWS 계정의 권한을 제거하려면 권한 있는 계정 목록에서 해당 계정의 AWS 계정 식별자에 대해 [Delete]를 선택합니다.



6. [Save]를 선택하여 변경 사항을 저장합니다.

AWS CLI

DB 스냅샷을 공유하려면 `aws rds modify-db-snapshot-attribute` 명령을 사용합니다. `--values-to-add` 파라미터를 사용하여 수동 스냅샷을 복원할 권한이 있는 AWS 계정의 ID 목록을 추가합니다.

다음 예제에서는 두 AWS 계정 식별자 123451234512 및 123456789012에게 `manual-snapshot1` 스냅샷 복원 권한을 부여하고 `all` 속성 값을 제거하여 DB 스냅샷을 비공개로 표시합니다.

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
--values-to-add '[ "111122223333", "444455556666" ]'
```

목록에서 AWS 계정 식별자를 제거하려면 `--values-to-remove` 파라미터를 사용합니다. 다음 예는 AWS 계정 ID 444455556666의 스냅샷 복원을 방지합니다.

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
--values-to-remove '[ "444455556666" ]'
```

API

Amazon RDS API를 사용하여 수동 DB 스냅샷을 다른 AWS 계정과 공유할 수도 있습니다. 이렇게 하려면 [ModifyDBSnapshotAttribute](#) 작업을 호출합니다. `AttributeName`에 대해 `restore`를 지정하고 `valuesToAdd` 파라미터를 사용하여 수동 스냅샷의 복원 권한이 있는 AWS 계정을 위한 ID 목록을 추가합니다.

수동 스냅샷을 퍼블릭으로 설정하고 모든 AWS 계정에서 복원할 수 있도록 하려면 `all` 값을 사용합니다. 단, 일부 AWS 계정에만 제공할 비공개 정보가 포함된 수동 스냅샷에 대해 `all` 값을 추가하지 않도록 유의합니다.

다. 또한 암호화된 스냅샷에 대해 `all`을 지정하지 마십시오. 그러한 스냅샷을 퍼블릭으로 설정하는 것은 지원되지 않습니다.

AWS 계정의 공유 권한을 제거하려면 [ModifyDBSnapshotAttribute](#) 작업을 사용하면서 `AttributeName`을 `restore`로 설정하고 `ValuesToRemove` 파라미터를 함께 사용합니다. 수동 스냅샷을 비공개로 하려면 `restore` 속성에 대한 값 목록에서 `all` 값을 제거합니다.

스냅샷을 복원할 수 있도록 허용된 모든 AWS 계정을 나열하려면 [DescribeDBSnapshotAttributes](#) API 작업을 사용합니다.

DB 인스턴스를 지정된 시간으로 복원

DB 인스턴스를 특정 시점으로 복원하여 새로운 DB 인스턴스를 만들 수 있습니다. DB 인스턴스를 특정 시점으로 복원할 경우, 새 DB 인스턴스에는 기본 DB 보안 그룹이 적용됩니다. 사용자 지정 DB 보안 그룹을 DB 인스턴스에 적용해야 한다면, DB 인스턴스를 사용할 수 있게 된 후 AWS Management 콘솔, AWS CLI `modify-db-instance` 명령 또는 Amazon RDS API `ModifyDBInstance` 작업을 사용하여 명시적으로 적용해야 합니다.

RDS는 DB 인스턴스에 대한 트랜잭션 로그를 Amazon S3에 5분마다 업로드합니다. DB 인스턴스의 최근 복원 가능 시간을 확인하려면 AWS CLI `describe-db-instances` 명령을 사용한 후 DB 인스턴스의 `LatestRestorableTime` 필드에 반환되는 값을 살펴보십시오. AWS Management 콘솔에서 이 속성은 DB 인스턴스의 최근 복원 시간으로 표시됩니다. 백업 보존 기간 중 어느 특정 시점으로든 복원할 수 있습니다.

Amazon RDS가 사용하는 여러 가지 데이터베이스 엔진에는 특정 시점에서 복원할 때 특별히 고려할 사항이 있습니다. Oracle DB 인스턴스를 특정 시점으로 복원할 때, 새 DB 인스턴스가 사용할 다른 Oracle DB 엔진, 라이선스 모델 및 DBName(SID)을 지정할 수 있습니다. SQL Server DB 인스턴스를 특정 시점으로 복원할 때 그 인스턴스 내의 각 데이터베이스는 인스턴스 내에 있는 각각의 다른 데이터베이스와 1초 이내의 시점으로 복원됩니다. 인스턴스 내에 있는 여러 데이터베이스에 걸쳐 이루어지는 트랜잭션은 일관되게 복원되지 않을 수 있습니다. SQL Server DB 인스턴스의 경우 `OFFLINE`, `EMERGENCY` 및 `SINGLE_USER` 모드는 현재 지원되지 않습니다. 데이터베이스를 이들 모드 중 하나로 설정하면 최근 복원 가능 시간이 전체 인스턴스를 앞서 가는 동작이 중지됩니다.

SQL Server 데이터베이스의 복구 모델 변경과 같은 동작으로 특정 시점으로 복구에 사용되는 로그 시퀀스가 중단될 수 있습니다. 어떤 경우에는 Amazon RDS가 이 문제를 탐지하여 최근 복원 가능 시간이 앞서 가지 못하게 할 수 있고, SQL Server 데이터베이스가 `BULK_LOGGED` 복구 모델을 사용할 때와 같은 그 밖의 경우에는 로그 시퀀스 중단이 탐지되지 않습니다. 로그 시퀀스가 중단될 경우 SQL Server DB 인스턴스를 특정 시점으로 복원하지 못할 수도 있습니다. 이런 이유로, Amazon RDS는 SQL Server 데이터베이스의 복구 모델 변경을 지원하지 않습니다.

AWS Management 콘솔, AWS CLI 또는 RDS API를 사용하여 DB 인스턴스를 특정 시점으로 복원할 수 있습니다.

AWS Management 콘솔

DB 인스턴스를 지정된 시간으로 복원하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 복구하려는 DB 인스턴스를 선택합니다.
4. 작업에서 특정 시점으로 복구를 선택합니다.

DB 인스턴스 시작 창이 나타납니다.

5. 최근 복원 가능 시간을 선택하여 가능한 최근 시간으로 복원하거나, 사용자 지정을 선택하여 시간을 선택합니다.

사용자 지정을 선택한 경우 인스턴스를 복원하려는 날짜와 시간을 입력합니다.

6. DB 인스턴스 식별자에 복원된 DB 인스턴스의 이름을 입력한 후, 다른 옵션을 설정합니다.
7. [Launch DB Instance]를 선택합니다.

CLI

DB 인스턴스를 특정 시점으로 복원하려면, AWS CLI 명령 [restore-db-instance-to-point-in-time](#)을 사용하여 새 DB 인스턴스를 만듭니다.

Example

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

API

DB 인스턴스를 특정 시간으로 복원하려면, Amazon RDS API [RestoreDBInstanceToPointInTime](#) 작업을 다음 파라미터와 함께 호출합니다.

- `SourceDBInstanceIdentifier`
- `TargetDBInstanceIdentifier`
- `RestoreTime`

스냅샷 삭제

Amazon RDS에서 관리하는 DB 스냅샷이 더 이상 필요하지 않으면 삭제할 수 있습니다.

Note

AWS Backup에서 관리하는 백업을 삭제하려면 AWS Backup 콘솔을 사용하십시오. AWS Backup에 대한 자세한 내용은 [AWS 백업 개발자 안내서](#)를 참조하십시오.

DB 스냅샷 삭제

AWS Management 콘솔, AWS CLI 또는 RDS API를 사용해 수동, 공유 또는 퍼블릭 DB 스냅샷을 삭제할 수 있습니다.

공유 또는 퍼블릭 스냅샷을 삭제하려면 해당 스냅샷을 소유하는 AWS 계정에 로그인해야 합니다.

DB 인스턴스를 삭제하지 않은 채로 삭제하고 싶은 자동화된 DB 스냅샷이 있는 경우 DB 인스턴스의 백업 보존 기간을 0으로 변경하십시오. 자동화된 스냅샷은 변경 내용이 적용되면 삭제됩니다. 그다음 유지 관리 기간까지 기다리고 싶지 않다면 변경 내용을 즉시 적용할 수 있습니다. 변경이 완료된 후에는 백업 보존 기간을 0보다 큰 수로 설정하여 자동 백업을 다시 활성화할 수 있습니다. DB 인스턴스 수정에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#) 단원을 참조하십시오.

DB 인스턴스를 삭제한 경우 DB 인스턴스에 대한 자동 백업을 제거하여 자동화된 DB 스냅샷을 삭제할 수 있습니다. 자동 백업에 대한 자세한 내용은 [백업 작업 \(p. 196\)](#) 단원을 참조하십시오.

콘솔

DB 스냅샷을 삭제하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 스냅샷을 선택합니다.
3. 삭제하고 싶은 DB 스냅샷을 선택합니다.
4. 작업에서 스냅샷 삭제를 선택합니다.
5. 확인 페이지에서 삭제를 선택합니다.

AWS CLI

AWS CLI 명령 `copy-db-snapshot`을 사용하여 DB 스냅샷을 삭제할 수 있습니다.

다음 옵션을 사용하여 DB 스냅샷을 삭제할 수 있습니다.

- `--db-snapshot-identifier` – DB 스냅샷 식별자입니다.

Example

다음 코드는 `mydbsnapshot` DB 스냅샷을 삭제합니다.

Linux, OS X, Unix의 경우:

```
aws rds delete-db-snapshot \
    --db-snapshot-identifier mydbsnapshot
```

Windows의 경우:

```
aws rds delete-db-snapshot ^
    --db-snapshot-identifier mydbsnapshot
```

RDS API

Amazon RDS API 연산 `DeleteDBSnapshot`을 사용하여 DB 스냅샷을 삭제할 수 있습니다.

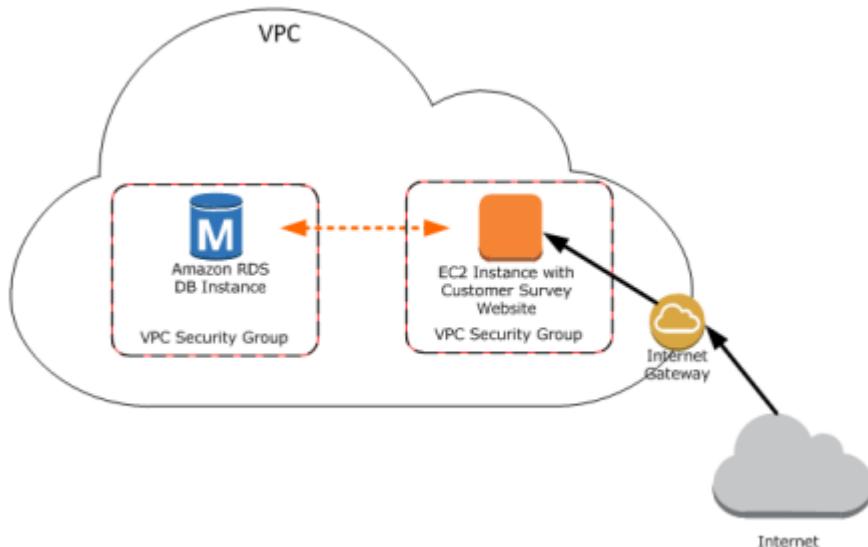
다음 파라미터를 사용하여 DB 스냅샷을 삭제할 수 있습니다.

- `DBSnapshotIdentifier` – DB 스냅샷 식별자입니다.

자습: DB 스냅샷에서 DB 인스턴스 복원

Amazon RDS를 사용하여 작업할 때의 일반적인 시나리오는 항상 필요하지는 않지만 때때로 DB 인스턴스를 사용하여 작업하는 상황입니다. 예를 들어, Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 사용하여 고객 설문 웹 사이트를 호스트하는 분기별 고객 설문 작업이 있고 이러한 설문 결과를 저장하는 데 사용되는 DB 인스턴스를 가지고 있을 수 있습니다. 이러한 시나리오에서 비용을 절약하는 한 가지 방법은 설문이 완료된 후 DB 인스턴스의 DB 스냅샷을 만들고 DB 인스턴스를 삭제한 다음, 설문을 다시 수행해야 할 때 해당 DB 인스턴스를 복원하는 것입니다.

다음 그림은 고객 설문 웹 사이트를 호스팅하는 EC2 인스턴스가 고객 설문 데이터를 보관하는 DB 인스턴스와 동일한 Amazon Virtual Private Cloud(Amazon VPC)에 있는 경우 가능한 시나리오입니다. 단, 각 인스턴스에는 고유의 보안 그룹이 있으며, EC2 인스턴스 보안 그룹은 인터넷에서의 액세스를 허용하는 반면, DB 인스턴스 보안 그룹은 EC2 인스턴스와의 액세스만 허용합니다. 설문이 완료되면, EC2 인스턴스는 중지될 수 있으며 DB 인스턴스는 최종 DB 스냅샷이 생성된 후 삭제될 수 있습니다. 나중에 다시 설문을 수행해야 하는 경우 EC2 인스턴스를 재시작하고 DB 스냅샷에서 DB 인스턴스를 복원할 수 있습니다.



EC2 인스턴스가 DB 인스턴스에 연결되도록 허용하는 이 시나리오에 필요한 VPC 보안 그룹 설정 방법에 관한 자세한 내용은 [동일한 VPC에 있는 EC2 인스턴스가 VPC 내에 있는 DB 인스턴스에 액세스 \(p. 409\)](#) 단원을 참조하십시오.

DB 인스턴스를 복원하기 전에 DB 스냅샷을 생성해야 합니다. DB 인스턴스를 복원하는 경우 복원 원본으로 사용할 DB 스냅샷의 이름을 입력하고 이 복원 작업에서 생성되는 새 DB 인스턴스의 이름을 입력합니다. DB 스냅샷에서 기존 DB 인스턴스로 복원할 수는 없습니다. 복원할 때 새 DB 인스턴스가 생성됩니다.

DB 스냅샷에서 DB 인스턴스 복원 필수 조건

복원된 DB 인스턴스의 일부 설정은 인스턴스가 복원될 때 다시 설정되므로 원래의 리소스를 보관하고 있어야 DB 인스턴스를 이전 설정으로 복원할 수 있습니다. 예를 들어, DB 스냅샷으로부터 DB 인스턴스를 복원하는 경우 기본 DB 파라미터와 기본 보안 그룹만 복원된 인스턴스와 연결됩니다. 여기서 연결이 의미하는 것은 기본 보안 그룹은 DB 인스턴스에 대한 액세스를 허용하지 않으며, 기본 파라미터 그룹에서 사용할 수 있는 사용자 지정 파라미터 설정이 없다는 것입니다. 사용자가 DB 스냅샷을 생성할 때 사용한 DB 인스턴스와 연결된 보안 그룹 및 DB 파라미터 그룹을 보관해야 합니다.

DB 스냅샷으로부터 DB 인스턴스를 복원하려면 다음 사항이 필요합니다.

- DB 인스턴스의 DB 스냅샷을 생성해 둬야 DB 스냅샷에서 DB 인스턴스를 복원할 수 있습니다. DB 스냅샷 생성에 대한 자세한 내용은 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.
- 사용자가 DB 스냅샷을 생성할 때 원본으로 사용한 DB 인스턴스와 연결된 파라미터 그룹 및 보안 그룹을 보관해야 합니다.
- 복원된 DB 인스턴스에 대한 올바른 옵션 그룹을 결정해야 합니다.
 - 복원할 때 원본으로 사용한 DB 스냅샷과 연결된 옵션 그룹은 생성되고 나면 복원된 DB 인스턴스와 연결됩니다. 예를 들어, 복원 원본으로 사용하려는 DB 스냅샷에서 Oracle Transparent Data Encryption(TDE)을 사용하는 경우 복원된 DB 인스턴스에서도 TDE 옵션이 포함된 동일한 옵션 그룹을 사용합니다.
 - 다른 VPC나 다른 플랫폼으로 해당 인스턴스를 복구하려고 해도 원래 DB 인스턴스에 연결된 옵션 그룹은 사용할 수 없습니다. 옵션 그룹이 DB 인스턴스에 배정되는 경우 DB 인스턴스가 위치한 플랫폼(VPC

또는 EC2-Classic(VPC 아님)에도 연결되기 때문에 이러한 제한 사항이 발생합니다. DB 인스턴스가 VPC에 속한 경우 해당 인스턴스와 연결된 옵션 그룹은 바로 이 VPC에도 연결됩니다.

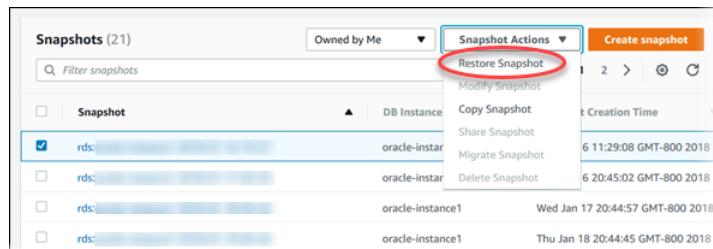
- DB 인스턴스를 다른 VPC나 다른 플랫폼으로 복구할 경우 기본 옵션 그룹을 인스턴스에 할당하거나, 해당 VPC 또는 플랫폼에 연결된 옵션 그룹을 할당하거나, 새 옵션 그룹을 생성하여 DB 인스턴스에 할당해야 합니다. Oracle TDE와 같은 지속적 또는 영구적 옵션을 사용하여 DB 인스턴스를 다른 VPC로 복구할 때는 지속적 또는 영구적 옵션을 포함하는 새 옵션 그룹을 생성해야 합니다. 옵션 그룹 작업에 대한 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#)을(를) 참조하십시오.

DB 스냅샷에서 DB 인스턴스 복원

AWS Management 콘솔에서 다음 절차를 통해 스냅샷으로부터 복원할 수 있습니다.

DB 스냅샷에서 DB 인스턴스를 복원하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 복원 원본으로 사용할 DB 스냅샷을 선택합니다.
4. 작업에서 스냅샷 복원을 선택합니다.



[Restore DB Instance] 페이지가 나타납니다.

5. 설정 아래의 DB 인스턴스 식별자에 복원된 DB 인스턴스에 사용하려는 이름을 입력합니다. DB 스냅샷을 만든 후 삭제한 DB 인스턴스로부터 복원하는 경우 해당 DB 인스턴스의 이름을 사용할 수 있습니다.
6. [Restore DB Instance]를 선택합니다.

복원된 DB 인스턴스 수정

복원 작업이 완료되는 즉시, 복원 원본 인스턴스에서 사용하는 사용자 지정 보안 그룹을 사용자가 가지고 있을 수 있는 사용자 지정 DB 파라미터 그룹과 연결해야 합니다. 기본 DB 파라미터 및 보안 그룹만 복원된 인스턴스와 연결됩니다. DB 인스턴스의 기능을 스냅샷을 생성한 DB 인스턴스의 기능과 동일하게 복원하려면 DB 인스턴스를 이전 DB 인스턴스가 사용하는 보안 그룹 및 파라미터 그룹을 사용하도록 수정해야 합니다.

DB 인스턴스를 사용할 수 있게 되면 RDS 콘솔의 `Modify` 명령, `ModifyDBInstance` API 또는 `aws rds modify-db-instance` 명령줄 도구를 명시적으로 사용하여 이러한 변경 내용을 적용해야 합니다. 복원된 인스턴스와 올바른 파라미터 파일을 연결할 수 있도록 보유하고 있는 DB 스냅샷에 대한 파라미터 그룹을 유지하는 것이 좋습니다.

복원된 DB 인스턴스의 다른 설정을 수정할 수 있습니다. 예를 들어, 원본 DB 스냅샷과 다른 스토리지 유형을 사용할 수 있습니다. 이 경우 데이터를 새로운 스토리지 유형으로 마이그레이션하기 위해 추가 작업이 필요하기 때문에 복원 프로세스가 오래 걸립니다. Magnetic(스탠다드) 스토리지로 또는 이 스토리지에서 복원할 경우 Magnetic 스토리지에는 프로비저닝된 IOPS나 범용(SSD) 스토리지의 IOPS 기능이 없기 때문에 마이그레이션이 가장 오래 걸립니다.

다음 단계부터는 사용자의 DB 인스턴스가 VPC에 있는 것으로 가정합니다. DB 인스턴스가 VPC에 있지 않은 경우에는 AWS Management 콘솔을 사용해 DB 인스턴스에 대해 필요한 DB 보안 그룹을 찾아야 합니다.

복원된 DB 인스턴스의 설정을 원래 DB 인스턴스의 설정과 같게 하는 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 세부 정보를 표시하고자 DB 스냅샷에서 복원할 때 생성한 DB 인스턴스의 이름을 선택합니다. 연결 탭을 선택합니다. DB 인스턴스에 할당된 보안 그룹은 액세스를 허용하지 않을 수 있습니다. 인바운드 규칙이 없는 경우 인바운드 액세스를 허용하는 권한이 없습니다.

Security group	Type	Rule
default ()	No applicable security group roles	N/A

4. [Modify]를 선택합니다.
5. Network & Security(네트워크 및 보안) 섹션에서 DB 인스턴스에 사용할 보안 그룹을 선택합니다. EC2 인스턴스와 함께 사용할 새 보안 그룹을 생성하는 규칙을 추가해야 하는 경우 [동일한 VPC에 있는 EC2 인스턴스가 VPC 내에 있는 DB 인스턴스에 액세스 \(p. 409\)](#) 단원에서 자세한 내용을 참조하십시오.

보안 그룹과 연결된 X를 선택하여 보안 그룹을 제거할 수도 있습니다.

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. [Continue]를 선택한 후 [Apply immediately]를 선택합니다.
7. [Modify DB Instance]를 선택합니다.

인스턴스 상태가 제공된 이후 DB 인스턴스 이름을 선택하면 세부 정보가 표시됩니다. 연결 탭을 선택하고, 새 보안 그룹이 적용되어 DB 인스턴스에 대한 액세스가 허가되었는지 확인합니다.

Details	
Configurations	Security and network
ARN arn:aws:rds:us-east-1:814387698303:db:restored-db-instance	Availability zone us-east-1a
Engine Oracle Enterprise Edition 12.1.0.2.v10	VPC gs-cluster-vpc ([])
License Model Bring Your Own License	Subnet group gs-subnetgroup1
Created Time Wed Jan 24 15:34:07 GMT-800 2018	Subnets subnet-[] subnet-[]
DB Name ORCL	Security groups gs-securitygroup1 ([]) (active)
Username oracleadmin	Publicly accessible Yes

Amazon RDS 모니터링

이 단원에서는 Amazon RDS를 모니터링하는 방법을 설명합니다.

주제

- [Amazon RDS 모니터링 개요 \(p. 228\)](#)
- [확장 모니터링 \(p. 238\)](#)
- [Amazon RDS 성능 개선 도우미 사용 \(p. 249\)](#)
- [Amazon RDS 권장 사항 사용 \(p. 286\)](#)
- [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#)
- [Amazon RDS 이벤트 보기 \(p. 305\)](#)
- [Amazon RDS 데이터베이스 로그 파일 \(p. 306\)](#)
- [AWS CloudTrail을 사용하여 Amazon RDS API 호출 로깅 \(p. 335\)](#)

Amazon RDS 모니터링 개요

모니터링은 Amazon RDS와 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. 발생하는 다중 지점 실패를 보다 쉽게 디버깅할 수 있도록 AWS 솔루션의 모든 부분으로부터 모니터링 데이터를 수집해야 합니다. Amazon RDS 모니터링을 시작하기 전에 다음 질문에 대한 답변을 포함하는 모니터링 계획을 작성하는 것이 좋습니다.

- 모니터링의 목표
- 모니터링할 리소스
- 이러한 리소스를 모니터링하는 빈도
- 사용할 모니터링 도구
- 모니터링 작업을 수행할 사람
- 문제 발생 시 알려야 할 대상

다음 단계에서는 다양한 시간과 다양한 부하 조건에서 성능을 측정하여 환경에서 일반 Amazon RDS 성능의 기준선을 설정합니다. Amazon RDS를 모니터링할 때 과거 모니터링 데이터를 저장할 것을 고려해야 합니다. 이 저장된 데이터는 현재 성능 데이터와 비교하고, 일반 성능 패턴과 성능 이상을 식별하고, 문제 해결 방법을 제안하는 기준이 됩니다.

예를 들어 Amazon RDS에서 네트워크 처리량, 읽기, 쓰기 및/또는 메타데이터 작업에 대한 I/O, 클라이언트 연결, DB 인스턴스에 대한 버스트 크레딧 밸런스 등을 모니터링할 수 있습니다. 성능이 설정된 기준을 벗어날 경우 워크로드에 대한 데이터베이스 가용성을 최적화하기 위해 클라이언트에 사용 가능한 DB 인스턴스 및 읽기 전용 복제본의 수 또는 DB 인스턴스의 인스턴스 클래스를 변경해야 할 수 있습니다.

일반적으로 성능 지표에 허용되는 값은 기준이 무엇인지 그리고 애플리케이션 무엇을 수행하는지에 따라 달립니다. 기준과의 일관된 차이 또는 추세를 조사하십시오. 특정 지표 유형에 대한 참고 정보는 다음과 같습니다.

- CPU 또는 RAM 사용량이 많음 – CPU 또는 RAM 사용량이 많을 경우 해당 애플리케이션의 목표와 일치하고 예상되는 결과라면 문제가 되지 않을 수 있습니다.
- 디스크 공간 사용량 – 총 디스크 용량의 85퍼센트 이상이 계속 사용될 경우 디스크 공간 사용량을 검사합니다. 인스턴스에서 데이터를 삭제할 수 있는지 또는 다른 시스템에 데이터를 아카이브하여 공간을 확보할 수 있는지 확인합니다.

- 네트워크 트래픽 – 네트워크 트래픽의 경우 시스템 관리자에게 문의하여 해당 도메인 네트워크 및 인터넷 연결의 기대 처리량을 확인합니다. 처리량이 기대값보다 항상 낮으면 네트워크 트래픽을 검사합니다.
- 데이터베이스 연결 – 인스턴스 성능 저하 및 응답 시간 지연과 함께 사용자 연결 수가 많을 경우 데이터베이스 연결 제한을 고려해 봅니다. DB 인스턴스에 대한 최적의 사용자 연결 수는 해당 인스턴스 클래스와, 수행하는 작업의 복잡성에 따라 다릅니다. DB 인스턴스를 User Connections 파라미터가 0(무제한)이 아닌 다른 값으로 설정된 파라미터 그룹과 연결하여 데이터베이스 연결 수를 지정할 수 있습니다. 기존 파라미터 그룹을 사용하거나 새로 하나 만들 수 있습니다. 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.
- IOPS 지표 – IOPS 지표의 기대값은 디스크 사양 및 서버 구성에 따라 다르므로 해당 기준에 일반적인 값을 파악합니다. 값이 기준과 계속 차이가 나는지 검사합니다. 최적의 IOPS 성능을 위해, 일반적인 작업 세트가 메모리에 적합하고 읽기 및 쓰기 작업을 최소화하는지 확인합니다.

모니터링 도구

AWS는 Amazon RDS를 모니터링하는 데 사용할 수 있는 다양한 도구를 제공합니다. 이들 도구 중에는 모니터링을 자동으로 수행하도록 구성할 수 있는 도구도 있지만, 수동 작업이 필요한 도구도 있습니다. 모니터링 작업을 최대한 자동화하는 것이 좋습니다.

자동 모니터링 도구

다음과 같은 자동 모니터링 도구를 사용하여 Amazon RDS를 관찰하고 문제 발생 시 보고할 수 있습니다.

- Amazon RDS 이벤트 – DB 인스턴스, , DB 스냅샷, , DB 파라미터 그룹 또는 DB 보안 그룹에 변경 사항이 있을 경우 알려주는 Amazon RDS 이벤트에 가입합니다. 자세한 내용은 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오.
- 데이터베이스 로그 파일 – Amazon RDS 콘솔 또는 Amazon RDS API 작업을 사용해 데이터베이스 로그 파일을 보거나, 다운로드하거나, 모니터링합니다. 또한 데이터베이스 테이블에 로드된 데이터베이스 로그 파일 중 일부에 쿼리를 요청할 수도 있습니다. 자세한 내용은 [Amazon RDS 데이터베이스 로그 파일 \(p. 306\)](#) 단원을 참조하십시오.
- Amazon RDS 확장된 모니터링 - 운영 체제에 대한 지표를 실시간으로 확인합니다. 자세한 내용은 [확장 모니터링 \(p. 238\)](#) 단원을 참조하십시오.

또한 Amazon RDS는 추가 모니터링 기능을 위해 Amazon CloudWatch와 통합됩니다.

- Amazon CloudWatch 지표 - Amazon RDS는 각각의 활성 데이터베이스 인스턴스에 대해 1분마다 CloudWatch로 지표를 전송합니다. CloudWatch에서 Amazon RDS 지표에 대해서는 추가 요금이 청구되지 않습니다. 자세한 내용은 [the section called “DB 인스턴스 측정치 보기” \(p. 236\)](#) 단원을 참조하십시오.
- Amazon CloudWatch 경보 - 특정 기간 동안 단일 Amazon RDS 지표를 감시하고, 설정한 임계값을 기준으로 측정치의 값에 따라 하나 이상의 작업을 수행할 수 있습니다. 자세한 내용은 [Amazon CloudWatch를 사용하여 모니터링 \(p. 230\)](#) 단원을 참조하십시오.
- Amazon CloudWatch Logs - 대부분의 DB 엔진에서 CloudWatch Logs의 데이터베이스 로그 파일을 모니터링 및 저장하고 액세스할 수 있습니다. 자세한 내용은 [Amazon CloudWatch Logs User Guide](#) 단원을 참조하십시오.

수동 모니터링 도구

Amazon RDS 모니터링의 또 한 가지 중요한 부분은 CloudWatch 경보에 포함되지 않는 항목을 수동으로 모니터링해야 한다는 점입니다. Amazon RDS, CloudWatch, AWS Trusted Advisor 및 다른 AWS 콘솔 대시보드에서 AWS 환경의 상태를 한눈에 파악할 수 있습니다. 또한 DB instance에서 로그 파일을 확인하는 것이 좋습니다.

- Amazon RDS 콘솔에서 리소스에 대해 다음과 같은 항목을 모니터링할 수 있습니다.

- DB 인스턴스에 대한 연결 수
- DB 인스턴스에 대한 읽기 및 쓰기 작업량
- DB 인스턴스에서 현재 사용 중인 스토리지의 양
- DB 인스턴스에 대해 사용 중인 메모리 및 CPU 양
- DB 인스턴스에서 주고 받는 네트워크 트래픽 양
- AWS Trusted Advisor 대시보드에서는 다음과 같은 비용 최적화, 보안, 내결함성과 성능 개선 확인을 살펴 볼 수 있습니다.
 - Amazon RDS 유 휴 DB 인스턴스
 - Amazon RDS 보안 그룹 액세스 위험
 - Amazon RDS 백업
 - Amazon RDS 다중 AZ

이러한 사항에 대한 자세한 정보를 알고 싶다면 [Trusted Advisor Best Practices \(Checks\)](#) 단원을 참조하십시오.

- CloudWatch 험 페이지에 표시되는 항목은 다음과 같습니다.
 - 현재 경보 및 상태
 - 경보 및 리소스 그래프
 - 서비스 상태

또한 CloudWatch를 사용하여 다음 작업을 수행할 수도 있습니다.

- [사용자 정의 대시보드](#)를 만들어 원하는 서비스 모니터링
- 지표 데이터를 그래프로 작성하여 문제를 해결하고 추세 파악
- 모든 AWS 리소스 지표 검색 및 찾아보기
- 문제에 대해 알려주는 경보 생성 및 편집

Amazon CloudWatch를 사용하여 모니터링

Amazon RDS로부터 원시 데이터를 수집하여 읽기 가능한 실시간 지표로 처리하는 Amazon CloudWatch를 사용하여 DB 인스턴스를 모니터링할 수 있습니다. 이러한 통계는 2주간 기록되므로 기록 정보를 보고 웹 애플리케이션이나 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다. Amazon RDS 지표 데이터는 기본적으로 1분 단위로 CloudWatch에 자동 전송됩니다. CloudWatch에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 [Amazon CloudWatch](#), [Amazon CloudWatch Events](#), [Amazon CloudWatch Logs](#)이란 무엇인가?를 참조하십시오.

Amazon RDS 지표 및 차원

Amazon RDS 리소스를 사용할 때 Amazon RDS에서 1분마다 지표 및 차원을 Amazon CloudWatch에 보냅니다. 다음 절차에 따라 Amazon RDS에 대한 지표를 볼 수 있습니다.

Amazon CloudWatch 콘솔을 사용한 메트릭 확인

측정치는 먼저 서비스 네임스페이스별로 그룹화된 다음, 각 네임스페이스 내에서 다양한 차원 조합별로 그룹화됩니다.

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 필요한 경우 AWS 리전을 변경합니다. 탐색 모음에서 AWS 리소스가 상주하는 AWS 리전을 선택합니다. 자세한 내용은 [리전 및 앤드포인트](#) 단원을 참조하십시오.
3. 탐색 창에서 [Metrics]를 선택합니다. RDS 측정치 네임스페이스를 선택합니다.

The screenshot shows the AWS CloudWatch Metrics console. At the top, there are three tabs: "All metrics" (selected), "Graphed metrics", and "Graph options". Below the tabs, the navigation path is "All > RDS". A search bar contains the placeholder text "Search for any metric, dimension or resource id". The main area displays "168 Metrics" under three categories:

- DbClusterIdentifier, EngineName**: 3 Metrics
- Per-Database Metrics**: 45 Metrics
- By Database Class**: 45 Metrics

4. 측정치 차원(예: 데이터베이스 클래스별)을 선택합니다.
5. 지표를 정렬하려면 열 머리글을 사용합니다. 측정치를 그래프로 표시하려면 측정치 옆에 있는 확인란을 선택합니다. 리소스로 필터링하려면 리소스 ID를 선택한 후 검색에 추가를 선택합니다. 지표로 필터링하려면 지표 이름을 선택한 후 검색에 추가를 선택합니다.

The screenshot shows the AWS CloudWatch Metrics console with the "Graphed metrics (1)" tab selected. The navigation path is "All > RDS > By Database Class". The list shows metrics for "DatabaseClass (45)". The "db.r4.large" metric is selected, highlighted with a blue background. A context menu is open over this row, listing the following options:

- Add to search
- Search for this only
- Remove from graph
- Graph this metric only
- Graph all search results
- What is this?

AWS CLI를 사용하여 지표를 보려면

- 명령 프롬프트에서 다음 명령을 사용합니다.

```
aws cloudwatch list-metrics --namespace AWS/RDS
```

Amazon RDS 지표

AWS/RDS 네임스페이스에는 다음 지표가 포함되어 있습니다.

지표	설명
BinLogDiskUsage	마스터에서 이진 로그가 차지하는 디스크 공간 크기. MySQL 읽기 전용 복제본에 적용. 단위: 바이트
BurstBalance	사용할 수 있는 범용 SSD(gp2) 버스트-버킷 I/O 크레딧 비율 단위: 백분율
CPUUtilization	CPU 사용 백분율. 단위: 백분율
CPUCreditUsage	[T2 인스턴스] CPU 사용률을 위해 인스턴스에서 소비되는 CPU 크레딧의 수입니다. CPU 크레딧 하나는 1분 동안 100%의 사용률로 실행되는 vCPU 1개 또는 이와 동등한 vCPU, 사용률 및 시간의 조합과 동일합니다(예를 들어 2분 동안 50%의 사용률로 실행되는 vCPU 1개 또는 2분 동안 25%의 사용률로 실행되는 vCPU 2개). CPU 크레딧 측정치는 5분 간격으로만 제공됩니다. 5분 이상의 시간을 지정할 경우 Sum 통계 대신 Average 통계를 사용하십시오. 단위: 크레딧 (vCPU-분)
CPUCreditBalance	[T2 인스턴스] 시작 이후 인스턴스가 누적한 획득 CPU 크레딧 수입니다. T2 스탠다드의 경우 CPUCreditBalance에 누적된 시작 크레딧 수도 포함됩니다. 크레딧은 획득 이후에 크레딧 밸런스에 누적되고, 소비 시 크레딧 밸런스에서 소멸됩니다. 크레딧 밸런스는 최대 한도(인스턴스 크기에 따라 결정)가 있습니다. 한도에 도달하고 하면 새로 획득한 모든 크레딧은 삭제됩니다. T2 스탠다드의 경우 시작 크레딧은 한도에 포함되지 않습니다. CPUCreditBalance의 크레딧은 인스턴스가 기준 CPU 사용률 이상으로 버스터를 하는 데 소비할 수 있습니다. 인스턴스가 실행 중인 동안 CPUCreditBalance의 크레딧은 만료되지 않습니다. 인스턴스가 종지되면 CPUCreditBalance는 지속되지 않고 모든 누적된 크레딧이 삭제됩니다. CPU 크레딧 측정치는 5분 간격으로만 제공됩니다. 단위: 크레딧 (vCPU-분)
DatabaseConnections	사용 중인 데이터베이스 연결 수.

지표	설명
	단위: 개수
DiskQueueDepth	디스크 액세스를 대기 중인 I/O(읽기/쓰기 요청) 수. 단위: 개수
FailedSQLServerAgentJobs	최근 1분간 실패한 SQL Server 에이전트 작업의 수입니다. 단위: 수/분
FreeableMemory	사용 가능한 RAM 크기. 단위: 바이트
FreeStorageSpace	사용 가능한 스토리지 공간 크기. 단위: 바이트
MaximumUsedTransactionIDs	사용된 최대 전역 트랜잭션 ID. PostgreSQL에 적용됩니다. 단위: 개수
NetworkReceiveThroughput	DB 인스턴스 수신 네트워크 트래픽(고객 데이터베이스 트래픽과 모니터링 및 복제에 사용된 Amazon RDS 트래픽을 모두 포함). 단위: 바이트/초
NetworkTransmitThroughput	DB 인스턴스 송신 네트워크 트래픽(고객 데이터베이스 트래픽과 모니터링 및 복제에 사용된 Amazon RDS 트래픽을 모두 포함). 단위: 바이트/초
OldestReplicationSlotLag	수신된 WAL 데이터를 기준으로 가장 지연된 복제본의 지연 크기. PostgreSQL에 적용됩니다. 단위: 메가바이트
ReadIOPS	초당 평균 디스크 읽기 I/O 연산 수 단위: 개수/초
ReadLatency	디스크 I/O 연산당 평균 처리 시간. 단위: 초
ReadThroughput	초당 디스크에서 읽은 평균 바이트 수. 단위: 바이트/초
ReplicaLag	소스 DB 인스턴스를 기준으로 읽기 전용 복제본 DB 인스턴스의 지연 시간. MySQL, MariaDB 및 PostgreSQL 읽기 전용 복제본에 적용. 단위: 초
ReplicationSlotDiskUsage	복제 슬롯 파일에 사용된 디스크 공간. PostgreSQL에 적용됩니다. 단위: 메가바이트

지표	설명
<code>SwapUsage</code>	DB 인스턴스에서 사용된 스왑 공간 크기. SQL 서버에는 이 지표를 사용할 수 없습니다. 단위: 바이트
<code>TransactionLogsDiskUsage</code>	트랜잭션 로그에 사용된 디스크 공간. PostgreSQL에 적용됩니다. 단위: 메가바이트
<code>TransactionLogsGeneration</code>	초당 생성되는 트랜잭션 로그의 크기. PostgreSQL에 적용됩니다. 단위: 메가바이트/초
<code>WriteIOPS</code>	초당 평균 디스크 쓰기 I/O 연산 수 단위: 개수/초
<code>WriteLatency</code>	디스크 I/O 연산당 평균 처리 시간. 단위: 초
<code>WriteThroughput</code>	초당 디스크에 쓴 평균 바이트 수. 단위: 바이트/초

Amazon RDS 차원

다음 표의 차원을 사용하여 Amazon RDS 지표 데이터를 필터링할 수 있습니다.

차원	설명
<code>DBInstanceIdentifier</code>	이 차원은 특정 데이터베이스 인스턴스에 대해 요청하는 데이터를 필터링합니다.
<code>DBClusterIdentifier</code>	이 차원은 특정 Amazon Aurora DB 클러스터에 대해 요청하는 데이터를 필터링합니다.
<code>DBClusterIdentifier, Role</code>	이 차원은 인스턴스 역할(WRITER/READER)별로 지표를 집계하여 특정 Aurora DB 클러스터에 대해 요청하는 데이터를 필터링합니다. 예를 들어 클러스터에 속하는 모든 READER 인스턴스에 대한 지표를 집계할 수 있습니다.
<code>DatabaseClass</code>	이 차원은 특정 데이터베이스 클래스의 모든 인스턴스에 대해 요청하는 데이터를 필터링합니다. 예를 들어 데이터베이스 클래스 db.m1.small에 속하는 모든 인스턴스에 대한 지표를 집계할 수 있습니다.
<code>EngineName</code>	이 차원은 식별된 엔진 이름에 대해 요청하는 데이터만 필터링합니다. 예를 들어 엔진 이름이 mysql인 모든 인스턴스에 대한 지표를 집계할 수 있습니다.
<code>SourceRegion</code>	이 차원은 지정된 리전에 한해 요청한 데이터를 필터링합니다. 예를 들어 us-east-1 리전의 모든 인스턴스에 대한 지표를 집계할 수 있습니다.

Amazon RDS를 모니터링하는 CloudWatch 경보 생성

경보가 상태를 변경하면 Amazon SNS 메시지를 보내는 CloudWatch 경보를 만들 수 있습니다. 경보는 지정한 기간에 단일 지표를 감시하고 여러 기간에 지정된 임계값에 대한 지표 값을 기준으로 작업을 하나 이상 수행합니다. 이 작업은 Amazon SNS 주제나 Auto Scaling 정책으로 전송되는 알림입니다.

경보는 지속적인 상태 변경에 대해서만 작업을 호출합니다. CloudWatch 경보는 특정 상태가 되었다고 해서 작업을 호출하지는 않습니다. 이러한 상태가 변경되어 지정한 기간 동안 유지되어야 합니다. 다음 절차에서는 Amazon RDS에 대한 경보를 만드는 방법을 보여 줍니다.

CloudWatch 콘솔을 사용한 경보 설정

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. [Alarms]를 선택한 다음 [Create Alarm]을 선택합니다. 그러면 [Create Alarm Wizard]가 시작됩니다.
3. RDS 지표를 선택하고 Amazon RDS 지표를 스크롤하여 경보를 생성할 지표를 찾습니다. 이 대화 상자에서 Amazon RDS 지표를 표시하려면 리소스의 식별자를 검색합니다. 지표를 선택하여 경보를 생성한 다음 다음을 선택합니다.
4. 지표에 대한 이름, 설명 및 다음 경우 항상 값을 입력합니다.
5. 경보 상태에 도달하면 CloudWatch에서 이메일을 보내도록 하려면 이 경보가 발생할 경우 항상:에서 상태가 ALARM입니다.를 선택합니다. Send a notification to:(다음 주소로 알림 전송)에서 기존 SNS 주제를 선택합니다. Create topic(주제 생성)을 선택한 경우 새 이메일 구독 목록에 대한 이름 및 이메일 주소를 설정할 수 있습니다. 이 목록은 향후 경보를 위해 필드에 저장되고 표시됩니다.

Note

새 Amazon SNS 주제를 생성하기 위해 주제 생성을 사용할 경우 이메일 주소는 알림을 받기 전에 검증되어야 합니다. 이메일은 경보가 경보 상태에 입력될 때만 전송됩니다. 이러한 경보 상태 변경이 이메일이 검증되기 전에 발생할 경우에는 알림을 받지 못합니다.

6. 이제 경보 미리 보기 영역에서 생성할 경보를 미리 볼 수 있습니다. 경보 생성을 선택합니다.

AWS CLI를 사용하여 경보를 설정하려면

- `put-metric-alarm`을 호출합니다. 자세한 내용은 [AWS CLI Command Reference](#) 단원을 참조하십시오.

CloudWatch API를 사용하여 경보를 설정하려면

- `PutMetricAlarm`을 호출합니다. 자세한 내용은 [Amazon CloudWatch API Reference](#) 단원을 참조하십시오.

Amazon CloudWatch Logs에 데이터베이스 엔진 로그 게시

Amazon CloudWatch Logs의 로그 그룹에 로그 데이터를 게시하도록 Amazon RDS 데이터베이스 엔진을 구성할 수 있습니다. CloudWatch Logs를 통해 로그 데이터에 대한 실시간 분석을 수행할 수 있고, CloudWatch를 사용하여 경보를 만들고 지표를 볼 수 있습니다. CloudWatch Logs를 사용하여 CloudWatch Logs 에이전트로 관리할 수 있는 내구성이 뛰어난 스토리지에 로그 레코드를 저장할 수 있습니다. 예를 들어 로그 레코드를 호스트에서 로그 서비스로 교체할 시기를 결정할 수 있으므로 필요할 때 원시 로그에 액세스할 수 있습니다.

Amazon RDS MariaDB(모든 버전)와 Amazon RDS MySQL(버전 5.6, 5.7, 8.0)에 대한 로그를 내보낼 수 있습니다.

Note

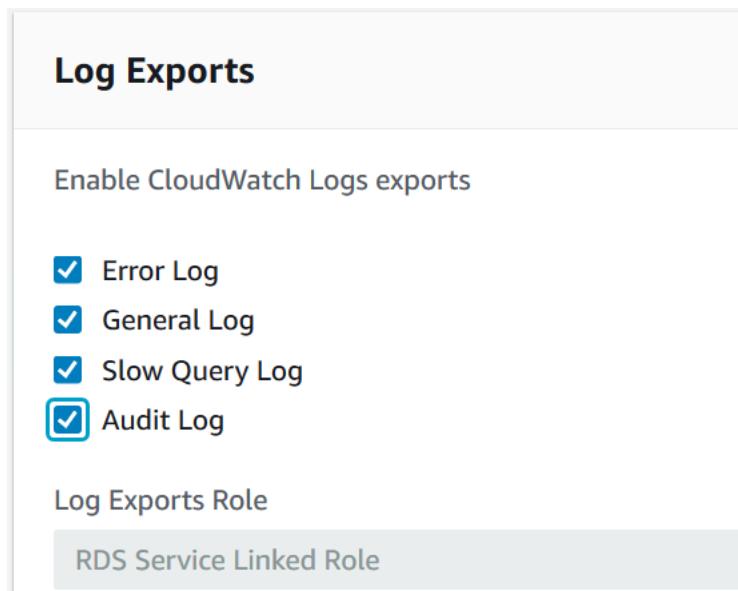
로그 데이터 게시를 활성화하려면 서비스 연결 역할이 있어야 합니다. 서비스 연결 역할에 대한 자세한 내용은 [Amazon RDS에 서비스 연결 역할 사용 \(p. 403\)](#) 단원을 참조하십시오.

이러한 엔진의 특정 요구 사항은 다음을 참조하십시오.

- the section called “Amazon CloudWatch Logs에 MariaDB 로그 게시” (p. 311)
- the section called “CloudWatch Logs에 MySQL 로그 게시” (p. 320)

CloudWatch 로그 통합 구성

CloudWatch Logs에 데이터베이스 로그 파일을 게시하려면 게시할 로그를 선택합니다. 새 DB 인스턴스를 만들 때 [Advanced Settings] 섹션에서 이를 선택합니다. 기존 DB 인스턴스를 수정하여 게시를 시작해도 됩니다.



게시를 활성화하면 Amazon RDS는 모든 DB 인스턴스 로그 레코드를 로그 그룹으로 계속 스트리밍합니다. 예를 들어 게시하는 각 로그 유형에 대해 /aws/rds/instance/log_type 로그 그룹이 있습니다. 이 로그 그룹은 로그를 생성하는 데이터베이스 인스턴스와 동일한 AWS 리전에 있습니다.

로그 레코드를 게시한 후 CloudWatch Logs를 사용하여 레코드를 검색 및 필터링할 수 있습니다. 로그 검색 및 필터링에 관한 자세한 내용은 [로그 데이터 검색 및 필터링](#)을 참조하십시오.

DB 인스턴스 측정치 보기

Amazon RDS는 DB 인스턴스의 상태를 모니터링할 수 있도록 지표를 제공합니다. DB 인스턴스 측정치와 운영 체제(OS) 측정치를 모두 모니터링할 수 있습니다.

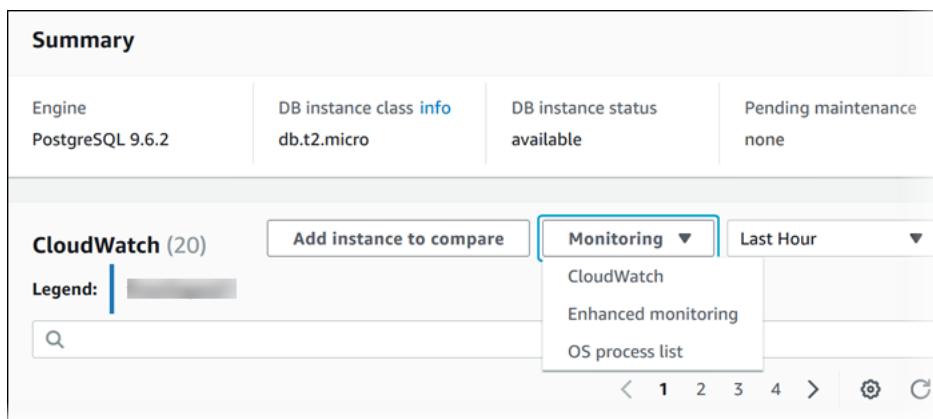
이 단원에서는 RDS 콘솔 및 CloudWatch를 사용하여 DB 인스턴스에 대한 지표를 확인하는 방법을 자세히 설명합니다. CloudWatch Logs를 사용하여 DB 인스턴스의 운영 체제에 대한 지표를 모니터링하는 방법에 대한 자세한 내용은 [확장 모니터링 \(p. 238\)](#) 단원을 참조하십시오.

콘솔을 사용하여 측정치 보기

DB 인스턴스에 대한 DB 및 OS 측정치를 보려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.

2. 탐색 창에서 데이터베이스를 선택합니다.
3. 정보가 필요한 DB 인스턴스의 이름을 선택하여 세부 정보를 표시합니다.
4. [Monitoring] 탭을 선택합니다.
5. 모니터링에서 측정치를 표시할 방법을 지정하는 옵션을 선택합니다.
 - CloudWatch - Amazon CloudWatch에서 사용 가능한 DB 인스턴스 측정치를 요약하여 표시합니다. 각 지표에는 특정 시간대에서 지표를 모니터링한 그래프도 포함되어 있습니다.
 - 확장 모니터링 - 확장 모니터링을 활성화한 상태로 DB 인스턴스에 대해 사용 가능한 OS 지표를 요약하여 표시합니다. 각 지표에는 특정 시간대에서 지표를 모니터링한 그래프도 포함되어 있습니다.
 - OS 프로세스 목록 - 선택한 인스턴스에서 실행되는 각 프로세스의 세부 정보를 표시합니다.



Tip

시간 범위 목록을 사용하여 그래프로 표시된 측정치의 시간 범위를 선택할 수 있습니다. 그래프를 선택하여 더 세부적인 보기 를 불러올 수 있습니다. 측정치별 필터를 데이터에 적용할 수도 있습니다.

CLI 또는 API를 사용하여 DB 인스턴스 지표 보기

Amazon RDS는 CloudWatch 측정치와 통합되어 다양한 DB 인스턴스 측정치를 제공합니다. CloudWatch 지표는 RDS 콘솔, AWS CLI 또는 API를 사용하여 볼 수 있습니다.

전체 Amazon RDS 지표 목록은 Amazon CloudWatch 사용 설명서의 [Amazon RDS 차원 및 지표](#)에서 확인할 수 있습니다.

CloudWatch CLI를 사용하여 DB 측정치 보기

Note

다음 CLI 예제를 실행하려면 CloudWatch 명령줄 도구가 필요합니다. CloudWatch에 대한 자세한 내용을 보거나 개발자 도구를 다운로드하려면 [Amazon CloudWatch 제품 페이지](#)를 참조하십시오. 이 예제에 나온 StartTime 및EndTime 값은 설명을 돋기 위해 지정되었습니다. 따라서 DB 인스턴스의 올바른 시작 및 종료 시간 값으로 대체해야 합니다.

DB 인스턴스의 사용량 및 성능 통계를 보는 방법

- CloudWatch 명령 mon-get-stats를 다음 파라미터와 함께 사용합니다.

```
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
```

CloudWatch API를 사용하여 DB 측정치 보기

이 예제에 나온 StartTime 및 EndTime 값은 설명을 돋기 위해 지정되었습니다. 따라서 DB 인스턴스의 올바른 시작 및 종료 시간 값으로 대체해야 합니다.

DB 인스턴스의 사용량 및 성능 통계를 보는 방법

- 다음 파라미터와 함께 CloudWatch API GetMetricStatistics를 호출합니다.
 - Statistics.member.1 = Average
 - Namespace = AWS/RDS
 - StartTime = 2009-10-16T00:00:00
 - EndTime = 2009-10-16T00:02:00
 - Period = 60
 - MeasureName = FreeStorageSpace

확장 모니터링

Amazon RDS는 DB 인스턴스가 실행되는 운영 체제(OS)에 대한 측정치를 실시간으로 제공합니다. 콘솔을 사용하여 DB 인스턴스에 대한 측정치를 보거나, 선택한 모니터링 시스템의 Amazon CloudWatch Logs에서 Enhanced Monitoring JSON 출력을 사용할 수 있습니다.

기본적으로 확장 모니터링 지표는 CloudWatch Logs 로그에 30일간 저장됩니다. 지표가 CloudWatch Logs에 저장되는 시간을 수정하려면 CloudWatch 콘솔에서 RDSOSMetrics 로그 그룹의 보존을 변경하십시오. 자세한 내용은 Amazon CloudWatch Logs User Guide의 [CloudWatch에서 로그 데이터 보존 기간을 변경](#)을 참조하십시오.

확장 모니터링 사용 비용은 다음과 같은 몇 가지 요인에 따라 달라집니다.

- Amazon CloudWatch Logs이 제공하는 프리 티어를 초과하는 Enhanced Monitoring에 대해서만 비용이 청구됩니다.
요금에 대한 자세한 내용은 [Amazon CloudWatch 요금](#)을 참조하십시오.
- 모니터링 간격이 작을수록 OS 측정치가 더 자주 보고되고 모니터링 비용이 증가합니다.
- Enhanced Monitoring 사용 비용은 Enhanced Monitoring을 활성화한 각 DB 인스턴스에 대해 적용됩니다. 모니터링하는 DB 인스턴스의 수가 많을수록 더 많은 비용이 청구됩니다.
- 컴퓨팅 집약적인 워크로드를 지원하는 DB 인스턴스는 많은 OS 프로세스 활동이 보고되고 Enhanced Monitoring에 대한 높은 비용이 청구됩니다.

Enhanced Monitoring 가용성

Enhanced Monitoring은 다음 데이터베이스 엔진에 사용할 수 있습니다.

- MariaDB
- Microsoft SQL Server
- MySQL 버전 5.5 이상
- Oracle
- PostgreSQL

확장 모니터링은 db.m1.small을 제외한 모든 DB 인스턴스 클래스에서 사용할 수 있습니다.

CloudWatch 측정치와 Enhanced Monitoring 측정치의 차이점

CloudWatch는 DB 인스턴스의 하이퍼바이저에서 CPU 사용률에 대한 측정치를 수집하고, Enhanced Monitoring에서는 인스턴스의 에이전트에서 측정치를 수집합니다. 하이퍼바이저 계층에서는 소량의 작업만 수행하므로 두 측정치 간의 차이점을 확인할 수 있습니다. DB 인스턴스에서 사용하는 인스턴스 클래스가 적을수록 단일 물리적 인스턴스에서 하이퍼바이저 계층에 의해 관리되는 가상 머신(VM)의 수가 더 많아지므로 차이가 더 커질 수 있습니다. Enhanced Monitoring 측정치는 DB 인스턴스의 여러 프로세스 또는 스레드에서 CPU를 사용하는 방법을 확인하려는 경우에 유용합니다.

확장 모니터링 설정 및 활성화

확장 모니터링을 설정하고 활성화하려면 아래 나열된 단계를 수행하십시오.

시작하기 전

Enhanced Monitoring은 사용자를 대신하여 CloudWatch Logs에 OS 측정치 정보를 보낼 수 있는 권한이 필요합니다. AWS Identity and Access Management(IAM) 역할을 사용하여 확장 모니터링에 필요한 권한을 부여합니다.

콘솔에서 Enhanced Monitoring을 처음으로 활성화할 때 [Monitoring Role] 속성에 대한 [Default] 옵션을 선택하여 RDS에서 필요한 IAM 역할을 생성하도록 할 수 있습니다. 그러면 RDS에서 rds-monitoring-role 역할을 자동으로 생성하여 지정된 DB 인스턴스 또는 읽기 전용 복제본에 대해 사용합니다.

Enhanced Monitoring을 활성화하기 전에 필요한 역할을 생성한 다음 Enhanced Monitoring을 활성화할 때 새 역할을 이름을 지정할 수도 있습니다. AWS CLI 또는 RDS API를 사용하여 Enhanced Monitoring을 활성화할 경우 이 필수 역할을 생성해야 합니다.

적절한 IAM 역할을 생성하여 Amazon RDS에게 사용자를 대신하여 Amazon CloudWatch Logs 서비스와 통신하도록 허용하려면 다음 단계를 수행합니다.

확장 모니터링을 활성화하는 사용자는 PassRole 권한을 부여받아야 합니다. 자세한 내용은 IAM 사용 설명서의 [사용자에게 AWS 서비스에 역할을 전달할 수 있는 권한 부여](#)에 있는 예제 2를 참조하십시오.

Amazon RDS Enhanced Monitoring에 대한 IAM 역할을 생성하려면

1. <https://console.aws.amazon.com>에서 **IAM** 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. [Create role]을 선택합니다.
4. [AWS service] 템을 선택한 다음 서비스 목록에서 [RDS]를 선택합니다.
5. RDS - Enhanced Monitoring(RDS - 확장 모니터링)을 선택한 다음 Next: Permissions(다음: 권한)를 선택합니다.
6. Attached permissions policy(연결된 권한 정책) 페이지에 AmazonRDSEnhancedMonitoringRole이 표시되었는지 확인한 다음 Next: Tags(다음: 태그)를 선택합니다.
7. Add tags(태그 추가) 페이지에서 Next: Review(다음: 검토)를 선택합니다.
8. Role Name(역할 이름)에 **emaccess**와 같은 역할의 이름을 입력한 다음 Create role(역할 생성)을 선택합니다.

Enhanced Monitoring 활성화 및 비활성화

DB 인스턴스 또는 읽기 전용 복제본을 생성하거나 DB 인스턴스를 수정할 때 확장 모니터링을 활성화할 수 있습니다. 확장 모니터링을 활성화하기 위해 DB 인스턴스를 수정하는 경우 DB 인스턴스를 재부팅하지 않아도 변경 내용이 적용됩니다.

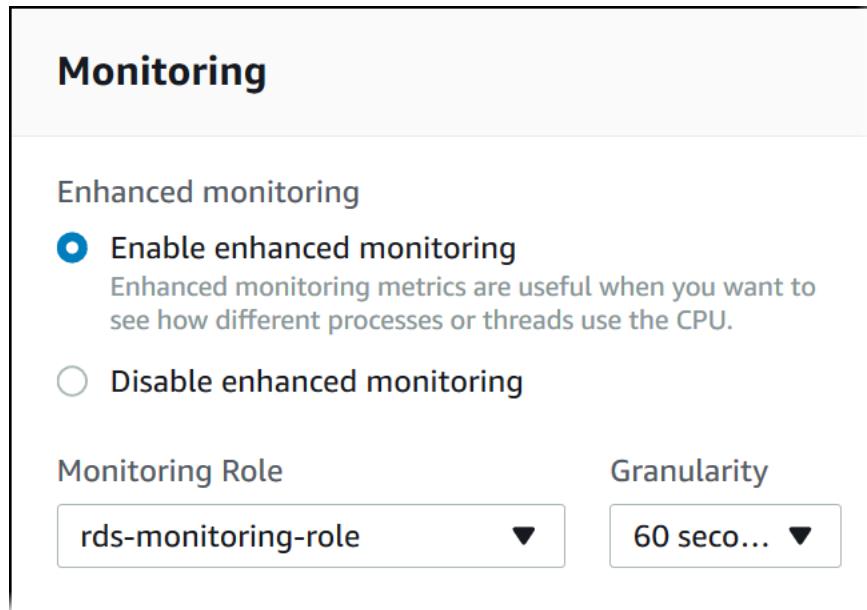
다음 작업 중 하나를 수행할 때 RDS 콘솔에서 확장 모니터링을 활성화할 수 있습니다.

- DB 인스턴스 생성 – 고급 설정 구성 페이지에서 확장 모니터링을 활성화할 수 있습니다.
- 읽기 전용 복제본 생성 – 고급 설정 구성 페이지에서 확장 모니터링을 활성화할 수 있습니다.
- [Modify a DB Instance] – [Modify DB Instance] 페이지에서 Enhanced Monitoring을 활성화할 수 있습니다.

RDS 콘솔을 사용하여 Enhanced Monitoring을 활성화하려면 [Monitoring] 섹션으로 스크롤한 후 다음 작업을 수행합니다.

1. DB 인스턴스 또는 읽기 전용 복제본에 대해 [Enable enhanced monitoring]을 선택합니다.
2. Amazon RDS에 사용자를 대신하여 Amazon CloudWatch Logs와 통신하도록 허용하기 위해 생성한 IAM 역할에 대한 [Monitoring Role] 속성을 설정하거나, [Default]를 선택하여 RDS에서 rds-monitoring-role 역할을 자동으로 생성하도록 합니다.
3. [Granularity] 속성은 DB 인스턴스 또는 읽기 전용 복제본에 대해 측정치를 수집하는 시점 간의 간격(초)으로 설정합니다. [Granularity] 속성을 1, 5, 10, 15, 30 또는 60 값 중 하나로 설정할 수 있습니다.

Enhanced Monitoring을 비활성화하려면 [Disable enhanced monitoring]을 선택합니다.



Enable Enhanced Monitoring의 경우 DB 인스턴스를 다시 시작하지 않아도 됩니다.

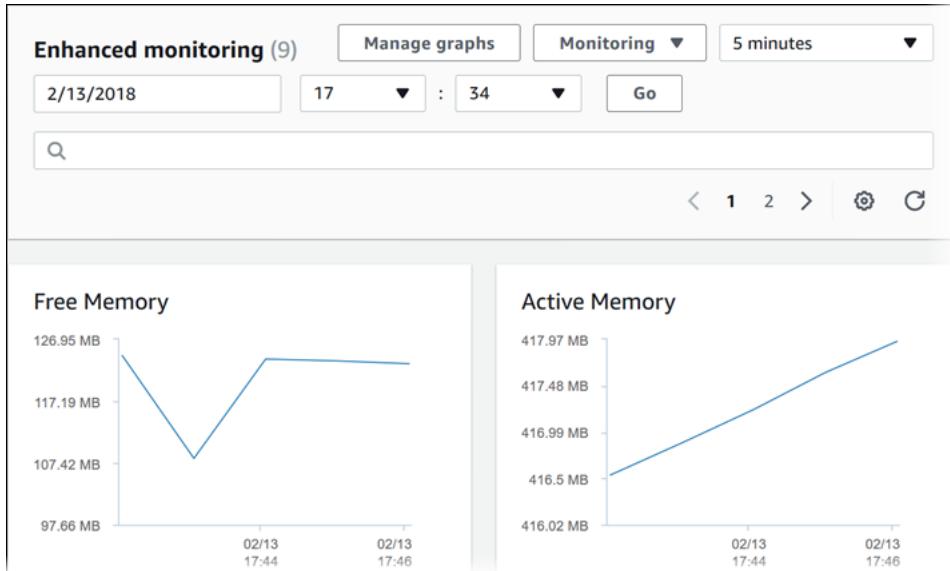
Note

RDS 콘솔을 새로 고치는 최소 간격은 5초입니다. RDS 콘솔에서 단위를 1초로 설정한 경우에도 업데이트된 측정치는 5초마다 표시됩니다. CloudWatch Logs를 사용하여 1초 측정치 업데이트를 검색할 수 있습니다.

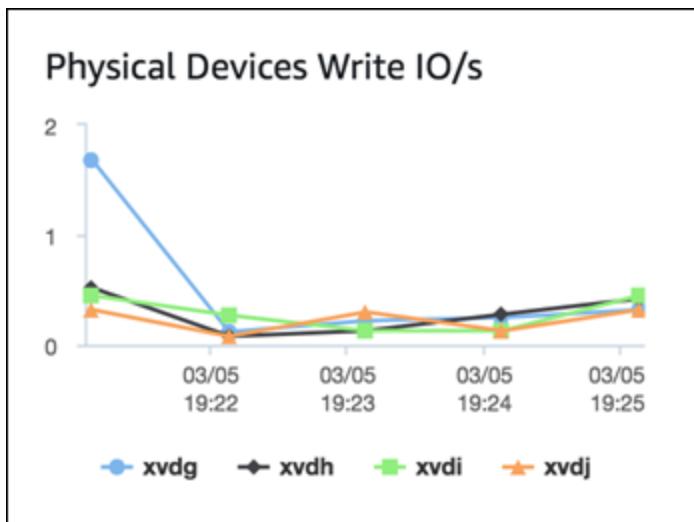
확장 모니터링 보기

모니터링에서 확장 모니터링을 선택하면 RDS 콘솔에서 확장 모니터링이 보고하는 OS 측정치를 볼 수 있습니다.

확장 모니터링 페이지가 다음과 같이 표시됩니다.



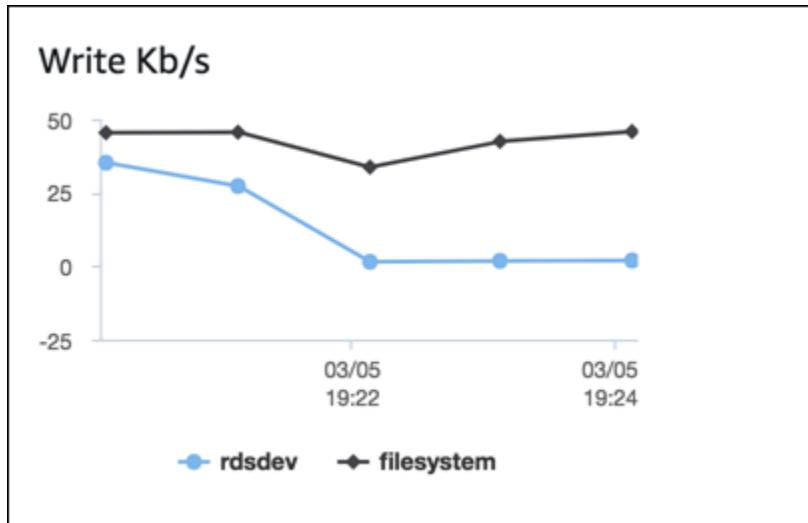
일부 DB 인스턴스에서는 DB 인스턴스의 데이터 스토리지 볼륨에 대해 한 개 이상의 디스크를 사용합니다. 이 DB 인스턴스의 Physical Devices(물리적 디바이스) 그래프에는 각 디스크에 대한 지표가 표시됩니다. 예를 들어 다음 그래프에는 디스크 4개에 대한 지표가 표시되어 있습니다.



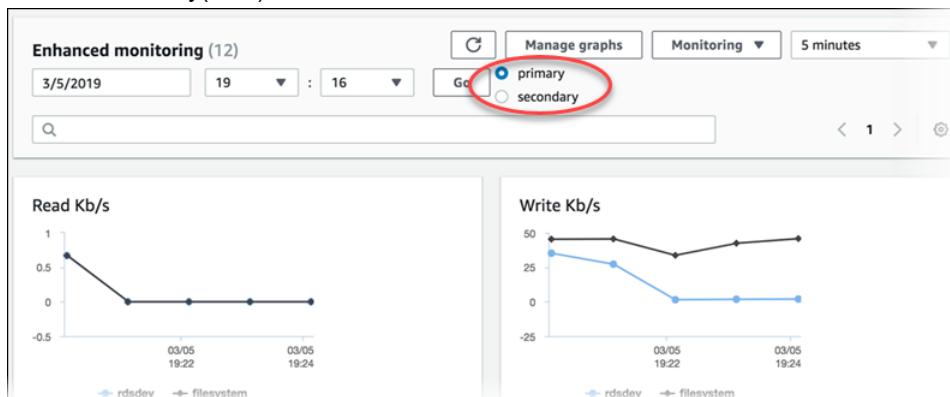
Note

현재 Physical Devices(물리적 디바이스) 그래프는 Microsoft SQL Server DB 인스턴스에는 제공되지 않습니다.

집계된 디스크 I/O 및 파일 시스템 그래프를 볼 때 rdsdev 디바이스는 모든 데이터베이스 파일 및 로그가 저 장되는 /rdsdbdata 파일 시스템과 관련이 있습니다. filesystem 디바이스는 운영 체제와 관련된 파일이 저 장되는 / 파일 시스템(루트라고도 함)과 관련이 있습니다.



DB 인스턴스가 다중 AZ 배포인 경우 기본 DB 인스턴스 및 다중 AZ 스탠바이 복제본에 대한 OS 지표를 볼 수 있습니다. 확장 모니터링 보기에서 primary(기본)을 선택하여 기본 DB 인스턴스에 대한 OS 지표를 확인하거나 secondary(보조)를 선택하여 스탠바이 복제본에 대한 OS 지표를 확인하십시오.



다중 AZ 배포에 대한 자세한 정보는 [Amazon RDS를 위한 고가용성\(다중 AZ\) \(p. 109\)](#) 단원을 참조하십시오.

Note

현재 다중 AZ 스탠바이 복제본에 대한 OS 지표를 보는 기능은 MariaDB, Microsoft SQL Server 또는 PostgreSQL DB 인스턴스에서는 지원되지 않습니다.

DB 인스턴스에서 실행 중인 프로세스에 대한 자세한 정보를 보려면 [Monitoring]에 대해 [OS process list]를 선택합니다.

프로세스 목록 보기는 다음과 같이 표시됩니다.

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						

프로세스 목록 보기에서 표시되는 확장 모니터링 지표는 다음과 같이 구성됩니다.

- RDS child processes(RDS 하위 프로세스) – DB 인스턴스를 지원하는 RDS 프로세스(예: MySQL DB 인스턴스의 경우 mysqld)를 요약하여 표시합니다. 프로세스 스레드는 상위 프로세스 아래에 중첩되어 표시됩니다. 프로세스 스레드에는 CPU 사용률만 표시됩니다. 다른 측정치는 프로세스의 모든 스레드에 대해 동일합니다. 콘솔에는 최대 100개의 프로세스와 스레드가 표시됩니다. 결과에는 CPU와 메모리를 소비하는 상위 프로세스 및 스레드가 함께 표시됩니다. 프로세스와 스레드가 각각 50개 이상씩 있는 경우 콘솔에는 각 범주의 상위 50개 소비자가 표시됩니다. 이 표시를 통해 성능에 가장 큰 영향을 미치고 있는 프로세스를 식별할 수 있습니다.
- [RDS processes] – RDS DB 인스턴스를 지원하는 데 필요한 RDS 관리 에이전트, 진단 모니터링 프로세스 및 기타 AWS 프로세스에서 사용되는 리소스를 요약하여 표시합니다.
- [OS processes] – 일반적으로 성능에 최소한의 영향만 미치는 커널 및 시스템 프로세스를 요약하여 표시합니다.

각 프로세스에 대해 나열되는 항목은 다음과 같습니다.

- VIRT – 프로세스의 가상 크기를 표시합니다.
- RES – 프로세스에서 사용 중인 실제 물리적 메모리를 표시합니다.
- CPU% – 프로세스에서 사용하는 CPU 대역폭의 비율을 표시합니다.
- MEM% – 프로세스에서 사용하는 총 메모리의 비율을 표시합니다.

RDS 콘솔에 표시되는 모니터링 데이터는 Amazon CloudWatch Logs으로부터 검색됩니다. CloudWatch Logs로부터 로그 스트림으로 DB 인스턴스용 측정치를 검색할 수도 있습니다. 자세한 내용은 [CloudWatch Logs를 사용하여 Enhanced Monitoring 보기 \(p. 244\)](#) 단원을 참조하십시오.

다음 기간 중에는 확장 모니터링 지표가 반환되지 않습니다.

- DB 인스턴스의 장애 조치 동안.
- DB 인스턴스의 인스턴스 클래스 변경(컴퓨팅 확장) 중.

Enhanced Monitoring 측정치는 데이터베이스 엔진이 재부팅되는 이유로만 DB 인스턴스의 재부팅 동안 반환됩니다. 운영 체제의 측정치는 계속 보고됩니다.

CloudWatch Logs를 사용하여 Enhanced Monitoring 보기

DB 인스턴스에 대한 Enhanced Monitoring을 활성화한 후 CloudWatch Logs를 사용하여 DB 인스턴스에 대한 측정치를 볼 수 있습니다. 각 로그 스트림에는 모니터링 중인 단일 DB 인스턴스가 표시됩니다. 로그 스트림 식별자는 DB 인스턴스에 대한 리소스 식별자(DBiResourceId)입니다.

Enhanced Monitoring 로그 데이터를 보려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 필요한 경우 DB 인스턴스가 있는 리전을 선택합니다. 자세한 내용은 Amazon Web Services 일반 참조의 [리전 및 앤드포인트](#)를 참조하십시오.
3. 탐색 창에서 로그를 선택합니다.
4. 로그 그룹 목록에서 RDSOSMetrics를 선택합니다.

다중 AZ 배포에서 이름에 -secondary가 추가된 로프 파일은 다중 AZ 스탠바이 복제복입니다.

The screenshot shows the AWS CloudWatch Log Groups interface. At the top, there are buttons for 'Search Log Group', 'Create Log Stream', and 'Delete Log Stream'. Below that is a 'Filter' input field labeled 'Log Stream Name Prefix'. The main area displays a list of log streams. One stream is highlighted with a red box: 'db-SORJBHOPBSMWGRI5EJW3KMYOTU-secondary'. To the right of the stream name, the 'Last Event Time' is listed as '2019-03-05 12:12 UTC-8'. Another stream, 'db-SORJBHOPBSMWGRI5EJW3KMYOTU', is listed below it with a 'Last Event Time' of '2019-03-05 12:07 UTC-8'.

5. 로그 스트림 목록에서 보려는 로그 스트림을 선택합니다.

사용 가능한 OS 측정치

다음 표에는 Amazon CloudWatch Logs에서 사용 가능한 OS 측정치가 나와 있습니다.

MariaDB, MySQL, Oracle 및 PostgreSQL DB 인스턴스의 지표

그룹	측정치	설명
General	engine	DB 인스턴스에 대한 데이터베이스 엔진
	instanceID	DB 인스턴스 식별자
	instanceResourceIdentifier	DB 인스턴스에 대해 리전별로 고유하고 변경 불가능한 식별자이며, 로그 스트림 식별자로도 사용됩니다.
	numVCpus	DB 인스턴스의 가상 CPU 수
	timestamp	측정치를 가져온 시간
	uptime	DB 인스턴스가 활성 상태로 유지된 시간
	version	OS 측정치 스트림 JSON 형식의 버전

그룹	측정치	설명
cpuUtilization	guest	게스트 프로그램에서 사용 중인 CPU의 비율
	idle	유류 상태인 CPU의 비율
	irq	소프트웨어 인터럽트에서 사용 중인 CPU의 비율
	nice	가장 낮은 우선순위로 실행 중인 프로그램에서 사용 중인 CPU의 비율
	steal	다른 가상 머신에서 사용 중인 CPU의 비율
	system	커널에서 사용 중인 CPU의 비율
	total	사용 중인 CPU의 총 비율입니다. 이 값에는 nice 값이 포함됩니다.
	user	사용자 프로그램에서 사용 중인 CPU의 비율
	wait	I/O 액세스를 대기 중인 동안 사용되지 않은 CPU의 비율
diskIO	avgQueueLen	I/O 디바이스의 대기열에서 대기 중인 요청 수입니다.
	avgReqSz	평균 요청 크기(KB)
	await	대기열 시간과 서비스 시간을 포함하여 요청에 응답하는 데 필요한 시간(밀리초)
	device	사용 중인 디스크 디바이스의 식별자
	readIOsPS	초당 읽기 작업 수
	readKb	읽은 총 KB 수
	readKbPS	초당 읽은 KB 수
	rrqmPS	초당 대기 중인 병합 읽기 요청 수
	tps	초당 I/O 트랜잭션 수
	util	요청이 발급된 CPU 시간의 비율
	writeIOsPS	초당 쓰기 작업 수
	writeKb	기록한 총 KB 수
	writeKbPS	초당 기록한 KB 수
	wrqmPS	초당 대기 중인 병합 쓰기 요청 수
fileSys	maxFiles	파일 시스템에 대해 생성될 수 있는 최대 파일 수
	mountPoint	파일 시스템의 경로
	name	파일 시스템의 이름
	total	파일 시스템에 사용 가능한 총 디스크 공간(KB)
	used	파일 시스템에서 파일에 사용된 디스크 공간의 양(KB)
	usedFilePercent	사용 중인 사용 가능한 파일의 비율

그룹	측정치	설명
	usedFiles	파일 시스템의 파일 수
	usedPercent	사용 중인 파일 시스템 디스크 공간의 비율
loadAverageMinutes	fifteen	마지막 15분 동안 CPU 시간을 요청한 프로세스 수
	five	마지막 5분 동안 CPU 시간을 요청한 프로세스 수
	one	마지막 1분 동안 CPU 시간을 요청한 프로세스 수
memory	active	할당된 메모리의 양(KB)
	buffers	스토리지 디바이스에 쓰기 이전에 I/O 요청을 버퍼링하는 데 사용되는 메모리의 양(KB)
	cached	파일 시스템 기반 I/O를 캐시하는 데 사용된 메모리의 양
	dirty	수정되었지만 스토리지의 관련 데이터 블록에 기록되지 않은 RAM의 메모리 페이지 양(KB)
	free	할당되지 않은 메모리의 양(KB)
	hugePagesFree	사용 가능한 방대한 페이지 수입니다. 방대한 페이지는 Linux 커널의 기능입니다.
	hugePagesRsvd	커밋된 방대한 페이지의 수
	hugePagesSize	각 방대한 페이지 단위의 크기(KB)
	hugePagesSurp	총계 대비 사용 가능한 초과 방대한 페이지 수
	hugePagesTotal	시스템의 총 방대한 페이지 수
	inactive	가장 적게 사용되는 메모리 페이지의 양(KB)
	mapped	프로세스 주소 공간 내에 메모리 매핑되는 총 파일 시스템 컨텐츠 양(KB)
	pageTables	페이지 표에 사용된 메모리의 양(KB)
	slab	재사용 가능한 커널 데이터 구조의 양(KB)
	total	총 메모리 양(KB)
	writeback	RAM에서 지원 스토리지에 아직 기록 중인 더티 페이지의 양(KB)
network	interface	DB 인스턴스에 대해 사용 중인 네트워크 인터페이스의 식별자
	rx	초당 수신된 바이트 수
	tx	초당 업로드된 바이트 수
processList	cpuUsedPc	프로세스에서 사용된 CPU 비율
	id	프로세스의 식별자
	memoryUsedPc	프로세스에 사용된 메모리의 양(KB)
	name	프로세스의 이름입니다.

그룹	측정치	설명
	parentID	프로세스의 상위 프로세스에 대한 프로세스 식별자
	rss	프로세스에 할당된 RAM의 양(KB)
	tgid	스레드 그룹 식별자이며, 스레드가 속한 프로세스 ID를 나타내는 숫자입니다. 이 식별자는 동일한 프로세스에서 스레드를 그룹화하는 데 사용됩니다.
	VIRT	프로세스에 할당된 가상 메모리의 양(KB)
swap	swap	사용 가능한 스왑 메모리 양(KB)
	swap in	디스크에서 스왑된 메모리 양(KB)
	swap out	디스크로 스왑된 총 메모리 양(KB)
	free	사용 가능한 스왑 메모리 양(KB)
	committed	캐시 메모리로 사용된 스왑 메모리의 양(KB)
tasks	blocked	차단되는 작업 수
	running	실행 중인 작업 수
	sleeping	절전 상태인 작업 수
	stopped	중단된 작업 수
	total	총 작업 수
	zombie	상위 작업은 활성화되었지만 비활성 상태인 하위 작업 수

Microsoft SQL Server DB 인스턴스에 대한 측정치

그룹	측정치	설명
General	engine	DB 인스턴스에 대한 데이터베이스 엔진
	instanceID	DB 인스턴스 식별자
	instanceResourceID	DB 인스턴스에 대해 리전별로 고유하고 변경 불가능한 식별자이며, 로그 스트림 식별자로도 사용됩니다.
	numVCpus	DB 인스턴스의 가상 CPU 수
	timestamp	측정치를 가져온 시간
	uptime	DB 인스턴스가 활성 상태로 유지된 시간
	version	OS 측정치 스트림 JSON 형식의 버전
cpuUtilization	idle	유휴 상태인 CPU의 비율
	kern	커널에서 사용 중인 CPU의 비율
	user	사용자 프로그램에서 사용 중인 CPU의 비율
disks	name	디스크에 대한 식별자

그룹	측정치	설명
	totalKb	디스크의 총 공간(KB)
	usedKb	디스크에서 사용된 공간의 양(KB)
	usedPc	디스크에서 사용된 공간의 비율(%)
	availKb	디스크에서 사용 가능한 공간(KB)
	availPc	디스크에서 사용 가능한 공간의 비율(%)
	rdCountPS	초당 읽기 작업 수
	rdBytesPS	초당 읽은 바이트 수
	wrCountPS	초당 쓰기 작업 수
	wBytesPS	초당 기록한 바이트 양
memory	commitTotKb	사용 중인 페이지 파일 지원 가상 주소 공간의 양 즉, 현재 약정 요금입니다. 이 값은 기본 메모리(RAM)와 디스크(페이지 파일)로 구성됩니다.
	commitLimitKb	commitTotKb 측정치에 대해 가능한 최대값입니다. 이 값은 현재 페이지 파일 크기와 페이지 가능 가능한 콘텐츠에 사용할 수 있는 물리적 메모리(페이지 불가능 영역에 할당된 RAM 제외)의 합계입니다.
	commitPeakKb	운영 체제를 마지막으로 시작한 이후의 commitTotKb 측정치 중 가장 큰 값
	kernTotKb	페이지된 풀과 페이지되지 않은 풀의 메모리 합계(KB)
	kernPagedKb	페이지된 커널 풀의 메모리 양(KB)
	kernNonpagedKb	페이지되지 않은 커널 풀의 메모리 양(KB)
	pageSize	페이지 크기(KB)
	physTotKb	물리적 메모리의 양(KB)
	physAvailKb	사용 가능한 물리적 메모리의 양(KB)
	sqlServerTotKb	Microsoft SQL Server에 커밋된 메모리의 양(KB)
	sysCacheKb	시스템 캐시 메모리의 양(KB)
network	interface	DB 인스턴스에 대해 사용 중인 네트워크 인터페이스의 식별자
	rdBytesPS	초당 수신된 바이트 수
	wrBytesPS	초당 보낸 바이트 수
processList	cpuUsedPc	프로세스에서 사용된 CPU 비율
	memUsedPc	프로세스에서 사용된 총 메모리 비율
	name	프로세스의 이름입니다.
	pid	프로세스의 식별자 Amazon RDS에서 소유한 프로세스의 경우 이 값이 표시되지 않습니다.

그룹	측정치	설명
	ppid	이 프로세스의 상위에 대한 프로세스 식별자입니다. 이 값은 하위 프로세스에 대해서만 표시됩니다.
	tid	스레드 식별자입니다. 이 값은 스레드에 대해서만 표시됩니다. 소유 프로세스는 pid 값을 사용하여 식별할 수 있습니다.
	workingSetKb	프라이빗 작업 세트의 메모리 양과 프로세스에서 사용 중이고 다른 프로세스와 공유 가능한 메모리의 양을 더한 값(KB)
	workingSetPrivKb	프로세스에서 사용 중이지만 다른 프로세스와 공유할 수 없는 메모리의 양(KB)
	workingSetSharedKb	프로세스에서 사용 중이고 다른 프로세스와 공유할 수 있는 메모리의 양(KB)
	virtKb	프로세스에서 사용 중인 가상 주소 공간의 양(KB)입니다. 가상 주소 공간을 사용하는 것이 반드시 디스크 또는 기본 메모리 페이지를 사용하는 것을 의미하지는 않습니다.
system	handles	시스템에서 사용 중인 핸들 수
	processes	시스템에서 실행 중인 프로세스 수
	threads	시스템에서 실행 중인 스레드 수

Amazon RDS 성능 개선 도우미 사용

Amazon RDS 성능 개선 도우미는 데이터베이스 성능을 분석하고 문제를 해결할 수 있도록 Amazon RDS DB 인스턴스 로드를 모니터링합니다. Amazon RDS 성능 개선 도우미는 현재 다음 DB 엔진에서 사용할 수 있습니다.

- MySQL과 호환되는 Amazon Aurora 버전 2.04.2 및 2.x 상위 버전(MySQL 5.7과 호환됨)
- MySQL과 호환되는 Amazon Aurora 버전 1.17.3 및 1.x 상위 버전(MySQL 5.6과 호환)
- PostgreSQL과 호환되는 Amazon Aurora
- Amazon RDS for MariaDB 버전 10.2.21 및 10.2 상위 버전
- Amazon RDS for MySQL 버전 5.7.22 및 5.7 상위 버전, 버전 5.6.41 및 5.6 상위 버전
- Amazon RDS for Microsoft SQL Server(SQL Server 2008을 제외한 모든 버전)
- Amazon RDS for PostgreSQL 버전 10
- Amazon RDS for Oracle(모든 버전)

Important

Amazon RDS 성능 개선 도우미를 Amazon Aurora와 함께 사용하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 [Amazon RDS 성능 개선 도우미 사용](#)을 참조하십시오.

Note

Amazon RDS 성능 개선 도우미는 MariaDB 버전 10.0, 10.1, 또는 10.3, MySQL 버전 5.5 또는 8.0에서는 지원되지 않습니다.

MariaDB 및 MySQL용 Amazon RDS의 경우 db.t2.micro, db.t2.small, db.t3.micro, db.t3.small DB 인스턴스 클래스에서는 성능 개선 도우미가 지원되지 않습니다.

성능 개선 도우미(Performance Insights)는 기존 Amazon RDS 모니터링 기능을 확장한 것으로서 데이터베이스 성능을 표시하여 성능 문제를 분석하는 데 효과적입니다. 성능 개선 도우미 대시보드가 데이터베이스 부하를 시각화하여 대기 시간, SQL 문, 호스트 또는 사용자를 기준으로 부하를 필터링합니다.

성능 개선 도우미는 Amazon RDS를 사용해 작업하는 모든 DB 엔진을 위한 콘솔 생성 마법사 내에서 기본적으로 활성화되어 있습니다. DB 인스턴스의 데이터베이스가 2개 이상인 경우에는 모든 데이터베이스에 대한 성능 데이터가 DB 인스턴스에 집계됩니다.

성능 개선 도우미의 중앙 지표는 DB 엔진에서 활성 세션의 평균 수를 의미하는 DB Load입니다. DB Load 지표는 1초마다 수집됩니다. 활성 세션이란 DB 엔진에게 작업을 제출하여 현재 응답 대기 중인 연결 세션을 말합니다. 예를 들어 DB 엔진에게 SQL 쿼리를 제출하면 DB 엔진이 이 쿼리를 처리하는 동안 해당하는 데이터베이스 세션이 활성화됩니다.

DB Load와 대기 이벤트 데이터를 결합하여 활성 세션의 전체 상태를 이해할 수 있습니다. 대기 이벤트는 DB 엔진마다 다릅니다.

- 모든 MariaDB 및 MySQL 대기 이벤트에 대한 자세한 내용은 MySQL 설명서의 [대기 이벤트 요약 테이블](#)을 참조하십시오.
- 모든 PostgreSQL 대기 이벤트에 대한 자세한 내용은 PostgreSQL 설명서의 [PostgreSQL 대기 이벤트](#)를 참조하십시오.
- 모든 Oracle 대기 이벤트에 대한 자세한 내용은 Oracle 설명서의 [대기 이벤트 설명](#)을 참조하십시오.
- 모든 SQL Server 대기 이벤트에 대한 자세한 내용은 SQL Server 설명서의 [대기 유형](#)을 참조하십시오.

Note

Oracle의 경우 연결된 SQL 없이 백그라운드 프로세스가 때때로 수행됩니다. 이 경우 성능 개선 도우미는 쿠론으로 연결된 백그라운드 프로세스의 유형과 이 백그라운드 프로세스에 연결된 대기 클래스를 보고합니다. 백그라운드 프로세스의 유형으로는 LGWR, ARC0, PMON 등이 있습니다.

예를 들어 아카이버가 I/O를 수행할 경우 성능 개선 도우미는 ARC1:System I/O와 같이 보고합니다. 경우에 따라 백그라운드 프로세스 유형이 누락되고 성능 개선 도우미가 대기 클래스만 보고할 때도 있습니다(예: :System I/O).

세션 정보는 수집 및 집계 후 대시보드에 평균 활성 세션 차트로 표시됩니다. 평균 활성 세션 차트에는 최대 CPU 값이 선으로 표시되기 때문에 활성 세션이 최대 값을 초과하는지 알 수 있습니다. 최대 CPU 값은 DB 인스턴스에서 vCPU(가상 CPU) 코어의 수로 결정됩니다.

평균 활성 세션 차트의 로드가 최대 CPU 선을 상회하는 경우가 잦아지고 CPU가 기본 대기 상태라면 시스템 CPU에서 과부하가 발생한 것입니다. 이러한 경우 연결 수를 인스턴스에 맞게 조절하거나, CPU 부하가 높은 SQL 쿼리를 모두 조정하거나, 인스턴스 클래스의 크기를 늘리는 것이 좋습니다. 대기 상태가 잦고 일관성 있게 나타난다면 병목 현상 또는 리소스 유지 문제가 있을 수 있으므로 이를 해결해야 합니다. 로드가 최대 CPU 선을 넘지 않는다 하더라도 이러한 문제가 나타날 수 있습니다.

다음 비디오에서 성능 개선 도우미의 개요를 볼 수 있습니다.

[성능 개선 도우미를 사용하여 Amazon Aurora PostgreSQL의 성능 분석](#)

주제

- [성능 개선 도우미 활성화 \(p. 251\)](#)
- [성능 개선 도우미의 액세스 제어 \(p. 254\)](#)
- [성능 개선 도우미 대시보드 사용 \(p. 256\)](#)
- [추가 사용자 인터페이스 기능 \(p. 264\)](#)
- [성능 개선 도우미 API \(p. 265\)](#)
- [Amazon CloudWatch에 게시되는 성능 개선 도우미 지표 \(p. 276\)](#)
- [성능 개선 도우미 카운터 \(p. 278\)](#)
- [AWS CloudTrail을 사용하여 성능 개선 도우미 작업 로깅 \(p. 286\)](#)

성능 개선 도우미 활성화

성능 개선 도우미를 사용하려면 DB 인스턴스에서 이 기능을 활성화해야 합니다.

콘솔

새 DB 인스턴스를 생성할 때 콘솔을 사용하여 성능 개선 도우미를 활성화할 수 있습니다. 성능 개선 도우미를 활성화하도록 DB 인스턴스를 수정할 수도 있습니다.

주제

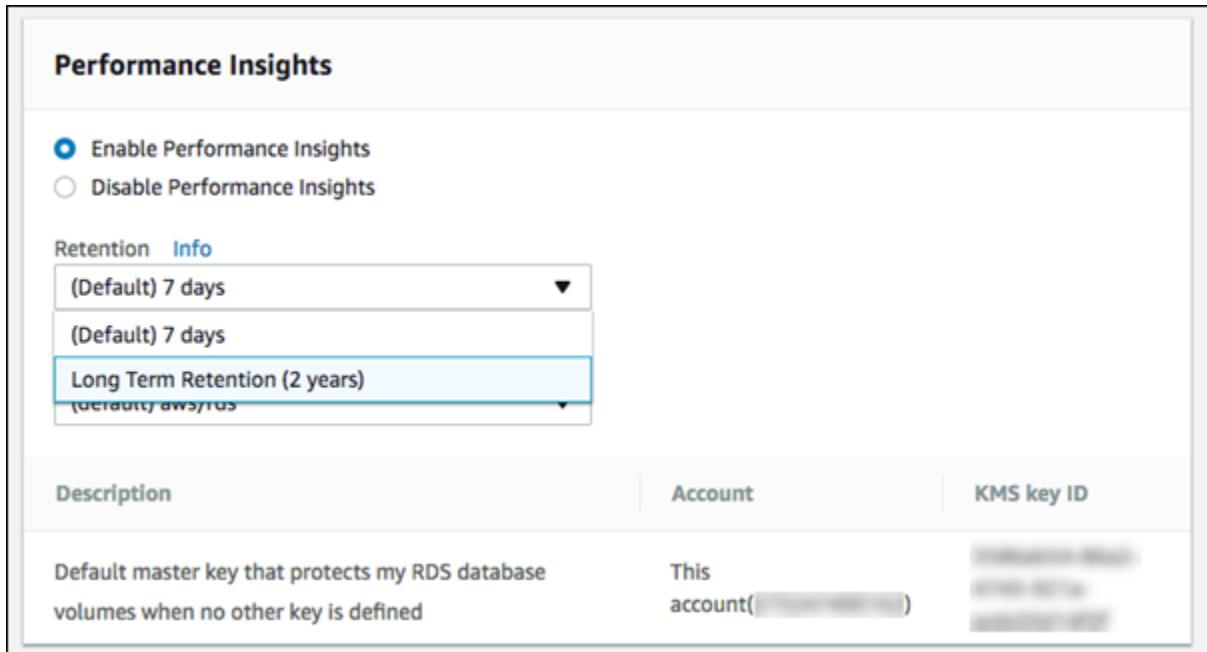
- [DB 인스턴스를 생성할 때 콘솔을 사용하여 성능 개선 도우미 활성화 \(p. 251\)](#)
- [DB 인스턴스를 수정 때 콘솔을 사용하여 성능 개선 도우미 활성화 \(p. 251\)](#)

DB 인스턴스를 생성할 때 콘솔을 사용하여 성능 개선 도우미 활성화

새 DB 인스턴스를 생성할 때 성능 개선 도우미(성능 개선 도우미) 섹션에서 Enable 성능 개선 도우미(성능 개선 도우미)를 선택하면 성능 개선 도우미가 활성화됩니다.

DB 인스턴스를 생성하려면 [Amazon RDS DB 인스턴스 생성 \(p. 111\)](#)의 DB 엔진에 대한 지침을 따르십시오.

다음 이미지는 성능 개선 도우미(성능 개선 도우미) 섹션을 보여줍니다.



Enable 성능 개선 도우미(성능 개선 도우미 활성화)를 선택할 때 다음 옵션이 있습니다.

- 보존 – 성능 개선 도우미 데이터를 보존할 시간입니다. 7일(기본값) 또는 2년을 선택합니다.
- 마스터 키 – AWS Key Management Service(AWS KMS) 키를 지정합니다. 성능 개선 도우미는 AWS KMS 키를 사용하여 잠재적으로 민감한 모든 데이터를 암호화합니다. 데이터는 암호화된 상태로 전송 및 저장됩니다. 자세한 내용은 [Amazon RDS 리소스 암호화 \(p. 385\)](#) 단원을 참조하십시오.

DB 인스턴스를 수정 때 콘솔을 사용하여 성능 개선 도우미 활성화

콘솔을 사용하여 성능 개선 도우미를 활성화하도록 DB 인스턴스를 수정할 수 있습니다.

콘솔을 사용하여 DB 인스턴스의 성능 개선 도우미를 활성화하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 데이터베이스를 선택합니다.
3. 수정하려는 DB 인스턴스를 선택한 후 수정을 선택합니다.
4. 성능 개선 도우미(성능 개선 도우미) 섹션에서 Enable 성능 개선 도우미(성능 개선 도우미 활성화).

Enable 성능 개선 도우미(성능 개선 도우미 활성화)를 선택할 때 다음 옵션이 있습니다.

- 보존 – 성능 개선 도우미 데이터를 보존할 시간입니다. 7일(기본값) 또는 2년을 선택합니다.
 - 마스터 키 – AWS Key Management Service(AWS KMS) 키를 지정합니다. 성능 개선 도우미는 AWS KMS 키를 사용하여 잠재적으로 민감한 모든 데이터를 암호화합니다. 데이터는 암호화된 상태로 전송 및 저장됩니다. 자세한 내용은 [Amazon RDS 리소스 암호화 \(p. 385\)](#) 단원을 참조하십시오.
5. [Continue]를 선택합니다.
 6. Scheduling of Modifications(수정 사항 예약)에서 다음 중 하나를 선택합니다.
 - 예약된 다음 유지 관리 기간에 적용 – 성능 개선 도우미 수정 사항을 다음 유지 관리 기간에 적용합니다.
 - 즉시 적용 – 성능 개선 도우미 수정 사항을 최대한 빨리 적용합니다.
 7. Modify Instance(DB 인스턴스)를 선택합니다.

AWS CLI

`create-db-instance` AWS CLI 명령을 사용하여 새 DB 인스턴스를 생성하면 `--enable-performance-insights`를 지정할 때 성능 개선 도우미가 활성화됩니다.

다음 AWS CLI 명령을 사용해 `--enable-performance-insights` 값을 지정할 수도 있습니다.

- `create-db-instance-read-replica`
- `modify-db-instance`
- `restore-db-instance-from-s3`

다음 절차에서는 AWS CLI를 사용하여 DB 인스턴스의 성능 개선 도우미를 활성화하는 방법을 설명합니다.

AWS CLI를 사용하여 DB 인스턴스의 성능 개선 도우미를 활성화하려면

- `modify-db-instance` AWS CLI 명령을 호출하고 다음 값을 입력합니다.
 - `--db-instance-identifier` – DB 인스턴스의 이름.
 - `--enable-performance-insights`

다음 예제에서는 `sample-db-instance`에 대한 성능 개선 도우미를 활성화합니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
--db-instance-identifier sample-db-instance \
--enable-performance-insights
```

Windows의 경우:

```
aws rds modify-db-instance ^
--db-instance-identifier sample-db-instance ^
--enable-performance-insights
```

성능 개선 도우미를 활성화할 때 선택적으로 `--performance-insights-retention-period` 옵션을 사용하여 성능 개선 도우미 데이터를 보존할 시간을 일 단위로 지정할 수 있습니다. 유효한 값은 7(기본값) 또는 731(2년)입니다.

다음 예제에서는 `sample-db-instance`의 성능 개선 도우미를 활성화하고 성능 개선 도우미 데이터가 2년 동안 보존되도록 지정합니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
--db-instance-identifier sample-db-instance \
--enable-performance-insights \
--performance-insights-retention-period 731
```

Windows의 경우:

```
aws rds modify-db-instance ^
--db-instance-identifier sample-db-instance ^
--enable-performance-insights ^
--performance-insights-retention-period 731
```

API

[CreateDBInstance](#) 작업 Amazon RDS API 작업을 사용하여 새 DB 인스턴스를 생성할 때 `EnablePerformanceInsights`를 `True`로 설정하면 성능 개선 도우미가 활성화됩니다.

다음 API 작업으로 `EnablePerformanceInsights` 값을 지정할 수도 있습니다.

- [ModifyDBInstance](#)
- [CreateDBInstanceReadReplica](#)
- [RestoreDBInstanceFromS3](#)

성능 개선 도우미를 활성화할 때 선택적으로 `PerformanceInsightsRetentionPeriod` 파라미터를 사용하여 성능 개선 도우미 데이터를 보존할 시간을 일 단위로 지정할 수 있습니다. 유효한 값은 7(기본값) 또는 731(2년)입니다.

MariaDB 또는 MySQL용 Amazon RDS에서 성능 개선 도우미에 대해 성능 스키마 활성화

MariaDB 또는 MySQL용 Amazon RDS MySQL의 경우, 성능 개선 도우미는 성능 스키마 기능이 활성화되면 자세한 정보를 제공합니다. 성능 개선 도우미를 활성화한 상태에서 Amazon RDS MySQL DB 인스턴스를 생성하면 성능 스키마가 자동으로 활성화됩니다. 성능 개선 도우미를 활성화한 상태에서 DB 인스턴스를 생성하면 성능 스키마 파라미터의 다음 하위 집합이 지정된 값으로 자동으로 설정됩니다.

- `performance_schema=1`

- performance-schema-consumer-events-waits-current=ON
- performance-schema-instrument='wait/=%=ON'
- performance-schema-consumer-global-instrumentation=ON
- performance-schema-consumer-thread-instrumentation=ON

파라미터 그룹에 `performance_schema` 파라미터에 대해 명시적으로 설정된 값이 없는 경우에만 성능 스키마가 자동으로 활성화됩니다. `performance_schema` 파라미터를 검사한 후 원본 값이 `user`인 경우 값을 설정할 수 있습니다. 성능 스키마 파라미터를 자동으로 설정하려면 `performance_schema` 파라미터 값을 설정 해제합니다. AWS Management 콘솔에서 파라미터를 확인하거나 AWS CLI `describe-db-parameters` 명령을 실행하여 파라미터 값의 원본을 확인할 수 있습니다.

`performance_schema` 파라미터 값을 변경할 때 DB 인스턴스를 재부팅해야 합니다. 성능 개선 도우미를 활성화한 상태에서 새 DB 인스턴스를 생성할 경우 `performance_schema` 파라미터는 기본적으로 1(활성화)로 설정됩니다.

성능 스키마를 활성화하지 않은 상태에서는 성능 개선 도우미에 데이터베이스 부하가 MySQL 프로세스의 독특 상태별로 구분되어 표시됩니다. 성능 스키마를 활성화한 상태에서는 성능 개선 도우미에 데이터베이스 부하가 세부 대기 이벤트별로 구분되어 표시됩니다.

자세한 내용은 [성능 개선 도우미 대시보드 사용 \(p. 256\)](#) 단원을 참조하십시오.

성능 개선 도우미의 액세스 제어

성능 개선 도우미에 액세스하려면 AWS Identity and Access Management(IAM)의 적절한 권한이 있어야 합니다. 액세스 권한은 다음 두 가지 옵션을 사용해 부여할 수 있습니다.

1. `AmazonRDSFullAccess` 관리 정책을 IAM 사용자 또는 역할에 연결합니다.
2. 사용자 지정 IAM 정책을 생성해 IAM 사용자 또는 역할에 연결합니다.

AmazonRDSFullAccess 관리 정책

`AmazonRDSFullAccess`는 모든 Amazon RDS API 작업에 대한 액세스 권한을 부여하는 AWS 관리 정책입니다. 이 정책은 Amazon RDS 콘솔에서 사용하는 관련 서비스(예: Amazon SNS를 사용한 이벤트 알림)에 대한 액세스 권한도 부여합니다.

또한 `AmazonRDSFullAccess`에는 성능 개선 도우미 사용에 필요한 모든 권한이 들어 있습니다. 이 정책을 IAM 사용자 또는 역할에 연결하면 수신자가 기타 콘솔 기능과 함께 성능 개선 도우미를 사용할 수 있습니다.

사용자 지정 IAM 정책 사용

`AmazonRDSFullAccess` 정책이 포함된 완전한 액세스 권한이 없는 사용자의 경우, 사용자 관리형 IAM 정책을 생성 또는 수정하여 성능 개선 도우미에 대한 액세스 권한을 부여할 수 있습니다. IAM 사용자 또는 역할에 이 정책을 연결하면 수신자가 성능 개선 도우미를 사용할 수 있습니다.

사용자 지정 정책을 생성하는 방법

1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. [Create policy]를 선택합니다.
4. [Create Policy] 페이지에서 JSON 탭을 선택합니다.
5. 다음을 복사하여 붙여 넣습니다.

{

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": "pi:*",
        "Resource": "arn:aws:pi:/*:metrics/rds/*"
    }
]
```

- [Review policy]를 선택합니다.

Note

현재 이 정책을 입력하면 [Visual editor] 탭에 pi 리소스가 인식되지 않는다는 경고가 표시됩니다. 이 경고는 무시해도 됩니다.

- 정책의 이름과 설명(선택 사항)을 지정한 다음 [Create policy]를 선택합니다.

이제 IAM 사용자 또는 역할에 해당 정책을 연결할 수 있습니다. 다음 절차에서는 이러한 목적에 사용할 수 있는 IAM 사용자가 이미 있다고 가정합니다.

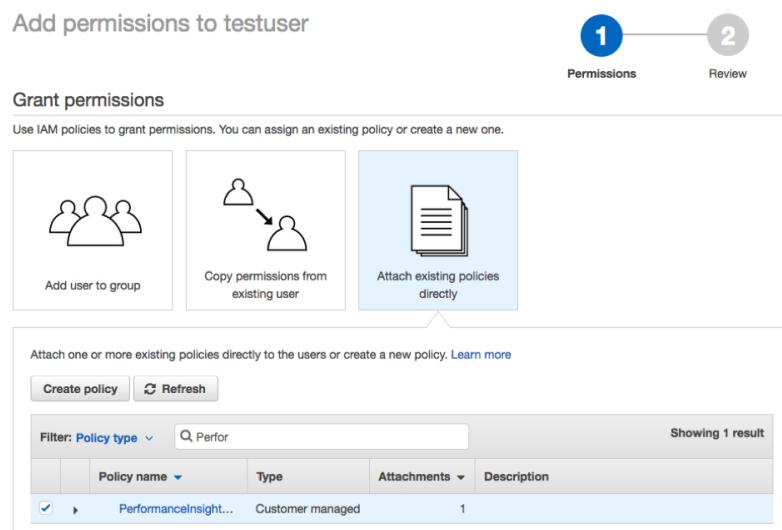
IAM 사용자에게 정책을 연결하려면

- <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
- 탐색 창에서 사용자를 선택합니다.
- 목록에서 기존 사용자를 선택합니다.

Important

사용자가 성능 개선 도우미를 사용하려면 사용자 지정 정책 외에 Amazon RDS에 대한 액세스 권한이 있어야 합니다. 예를 들어 AmazonRDSReadOnlyAccess 사전 정의 정책은 Amazon RDS에 대한 읽기 전용 액세스를 제공합니다. 자세한 내용은 [Amazon RDS에 대한 AWS 관리형\(미리 정의된\) 정책 \(p. 346\)](#) 단원을 참조하십시오.

- [Summary] 페이지에서 [Add permissions]를 선택합니다.
- [Attach existing policies directly]를 선택합니다. [Search]에 다음과 같이 정책 이름의 첫 문자 몇 개를 입력합니다.



- 정책을 선택하고 [Next: Review]를 선택합니다.
- [Add permissions]를 선택합니다.

성능 개선 도우미 대시보드 사용

성능 개선 도우미 대시보드에는 성능 문제를 분석하여 해결할 수 있는 데이터베이스 성능 정보가 포함됩니다. 메인 대시보드 페이지에서 데이터베이스 부하에 대한 정보를 확인할 수 있습니다. 또한 특정 대기 상태, SQL 쿼리, 호스트 또는 사용자에 대한 세부 정보를 알아볼 수 있습니다.

주제

- [성능 개선 도우미 대시보드 열기 \(p. 256\)](#)
- [성능 개선 도우미 대시보드 구성 요소 \(p. 257\)](#)
- [성능 개선 도우미 대시보드를 사용한 데이터베이스 부하 분석 \(p. 261\)](#)
- [성능 개선 도우미 대시보드에서 더 많은 SQL 텍스트 보기 \(p. 262\)](#)

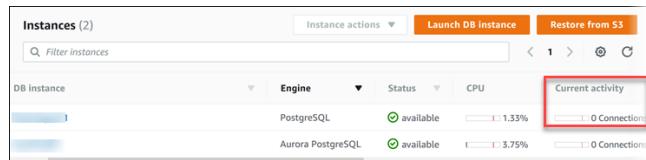
성능 개선 도우미 대시보드 열기

성능 개선 도우미 대시보드는 다음 절차에 따라 확인할 수 있습니다.

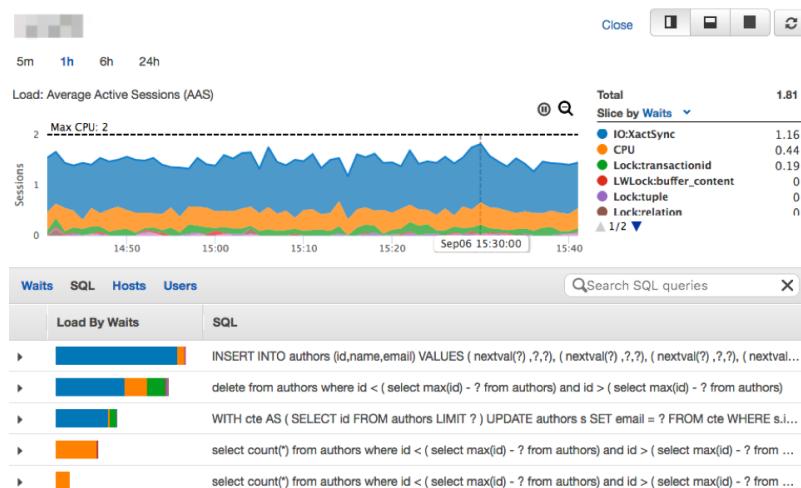
AWS Management Console에서 성능 개선 도우미 대시보드를 보려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [성능 개선 도우미]를 선택합니다.
3. DB 인스턴스를 선택합니다. 선택한 DB 인스턴스에 대한 성능 개선 도우미 대시보드가 표시됩니다.

성능 개선 도우미가 활성화되면 DB 인스턴스의 경우 DB 인스턴스 목록에서 세션 항목을 선택하여 대시보드에 접속할 수도 있습니다. 현재 활동에서 세션 항목은 지난 5분 동안 평균 활동 세션의 데이터베이스 로드를 보여 줍니다. 로드가 막대 모양으로 표시됩니다. 막대가 비어 있으면 DB 인스턴스가 유지 상태입니다. 로드가 증가하면 막대가 파란색으로 채워집니다. 로드에서 DB 인스턴스 클래스의 가상 CPU(vCPU) 수를 전달하면 막대가 빨간색으로 바뀌고 병목 가능성을 나타냅니다.



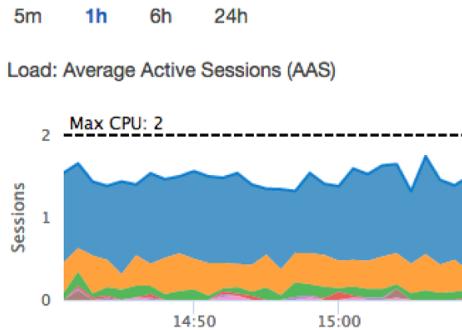
다음 이미지는 DB 인스턴스의 대시보드를 보여 줍니다.



성능 개선 도우미 대시보드는 기본적으로 마지막 60분 동안 수집된 데이터를 표시합니다. 하지만 마지막 5분, 60분, 5시간, 24시간 또는 1주 동안 데이터를 표시하도록 수정할 수 있습니다. 사용 가능한 모든 데이터를 볼 수도 있습니다.

Performance Insight 대시보드는 새 데이터로 자동으로 고쳐집니다. 새로 고침 속도는 표시되는 데이터의 양에 따라 다릅니다.

- 5분은 5초마다 새로 고칩니다.
- 1시간 및 5시간은 1분마다 새로 고칩니다.
- 24시간은 5분마다 새로 고칩니다.
- 1주는 1시간마다 새로 고칩니다.



성능 개선 도우미 대시보드 구성 요소

대시보드는 세 부분으로 나뉩니다.

1. 카운터 지표 차트 – 특정 성능 카운터 지표의 데이터를 보여 줍니다.
2. 평균 활성 세션 차트 - 데이터베이스 부하와 DB 인스턴스 용량을 비교하여 최대 CPU 선으로 표시합니다.
3. 상위 부하 항목 테이블 - 데이터베이스 부하에 가장 많이 영향을 미치는 항목을 보여 줍니다.

카운터 지표 차트

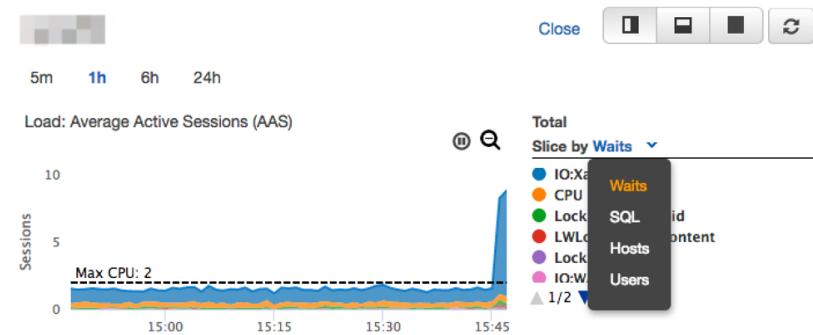
카운터 지표 차트에는 성능 카운터의 데이터가 표시됩니다. 표시되는 기본 지표는 `blk.read.avg` 및 `xact_commit.avg`입니다. 차트 오른쪽 위 모서리에 있는 기어 모양 아이콘을 선택하여 표시할 성능 카운터를 선택할 수 있습니다.



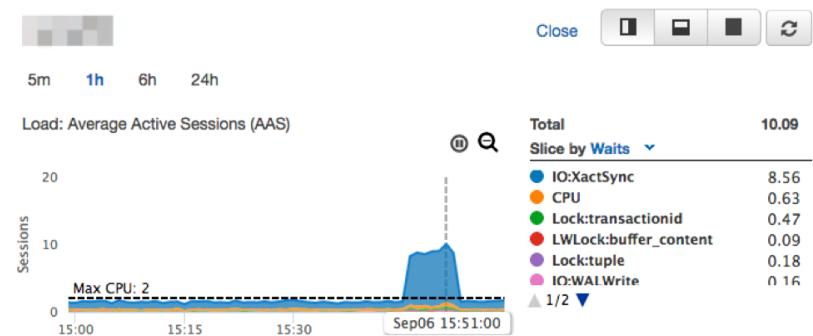
자세한 내용은 [성능 개선 도우미 카운터 \(p. 278\)](#) 단원을 참조하십시오.

평균 활성 세션 차트

평균 활성 세션 차트는 데이터베이스 부하와 DB 인스턴스 용량의 비교 방식을 최대 CPU 선으로 표시합니다. 기본적으로 부하는 대기 상태를 기준으로 구분된 활성 세션으로 표시됩니다. 또한 SQL 쿼리, 호스트 또는 사용자로 구분된 활성 세션으로 부하를 표시하도록 선택할 수도 있습니다.

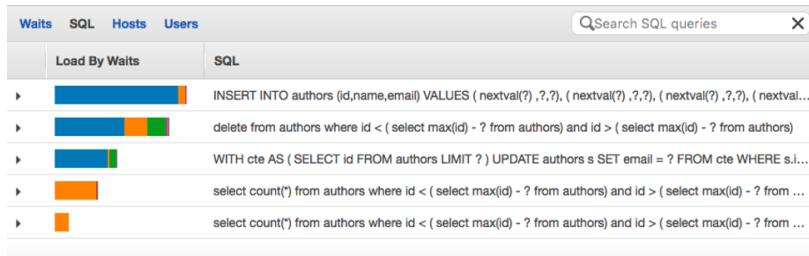


선택한 기간의 항목 세부 정보를 범례에 표시하려면 평균 활성 세션 차트의 임의 항목 위로 마우스 포인터를 가져가면 됩니다.



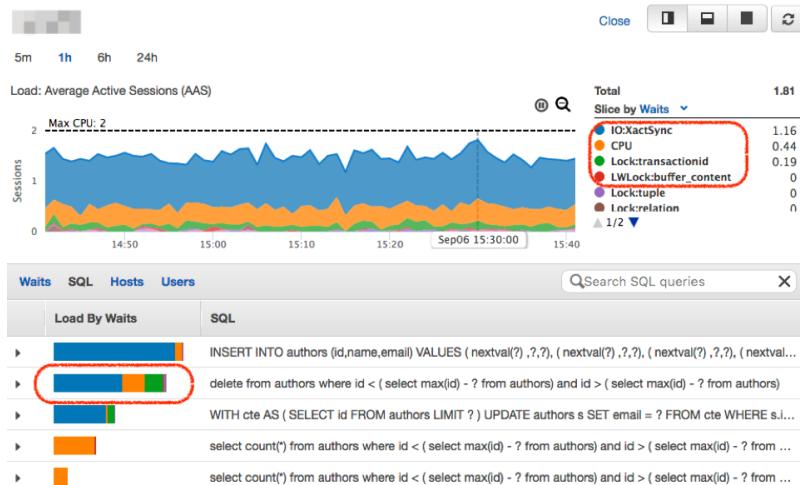
상위 부하 항목 테이블

[Top Load Items] 테이블에는 데이터베이스 부하에 영향을 끼치는 상위 항목이 표시됩니다. 기본적으로 데이터베이스 부하에 영향을 미치는 상위 SQL 쿼리가 표시됩니다. 쿼리는 구조적으로 유사한 실제 쿼리가 다수 요약되어 표시되지만 다른 파라미터가 존재할 가능성도 있습니다. 그 밖에 최상위 대기 상태, 호스트 또는 사용자를 표시하도록 선택할 수도 있습니다.



각 상위 로드 항목과 연결된 데이터베이스 로드 비율(%)은 DB Load by Waits(대기별 DB 로드) 열에 표시됩니다. 이 열에는 현재 평균 활성 세션 차트에서 어떤 구분 기준을 선택하든 그 기준에 따라 해당 항목의 로드가 반영됩니다. 예를 들어, 평균 활성 세션 차트가 호스트별로 구분되어 있고 상위 부하 항목 테이블에서 SQL 쿼리를 살펴보고 있다고 가정합니다. 이 경우 DB Load by Waits(대기별 DB 로드) 막대는 관련 호스트에서 쿼리가 나타내는 로드를 반영합니다. 평균 활성 세션 차트의 해당 호스트에 컬러 코드로 매핑합니다.

또 다른 예로, 평균 활성 세션 차트가 대기 상태별로 구분되어 있고 상위 부하 항목 테이블에서 SQL 쿼리를 살펴보고 있다고 가정합니다. 이 경우 DB Load by Waits(대기별 DB 로드) 막대는 쿼리가 영향을 미치는 대기 상태의 정도를 크기, 세그먼트 및 컬러 코드로 표시합니다. 해당 쿼리에 영향을 미치는 대기 상태도 표시합니다.



Top Load Items(상위 부하 항목) 테이블에서 SQL 문과 관련된 다음과 같은 식별자(ID) 유형을 볼 수 있습니다.

- SQL ID – 데이터베이스가 SQL 문을 고유하게 식별하기 위해 사용하는 ID입니다.

Oracle 및 SQL Server DB 인스턴스의 경우, SQL ID를 사용하여 특정 SQL 문을 찾을 수 있습니다.

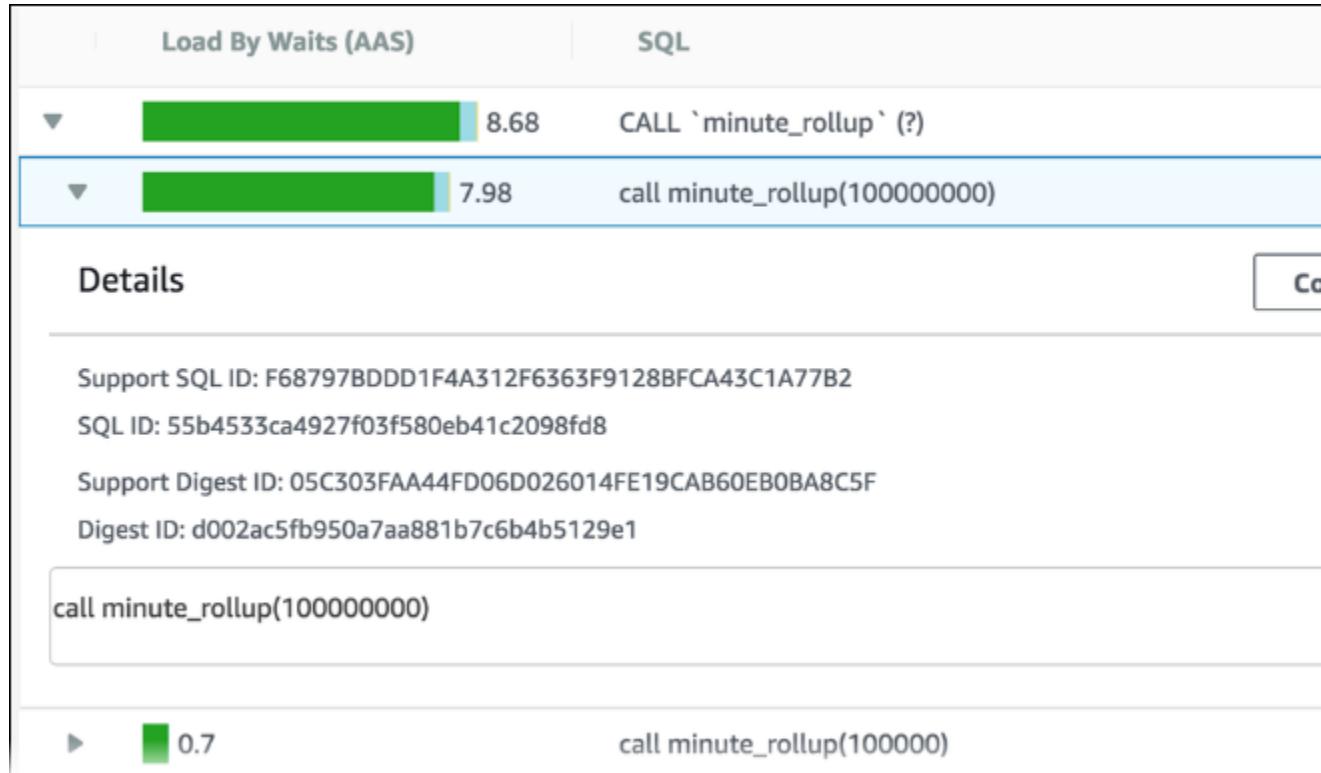
- SQL ID 지원 – SQL ID의 해시 값입니다. 이 값은 AWS Support를 사용할 때 SQL ID를 참조하는 용도로만 사용됩니다. AWS Support는 실제 SQL ID 및 SQL 텍스트에 액세스할 수 없습니다.
- 디아제스트 ID – 데이터베이스가 SQL 디아제스트를 고유하게 식별하기 위해 사용하는 ID입니다. SQL 디아제스트에는 리터럴이 제거되고 공백이 표준화된 SQL 문이 하나 이상 포함될 수 있습니다. 리터럴은 물음표(?)로 대체됩니다.

MariaDB, MySQL 및 PostgreSQL을 Amazon RDS DB 인스턴스의 경우, 다이제스트 ID를 사용하여 특정 SQL 다이제스트를 찾을 수 있습니다.

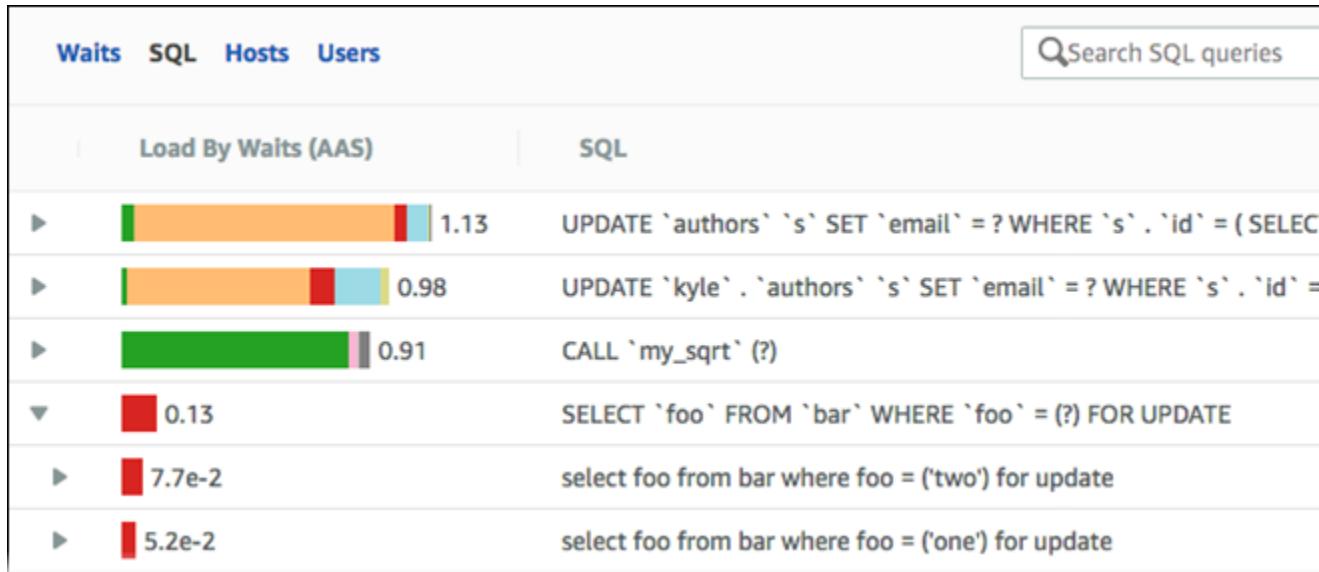
Oracle 및 SQL Server DB 인스턴스의 경우 다이제스트 ID는 SQL ID와 동일합니다. Top Load Items(상위 부하 항목) 표의 맨 위 행은 실제 SQL 문으로서 리터럴이 포함되어 있습니다.

- Support Digest ID(다이제스트 ID 지원) – 다이제스트 ID의 해시 값입니다. 이 값은 AWS Support를 사용할 때 다이제스트 ID를 참조하는 용도로만 사용됩니다. AWS Support는 실제 다이제스트 ID 및 SQL 텍스트에 액세스할 수 있는 권한이 없습니다.

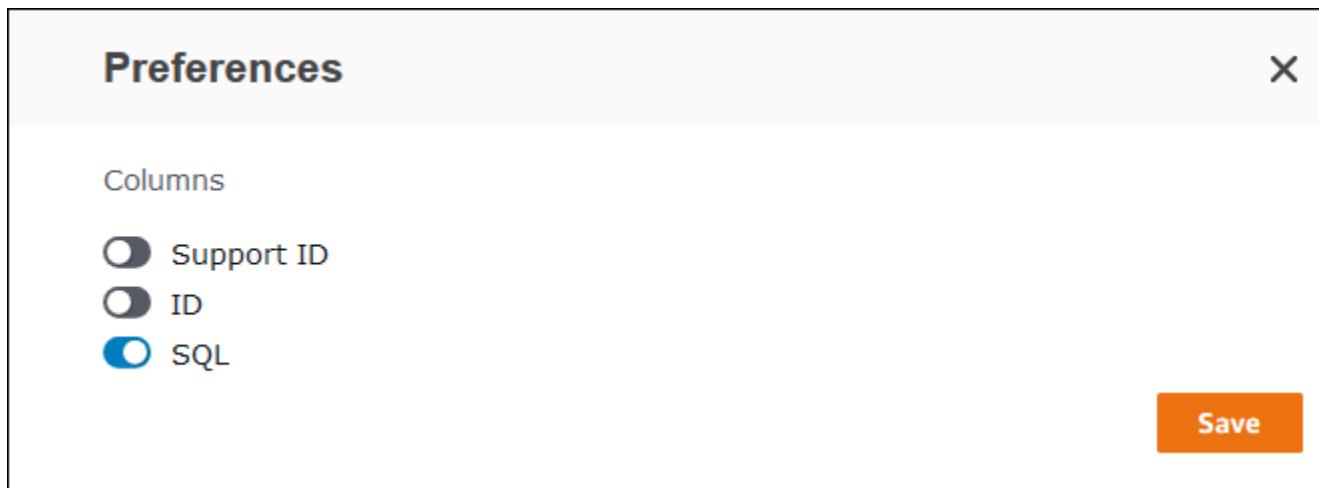
Top Load Items(상위 부하 항목) 테이블에서 상위 문을 열어 ID를 볼 수 있습니다. 다음 이미지는 열린 상위 문을 보여 줍니다.



기본 설정 아이콘을 선택하여 Top Load Items(상위 부하 항목) 테이블에 표시된 ID를 제어할 수 있습니다.



기본 설정 아이콘을 선택하면 기본 설정 창이 열립니다.



Top Load Items(상위 부하 항목) 테이블에 표시하려는 ID를 활성화한 후, 저장을 선택합니다.

성능 개선 도우미 대시보드를 사용한 데이터베이스 부하 분석

평균 활성 세션 차트가 병목 현상을 보일 때는 부하가 발생하는 위치를 찾아낼 수 있습니다. 이렇게 하려면 평균 활성 세션 차트 아래에 있는 상위 부하 항목 테이블을 살펴봅니다. SQL 쿼리나 사용자 같은 특정 항목을 선택하여 드릴다운을 통해 세부 정보까지 확인할 수 있습니다.

대기 상태와 상위 SQL 쿼리를 기준으로 구분된 DB 부하가 기본 Performance Insights 대시보드 보기입니다. 이 보기는 일반적으로 성능 문제를 가장 정확하게 파악할 수 있는 조합입니다. 대기 상태를 기준으로 구분된 DB 부하는 데이터베이스의 리소스 또는 동시성 병목 현상 유무를 표시합니다. 이 경우 상위 항목 테이블의 [SQL] 부하를 야기하는 쿼리를 표시합니다.

성능 문제를 진단하는 일반 워크플로우는 다음과 같습니다.

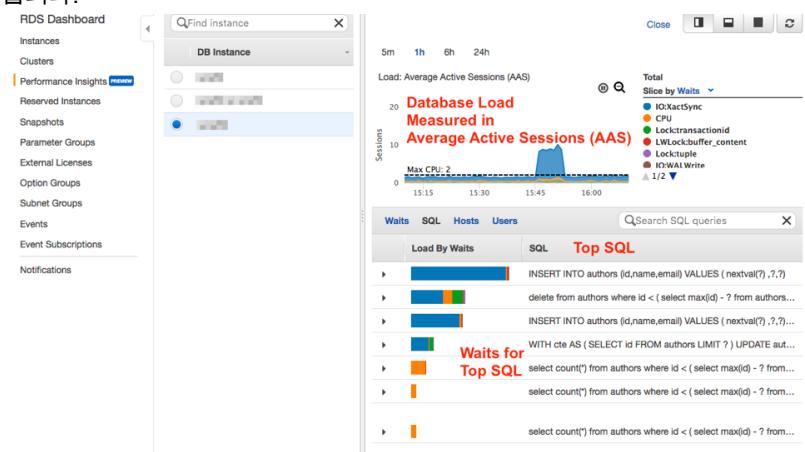
1. 평균 활성 세션 차트를 보면서 데이터베이스 부하가 최대 CPU 선을 상회하는지 모니터링합니다.
2. 상회하는 경우가 있으면 평균 활성 세션 차트를 보면서 원인이 되는 대기 상태를 식별합니다.

3. 상위 부하 항목 테이블의 [SQL] 탭에서 어떤 쿼리가 대기 상태에 가장 큰 영향을 미치는지 모니터링하면서 부하를 야기하는 요약 쿼리를 식별합니다. DB Load by Wait(대기별 DB 로드) 열을 보면 이러한 요약 쿼리를 식별할 수 있습니다.
4. [SQL] 탭에서 요약 쿼리 중 하나를 선택하여 확장한 다음 구성하고 있는 하위 쿼리를 확인합니다.

예를 들어 다음 대시보드에서 IO:XactSync 대기 시간은 빈번한 문제입니다. [CPU] 대기 시간은 비교적 빈번하지는 않지만 부하에 여전히 중요한 역할을 합니다.

상위 부하 항목 테이블의 [SQL] 탭에서 처음 4개의 룰업 쿼리는 첫 번째 상태와 매우 밀접한 관계가 있습니다. 따라서 이러한 쿼리의 하위 쿼리까지 세부 정보를 확인해야 합니다. 이러한 쿼리가 성능 문제에 어떻게 영향을 끼치고 있는지 알아보려면 이렇게 확인해야 합니다.

마지막 3개의 룰업 쿼리는 CPU 부하에 중요한 역할을 하며, CPU 부하가 문제인지 여부를 조사하는 쿼리가 됩니다.



성능 개선 도우미 대시보드에서 더 많은 SQL 텍스트 보기

기본적으로 Top Load Items(상위 부하 항목) 표의 각 행에는 각 SQL 문에 대해 500바이트의 SQL 텍스트가 표시됩니다. SQL 문이 500바이트 이상인 경우 성능 개선 도우미 대시보드에서 해당 문을 열어 더 많은 SQL 문을 볼 수 있습니다. 성능 개선 도우미 대시보드는 SQL 문 하나에 최대 4,096바이트를 표시할 수 있습니다. 또한 표시된 SQL 문을 복사하거나 다운로드할 수 있습니다.

SQL 텍스트의 한도는 DB 엔진에 달려 있습니다. 다음과 같은 제한이 적용됩니다.

- MySQL 및 MariaDB용 Amazon RDS – 1,024바이트
- Amazon RDS PostgreSQL – `track_activity_query_size` DB 인스턴스 파라미터로 설정

Amazon RDS PostgreSQL DB 인스턴스의 경우, `track_activity_query_size` DB 인스턴스 파라미터를 최대 102,400바이트까지 설정하여 SQL 텍스트의 한도를 제어할 수 있습니다. AWS Management 콘솔을 사용해 이 파라미터로 설정한 한도까지 SQL 텍스트를 다운로드할 수 있습니다. 자세한 내용은 [Amazon RDS PostgreSQL DB 인스턴스에 대한 SQL 텍스트 한도 설정 \(p. 264\)](#) 단원을 참조하십시오.

Important

현재는 AWS Management 콘솔에서 더 많은 SQL 텍스트를 보고 다운로드할 수만 있습니다. AWS 성능 개선 도우미 CLI 및 API는 최대 500바이트의 텍스트를 반환할 수 있습니다.

Note

현재는 Amazon RDS Oracle 또는 SQL Server DB 인스턴스에 대해 500바이트 이상의 SQL 텍스트를 보거나 복사하거나 다운로드할 수 없습니다.

MySQL용 Amazon RDS DB 인스턴스의 경우, EU(스톡홀름) 리전에서는 더 많은 SQL 텍스트를 보는 기능이 지원되지 않습니다.

성능 개선 도우미 대시보드에서 더 많은 SQL 텍스트를 보려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [성능 개선 도우미]를 선택합니다.
3. DB 인스턴스를 선택합니다. 선택한 DB 인스턴스에 대한 성능 개선 도우미 대시보드가 표시됩니다.

500바이트 이상의 텍스트가 있는 SQL 문은 다음 이미지와 유사합니다.

The screenshot shows the AWS RDS Performance Advisor interface. The top navigation bar has tabs for 'Waits', 'SQL', 'Hosts', and 'Users'. The 'SQL' tab is selected. Below the tabs is a search bar with the placeholder 'Search SQL query...'. The main area is titled 'Load By Waits (AAS)' and contains a table with several rows. Each row has a small orange bar chart icon, a wait time value (e.g., 1.6e-2, 3.3e-3), and an SQL query. The queries are identical: 'INSERT INTO large_query_test (i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_10, i_11, i_12, i_13, i_14, i_15, i_16, i_17, i_18, i_19, i_20, i_21, i_22, i_23, i_24, i_25, i_26, i_27, i_28, i_29, i_30, i_31, i_32, i_33, i_34, i_35, i_36, i_37, i_38, i_39, i_40, i_41, i_42, i_43, i_44, i_45, i_46, i_47, i_48) VALUES (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48)'. The table has a header row and approximately 10 data rows.

4. 더 많은 SQL 텍스트를 볼 SQL 문을 엽니다.

The screenshot shows the same AWS RDS Performance Advisor interface as the previous one, but with a specific SQL snippet highlighted. The highlighted row has an orange bar chart icon and the value '3.3e-3'. The SQL query is: 'INSERT INTO large_query_test (i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_10, i_11, i_12, i_13, i_14, i_15, i_16, i_17, i_18, i_19, i_20, i_21, i_22, i_23, i_24, i_25, i_26, i_27, i_28, i_29, i_30, i_31, i_32, i_33, i_34, i_35, i_36, i_37, i_38, i_39, i_40, i_41, i_42, i_43, i_44, i_45, i_46, i_47, i_48) VALUES (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48)'. Below the table, there is a 'Details' section containing support information and a 'Copy snippet' button.

성능 개선 도우미 대시보드는 각 SQL 문에 최대 4,096바이트를 표시할 수 있습니다.

5. (선택 사항) Copy snippet(코드 조각 복사)을 선택하여 표시된 SQL 문을 복사하거나 Download full SQL(전체 SQL 다운로드)를 선택하여 SQL 문을 다운로드합니다.

Note

SQL 문을 복사하거나 다운로드하려면 팝업 차단 기능을 비활성화하십시오.

Amazon RDS PostgreSQL DB 인스턴스에 대한 SQL 텍스트 한도 설정

Amazon RDS PostgreSQL DB 인스턴스의 경우, 성능 개선 도우미 대시보드에 표시될 수 있는 SQL 텍스트의 한도를 제어할 수 있습니다.

이를 위해서는 `track_activity_query_size` DB 인스턴스 파라미터를 수정해야 합니다. `track_activity_query_size` 파라미터에 대한 기본 설정은 1,024바이트입니다.

바이트 수를 늘려 성능 개선 도우미 대시보드에서 볼 수 있는 SQL 텍스트 크기를 늘릴 수 있습니다. 파라미터의 한도는 10,240바이트입니다. `track_activity_query_size` DB 파라미터에 대한 자세한 내용은 PostgreSQL 설명서에서 [런타임 통계](#)를 참조하십시오.

파라미터를 수정하려면 Amazon RDS PostgreSQL DB 인스턴스와 연결된 파라미터 그룹에서 파라미터 설정을 변경하십시오.

Amazon RDS PostgreSQL DB 인스턴스에서 기본 파라미터 그룹을 사용 중인 경우, 다음 절차를 완료하십시오.

- 적절한 DB 엔진 및 DB 엔진 버전에 대해 새로운 DB 인스턴스 파라미터 그룹을 생성합니다.
- 새 파라미터 그룹에 파라미터를 설정합니다.
- 새 파라미터 그룹을 DB 인스턴스에 연결합니다.

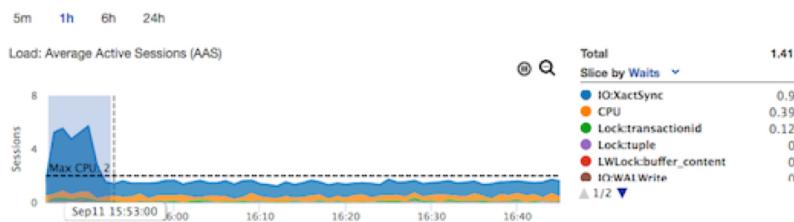
DB 인스턴스 파라미터 설정에 대한 자세한 내용은 [DB 파라미터 그룹의 파라미터 수정 \(p. 164\)](#) 단원을 참조하십시오.

추가 사용자 인터페이스 기능

성능 개선 도우미 사용자 인터페이스의 다른 기능을 사용해 성능 데이터를 분석할 수 있습니다.

클릭하여 끌어 확대

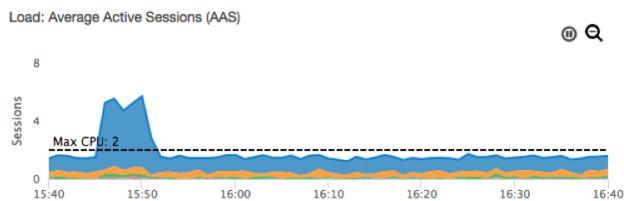
성능 개선 도우미 인터페이스에서 부하 차트의 작은 부분을 선택하여 확대해 자세히 볼 수 있습니다.



부하 차트의 한 부분을 확대하려면 시작 시간을 선택하고 원하는 기간 끝까지 끕니다. 이렇게 하면 선택한 영역이 강조 표시됩니다. 마우스를 놓으면 부하 차트의 선택한 영역이 확대되고 [Top N] 테이블이 다시 계산됩니다.

일시 중지 및 축소

부하 차트의 오른쪽 맨 위에는 [Pause] 및 [Zoom out] 도구가 있습니다.



[Pause]를 선택하면 부하 차트가 자동 새로 고침을 중지합니다. [Pause]를 다시 선택하면 부하 차트가 자동 새로 고침을 다시 시작합니다.

[Zoom out]을 선택하면 부하 차트가 다음으로 가장 큰 시간 간격으로 축소됩니다.

성능 개선 도우미 API

지원되는 엔진 유형에 대해 성능 개선 도우미가 활성화된 경우 Amazon RDS 성능 개선 도우미 API는 RDS 인스턴스의 성능에 대한 가시성을 제공합니다. Amazon CloudWatch Logs는 AWS 서비스에 대해 판매된 모니터링 지표를 위한 신뢰할 수 있는 소스를 제공합니다. 성능 개선 도우미는 평균 활성 세션 수로 측정되어 API 소비자에게 2D 시계열 데이터 세트로 제공되는 도메인별 데이터베이스 부하 보기를 제공합니다. 데이터의 시간 차원은 쿼리된 시간 범위 내 각 시점에 대한 데이터베이스 부하 데이터를 제공합니다. 각 시점에서 요청된 차원에 관해 해당 시점에서 측정되는 전체 부하를 분해합니다(예: SQL, Wait-event, User 또는 Host).

Amazon RDS 성능 개선 도우미는 데이터베이스 성능을 분석하고 관련 문제를 해결할 수 있도록 Amazon RDS DB 인스턴스를 모니터링합니다. 성능 개선 도우미 데이터를 볼 수 있는 한 가지 방법은 AWS Management 콘솔에서 보는 것입니다. 또한 성능 개선 도우미는 사용자가 자신의 데이터를 쿼리할 수 있도록 퍼블릭 API도 제공합니다. API는 데이터를 데이터베이스에 오프로드하거나 성능 개선 도우미 데이터를 기준 모니터링 대시보드에 추가하거나 모니터링 도구를 빌드하는 데 사용할 수 있습니다. 성능 개선 도우미 API를 사용하려면 Amazon RDS DB 인스턴스 중 하나에서 성능 개선 도우미를 활성화하십시오. 성능 개선 도우미 활성화에 대한 자세한 내용은 [성능 개선 도우미 활성화 \(p. 251\)](#) 단원을 참조하십시오.

성능 개선 도우미 API에 대한 자세한 내용은 [Amazon RDS 성능 개선 도우미 API 참조](#)를 참조하십시오.

성능 개선 도우미용 AWS CLI

AWS CLI를 사용해 성능 개선 도우미 데이터를 볼 수 있습니다. 명령줄에 다음과 같이 입력하여 성능 개선 도우미용 AWS CLI 명령에 대한 도움말을 볼 수 있습니다.

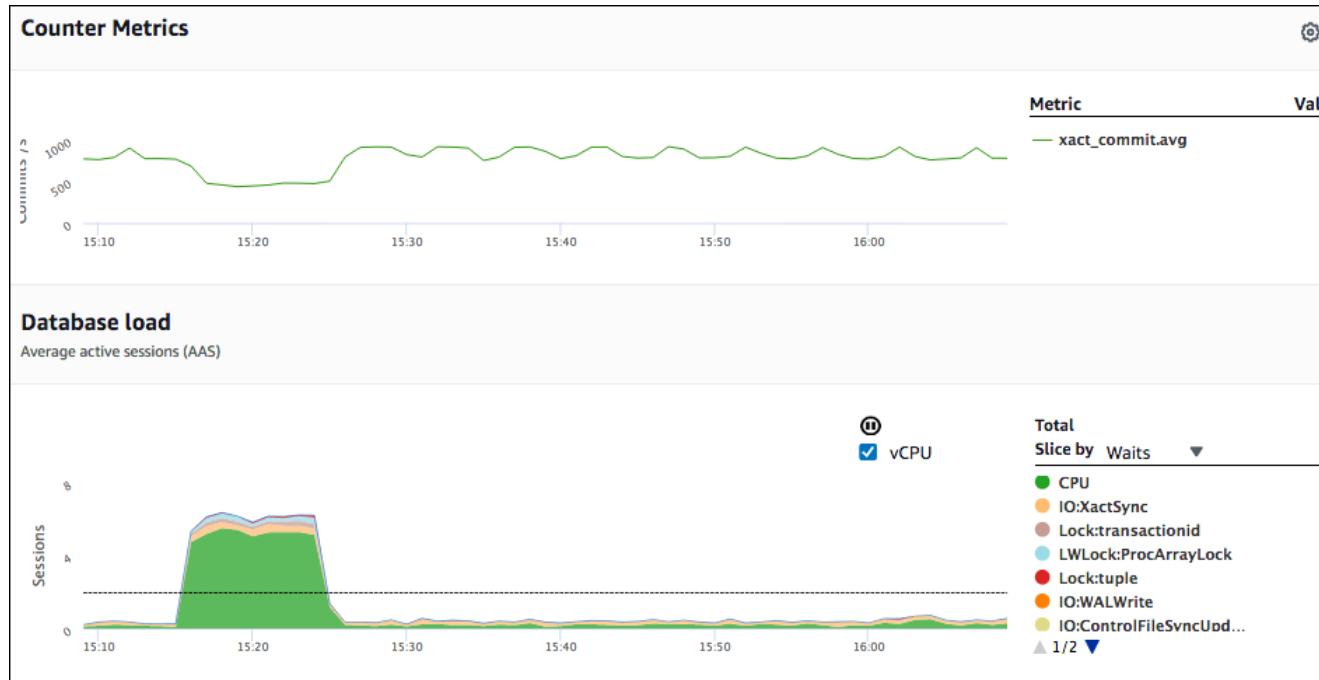
```
aws pi help
```

AWS CLI가 설치되어 있지 않은 경우 설치에 관한 자세한 정보는 AWS CLI 사용 설명서의 [AWS 명령줄 인터페이스 설치](#)를 참조하십시오.

시계열 지표 조회

`GetResourceMetrics` 연산은 성능 개선 도우미 데이터에서 시계열 지표를 하나 이상 조회합니다. `GetResourceMetrics`에는 지표 및 기간이 필요하고 데이터 포인트 목록이 포함된 응답을 반환합니다.

예를 들어 AWS Management 콘솔에서는 성능 개선 도우미 대시보드의 두 곳에서 `GetResourceMetrics`를 사용합니다. `GetResourceMetrics`는 Counter Metrics(카운터 지표) 차트를 입력하는 데 사용되고 Database Load(데이터베이스 로드)에서는 다음 이미지처럼 보입니다.



`GetResourceMetrics`에서 반환하는 모든 지표는 표준 시계열 지표이며 한 가지 예외가 있습니다. 그 예외는 성능 개선 도우미의 핵심 지표인 `db.load`입니다. 이 지표는 Database Load(데이터베이스 부하) 차트에 표시됩니다. `db.load` 지표는 ‘차원’이라는 하위 구성 요소로 구분할 수 있기 때문에 다른 시계열 지표와 달립니다. 앞의 이미지에서 `db.load`는 `db.load`를 구성하는 대기 상태에 따라 구분되고 그룹화됩니다.

`get-resource-metrics` AWS CLI 명령 사용에 대한 자세한 내용은 [get-resource-metrics](#) 단원을 참조하십시오.

--metric-queries 옵션의 경우 결과를 얻고자 하는 쿼리를 한 개 이상 지정하십시오. 각 쿼리는 필수인 Metric과 선택 사항인 GroupBy 및 Filter 파라미터로 구성됩니다. 다음은 --metric-queries 옵션 사용을 보여주는 예입니다.

```
{
  "Metric": "string",
  "GroupBy": {
    "Group": "string",
    "Dimensions": ["string", ...],
    "Limit": integer
  },
  "Filter": {"string": "string"
  ...
}
```

성능 개선 도우미에 대한 AWS CLI 예시

다음은 성능 개선 도우미에 대한 AWS CLI 사용을 보여주는 예입니다.

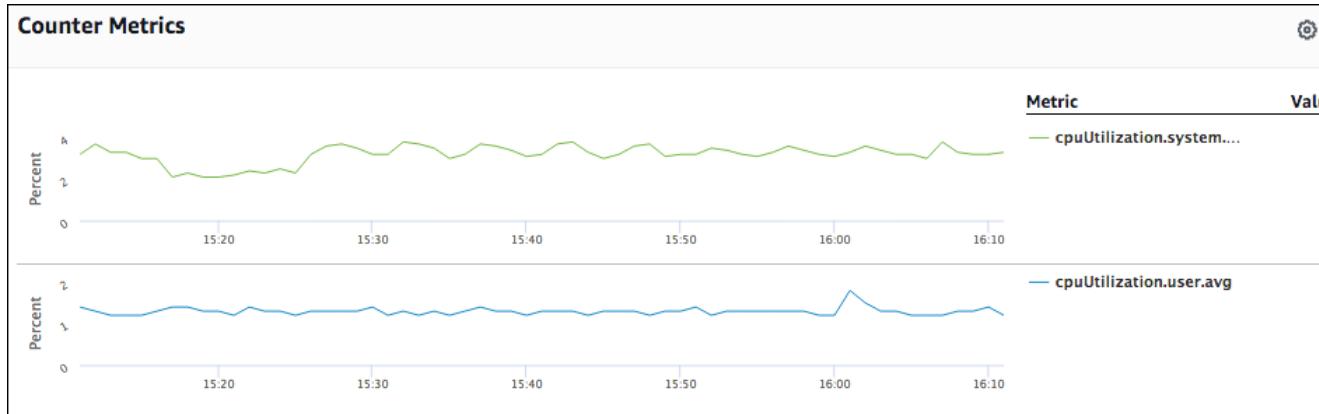
주제

- 카운터 지표 검색 (p. 267)
- 상위 대기 이벤트에 대한 DB 평균 로드 검색 (p. 269)
- 상위 SQL에 대한 DB 평균 로드 검색 (p. 271)

- SQL을 기준으로 필터링된 DB 평균 로드 검색 (p. 273)

카운터 지표 검색

다음 이미지는 AWS Management 콘솔에 표시되는 카운터 지표 차트 2개를 나타낸 것입니다.



다음 예에서는 카운터 지표 차트 2개를 생성하기 위해 AWS Management 콘솔이 사용하는 것과 동일한 데이터를 수집하는 방법을 보여줍니다.

Linux, OS X, Unix의 경우:

```
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": "sys.cpu.user.avg"}, {"Metric": "sys.cpu.system.avg"}]'
```

Windows의 경우:

```
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": "sys.cpu.user.avg"}, {"Metric": "sys.cpu.system.avg"}]'
```

--metrics-query 옵션에 대해 파일을 지정하면 명령이 더 쉽게 읽히도록 할 수 있습니다. 다음 예에서는 옵션에 대해 query.json이라는 파일을 사용합니다. 이 파일의 콘텐츠는 다음과 같습니다.

```
[{"Metric": "sys.cpu.user.avg"}, {"Metric": "sys.cpu.system.avg"}]
```

```
    }  
]
```

다음 명령을 실행하여 파일을 사용합니다.

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

앞의 예에서는 옵션에 다음 값을 지정합니다.

- **--service-type** – Amazon RDS용 RDS
- **--identifier** – DB 인스턴스에 대한 리소스 ID입니다.
- **--start-time** 및 **--end-time** – 쿼리할 기간에 대한 ISO 8601 DateTime 값으로서, 지원되는 형식은 여러 가지입니다.

다음과 같이 1시간 범위로 쿼리합니다.

- **--period-in-seconds** – 1분당 쿼리의 경우 60
- **--metric-queries** – 쿼리 2개의 배열, 각 쿼리는 지표 1개에만 해당됨.

지표 이름에는 지표를 유용한 범주로 분류하기 위해 점이 사용되고, 마지막 요소는 함수입니다. 예시에서 함수는 각 쿼리에 대해 avg입니다. Amazon CloudWatch와 마찬가지로 지원되는 함수는 min, max, total 및 avg입니다.

응답은 다음과 비슷합니다.

```
{  
  "Identifier": "db-XXX",  
  "AlignedStartTime": 1540857600.0,  
  "AlignedEndTime": 1540861200.0,  
  "MetricList": [  
    { //A list of key/datapoints  
      "Key": {  
        "Metric": "sys.cpu.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 sys.cpu.user.avg metric
        ]
    },
    {
        "Key": {
            "Metric": "sys.cpu.system.avg" //Metric2
        },
        "DataPoints": [
            {
                "Timestamp": 1540857660.0, //Minute1
                "Value": 12.0
            },
            {
                "Timestamp": 1540857720.0, //Minute2
                "Value": 13.5
            },
            //... 60 datapoints for the sys.cpu.system.avg metric
        ]
    }
]
] //end of MetricList
} //end of response

```

응답에는 Identifier, AlignedStartTime 및 AlignedEndTime이 있습니다. --period-in-seconds 값이 60인 경우 시작 및 종료 시간은 분 단위로 맞춰져 있습니다. --period-in-seconds 값이 3600인 경우 시작 및 종료 시간은 시간 단위로 맞춰져 있습니다.

응답의 MetricList에는 다수의 항목이 있는데, 각각 Key 및 DataPoints 항목이 포함되어 있습니다. 각 DataPoint에는 Timestamp 및 Value이 있습니다. 쿼리는 1시간에 걸친 분당 데이터에 대한 것이므로 각 Datapoints 목록에는 Timestamp1/Minute1, Timestamp2/Minute2 등에서 최대 Timestamp60/Minute60까지 60개의 데이터 포인트가 있습니다.

쿼리는 두 가지 카운터 지표에 대한 것이므로 MetricList 응답에는 두 개의 요소가 있습니다.

상위 대기 이벤트에 대한 DB 평균 로드 검색

다음 예는 AWS Management 콘솔에서 누적 영역 선 그래프를 생성하는 데 사용하는 것과 동일한 쿼리입니다. 이 예에서는 최상위 7개 대기 이벤트에 따라 구분된 로드의 마지막 한 시간 db.load.avg를 검색합니다. 명령은 [카운터 지표 검색 \(p. 267\)](#)의 명령과 동일합니다. 그러나 query.json 파일의 컨텐츠는 다음과 같습니다.

```

[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_event", "Limit": 7 }
    }
]

```

다음 명령을 실행합니다.

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

이 예시에서는 최상위 7개 대기 이벤트의 db.load.avg 및 GroupBy에 대한 지표를 지정합니다. 이 예의 유효 값에 대한 자세한 내용은 성능 개선 도우미 API 참조의 [DimensionGroup](#) 단원을 참조하십시오.

응답은 다음과 비슷합니다.

```
{
    "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": [
                    {
                        "Name": "CPU", "Value": "Core 0"
                    },
                    {
                        "Name": "CPU", "Value": "Core 1"
                    }
                ]
            },
            "DataPoints": [
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 0.2583333333333333
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 0.19166666666666667
                },
                {
                    "Timestamp": 1540857780.0, //Minute 3
                    "Value": 0.1483333333333333
                }
                //... 60 datapoints for the db.load.avg by CPU key
            ]
        }
    ]
}
```

```
        "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
```

이 응답에는 MetricList에 항목이 8개 있습니다. 총 db.load.avg에는 항목이 1개 있고, 최상위 7개 대기 이벤트 중 하나에 따라 구분된 db.load.avg에 각각에 대해서는 항목이 7개 있습니다. 첫 번째 예시와 달리 그룹화 차원이 있었기 때문에 지표에 대한 각 그룹화에는 키가 1개 있어야 합니다. 기본 카운터 지표 사용 사례처럼 각 지표에 키가 한 개만 있을 수는 없습니다.

상위 SQL에 대한 DB 평균 로드 검색

다음 예에서는 최상위 10개 SQL 문을 기준으로 db.wait_events를 그룹화합니다. SQL 문에는 두 가지 그룹이 있습니다.

- db.sql-select * from customers where customer_id = 123와 같은 SQL 문
- db.sql_tokenized-select * from customers where customer_id = ?와 같은 토큰화된 SQL 문

데이터베이스 성능 분석 시 파라미터만 다른 SQL 문은 하나의 로직 항목으로 간주하는 것이 도움이 될 수 있습니다. 따라서 쿼리 시에는 db.sql_tokenized를 사용할 수 있습니다. 그러나 특히 설명 계획에 관심이 있는 경우에는 때로 파라미터가 있는 전체 SQL 문을 검토하고 db.sql로 그룹화를 쿼리하는 것이 유용합니다. 토큰화된 SQL과 전체 SQL 간에는 상위-하위 관계가 있는데, 여러 개의 전체 SQL(하위)은 토큰화된 동일한 SQL(상위) 아래에 그룹화됩니다.

이 예의 명령은 [상위 대기 이벤트에 대한 DB 평균 로드 검색 \(p. 269\)](#)의 명령과 유사합니다. 그러나 query.json 파일의 컨텐츠는 다음과 같습니다.

```
[  
    {  
        "Metric": "db.load.avg",  
        "GroupBy": { "Group": "db.sql_tokenized", "Limit": 10 }  
    }  
]
```

다음 예에는 db.sql_tokenized가 사용됩니다.

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

이 예에서는 24시간 동안 쿼리를 실행하는데 1시간은 초 단위로 구성됩니다.

이 예에서는 최상위 7개 대기 이벤트의 db.load.avg 및 GroupBy에 대한 지표를 지정합니다. 이 예의 유효 값에 대한 자세한 내용은 성능 개선 도우미 API 참조의 [DimensionGroup](#) 단원을 참조하십시오.

응답은 다음과 비슷합니다.

```
{
    "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
```

이 응답은 MetricList에 11개의 항목이 있는데(전체 1개, 최상위 토큰화 SQL 10개) 각 항목에는 시간당 DataPoints가 24개입니다.

토큰화된 SQL의 경우 각 차원 목록에 3개의 항목이 있습니다.

- db.sql_tokenized.statement – 토큰화된 SQL 문입니다.
- db.sql_tokenized.db_id – SQL 참조에 사용되는 기본 데이터베이스 ID 또는 기본 데이터베이스 ID를 사용할 수 없는 경우 성능 개선 도우미가 생성하는 합성 ID입니다. 이 예에서는 pi-2372568224 합성 ID를 반환합니다.
- db.sql_tokenized.statement – 성능 개선 도우미 내부의 쿼리에 대한 ID입니다.

AWS Management 콘솔에서는 이 ID를 지원 ID라고 합니다. 이렇게 부르는 이유는 이 ID가 데이터베이스 관련 문제 해결을 지원하기 위해 AWS Support가 검토하는 데이터이기 때문입니다. AWS는 데이터 보안 및 개인 정보 보호를 중대 사안으로 간주하므로 거의 모든 데이터는 AWS KMS 키로 암호화되어 저장됩니다. 그러므로 AWS 내부의 어느 누구도 이 데이터를 볼 수 없습니다. 앞의 예에서 tokenized_statement와 tokenized.db_id 모두 암호화되어 저장됩니다. 데이터베이스 관련 문제가 있는 경우 AWS Support가 지원 ID를 참조하여 도움을 드릴 수 있습니다.

쿼리 시 GroupBy에서 Group을 지정하면 편리할 수 있습니다. 그러나 반환되는 데이터에 대한 더 세분화된 제어를 위해서는 차원 목록을 지정하십시오. 예를 들어 db.sql_tokenized.statement만 필요한 경우에는 query.json file에 Dimensions 속성을 추가할 수 있습니다.

```
[  
  {  
    "Metric": "db.load.avg",  
    "GroupBy": {  
      "Group": "db.sql_tokenized",  
      "Dimensions": ["db.sql_tokenized.statement"],  
      "Limit": 10  
    }  
  }  
]
```

SQL을 기준으로 필터링된 DB 평균 로드 검색



앞의 이미지에서는 특정 쿼리가 선택되어 있고 상위 평균 활성 세션 누적 영역 선 그래프는 이 쿼리로 범위가 지정되어 있습니다. 쿼리가 여전히 최상위 7개 전체 대기 이벤트에 대한 것이라 하더라도 응답의 값은 필터링됩니다. 필터로 인해 특정 필터의 짹이 되는 세션만 고려합니다.

이 예에서 해당되는 API 쿼리는 [상위 SQL에 대한 DB 평균 로드 검색 \(p. 271\)](#) 단원의 명령과 유사합니다. 그러나 query.json 파일의 컨텐츠는 다음과 같습니다.

```
[  
 {  
   "Metric": "db.load.avg",  
   "GroupBy": { "Group": "db.wait_event", "Limit": 5 },  
   "Filter": { "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }  
 }  
]
```

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

응답은 다음과 비슷합니다.

```
{  
  "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

```

이 응답에서 모든 값은 query.json 파일에 지정된 토큰화된 SQL AKIAIOSFODNN7EXAMPLE의 기여에 따라 필터링됩니다. 키는 필터링된 SQL에 영향을 미친 상위 5개 대기 이벤트이므로 필터가 없는 쿼리와는 다른 순서를 따를 수 있습니다.

Amazon CloudWatch에 게시되는 성능 개선 도우미 지표

성능 개선 도우미는 Amazon CloudWatch에 지표를 자동으로 게시합니다. 동일한 데이터는 성능 개선 도우미에서 쿼리할 수 있지만 CloudWatch에 지표가 있으면 CloudWatch 경보를 더 쉽게 추가할 수 있습니다. 또한 기존 CloudWatch 대시보드에 지표를 더 쉽게 추가할 수 있습니다.

지표	설명
DBLoad	DB 엔진에 대한 활성 세션 수입니다. 일반적으로 사용자는 활성 세션의 평균 개수에 대한 데이터를 원합니다. 성능 개선 도우미에서 이 데이터는 db.load.avg로 쿼리됩니다.
DBLoadCPU	대기 이벤트 유형이 CPU인 활성 세션 수입니다. 성능 개선 도우미에서 이 데이터는 db.load.avg로 쿼리되며 대기 이벤트 유형인 CPU를 기준으로 필터링됩니다.

지표	설명
DBLoadNonCPU	대기 이벤트 유형이 CPU가 아닌 활성 세션 수입니다.

CloudWatch 콘솔, AWS CLI 또는 CloudWatch API를 사용하여 이러한 지표를 검사할 수 있습니다.

예를 들어, [get-metric-statistics](#) 명령을 실행하여 DBLoad 지표에 대한 통계를 가져올 수 있습니다.

```
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
```

이 예에서는 다음과 비슷한 출력이 생성됩니다.

```
{
  "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"
}
```

CloudWatch에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 [Amazon CloudWatch이란 무엇인가?](#) 단원을 참조하십시오.

성능 개선 도우미 카운터

계수기 지표를 통해 성능 개선 도우미 대시보드에 최대 10개의 추가 그래프가 포함되도록 사용자 지정할 수 있습니다. 이 그래프에는 수십 건의 운영 체제 및 데이터베이스 성능 지표 모음이 표시됩니다. 이 정보와 데이터베이스 로드를 연관 지으면 성능 문제를 식별하고 분석하는 데 도움이 됩니다.

주제

- [성능 개선 도우미 운영 체제 카운터 \(p. 278\)](#)
- [MariaDB 및 MySQL용 Amazon RDS의 성능 개선 도우미 카운터 \(p. 280\)](#)
- [Oracle용 Amazon RDS의 성능 개선 도우미 카운터 \(p. 283\)](#)
- [Amazon RDS PostgreSQLdyd 성능 개선 도우미 카운터 \(p. 284\)](#)

성능 개선 도우미 운영 체제 카운터

Aurora PostgreSQL용 성능 개선 도우미에서 다음 운영 체제 카운터를 사용할 수 있습니다. [CloudWatch Logs를 사용하여 Enhanced Monitoring 보기 \(p. 244\)](#)에서 이러한 지표의 정의를 확인할 수 있습니다.

카운터	유형
active	메모리
버퍼	메모리
캐시됨	메모리
더티	메모리
사용 가능	메모리
비활성	메모리
hugePagesFree	메모리
hugePagesRsvd	메모리
hugePagesSize	메모리
hugePagesSurp	메모리
hugePagesTotal	메모리
매팅됨	메모리
pageTables	메모리
슬래브	메모리
총합	메모리
writeback	메모리
게스트	cpuUtilization
유 휴	cpuUtilization

카운터	유형
irq	cpuUtilization
nice	cpuUtilization
도용	cpuUtilization
시스템	cpuUtilization
총합	cpuUtilization
사용자	cpuUtilization
대기	cpuUtilization
avgQueueLen	diskIO
avgReqSz	diskIO
await	diskIO
readIOsPS	diskIO
readKb	diskIO
readKbPS	diskIO
rrqmPS	diskIO
tps	diskIO
util	diskIO
writelOsPS	diskIO
writeKb	diskIO
writeKbPS	diskIO
wrqmPS	diskIO
차단됨	작업
실행 중	작업
절전	작업
중단됨	작업
총합	작업
좀비	작업
1	loadAverageMinute
15	loadAverageMinute
5	loadAverageMinute
캐시됨	스왑
사용 가능	스왑

카운터	유형
in	스왑
out	스왑
총합	스왑
maxFiles	fileSys
usedFiles	fileSys
usedFilePercent	fileSys
usedPercent	fileSys
used	fileSys
총합	fileSys
rx	network
tx	network
numVCpus	general

MariaDB 및 MySQL용 Amazon RDS의 성능 개선 도우미 카운터

MariaDB 및 MySQL용 Amazon RDS를 위한 성능 개선 도우미에서 다음 데이터베이스 카운터를 사용할 수 있습니다.

주제

- [RDS MariaDB 및 RDS MySQL용 기본 카운터 \(p. 280\)](#)
- [MariaDB 및 MySQL용 Amazon RDS를 위한 기본이 아닌 카운터 \(p. 282\)](#)

RDS MariaDB 및 RDS MySQL용 기본 카운터

MySQL 설명서의 [서버 상태 변수](#)에서 이러한 기본 지표에 대한 정의를 확인하실 수 있습니다.

카운터	Type	Unit
Com_analyze	SQL	초당 쿼리 수
Com_optimize	SQL	초당 쿼리 수
Com_select	SQL	초당 쿼리 수
Innodb_rows_deleted	SQL	초당 행
Innodb_rows_inserted	SQL	초당 행
Innodb_rows_read	SQL	초당 행
Innodb_rows_updated	SQL	초당 행
Select_full_join	SQL	초당 쿼리 수
Select_full_range_join	SQL	초당 쿼리 수

카운터	Type	Unit
Select_range	SQL	초당 쿼리 수
Select_range_check	SQL	초당 쿼리 수
Select_scan	SQL	초당 쿼리 수
Slow_queries	SQL	초당 쿼리 수
Sort_merge_passes	SQL	초당 쿼리 수
Sort_range	SQL	초당 쿼리 수
Sort_rows	SQL	초당 쿼리 수
Sort_scan	SQL	초당 쿼리 수
질문	SQL	초당 쿼리 수
Table_locks_immediate	잠금	초당 요청
Table_locks_waited	잠금	초당 요청
Innodb_row_lock_time	잠금	밀리초(평균)
Aborted_clients	Users	Connections
Aborted_connects	Users	Connections
Threads_created	Users	Connections
Threads_running	Users	Connections
Innodb_data_writes	IO	초당 연산 수
Innodb_dblwr_writes	IO	초당 연산 수
Innodb_log_write_requests	IO	초당 연산 수
Innodb_log_writes	IO	초당 연산 수
Innodb_pages_written	IO	초당 페이지
Created_tmp_disk_tables	Temp	초당 테이블
Created_tmp_tables	Temp	초당 테이블
Innodb_buffer_pool_pages_data	Cache	페이지
Innodb_buffer_pool_pages_total	Cache	페이지
Innodb_buffer_pool_read_requests	Cache	초당 페이지
Innodb_buffer_pool_reads	Cache	초당 페이지
Opened_tables	Cache	테이블
Opened_table_definitions	Cache	테이블
Qcache_hits	Cache	쿼리

MariaDB 및 MySQL용 Amazon RDS를 위한 기본이 아닌 카운터

기본이 아닌 카운터 지표는 Amazon RDS가 정의하는 카운터입니다. 기본이 아닌 지표는 특정 쿼리를 통해 얻는 지표일 수 있습니다. 기본이 아닌 지표는 파생 지표일 수 있습니다. 이 경우 비율, 적중률 또는 자연 시간에 대한 계산 시 2개 이상의 기본 카운터가 사용됩니다.

카운터	Type	설명	정의
innodb_buffer_pool_reads	Count	InnoDB가 버퍼 풀에서 충족할 수 있었던 읽기의 수입니다.	innodb_buffer_pool_read_requests - innodb_buffer_pool_reads
innodb_buffer_pool_read_requests	Count	InnoDB가 버퍼 풀에서 InnoDB가 충족할 수 있었던 읽기의 비율입니다.	100 * innodb_buffer_pool_read_requests / (innodb_buffer_pool_read_requests + innodb_buffer_pool_reads)
innodb_buffer_pool_pages_data	Count	데이터(페이지)를 포함하는 InnoDB 버퍼 풀의 비율입니다. Note 입출력 테이블을 사용하는 경우 이 값은 달라질 수 있습니다. 자세한 내용은 MySQL 설명서에서 서버 상태 변수 의 Innodb_buffer_pool_pages_data 및 Innodb_buffer_pool_pages_total에 대한 정보를 참조하십시오.	Innodb_buffer_pool_pages_data / Innodb_buffer_pool_pages_total * 100.0
query_cache_hit_ratio	Ratio	MySQL 결과 집합 캐시(쿼리 캐시) 적중률입니다.	QCache_hits / (QCache_hits + Com_select) * 100
innodb_datafile_writes_to_disk	Count	디스크에 대한 InnoDB 데이터 파일 쓰기의 수(이중 쓰기 및 재실행 로깅 쓰기 연산은 제외)입니다.	Innodb_data_writes - Innodb_log_writes - Innodb_dblwr_writes
innodb_rows_changed	Count	총 InnoDB 행 연산입니다.	db.SQL.Innodb_rows_inserted + db.SQL.Innodb_rows_deleted + db.SQL.Innodb_rows_updated
active_transactions	Count	총 활성 트랜잭션입니다.	SELECT COUNT(1) AS active_transactions FROM INFORMATION_SCHEMA.INNODB_TRX
innodb_deadlocks	Count	교착 상태의 총 개수입니다.	SELECT COUNT AS innodb_deadlocks FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_deadlocks'

카운터	Type	설명	정의
innodb_lock_timeouts	정수	시간을 초과한 교착 상태의 총 개수입니다.	<pre>SELECT COUNT AS innodb_lock_timeouts FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_timeouts'</pre>
innodb_row_lock_waits	정수	대기의 원인이 된 행 잠금의 총 개수입니다.	<pre>SELECT COUNT AS innodb_row_lock_waits FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_row_lock_waits'</pre>

Oracle용 Amazon RDS의 성능 개선 도우미 카운터

RDS Oracle용 성능 개선 도우미에서는 다음 데이터베이스 카운터를 사용할 수 있습니다.

RDS Oracle용 기본 카운터

Oracle 설명서의 [통계 설명](#)에서 이러한 기본 지표에 대한 정의를 확인하실 수 있습니다.

Note

CPU used by this session 카운터 지표의 경우 단위가 기본 100분의 1초에서 활성 세션으로 변환되어 값을 사용하기가 더 쉬워졌습니다. 예를 들어 DB 로드 차트의 CPU 전송은 CPU에 대한 수요를 나타냅니다. 카운터 지표 CPU used by this session은 Oracle 세션의 CPU 사용량을 나타냅니다. CPU 전송을 CPU used by this session 카운터 지표와 비교할 수 있습니다. CPU에 대한 수요가 사용된 CPU보다 높은 경우 세션은 CPU 시간을 기다리고 있습니다.

카운터	Type	Unit
이 세션에서 사용한 CPU	User	활성 세션 dl 경우 단위가 기본 100분의 1초에서 활성 세션으로 변환되어 값을 사용하기가 더 쉬워졌습니다.
클라이언트에게로 또는 클라이언트로의 SQL*Net 왕복	User	초당 왕복
클라이언트에게서 SQL*Net을 통해 수신한 바이트 수	User	초당 바이트
사용자 커밋	User	초당 커밋 수
누적 로그온 수	User	초당 로그온 수
사용자 호출	User	초당 호출 수
SQL*Net을 통해 클라이언트에게 전송된 바이트	User	초당 바이트
사용자 룰백	User	초당 룰백 수
재실행 크기	Redo	초당 바이트

카운터	Type	Unit
구문 분석 개수(합계)	SQL	초당 구문 분석 수
구문 분석 개수(하드)	SQL	초당 구문 분석 수
획득한 테이블 스캔 행	SQL	초당 행
정렬(메모리)	SQL	초당 정렬
정렬(디스크)	SQL	초당 정렬
정렬(행)	SQL	초당 정렬
물리적 읽기 바이트 수	Cache	초당 바이트
DB 블록 GET	Cache	초당 블록 수
DBWR 체크포인트	Cache	분당 체크포인트
물리적 읽기 수	Cache	초당 읽기 수
캐시에서 일관된 GET	Cache	초당 GET의 수
캐시의 DB 블록 GET	Cache	초당 GET의 수
일관된 GET	Cache	초당 GET의 수

Amazon RDS PostgreSQLdyd 성능 개선 도우미 카운터

Amazon RDS PostgreSQL용 성능 개선 도우미에서는 다음 데이터베이스 카운터를 사용할 수 있습니다.

주제

- [Amazon RDS PostgreSQL용 기본 카운터 \(p. 284\)](#)
- [기본이 아닌 Amazon RDS PostgreSQL용 카운터 \(p. 285\)](#)

Amazon RDS PostgreSQL용 기본 카운터

PostgreSQL 설명서의 [통계 보기](#)에서 이러한 기본 지표에 대한 정의를 확인하실 수 있습니다.

카운터	Type	Unit
blk_hit	Cache	초당 블록 수
buffers_alloc	Cache	초당 블록 수
buffers_checkpoint	체크포인트	초당 블록 수
checkpoint_sync_time	체크포인트	체크포인트당 밀리초
checkpoint_write_time	체크포인트	체크포인트당 밀리초
checkpoints_req	체크포인트	분당 체크포인트
checkpoints_timed	체크포인트	분당 체크포인트
maxwritten_clean	체크포인트	분당 Bgwriter 클린 스텝

카운터	Type	Unit
deadlocks	동시성	분당 교착 상태의 수
blk_read_time	IO	밀리초
blk_s_read	IO	초당 블록 수
buffers_backend	IO	초당 블록 수
buffers_backend_fsync	IO	초당 블록 수
buffers_clean	IO	초당 블록 수
tup_deleted	SQL	초당 튜플 수
tup_fetched	SQL	초당 튜플 수
tup_inserted	SQL	초당 튜플 수
tup_returned	SQL	초당 튜플 수
tup_updated	SQL	초당 튜플 수
temp_bytes	Temp	초당 바이트
temp_files	Temp	분당 파일 수
active_transactions	트랜잭션	트랜잭션
blocked_transactions	트랜잭션	트랜잭션
max_used_xact_ids	트랜잭션	트랜잭션
xact_commit	트랜잭션	초당 커밋 수
xact_rollback	트랜잭션	초당 롤백 수
numbackends	User	Connections
archived_count	WAL	분당 파일 수
archive_failed_count	WAL	분당 파일 수

기본이 아닌 Amazon RDS PostgreSQL용 카운터

기본이 아닌 카운터 지표는 Amazon RDS가 정의하는 카운터입니다. 기본이 아닌 지표는 특정 쿼리를 통해 얻는 지표일 수 있습니다. 기본이 아닌 지표는 파생 지표일 수 있습니다. 이 경우 비율, 적중률 또는 지연 시간에 대한 계산 시 2개 이상의 기본 카운터가 사용됩니다.

카운터	Type	설명	정의
checkpoint_sync_time	체크포인트	파일이 디스크에 동기화되는 체크포인트 처리 중 일부에서 소요된 총 시간입니다.	$\text{checkpoint_sync_time} / (\text{checkpoints_timed} + \text{checkpoints_req})$
checkpoint_write_time	체크포인트	파일이 디스크에 쓰이는 체크포인트 처리 중 일부에서 소요된 총 시간입니다.	$\text{checkpoint_write_time} / (\text{checkpoints_timed} + \text{checkpoints_req})$

카운터	Type	설명	정의
read_latency	IO	이 인스턴스의 백엔드에서 데이터 파일 블록을 읽는 데 소요된 시간입니다.	<code>blk_read_time / blks_read</code>

AWS CloudTrail을 사용하여 성능 개선 도우미 작업 로깅

성능 개선 도우미는 AWS CloudTrail과 통합됩니다. CloudTrail은 AWS 계정에서 성능 개선 도우미에 의해 또는 성능 개선 도우미를 대신하여 수행된 하위 수준 API 요청을 캡처합니다. 그런 다음 CloudTrail은 사용자가 지정한 Amazon S3 버킷에 로그 파일을 전송합니다. CloudTrail은 RDS 콘솔의 성능 개선 도우미 또는 성능 개선 도우미 하위 수준 API로부터 수행된 호출을 캡처합니다.

CloudTrail에서 수집된 정보를 사용하여 Performance Insights에 어떤 요청이 수행되었는지 알 수 있습니다. 또한 어떤 소스 IP 주소에서 요청했는지, 누가 언제 요청했는지 등을 확인할 수 있습니다. AWS 계정에서 자동으로 CloudTrail 로깅이 활성화됩니다. CloudTrail에 대한 자세한 내용은 [AWS CloudTrail User Guide](#)를 참조하십시오.

성능 개선 도우미에 수행되는 하위 수준 API 호출 작업은 로그 파일로 추적됩니다. 성능 개선 도우미 레코드는 다른 AWS 서비스 레코드와 함께 로그 파일에 기록됩니다. CloudTrail은 기간 및 파일 크기에 따라 새 파일을 언제 생성하고 쓸지를 결정합니다. 지원되는 API 작업은 다음과 같습니다.

- [DescribeDimensionKeys](#)
- [GetResourceMetrics](#)

Amazon RDS 권장 사항 사용

Amazon RDS DB 인스턴스, 읽기 복제본, DB 파라미터 그룹 등 데이터베이스에自動화된 권장 사항을 제공합니다. 이러한 권장 사항은 DB 인스턴스 구성, 사용량 및 성능 데이터에 대한 분석을 통해 모범 사례 지침을 제공합니다.

다음 표에서 이러한 권장 사항의 예를 확인할 수 있습니다.

Type	설명	권장 사항	추가 정보
오래된 엔진 버전	DB 인스턴스가 최신 마이너 엔진 버전을 실행하지 않습니다.	최신 보안 수정 사항과 기타 개선 사항이 포함된 최신 버전으로 업그레이드하는 것이 좋습니다.	DB 인스턴스 엔진 버전 업그레이드 (p. 118)
보류 중인 유지 관리 사용 가능	DB 인스턴스에서 가능한 보류 중인 유지 관리 작업이 있습니다.	DB 인스턴스에서 가능한 보류 중인 유지 관리 작업을 수행하는 것이 좋습니다. 운영 체제 업데이트는 보안상 가장 빈번하게 발생하며 최대한 빠른 시간 내에 실행해야 합니다.	DB 인스턴스 유지 관리 (p. 113)
자동 백업 비활성화됨	DB 인스턴스에서 자동 백업이 비활성화되었습니다.	DB 인스턴스에서 자동 백업을 활성화하는 것이 좋습니다. 자동 백업은 DB 인스턴스의 특정 시점 복구를 지원합니다. 추가 비용 없이 DB 인스턴스의 스토리지 크기만큼 최대 백업 스토리지가 제공됩니다.	백업 작업 (p. 196)

Type	설명	권장 사항	추가 정보
사용 중인 마그네티ック 볼륨	DB 인스턴스에서 마그네티ック 스토리지를 사용 중입니다.	대부분 DB 인스턴스에 대해 마그네티ック 스토리지는 권장되지 않습니다. 범용 (SSD) 스토리지 또는 프로비저닝된 IOPS 스토리지로 전환하는 것이 좋습니다.	Amazon RDS DB 인스턴스 스토리지 (p. 101)
사용 중인 EC2-Classic 플랫폼	DB 인스턴스에서 레거시 EC2-Classic 플랫폼을 사용 중입니다.	네트워크 액세스 제어를 항상 하려면 DB 인스턴스를 EC2-VPC 플랫폼으로 마이그레이션하는 것이 좋습니다. Amazon VPC는 AWS 클라우드의 다른 가상 네트워크와 논리적으로 격리된 가상 네트워크를 제공합니다.	EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 (p. 407)
비활성화된 개선된 모니터링	DB 인스턴스에 확장된 모니터링이 활성화되지 않았습니다.	확장된 모니터링을 활성화하는 것이 좋습니다. 확장된 모니터링은 모니터링 및 문제 해결을 위해 실시간 운영 체제 지표를 제공합니다.	확장 모니터링 (p. 238)
비활성화된 암호화	DB 인스턴스에 암호화가 활성화되지 않았습니다.	암호화를 활성화하는 것이 좋습니다. 암호화된 스냅샷에서 복원하여 기존 Amazon RDS DB 인스턴스를 암호화할 수 있습니다.	Amazon RDS 리소스 암호화 (p. 385)
사용 중인 이전 세대 DB 인스턴스 클래스	DB 인스턴스가 이전 세대 DB 인스턴스 클래스에서 실행 중입니다.	이전 세대 DB 인스턴스 클래스는 더 좋은 요금, 더 좋은 성능 또는 이들 모두를 갖춘 DB 인스턴스 클래스로 대체되었습니다. 최신 세대 DB 인스턴스 클래스에서 DB 인스턴스를 실행하는 것이 좋습니다.	DB 인스턴스 클래스 선택 (p. 77)
Oracle DB 인스턴스에 방대한 페이지 사용 안 함	DB 인스턴스에서 사용하는 DB 파라미터 그룹에서 <code>use_large_pages</code> 파라미터가 ONLY로 설정되어 있지 않습니다.	데이터베이스 확장성 향상을 위해서는 DB 인스턴스에서 사용하는 DB 파라미터 그룹에서 <code>use_large_pages</code> 를 ONLY로 설정하는 것이 좋습니다.	Oracle DB 인스턴스에 방대한 페이지 사용 (p. 721)
기본이 아닌 사용자 지정 메모리 파라미터	DB 파라미터 그룹에서는 기본값에서 너무 많이 벗어나는 설정으로 인해 성능이 떨어지고 오류가 발생할 수 있습니다. DB 인스턴스에서 사용하는 DB 파라미터 그룹에서는 사용자 지정 메모리 파라미터를 기본값으로 설정하는 것이 좋습니다.	기본값에서 너무 많이 벗어나는 설정으로 인해 성능이 떨어지고 오류가 발생할 수 있습니다. DB 인스턴스에서 사용하는 DB 파라미터 그룹에서는 사용자 지정 메모리 파라미터를 기본값으로 설정하는 것이 좋습니다.	DB 파라미터 그룹 작업 (p. 161)
MySQL DB 인스턴스에 대해 활성화된 변경 버퍼링	DB 파라미터 그룹에서 변경 버퍼링을 활성화 하였습니다.	변경 버퍼링을 통해 MySQL DB 인스턴스는 보조 인덱스 유지에 필요한 일부 쓰기를 연기할 수 있습니다. 이러한 구성은 성능을 약간 향상 시킬 수 있지만 충돌 복구가 크게 지연되는 원인이 될 수 있습니다. 충돌 복구 중에 보조 인덱스는 최신 상태이어야 합니다. 따라서 변경 버퍼링의 이점보다 매우 오래 지속될 가능성이 있는 충돌 복구 이벤트로 인한 손해가 더 크므로 변경 버퍼링을 비활성화하는 것이 좋습니다.	AWS 데이터베이스 블로그의 Amazon RDS for MySQL에 대한 파라미터 구성 모범 사례, PART 1: 성능 관련 파라미터

Type	설명	권장 사항	추가 정보
MySQL DB 인스턴스에 대해 활성화된 쿼리 캐시	DB 파라미터 그룹에서 쿼리 캐시 파라미터를 활성화하였습니다.	변경 시 캐시 제거가 필요한 경우 쿼리 캐시로 인해 DB 인스턴스가 교착 상태에 빠진 것처럼 보일 수 있습니다. 쿼리 캐시는 대부분의 워크로드에 이점이 되지 못합니다. 쿼리 캐시는 MySQL 버전 8.0에서 제거되었습니다. 쿼리 캐시 파라미터를 비활성화하는 것이 좋습니다.	AWS 데이터베이스 블로그의 Amazon RDS for MySQL에 대한 파라미터 구성 모범 사례, PART 1: 성능 관련 파라미터
테이블에 로깅 하기	DB 파라미터 그룹은 로깅 출력을 TABLE로 설정합니다.	로깅 출력을 TABLE로 설정하면 이 파라미터를 FILE로 설정하는 것보다 더 많은 스토리지를 사용합니다. 스토리지 한도에 도달하지 않게 하려면 로깅 출력 파라미터를 FILE로 설정하는 것이 좋습니다.	MySQL 데이터베이스 로그 파일 (p. 318)

Amazon RDS는 리소스가 생성되거나 수정될 때 리소스에 대해 권장 사항을 생성합니다. Amazon RDS 역시 리소스를 주기적으로 스캔하고 권장 사항을 생성합니다.

Amazon RDS 권장 사항에 대응

AWS Management 콘솔에서 권장 사항을 확인할 수 있습니다. 즉시 권장 조치를 수행하거나 다음 유지 관리 기간으로 예약하거나 무시할 수 있습니다.

Amazon RDS 권장 사항에 대응하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 Recommendations(권장 사항)를 선택합니다.

The screenshot shows the Amazon RDS Dashboard. On the left, there's a sidebar with links like Dashboard, Databases, Performance Insights, Snapshots, Automated backups, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, and Recommendations. The 'Recommendations' link is circled in red. On the right, there's a 'Resources' panel with links for DB Instances, Allocated, Click here, Reserved inst, Snapshots (1), Manual (2), Automate, Recent event, and Event subscri. Below that is a 'Create dat' section and the 'Amazon Rela' section.

Recommendations(권장 사항) 페이지가 표시됩니다.

The 'Recommendations' page displays three items:

- Enhanced monitoring disabled (2) - DB instances that do not have Enhanced Monitoring enabled. [Info](#)
- Magnetic volume in use (1) - DB instances using magnetic storage. [Info](#)
- Encryption disabled (1) - DB instances that do not have encryption enabled. [Info](#)

3. Recommendations(권장 사항) 페이지에서 다음 중 하나를 선택합니다.
- Active(활성) – 적용, 무시 또는 예약할 수 있는 현재 권장 사항을 표시합니다.
 - Dismissed(무시됨) – 무시된 권장 사항을 표시합니다. Dismissed(무시됨)를 선택할 때 무시된 권장 사항을 적용할 수 있습니다.
 - 예약됨 – 예약되었지만 아직 적용되지 않은 권장 사항을 표시합니다. 이러한 권장 사항은 다음 예약된 유지 관리 기간에 적용됩니다.
 - Applied(적용됨) – 현재 적용된 권장 사항을 표시합니다.

권장 사항 목록에서 섹션을 열어 해당 섹션의 권장 사항을 볼 수 있습니다.

The screenshot shows the 'Recommendations' page with the 'Active (4)' tab selected. Under the 'Enhanced monitoring disabled' section, there are two items for DB instances 'mymariadb' and 'mysqldb'. Each item has a detailed description and a timestamp. Below this section are other collapsed sections: 'Magnetic volume in use (1)' and 'Encryption disabled (1)'. At the top right of the recommendations table, there are 'Dismiss' and 'Apply now' buttons, and a navigation bar with arrows and a magnifying glass icon.

각 섹션의 권장 사항 표시에 대한 기본 설정을 구성하려면 기본 설정 아이콘을 선택합니다.

This screenshot is identical to the one above, but the 'Apply now' button in the top right corner of the recommendations table is circled in red to indicate it as the target for configuration.

표시되는 기본 설정 창에서 표시 옵션을 설정할 수 있습니다. 이러한 옵션에는 표시된 열과 페이지에 표시할 권장 사항 수가 포함됩니다.

4. 활성 권장 사항 관리:

- 활성을 선택하고 하나 이상의 섹션을 열어 권장 사항을 봅니다.
- 하나 이상의 권장 사항을 선택하고 Apply now(지금 적용)(즉시 적용), Schedule(일정)(다음 유지 관리 기간에 적용) 또는 Dismiss(무시)를 선택합니다.

권장 사항에 대해 지금 적용 버튼이 표시되지만 사용할 수 없는 경우(회색으로 표시) DB 인스턴스를 사용할 수 없습니다. DB 인스턴스 상태가 사용 가능한 경우 권장 사항을 즉시 적용할 수 있습니다. 예를 들어 DB 인스턴스의 상태가 수정 중인 경우 권장 사항을 즉시 적용할 수 없습니다. 이 경우 DB 인스턴스가 사용 가능할 때까지 기다렸다가 권장 사항을 적용합니다.

권장 사항에 대해 활성 버튼이 표시되지 않는 경우 Recommendations(권장 사항) 페이지를 사용하여 권장 사항을 적용할 수 없습니다. DB 인스턴스를 수정하여 수동으로 권장 사항을 적용할 수 있습니다.

DB 인스턴스 변경에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#) 단원을 참조하십시오.

Note

지금 적용을 선택하면 간단한 DB 인스턴스 종단이 발생할 수 있습니다.

Amazon RDS 이벤트 알림 서비스 사용

주제

- [Amazon RDS 이벤트 카테고리 및 이벤트 메시지 \(p. 292\)](#)
- [Amazon RDS 이벤트 알림 구독 \(p. 298\)](#)
- [Amazon RDS 이벤트 알림 구독의 목록 표시 \(p. 299\)](#)
- [Amazon RDS 이벤트 알림 구독 변경 \(p. 301\)](#)
- [Amazon RDS 이벤트 알림 구독에 대한 소스 식별자 추가 \(p. 302\)](#)
- [Amazon RDS 이벤트 알림 구독의 소스 식별자 제거 \(p. 303\)](#)
- [Amazon RDS 이벤트 알림 카테고리의 목록 표시 \(p. 303\)](#)
- [Amazon RDS 이벤트 알림 구독 삭제 \(p. 304\)](#)

Amazon RDS는 Amazon RDS 이벤트 발생 시 Amazon Simple Notification Service(Amazon SNS)를 사용하여 알림 서비스를 제공합니다. 이 서비스는 AWS 리전에 따라 Amazon SNS가 지원하는 알림 메시지 형식에 따라 이메일, 문자 또는 HTTP 엔드포인트 호출 등이 될 수 있습니다.

Amazon RDS는 구독 가능한 카테고리로 이벤트를 그룹화합니다. 따라서 해당 카테고리의 이벤트가 발생했을 때 이에 대한 알림 메시지를 받을 수 있습니다. 구독 가능한 이벤트 범주로는 DB 인스턴스, DB 스냅샷, DB 파라미터 그룹, DB 보안 그룹 등이 있습니다. 예를 들어 임의의 DB 인스턴스에 대한 백업 카테고리를 구독할 경우 백업 관련 이벤트가 발생하여 DB 인스턴스에 영향을 끼칠 때마다 알림 메시지가 수신됩니다. 혹은 DB 보안 그룹의 구성 변경 카테고리를 구독하면 DB 보안 그룹이 변경될 때마다 메시지가 수신됩니다. 또한 이벤트 알림 메시지 구독이 변경되어도 알림 메시지가 수신됩니다.

이벤트 알림 메시지는 구독 생성 시 입력한 주소로 보내집니다. 모든 이벤트 알림 메시지를 수신하거나 프로덕션 DB 인스턴스의 중요 이벤트만 수신하는 등 다른 구독을 복수로 생성하는 것도 가능합니다. Amazon RDS 콘솔에서 활성화 옵션을 아니요로 설정하거나, AWS CLI 또는 Amazon RDS API를 통해 Enabled 파라미터를 `false`로 설정하면 구독을 삭제하지 않고 알림 기능을 쉽게 끌 수 있습니다.

Note

SMS 문자 메시지를 사용한 Amazon RDS 이벤트 알림 서비스는 현재 미국 동부(버지니아 북부) 리전에서 주제 Amazon 리소스 이름(ARN)과 Amazon RDS 리소스로 제공되고 있습니다. SNS 문자 메시지 사용에 대한 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 [Amazon SNS를 통한 SMS 알림 메시지 전송 및 수신](#)을 참조하십시오.

Amazon RDS에서는 Amazon SNS 주제의 ARN을 사용하여 각 구독을 식별합니다. Amazon RDS 콘솔은 구독 생성 시 ARN을 생성합니다. 그 밖에 CLI 또는 API를 사용하는 경우에는 구독을 생성하면서 Amazon SNS 콘솔이나 Amazon SNS API를 통해 ARN을 생성합니다.

Amazon RDS 이벤트 알림에 대한 결제는 Amazon Simple Notification Service(Amazon SNS)를 통해 이루어집니다. 이벤트 알림 사용 시 Amazon SNS 요금이 적용됩니다. Amazon SNS 결제에 대한 자세한 내용은 [Amazon Simple Notification Service 요금](#)을 참조하십시오.

Amazon RDS 이벤트 알림 서비스를 구독하는 프로세스는 다음과 같습니다.

1. Amazon RDS 콘솔, AWS CLI 또는 API를 사용하여 Amazon RDS 이벤트 알림 서비스 구독을 생성합니다.
2. Amazon RDS가 구독 생성 시 제출한 이메일 주소로 승인 이메일 또는 SMS 메시지를 전송합니다. 구독 여부를 확인하려면 전송된 알림의 링크를 선택합니다.

3. 구독 여부를 확인하면 Amazon RDS 콘솔의 [My Event Subscriptions] 영역에 구독 상태가 업데이트됩니다.
4. 이제부터 이벤트 알림 메시지가 수신됩니다.

Note

Amazon SNS이 구독된 HTTP 또는 HTTPS 엔드포인트에 알림을 전송할 때 엔드포인트에 전송된 POST 메시지에는 JSON 문서를 포함하는 메시지 본문이 있습니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 [Amazon SNS 메시지 및 JSON 형식](#)을 참조하십시오.

다음 섹션부터는 수신되는 모든 카테고리와 이벤트에 대해 살펴보겠습니다. 또한 Amazon RDS 이벤트 구독 및 구독 작업에 대한 정보도 제공합니다.

Amazon RDS 이벤트 카테고리 및 이벤트 메시지

Amazon RDS는 Amazon RDS 콘솔, AWS CLI 또는 API를 사용하여 구독할 수 있는 이벤트 카테고리가 매우 많습니다. 이러한 카테고리는 각각 DB 인스턴스, DB 스냅샷, DB 보안 그룹 또는 DB 파라미터 그룹 등 다양한 소스 유형에 적용됩니다.

다음 표는 DB 인스턴스가 소스 유형일 때 이벤트 카테고리와 이벤트 목록을 나타냅니다.

Category	Amazon RDS 이벤트 ID	설명
가용성	RDS-EVENT-0006	재시작된 DB 인스턴스.
가용성	RDS-EVENT-0004	DB 인스턴스 종료.
가용성	RDS-EVENT-0022	MySQL 또는 MariaDB 재시작 중 오류가 발생했습니다.
backup	RDS-EVENT-0001	DB 인스턴스 백업.
backup	RDS-EVENT-0002	완료된 DB 인스턴스 백업.
구성 변경	RDS-EVENT-0009	DB 인스턴스가 보안 그룹에 추가되었습니다.
구성 변경	RDS-EVENT-0024	DB 인스턴스가 다른 AZ DB 인스턴스로 전환 중입니다.
구성 변경	RDS-EVENT-0030	DB 인스턴스가 단일 AZ DB 인스턴스로 전환 중입니다.
구성 변경	RDS-EVENT-0012	수정을 데이터베이스 인스턴스 클래스에 적용.
구성 변경	RDS-EVENT-0018	현재 이 DB 인스턴스의 스토리지 설정이 변경 중입니다.
구성 변경	RDS-EVENT-0011	이 DB 인스턴스의 파라미터 그룹이 변경되었습니다.
구성 변경	RDS-EVENT-0092	이 DB 인스턴스의 파라미터 그룹 업데이트가 완료되었습니다.
구성 변경	RDS-EVENT-0028	이 DB 인스턴스의 자동 백업이 비활성화되었습니다.
구성 변경	RDS-EVENT-0032	이 DB 인스턴스의 자동 백업이 활성화되었습니다.
구성 변경	RDS-EVENT-0033	마스터 사용자 이름과 일치하는 사용자가 [숫자]명 있습니다. 특정 호스트에 연결되지 않은 사용자가 재설정되었습니다.
구성 변경	RDS-EVENT-0025	DB 인스턴스가 다른 AZ DB 인스턴스로 전환되었습니다.

Category	Amazon RDS 이벤트 ID	설명
구성 변경	RDS-EVENT-0029	DB 인스턴스가 단일 AZ DB 인스턴스로 전환되었습니다.
구성 변경	RDS-EVENT-0014	이 DB 인스턴스의 클래스가 변경되었습니다.
구성 변경	RDS-EVENT-0017	이 DB 인스턴스의 스토리지 설정이 변경되었습니다.
구성 변경	RDS-EVENT-0010	DB 인스턴스가 보안 그룹에서 제거되었습니다.
구성 변경	RDS-EVENT-0016	DB 인스턴스의 마스터 암호가 재설정되었습니다.
구성 변경	RDS-EVENT-0067	DB 인스턴스의 마스터 암호 재설정 시도가 실패했습니다.
구성 변경	RDS-EVENT-0078	Enhanced Monitoring 구성이 변경되었습니다.
생성	RDS-EVENT-0005	생성된 DB 인스턴스.
삭제	RDS-EVENT-0003	DB 인스턴스가 삭제되었습니다.
장애 조치	RDS-EVENT-0034	최근에 DB 인스턴스에 장애 조치가 발생하였기 때문에 Amazon RDS가 요청한 장애 조치를 실행하지 않습니다.
장애 조치	RDS-EVENT-0013	예비 인스턴스의 승격 원인이었던 다른 AZ 장애 조치가 시작되었습니다.
장애 조치	RDS-EVENT-0015	예비 인스턴스의 승격 원인이었던 다른 AZ 장애 조치가 완료되었습니다. DNS가 새로운 기본 DB 인스턴스로 이전하는 데 몇 분 걸릴 수 있습니다.
장애 조치	RDS-EVENT-0065	인스턴스가 부분적 장애 조치에서 복구되었습니다.
장애 조치	RDS-EVENT-0049	다른 AZ 장애 조치가 완료되었습니다.
장애 조치	RDS-EVENT-0050	성공적인 인스턴스 복구 후 다른 AZ 활성화가 시작되었습니다.
장애 조치	RDS-EVENT-0051	다른 AZ 활성화가 완료되었습니다. 이제 데이터베이스에 액세스할 수 있습니다.
결함	RDS-EVENT-0031	호환되지 않는 구성 또는 기본 스토리지 문제로 인해 DB 인스턴스에 장애가 발생했습니다. DB 인스턴스에 대해 특정 시점으로 복구를 시작합니다.
결함	RDS-EVENT-0036	DB 인스턴스가 호환되지 않는 네트워크에 있습니다. 특정 서브넷 ID 중 일부가 잘못되었거나 존재하지 않습니다.
결함	RDS-EVENT-0035	DB 인스턴스에 잘못된 파라미터가 있습니다. 예를 들어 이 인스턴스 클래스의 메모리 관련 파라미터가 너무 높게 설정되어 MySQL이 시작하지 않습니다. 따라서 메모리 파라미터 설정을 변경한 후 DB 인스턴스를 재부팅하는 것이 바람직합니다.
결함	RDS-EVENT-0058	Statspack 사용자 계정인 PERFSTAT 생성 중 오류가 발생하였습니다. Statspack 옵션을 추가하기 전에 해당 계정을 삭제하십시오.

Category	Amazon RDS 이벤트 ID	설명
결함	RDS-EVENT-0079	Enhanced Monitoring을 활성화하려면 Enhanced Monitoring IAM 역할이 있어야 합니다. Enhanced Monitoring IAM 역할을 생성하는 방법에 대한 자세한 내용은 Amazon RDS Enhanced Monitoring에 대한 IAM 역할을 생성하려면 (p. 239) 단원을 참조하십시오.
결함	RDS-EVENT-0080	구성을 변경하는 동안 오류가 발생하여 Enhanced Monitoring이 비활성화되었습니다. Enhanced Monitoring IAM 역할이 잘못 구성된 것 같습니다. Enhanced Monitoring IAM 역할을 생성하는 방법에 대한 자세한 내용은 Amazon RDS Enhanced Monitoring에 대한 IAM 역할을 생성하려면 (p. 239) 단원을 참조하십시오.
결함	RDS-EVENT-0081	SQL Server 기본 백업 및 복원을 위해 Amazon S3 버킷에 액세스하는데 사용하는 IAM 역할이 잘못 구성되었습니다. 자세한 내용은 기본 백업 및 복원 설정 (p. 521) 단원을 참조하십시오.
적은 스토리지	RDS-EVENT-0089	DB 인스턴스가 할당된 스토리지의 90% 이상을 사용하였습니다. [Free Storage Space] 측정치를 사용하여 DB 인스턴스에 대한 스토리지 공간을 모니터링할 수 있습니다. 자세한 내용은 DB 인스턴스 측정치 보기 (p. 236) 단원을 참조하십시오.
적은 스토리지	RDS-EVENT-0007	DB 인스턴스에 할당된 스토리지를 모두 사용하였습니다. 이 문제를 해결하려면 DB 인스턴스에 스토리지를 추가 할당해야 합니다. 자세한 내용은 RDS FAQ 단원을 참조하십시오. [Free Storage Space] 측정치를 사용하여 DB 인스턴스에 대한 스토리지 공간을 모니터링할 수 있습니다. 자세한 내용은 DB 인스턴스 측정치 보기 (p. 236) 단원을 참조하십시오.
유지 관리	RDS-EVENT-0026	DB 인스턴스의 오프라인 유지 관리가 진행 중입니다. 따라서 현재 DB 인스턴스는 사용할 수 없습니다.
유지 관리	RDS-EVENT-0027	DB 인스턴스의 오프라인 유지 관리가 완료되었습니다. 이제 DB 인스턴스를 사용할 수 있습니다.
유지 관리	RDS-EVENT-0155	DB 인스턴스에 사용 가능한 DB 엔진 마이너 버전 업그레이드가 있습니다.
알림	RDS-EVENT-0044	연산자 관련 알림 메시지입니다. 자세한 내용은 이벤트 메시지 단원을 참조하십시오.
알림	RDS-EVENT-0047	DB 인스턴스의 패치 작업이 완료되었습니다.
알림	RDS-EVENT-0048	DB 인스턴스의 패치 작업이 지연되었습니다.
알림	RDS-EVENT-0054	사용 중인 MySQL 스토리지 엔진이 InnoDB가 아닙니다. Amazon RDS는 MySQL 스토리지 엔진으로 InnoDB를 권장합니다. MySQL 스토리지 엔진에 대한 자세한 내용은 Amazon RDS MySQL에 대해 지원되는 스토리지 엔진 을 참조하십시오.

Category	Amazon RDS 이벤트 ID	설명
알림	RDS-EVENT-0055	<p>DB 인스턴스의 테이블 수가 Amazon RDS의 권장 모범 사례를 초과하였습니다. DB 인스턴스의 테이블 수를 줄이십시오.</p> <p>권장 모범 사례에 대한 자세한 내용은 Amazon RDS 기본 운영 지침 (p. 66) 단원을 참조하십시오.</p>
알림	RDS-EVENT-0056	<p>DB 인스턴스의 데이터베이스 수가 Amazon RDS의 권장 모범 사례를 초과하였습니다. DB 인스턴스의 데이터베이스 수를 줄이십시오.</p> <p>권장 모범 사례에 대한 자세한 내용은 Amazon RDS 기본 운영 지침 (p. 66) 단원을 참조하십시오.</p>
알림	RDS-EVENT-0064	<p>TDE 키가 교체되었습니다. 권장 모범 사례에 대한 자세한 내용은 Amazon RDS 기본 운영 지침 (p. 66) 단원을 참조하십시오.</p>
알림	RDS-EVENT-0084	<p>DB 인스턴스를 다중 AZ로 전환하려고 시도하였으나 다중 AZ를 지원하지 않는 인 메모리 파일 그룹이 포함되어 있습니다. 자세한 내용은 Microsoft SQL Server의 다중 AZ 배포 (p. 537) 단원을 참조하십시오.</p>
알림	RDS-EVENT-0157	<p>대상 인스턴스 클래스는 원본 DB 인스턴스에 있는 데이터베이스의 수를 지원할 수 없기 때문에 RDS는 DB 인스턴스 클래스를 수정할 수 없습니다. “인스턴스에 N 데이터베이스가 있지만, 변환 후에는 N만 지원할 것입니다.”라는 오류 메시지가 표시됩니다.</p> <p>자세한 내용은 Microsoft SQL Server DB 인스턴스에 대한 제한 (p. 479) 단원을 참조하십시오.</p>
알림	RDS-EVENT-0087	DB 인스턴스가 종단되었습니다.
알림	RDS-EVENT-0088	DB 인스턴스가 시작되었습니다.
알림	RDS-EVENT-0154	DB 인스턴스가 종지 최대 허용 시간 초과로 인해 시작종입니다.
읽기 전용 복제본	RDS-EVENT-0045	<p>읽기 전용 복제 프로세스에서 오류가 발생하였습니다. 자세한 내용은 이벤트 메시지 단원을 참조하십시오.</p> <p>읽기 전용 복제본 오류의 문제 해결을 위한 자세한 내용은 MySQL 읽기 전용 복제본의 문제 해결 (p. 649) 단원을 참조하십시오.</p>
읽기 전용 복제본	RDS-EVENT-0046	읽기 전용 복제본이 복제를 재개하였습니다. 이 메시지는 읽기 전용 복제본을 처음 생성할 때, 혹은 복제 기능의 정상 여부를 확인하는 모니터링 메시지로서 나타납니다. RDS-EVENT-0045 알림 메시지 후에 이 메시지가 표시되면 오류 이후, 또는 복제가 종단되었다가 다시 시작된 것입니다.
읽기 전용 복제본	RDS-EVENT-0057	읽기 전용 복제본의 복제가 종료되었습니다.
읽기 전용 복제본	RDS-EVENT-0062	읽기 전용 복제본의 복제가 수동으로 종단되었습니다.
읽기 전용 복제본	RDS-EVENT-0063	읽기 전용 복제본의 복제가 재설정되었습니다.

Category	Amazon RDS 이벤트 ID	설명
복구	RDS-EVENT-0020	DB 인스턴스 복구가 시작되었습니다. 복구 시간은 복구 할 데이터 용량에 따라 달라집니다.
복구	RDS-EVENT-0021	DB 인스턴스 복구가 완료되었습니다.
복구	RDS-EVENT-0023	수동 백업을 요청하였지만 Amazon RDS가 현재 DB 스냅샷을 생성 중입니다. 따라서 Amazon RDS가 DB 스냅샷 생성을 완료한 후에 다시 요청하십시오.
복구	RDS-EVENT-0052	다중 AZ 인스턴스 복구가 시작되었습니다. 복구 시간은 복구 할 데이터 용량에 따라 달라집니다.
복구	RDS-EVENT-0053	다중 AZ 인스턴스 복구가 완료되었습니다.
복구	RDS-EVENT-0066	SQL Server DB 인스턴스가 미러를 재구성 중입니다. 이 때 미러가 재구성될 때까지 성능이 저하됩니다. 복구 모델이 FULL이 아닌 데이터베이스가 발견되었습니다. 복구 모델이 FULL로 다시 변경된 후 미러링 복구가 시작되었습니다(<dbname>: <recovery model found>[,...])
복원	RDS-EVENT-0008	DB 인스턴스가 DB 스냅샷에서 복원되었습니다.
복원	RDS-EVENT-0019	DB 인스턴스가 특정 시점으로 백업에서 복원되었습니다.

다음 표는 DB 파라미터 그룹이 소스 유형일 때 이벤트 카테고리와 이벤트 목록을 나타냅니다.

Category	RDS 이벤트 ID	설명
구성 변경	RDS-EVENT-0037	파라미터 그룹 설정이 변경되었습니다.

다음 표는 DB 보안 그룹이 소스 유형일 때 이벤트 카테고리와 이벤트 목록을 나타냅니다.

Category	RDS 이벤트 ID	설명
구성 변경	RDS-EVENT-0038	보안 그룹 설정이 변경되었습니다.
결합	RDS-EVENT-0039	[사용자]가 소유하고 있는 Amazon EC2 보안 그룹은 존재하지 않습니다. 보안 그룹에 대한 권한 부여가 취소되었습니다.

다음 표는 DB 스냅샷이 소스 유형일 때 이벤트 카테고리와 이벤트 목록을 나타냅니다.

Category	RDS 이벤트 ID	설명
생성	RDS-EVENT-0040	수동 DB 스냅샷이 생성 중입니다.
삭제	RDS-EVENT-0041	DB 스냅샷이 삭제되었습니다.
생성	RDS-EVENT-0042	수동 DB 스냅샷이 생성되었습니다.
복원	RDS-EVENT-0043	DB 인스턴스가 DB 스냅샷에서 복원 중입니다.

Category	RDS 이벤트 ID	설명
알림	RDS-EVENT-0059	원본 리전[리전 이름]에서 리전 간 DB 스냅샷[DB 스냅샷 이름]의 복사가 시작되었습니다.
알림	RDS-EVENT-0060	원본 리전[리전 이름]에서 리전 간 DB 스냅샷[DB 스냅샷 이름] 복사가 [시간]분이 걸려 완료되었습니다.
알림	RDS-EVENT-0061	리전 간 DB 스냅샷 복사가 실패하였습니다.
생성	RDS-EVENT-0090	자동화된 DB 스냅샷이 생성 중입니다.
생성	RDS-EVENT-0091	자동화된 DB 스냅샷이 생성되었습니다.

다음 표에는 Aurora DB 클러스터가 원본 유형일 때 이벤트 카테고리와 이벤트 목록이 나와 있습니다.

Category	RDS 이벤트 ID	설명
장애 조치	RDS-EVENT-0069	DB 클러스터에 대한 장애 조치가 실패했습니다.
장애 조치	RDS-EVENT-0070	DB 클러스터에 대한 장애 조치가 다시 시작되었습니다.
장애 조치	RDS-EVENT-0071	DB 클러스터에 대한 장애 조치를 마쳤습니다.
장애 조치	RDS-EVENT-0072	DB 클러스터에 대한 장애 조치가 동일한 가용 영역 내에서 시작되었습니다.
장애 조치	RDS-EVENT-0073	DB 클러스터에 대한 장애 조치가 가용 영역 전체에서 시작되었습니다.
결합	RDS-EVENT-0083	Aurora가 Amazon S3 버킷에서 백업 데이터를 복사할 수 없습니다. Amazon S3 버킷에 액세스하기 위한 Aurora의 권한이 잘못 구성된 것 같습니다. 자세한 내용은 외부 MySQL 데이터베이스에서 Amazon Aurora MySQL DB 클러스터로 데이터 마이그레이션 단원 을 참조하십시오.
유지 관리	RDS-EVENT-0156	DB 클러스터에 사용 가능한 DB 엔진 마이너 버전 업그레이드가 있습니다.
알림	RDS-EVENT-0076	Aurora DB 클러스터로 마이그레이션하지 못했습니다.
알림	RDS-EVENT-0077	Aurora DB 클러스터로 마이그레이션하는 중에 원본 데이터베이스의 테이블을 InnoDB로 변환하지 못했습니다.
알림	RDS-EVENT-0149	조정 지점이 없습니다.
알림	RDS-EVENT-0150	DB 클러스터가 종지되었습니다.
알림	RDS-EVENT-0151	DB 클러스터가 시작되었습니다.
알림	RDS-EVENT-0152	DB 클러스터를 종지하지 못했습니다.
알림	RDS-EVENT-0153	DB 클러스터가 종지 최대 허용 시간 초과로 인해 시작종입니다.

다음 표에는 Aurora DB 클러스터 스냅샷이 원본 유형일 때 이벤트 카테고리와 이벤트 목록이 나와 있습니다.

Category	RDS 이벤트 ID	설명
backup	RDS-EVENT-0074	수동 DB 클러스터 스냅샷 생성이 시작되었습니다.
backup	RDS-EVENT-0075	수동 DB 클러스터 스냅샷이 생성되었습니다.

Amazon RDS 이벤트 알림 구독

임의의 DB 인스턴스, DB 스냅샷, DB 보안 그룹 또는 DB 파라미터 그룹에 대한 이벤트 발생 여부를 알 수 있도록 Amazon RDS 알림 구독을 생성할 수 있습니다. 가장 간단한 구독 생성 방법은 RDS 콘솔입니다. CLI 또는 API를 사용하여 이벤트 알림 구독을 생성하려면 먼저 Amazon Simple Notification Service 주제를 만든 후 Amazon SNS 콘솔이나 Amazon SNS API를 통해 해당 주제를 구독해야 합니다. 또한 CLI 명령이나 API 작업을 제출할 때도 사용되기 때문에 해당 주제의 Amazon 리소스 이름(ARN)을 잊어서는 안 됩니다. SNS 주제를 새로 만들어 구독하는 방법에 대한 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 [Amazon SNS 시작하기](#) 단원을 참조하십시오.

알림 메시지를 받고 싶은 소스 유형과 이벤트를 트리거링하는 Amazon RDS 소스를 지정할 수 있습니다. 이 두 가지는 [SourceType](소스 유형)과 [SourceIdentifier](이벤트를 발생시키는 Amazon RDS 소스)에서 지정됩니다. SourceType 및 SourceIdentifier(예를 들면 SourceType = db-instance 및 SourceIdentifier = myDBInstance1)를 둘 다 지정하는 경우 지정된 원본에 대한 모든 DB 인스턴스 이벤트가 수신됩니다. 하지만 SourceType만 지정하고 SourceIdentifier를 지정하지 않으면 모든 Amazon RDS 원본 중 해당 원본 유형의 이벤트만 알림 메시지로 받게 됩니다. SourceType 또는 SourceIdentifier를 지정하지 않으면, 고객 계정에 속하는 모든 Amazon RDS 원본에서 생성된 이벤트에 대한 알림을 수신합니다.

Note

이벤트 알림을 전달하는 데 최대 5분 정도 걸릴 수 있습니다.

AWS Management 콘솔

RDS 이벤트 알림 구독 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 이벤트 구독을 선택합니다.
3. [Event subscriptions] 창에서 [Create event subscription]을 선택합니다.
4. [Create event subscription] 대화 상자에서 다음과 같이 실행합니다.
 - a. 이름에서 이벤트 알림 구독 이름을 입력합니다.
 - b. 알림 받을 대상에서 Amazon SNS 주제의 기존 Amazon SNS ARN을 선택하거나 주제 생성을 선택하여 주제와 수신자 목록의 이름을 입력합니다.
 - c. [Source type]에서 원본 형식을 선택합니다.
 - d. [Yes]를 선택하여 구독을 활성화합니다. 구독만 생성하고 알림 메시지 전송은 아직 원하지 않을 경우에는 [No]를 선택합니다.
 - e. 선택한 소스 유형에 따라 이벤트 알림 메시지를 수신하고자 하는 이벤트 카테고리와 소스를 선택합니다.
 - f. 생성을 선택합니다.

Amazon RDS 콘솔에 현재 구독 생성 중으로 나옵니다.

Event subscriptions (2)			
Name	Status	Source Type	Enabled
Configchangerdpgres	active	Instances	Yes
Test	creating	Instances	Yes

CLI

RDS 이벤트 알림을 구독하려면 AWS CLI [create-event-subscription](#) 명령을 사용합니다. 다음 필수 파라미터를 포함합니다.

- `--subscription-name`
- `--sns-topic-arn`

Example

Linux, OS X, Unix의 경우:

```
aws rds create-event-subscription \
  --subscription-name myeventsubscription \
  --sns-topic-arn arn:aws:sns:us-east-1:802#####:myawsuser-RDS \
  --enabled
```

Windows의 경우:

```
aws rds create-event-subscription ^
  --subscription-name myeventsubscription ^
  --sns-topic-arn arn:aws:sns:us-east-1:802#####:myawsuser-RDS ^
  --enabled
```

API

Amazon RDS 이벤트 알림을 구독하려면 Amazon RDS API 함수 [CreateEventSubscription](#)을 호출합니다. 다음 필수 파라미터를 포함합니다.

- `SubscriptionName`
- `SnsTopicArn`

Amazon RDS 이벤트 알림 구독의 목록 표시

현재 Amazon RDS 이벤트 알림 구독을 목록으로 표시할 수 있습니다.

AWS Management 콘솔

현재 Amazon RDS 이벤트 알림 구독을 목록으로 표시하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Event subscriptions]를 선택합니다. [Event subscriptions] 창에 이벤트 알림 구독이 모두 표시됩니다.

Event subscriptions (2)		
	Name	Status
<input type="checkbox"/>	Configchangerdpgres	active
<input type="checkbox"/>	Postgresnotification	active

CLI

현재 Amazon RDS 이벤트 알림 구독을 표시하려면 AWS CLI [describe-event-subscriptions](#) 명령을 사용합니다.

Example

다음은 모든 이벤트 구독을 설명하는 예제입니다.

```
aws rds describe-event-subscriptions
```

다음은 myfirsteventsubscription을 설명하는 예제입니다.

```
aws rds describe-event-subscriptions --subscription-name myfirsteventsubscription
```

API

현재 Amazon RDS 이벤트 알림 구독을 표시하려면 Amazon RDS API [DescribeEventSubscriptions](#) 작업을 사용합니다.

Example

다음은 최대 100개의 이벤트 구독을 나열하는 코드 예제입니다.

```
https://rds.us-east-1.amazonaws.com/  
?Action=DescribeEventSubscriptions  
&MaxRecords=100  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4  
&Version=2014-09-01  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Credential=AKIADQKE4SARGYLE/20140428/us-east-1/rds/aws4_request  
&X-Amz-Date=20140428T161907Z  
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date  
&X-Amz-Signature=4208679fe967783a1a149c826199080a066085d5a88227a80c6c0cadb3e8c0d4
```

다음은 myfirsteventsubscription을 설명하는 예제입니다.

```
https://rds.us-east-1.amazonaws.com/  
?Action=DescribeEventSubscriptions
```

```
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&SubscriptionName=myfirsteventsubscription
&Version=2014-09-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140428/us-east-1/rds/aws4_request
&X-Amz-Date=20140428T161907Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=4208679fe967783a1a149c826199080a066085d5a88227a80c6c0cadb3e8c0d4
```

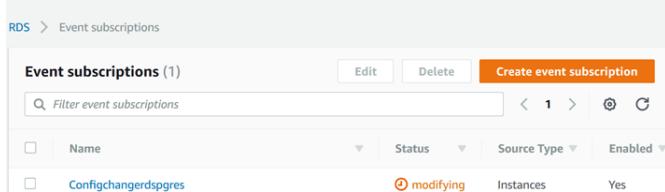
Amazon RDS 이벤트 알림 구독 변경

구독을 생성한 후에는 구독 이름, 소스 식별자, 카테고리 또는 주제 ARN을 변경할 수 있습니다.

AWS Management 콘솔

Amazon RDS 이벤트 알림 구독을 변경하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Event subscriptions]를 선택합니다.
3. [Event subscriptions] 창에서 수정할 구독을 선택한 다음 [Edit]를 선택합니다.
4. 대상 또는 원본 섹션에서 구독을 변경합니다.
5. [Edit]를 선택합니다. Amazon RDS 콘솔에 현재 구독 변경 중으로 나옵니다.



CLI

Amazon RDS 이벤트 알림 구독을 수정하려면, AWS CLI `modify-event-subscription` 명령을 사용합니다. 다음 필수 파라미터를 포함합니다.

- `--subscription-name`

Example

다음 코드를 사용하여 `myeventsSubscription`을 사용할 수 있습니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-event-subscription \
--subscription-name myeventsSubscription \
--enabled
```

Windows의 경우:

```
aws rds modify-event-subscription ^
--subscription-name myeventsSubscription ^
```

--enabled

API

Amazon RDS 이벤트를 수정하려면 Amazon RDS API 작업 [ModifyEventSubscription](#)을 호출합니다. 다음 필수 파라미터를 포함합니다.

- `SubscriptionName`

Amazon RDS 이벤트 알림 구독에 대한 소스 식별자 추가

소스 식별자(이벤트를 발생시키는 Amazon RDS 소스)를 기준 구독에 추가할 수 있습니다.

AWS Management 콘솔

소스 식별자는 Amazon RDS 콘솔에서 구독 관련 설정을 변경하면서 선택 또는 선택 해제를 통해 쉽게 추가하거나 제거할 수 있습니다. 자세한 내용은 [Amazon RDS 이벤트 알림 구독 변경 \(p. 301\)](#) 단원을 참조하십시오.

CLI

Amazon RDS 이벤트 알림 구독에 소스 식별자를 추가하려면 AWS CLI [add-source-identifier-to-subscription](#) 명령을 사용합니다. 다음 필수 파라미터를 포함합니다.

- `--subscription-name`
- `--source-identifier`

Example

다음 예제에서는 `myrdseventssubscription` 구독에 소스 식별자 `mysqldb`를 추가합니다.

Linux, OS X, Unix의 경우:

```
aws rds add-source-identifier-to-subscription \
--subscription-name myrdseventssubscription \
--source-identifier mysqldb
```

Windows의 경우:

```
aws rds add-source-identifier-to-subscription ^
--subscription-name myrdseventssubscription ^
--source-identifier mysqldb
```

API

Amazon RDS 이벤트 알림 구독에 소스 식별자를 추가하려면 Amazon RDS API [AddSourceIdentifierToSubscription](#)을 호출합니다. 다음 필수 파라미터를 포함합니다.

- `SubscriptionName`
- `SourceIdentifier`

Amazon RDS 이벤트 알림 구독의 소스 식별자 제거

임의 소스의 이벤트 알림 메시지를 더 이상 받고 싶지 않을 때는 소스 식별자(이벤트를 발생시키는 Amazon RDS 소스)를 구독에서 제거할 수 있습니다.

AWS Management 콘솔

소스 식별자는 Amazon RDS 콘솔에서 구독 관련 설정을 변경하면서 선택 또는 선택 해제를 통해 쉽게 추가하거나 제거할 수 있습니다. 자세한 내용은 [Amazon RDS 이벤트 알림 구독 변경 \(p. 301\)](#) 단원을 참조하십시오.

CLI

Amazon RDS 이벤트 알림 구독에서 소스 식별자를 제거하려면 AWS CLI `remove-source-identifier-from-subscription` 명령을 사용합니다. 다음 필수 파라미터를 포함합니다.

- `--subscription-name`
- `--source-identifier`

Example

다음 예제에서는 `myrdseventsSubscription` 구독에서 소스 식별자 `mysqldb`를 제거합니다.

Linux, OS X, Unix의 경우:

```
aws rds remove-source-identifier-from-subscription \
--subscription-name myrdseventsSubscription \
--source-identifier mysqldb
```

Windows의 경우:

```
aws rds remove-source-identifier-from-subscription ^
--subscription-name myrdseventsSubscription ^
--source-identifier mysqldb
```

API

Amazon RDS 이벤트 알림 구독에서 소스 식별자를 제거하려면 Amazon RDS API `RemoveSourceIdentifierFromSubscription` 명령을 사용합니다. 다음 필수 파라미터를 포함합니다.

- `SubscriptionName`
- `SourceIdentifier`

Amazon RDS 이벤트 알림 카테고리의 목록 표시

리소스 유형의 이벤트는 모두 여러 카테고리로 그룹화됩니다. 이용 가능한 카테고리 목록을 보려면 다음 절차를 따릅니다.

AWS Management 콘솔

이벤트 알림 구독을 생성 또는 변경할 때는 이벤트 카테고리가 Amazon RDS 콘솔에 표시됩니다. 자세한 내용은 [Amazon RDS 이벤트 알림 구독 변경 \(p. 301\)](#) 단원을 참조하십시오.

Source type
Source type of resource this subscription will consume event from

Instances

Instances to include
Instances that this subscription will consume events from

configuration change

creation

deletion

failover

failure

low storage

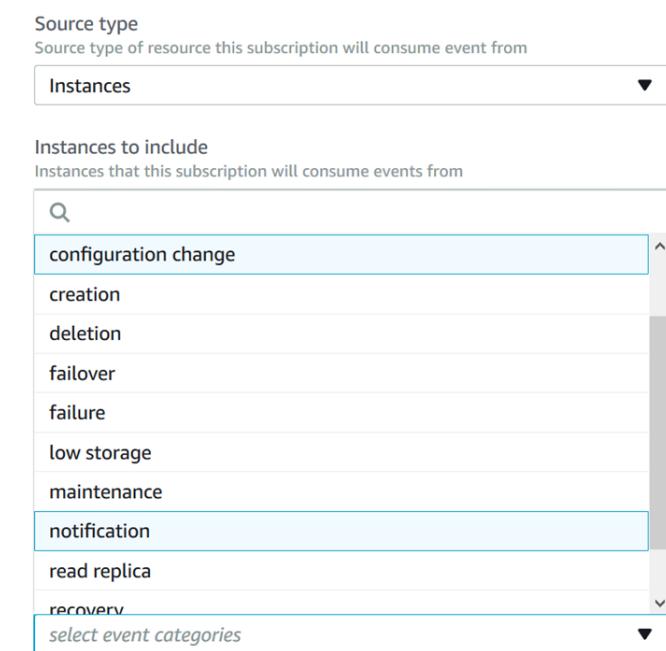
maintenance

notification

read replica

recovery

select event categories



CLI

Amazon RDS 이벤트 알림 범주를 표시하려면 AWS CLI [describe-event-categories](#) 명령을 사용합니다. 이 명령에는 필수 파라미터가 없습니다.

Example

```
aws rds describe-event-categories
```

API

Amazon RDS 이벤트 알림 범주를 표시하려면 Amazon RDS API [DescribeEventCategories](#) 명령을 사용합니다. 이 명령에는 필수 파라미터가 없습니다.

Amazon RDS 이벤트 알림 구독 삭제

필요 없는 구독은 삭제할 수 있습니다. 그러면 해당 주제의 모든 구독자에게는 구독 시 지정한 이벤트 알림 메시지가 발송되지 않습니다.

AWS Management 콘솔

Amazon RDS 이벤트 알림 구독을 삭제하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [DB Event Subscriptions]를 선택합니다.
3. [My DB Event Subscriptions] 창에서 삭제할 구독을 선택합니다.
4. 삭제를 선택합니다.
5. Amazon RDS 콘솔에 현재 구독 삭제 중으로 나옵니다.

The screenshot shows the 'Event subscriptions' section with a count of 2. There are three buttons at the top: 'Edit', 'Delete' (which is circled in red), and 'Create event subscription'. Below is a search bar and a navigation bar with icons for back, forward, and refresh. A table lists two entries:

	Name	Status
<input type="checkbox"/>	Configchangerdpgres	active
<input type="checkbox"/>	Postgresnotification	active

CLI

Amazon RDS 이벤트 알림 구독을 삭제하려면, AWS CLI [delete-event-subscription](#) 명령을 사용합니다. 다음 필수 파라미터를 포함합니다.

- `--subscription-name`

Example

다음은 `myrdssubscription` 구독을 삭제하는 예제입니다.

```
aws rds delete-event-subscription --subscription-name myrdssubscription
```

API

Amazon RDS 이벤트 알림 구독을 삭제하려면, RDS API [DeleteEventSubscription](#) 명령을 사용합니다. 다음 필수 파라미터를 포함합니다.

- `SubscriptionName`

Amazon RDS 이벤트 보기

Amazon RDS에서는 DB 인스턴스, DB 스냅샷, DB 보안 그룹 및 DB 파라미터 그룹과 관련된 이벤트 레코드를 유지합니다. 여기에는 이벤트 날짜 및 시간, 이벤트의 원본 이름 및 유형, 이벤트 관련 메시지 등의 정보가 포함됩니다.

지난 24시간 동안 발생한 이벤트를 표시하는 AWS Management 콘솔을 통해 RDS 리소스에 대한 이벤트를 검색할 수 있습니다. [describe-events](#) AWS CLI 명령 또는 [DescribeEvents](#) RDS API 작업을 사용하여 RDS 리소스에 대한 이벤트를 검색할 수도 있습니다. AWS CLI 또는 RDS API를 사용하여 이벤트를 볼 경우 최대 지난 14일 동안 발생한 이벤트를 검색할 수 있습니다.

AWS Management 콘솔

지난 24시간 동안 발생한 모든 Amazon RDS 인스턴스 이벤트를 보려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Events]를 선택합니다. 사용 가능한 이벤트가 목록에 표시됩니다.

3. [Filter] 목록을 사용하여 이벤트를 유형별로 필터링하고 [Filter] 목록 오른쪽에 있는 텍스트 상자를 사용하여 결과를 세부적으로 필터링할 수 있습니다. 예를 들어 다음 스크린샷에서는 DB 인스턴스 이벤트 유형별로 필터링되고 문자 **1318**을 포함된 이벤트 목록을 보여 줍니다.

The screenshot shows a search bar at the top containing '1318'. Below it is a table with two rows. The first row has 'Identifier' as 'feb1318' and 'Type' as 'DB cluster snapshots'. The second row also has 'Identifier' as 'feb1318' and 'Type' as 'DB cluster snapshots'. There are navigation controls like '< 1 >' and a refresh icon at the top right of the table area.

Identifier	Type
feb1318	DB cluster snapshots
feb1318	DB cluster snapshots

CLI

지난 7일 동안 발생한 모든 Amazon RDS 인스턴스 이벤트를 보려면

`describe-events` AWS CLI 명령을 호출하고 --duration 파라미터를 10080으로 설정하여 지난 7일 동안 발생한 모든 Amazon RDS 인스턴스 이벤트를 볼 수 있습니다.

```
aws rds describe-events --duration 10080
```

API

지난 14일 동안 발생한 모든 Amazon RDS 인스턴스 이벤트를 보려면

`DescribeEvents` RDS API 작업을 호출하고 Duration 파라미터를 20160으로 설정하여 지난 14일 동안 발생한 모든 Amazon RDS 인스턴스 이벤트를 볼 수 있습니다.

```
https://rds.us-west-2.amazonaws.com/
?Action=DescribeEvents
&Duration=20160
&MaxRecords=100
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-09-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140421/us-west-2/rds/aws4_request
&X-Amz-Date=20140421T194733Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=8e313cabcd9766c56a2886b5b298fd944e0b7cfa248953c82705fdd0374f27
```

Amazon RDS 데이터베이스 로그 파일

Amazon RDS 콘솔, AWS Command Line Interface(AWS CLI) 또는 Amazon RDS API를 사용하여 데이터베이스 로그를 보고 다운로드하고 조사할 수 있습니다. 트랜잭션 로그 보기, 다운로드 또는 조사는 지원되지 않습니다.

엔진별 정보는 다음을 참조하십시오.

- MariaDB 데이터베이스 로그 파일 (p. 310)
- Microsoft SQL Server 데이터베이스 로그 파일 (p. 318)
- MySQL 데이터베이스 로그 파일 (p. 318)
- Oracle 데이터베이스 로그 파일 (p. 326)
- PostgreSQL 데이터베이스 로그 파일 (p. 331)

데이터베이스 로그 파일 보기 및 나열

Amazon RDS 콘솔을 사용하여 DB 엔진에 대한 데이터베이스 로그 파일을 볼 수 있습니다. AWS CLI 또는 Amazon RDS API를 사용하여 다운로드하거나 모니터링할 수 있는 로그 파일을 나열할 수 있습니다.

Note

기존 Oracle DB 인스턴스에 대한 로그 파일 목록을 볼 수 없는 경우 목록을 보려면 인스턴스를 재부팅합니다.

AWS Management 콘솔

데이터베이스 로그 파일을 보려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 보고자 하는 로그 파일을 보유한 DB 인스턴스의 이름을 선택합니다.
4. 로그 및 이벤트 탭을 선택합니다.
5. 아래로 스크롤하여 [Logs] 섹션을 찾습니다.
6. 로그 섹션에서 표시할 로그를 선택한 다음 보기를 선택합니다.

AWS CLI

DB 인스턴스에 사용 가능한 데이터베이스 로그 파일을 나열하려면 AWS CLI `describe-db-log-files` 명령을 사용합니다.

다음 예에서는 `my-db-instance`라는 DB 인스턴스에 대한 로그 파일 목록을 반환합니다.

Example

```
aws rds describe-db-log-files --db-instance-identifier my-db-instance
```

API

DB 인스턴스에 사용 가능한 데이터베이스 로그 파일을 나열하려면 Amazon RDS API `DescribeDBLogFiles` 작업을 사용합니다.

데이터베이스 로그 파일 다운로드

Amazon RDS 콘솔, AWS CLI 또는 API를 사용하여 데이터베이스 로그 파일을 다운로드할 수 있습니다.

AWS Management 콘솔

데이터베이스 로그 파일을 다운로드하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.

2. 탐색 창에서 데이터베이스를 선택합니다.
3. 보고자 하는 로그 파일을 보유한 DB 인스턴스의 이름을 선택합니다.
4. 로그 및 이벤트 탭을 선택합니다.
5. 아래로 스크롤하여 [Logs] 섹션을 찾습니다.
6. 로그 섹션에서 다운로드 할 로그 옆에 있는 버튼을 선택한 다음 다운로드를 선택합니다.
7. 제공된 링크에 대한 컨텍스트(마우스 오른쪽 클릭) 메뉴를 열고 나서 [Save Link As]를 선택합니다. 로그 파일을 저장할 위치를 입력한 다음 저장을 선택합니다.



AWS CLI

데이터베이스 로그 파일을 다운로드 하려면 AWS CLI 명령 [download-db-log-file-portion](#)을 사용합니다. 기본적으로 이 명령은 로그 파일의 최신 부분만을 다운로드 합니다. 하지만 --starting-token 0 파라미터를 지정하여 전체 파일을 다운로드 할 수 있습니다.

다음 예제에서는 log/ERROR.4라는 로그 파일의 내용을 다운로드 하여 errorlog.txt라는 로컬 파일에 저장하는 방법을 보여줍니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds download-db-log-file-portion \
    --db-instance-identifier myexampledb \
    --starting-token 0 --output text \
    --log-file-name log/ERROR.4 > errorlog.txt
```

Windows의 경우:

```
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

데이터베이스 로그 파일을 다운로드 하려면 [API DownloadDBLogFilePortion](#) Amazon RDS 작업을 사용합니다.

데이터베이스 로그 파일 조사

Amazon RDS 콘솔을 사용하여 로그 파일의 내용을 모니터링 할 수 있습니다.

AWS Management 콘솔

데이터베이스 로그 파일을 조사하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 보고자 하는 로그 파일을 보유한 DB 인스턴스의 이름을 선택합니다.
4. 로그 및 이벤트 탭을 선택합니다.
5. 로그 섹션에서 로그 파일을 선택한 다음 보기 를 선택합니다.

Amazon CloudWatch Logs에 데이터베이스 로그 게시

DB 인스턴스 로그를 보고 다운로드하는 것 외에도 Amazon CloudWatch Logs에 로그를 게시할 수 있습니다. CloudWatch Logs에서는 로그 데이터에 대한 실시간 분석을 수행하고, 매우 내구력 있는 스토리지에 데이터를 저장하며, CloudWatch Logs 에이전트를 사용하여 데이터를 관리할 수 있습니다. 보존 기간을 지정하지 않는 한 AWS는 CloudWatch Logs에 게시된 로그 데이터를 무기한 보존합니다. 자세한 내용은 [CloudWatch Logs에서 로그 데이터 보존 기간 변경](#)을 참조하십시오.

엔진별 정보는 다음을 참조하십시오.

- the section called “Amazon CloudWatch Logs에 MariaDB 로그 게시” (p. 311)
- the section called “CloudWatch Logs에 MySQL 로그 게시” (p. 320)
- the section called “Amazon CloudWatch Logs에 Oracle 로그 게시” (p. 328)
- the section called “CloudWatch Logs에 PostgreSQL 로그 게시” (p. 333)

REST를 사용하여 로그 파일 내용 읽기

Amazon RDS는 DB 인스턴스 로그 파일 액세스를 허용하는 REST 엔드포인트를 제공합니다. 로그 파일 내용을 스트리밍하는 애플리케이션을 작성해야 하는 경우 유용합니다.

구문은 다음과 같습니다.

```
GET /v13/downloadCompleteLogFile/DBInstanceIdentifier/LogFileName HTTP/1.1
Content-type: application/json
host: rds.region.amazonaws.com
```

다음 파라미터는 필수 파라미터입니다.

- **DBInstanceIdentifier** — 다운로드하려는 로그 파일이 있는 DB 인스턴스에 고객이 할당하는 이름입니다.
- **LogFileName** — 다운로드할 로그 파일의 이름입니다.

응답에는 스트림으로 요청된 로그 파일의 내용이 포함됩니다.

다음 예제에서는 us-west-2 리전에 sample-sql로 명명된 DB 인스턴스에 대해 log/ERROR.6으로 명명된 로그 파일을 다운로드합니다.

```
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
```

존재하지 않는 DB 인스턴스를 지정하는 경우 응답에 다음 오류가 포함됩니다.

- `DBInstanceNotFound` — `DBInstanceIdentifier`는 기존 DB 인스턴스를 참조하지 않습니다. (HTTP 상태 코드: 404)

MariaDB 데이터베이스 로그 파일

MariaDB 오류 로그, 느린 쿼리 로그 및 일반 로그를 모니터링할 수 있습니다. MariaDB 오류 로그는 기본적으로 생성됩니다. DB 파라미터 그룹에서 파라미터를 설정하여 느린 쿼리와 일반 로그를 생성할 수 있습니다. Amazon RDS는 모든 MariaDB 로그 파일을 교체하며 각 유형에 대한 간격은 다음에 제공됩니다.

Amazon RDS 콘솔, Amazon RDS API, Amazon RDS CLI 또는 AWS SDK를 통해 MariaDB 로그를 직접 모니터링할 수 있습니다. 또한, 주 데이터베이스에 있는 데이터베이스 테이블로 로그를 전송하고 그 테이블을 쿼리하여 MariaDB 로그에 액세스할 수 있습니다. mysqlbinlog 유ти리티를 사용하여 이진 로그를 다운로드할 수 있습니다.

파일 기반 데이터베이스 로그 보기, 다운로드 및 조사 방법에 대한 자세한 내용은 [Amazon RDS 데이터베이스 로그 파일 \(p. 306\)](#) 단원을 참조하십시오.

MariaDB 오류 로그 액세스

MariaDB 오류 로그는 `<host-name>.err` 파일에 기록됩니다. Amazon RDS 콘솔을 사용하거나 Amazon RDS API, Amazon RDS CLI 또는 AWS SDK를 사용하여 로그를 검색함으로써 이 파일을 볼 수 있습니다. `<host-name>.err` 파일은 5분마다 폴러시되고 그 내용이 `mysql-error-running.log`에 추가됩니다. 그런 다음, `mysql-error-running.log` 파일은 1시간마다 순환되고 지난 24시간 동안 매시간 생성된 파일이 보존됩니다. 각 로그 파일이 생성된 시간(UTC)이 파일 이름에 추가됩니다. 로그 파일에는 타임스탬프도 포함되어 있어, 로그 항목이 작성된 시간을 확인하는 데 도움이 됩니다.

MariaDB에서는 시작, 종료 및 오류 발생 시에만 오류 로그에 데이터가 기록됩니다. DB 인스턴스는 오류로 그에 새 항목이 기록되지 않는 상태로 몇 시간이나 며칠씩 작동할 수 있습니다. 최근 항목이 보이지 않으면 이는 서버에서 로그에 입력된 오류가 발생하지 않았기 때문입니다.

MariaDB 느린 쿼리 및 일반 로그 액세스

DB 파라미터 그룹에서 파라미터를 설정하면 MariaDB 느린 쿼리 로그와 일반 로그가 파일이나 데이터베이스 테이블에 기록될 수 있습니다. DB 파라미터 그룹의 생성 및 변경에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오. Amazon RDS 콘솔에서 또는 Amazon RDS API, AWS CLI, AWS SDK를 사용하여 느린 쿼리 로그 또는 일반 로그를 보려면 먼저 이러한 파라미터를 설정해야 합니다.

이 목록에 있는 파라미터를 사용하여 MariaDB 로깅을 제어할 수 있습니다.

- `slow_query_log`: 느린 쿼리 로그를 만들려면 1로 설정합니다. 기본값은 0입니다.
- `general_log`: 일반 로그를 만들려면 1로 설정합니다. 기본값은 0입니다.
- `long_query_time`: 빠르게 실행되는 쿼리가 느린 쿼리 로그에 기록되지 않도록 하려면 로그에 기록할 쿼리의 최단 실행 시간 값(초)을 지정하십시오. 기본값은 10초이고, 최소값은 0초입니다. `log_output = FILE`인 경우에는 마이크로초 단위까지 부동 소수점 값을 지정할 수 있습니다. `log_output = TABLE`인 경우에는 초 단위로 정수 값을 지정해야 합니다. 실행 시간이 `long_query_time` 값을 초과하는 쿼리만 로그에 기록됩니다. 예를 들어 `long_query_time`을 0.1로 설정하면 100밀리초 미만의 시간 동안 작동하는 쿼리가 로그에 기록되지 않습니다.

- `log_queries_not_using_indexes`: 인덱스를 사용하지 않는 모든 쿼리를 느린 쿼리 로그에 기록하려면 이 파라미터를 1로 설정합니다. 기본값은 0입니다. 인덱스를 사용하지 않는 쿼리는 실행 시간이 `long_query_time` 파라미터의 값보다 짧아도 로그에 기록됩니다.
- `log_output option`: `log_output` 파라미터에 대해 다음 옵션 중 하나를 지정할 수 있습니다.
 - TABLE(기본값)– `mysql.general_log` 테이블에는 일반 쿼리를, `mysql.slow_log` 테이블에는 느린 쿼리를 씁니다.
 - FILE– 파일 시스템에 일반 쿼리 로그와 느린 쿼리 로그를 모두 씁니다. 로그 파일은 매시간 순환됩니다.
 - NONE– 로깅을 비활성화합니다.

로깅을 사용하는 경우, Amazon RDS는 테이블 로그를 순환하거나 로그 파일을 정기적으로 삭제합니다. 이러한 예방 조치를 취하면 데이터베이스 사용에 방해가 되거나 성능에 영향을 미치는 큰 로그 파일이 생성될 가능성을 줄일 수 있습니다. FILE 및 TABLE 로깅 접근 방식 교체 및 삭제는 다음과 같습니다.

- FILE 로깅을 사용하는 경우, 로그 파일은 매시간 검사되며 24시간 이상 지난 로그 파일은 삭제됩니다. 경우에 따라 삭제 후 나머지 로그 파일의 총 크기가 DB 인스턴스에 할당된 공간 중 2%의 임계값을 초과할 수 있습니다. 이러한 경우 로그 파일의 전체 크기가 임계값 이하로 작아질 때까지 가장 큰 로그 파일부터 차례대로 삭제됩니다.
- TABLE 로깅이 활성화된 경우 경우에 따라 로그 테이블이 24시간마다 순환됩니다. 테이블 로그에서 사용되는 공간이 할당된 스토리지 공간 중 20% 이상을 차지하거나 모든 로그의 총 크기가 10GB 이상일 경우 이 순환이 발생합니다. DB 인스턴스에 대해 사용된 공간의 양이 DB 인스턴스의 할당된 스토리지 공간 중 90% 이상을 차지할 경우, 로그 순환을 위한 임계값은 줄어듭니다. 테이블 로그에서 사용되는 공간이 할당된 스토리지 공간 중 10% 이상을 차지하거나 모든 로그의 총 크기가 5GB 이상일 경우, 로그 테이블은 순환됩니다.

로그 테이블이 순환되면 현재 로그 테이블은 백업 로그 테이블에 복사되며 현재 로그 테이블의 해당 항목들은 제거됩니다. 백업 로그 테이블이 이미 존재할 경우, 현재 로그 테이블이 백업으로 복사되기 전에 백업 로그 테이블이 삭제됩니다. 필요하다면 백업 로그 테이블을 쿼리할 수 있습니다.

`mysql.general_log` 테이블에 대한 백업 로그 테이블 이름은 `mysql.general_log_backup`으로 지정됩니다. `mysql.slow_log` 테이블에 대한 백업 로그 테이블 이름은 `mysql.slow_log_backup`으로 지정됩니다.

`mysql.rds_rotate_general_log` 절차를 호출하면 `mysql.general_log` 테이블을 순환할 수 있습니다. `mysql.rds_rotate_slow_log` 절차를 호출하면 `mysql.slow_log` 테이블을 순환할 수 있습니다.

데이터베이스 버전 업그레이드가 진행되는 동안 테이블 로그가 순환됩니다.

Amazon RDS는 TABLE 및 FILE 로그 순환을 Amazon RDS 이벤트에 기록하고 사용자에게 알림 메시지를 보냅니다.

Amazon RDS 콘솔, Amazon RDS API, Amazon RDS CLI 또는 AWS SDK에서 로그를 사용하여 작업하려면 `log_output` 파라미터를 FILE로 설정합니다. MariaDB 오류 로그와 같이, 이런 로그 파일은 매시간 순환됩니다. 이전의 24시간 동안 생성된 로그 파일이 보존됩니다.

느린 쿼리 및 일반 로그에 대한 자세한 내용은 MariaDB 문서에서 다음 단원을 참조하십시오.

- [느린 쿼리 로그](#)
- [일반 쿼리 로그](#)

Amazon CloudWatch Logs에 MariaDB 로그 게시

Amazon CloudWatch Logs의 로그 그룹에 로그 데이터를 게시하도록 Amazon RDS MariaDB DB 인스턴스를 구성할 수 있습니다. CloudWatch Logs를 통해 로그 데이터에 대한 실시간 분석을 수행할 수 있고, CloudWatch를 사용하여 경보를 만들고 지표를 볼 수 있습니다. CloudWatch Logs를 사용하여 내구성이 뛰어난 스토리지에 로그 레코드를 저장할 수 있습니다.

Amazon RDS는 각 MariaDB 데이터베이스 로그를 로그 그룹에 개별적인 데이터베이스 스트림으로 게시합니다. 예를 들어 느린 쿼리 로그를 포함하도록 내보내기 함수를 구성하면 느린 쿼리 데이터가 /aws/rds/instance/*my_instance*/slowquery 로그 그룹의 느린 쿼리 로그 스트림에 저장됩니다.

오류 로그는 기본적으로 활성화됩니다. 다음 표에는 기타 MariaDB 로그의 요구 사항이 요약되어 있습니다.

로그	요구 사항
감사 로그	DB 인스턴스는 MARIADB_AUDIT_PLUGIN 옵션과 함께 사용자 지정 옵션 그룹을 사용해야 합니다.
일반 로그	DB 인스턴스는 파라미터 설정 general_log = 1과 함께 사용자 지정 파라미터 그룹을 사용하여 일반 로그를 활성화해야 합니다.
느린 쿼리 로그	DB 인스턴스는 파라미터 설정 slow_query_log = 1과 함께 사용자 지정 파라미터 그룹을 사용하여 느린 쿼리 로그를 활성화해야 합니다.
로그 출력	DB 인스턴스는 파라미터 설정 log_output = FILE과 함께 사용자 지정 파라미터 그룹을 사용하여 로그를 파일 시스템에 쓰고 CloudWatch Logs에 게시해야 합니다.

AWS Management 콘솔

콘솔에서 CloudWatch Logs에 MariaDB 로그를 게시하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 변경하려는 DB 인스턴스를 선택합니다.
3. [Modify]를 선택합니다.
4. 로그 내보내기 섹션에서 CloudWatch Logs에 게시하기 시작할 로그를 선택합니다.
5. [Continue]를 선택한 후, 요약 페이지에서 [Modify DB Instance]를 선택합니다.

AWS CLI

AWS CLI를 사용하여 MariaDB 로그를 게시할 수 있습니다. 다음 파라미터로 `modify-db-instance` 명령을 호출할 수 있습니다.

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

`--cloudwatch-logs-export-configuration` 옵션에 대한 변경 사항은 항상 DB 인스턴스에 즉시 적용됩니다. 따라서 `--apply-immediately` 및 `--no-apply-immediately` 옵션은 지정해도 아무런 효과가 없습니다.

또 다음 AWS CLI 명령을 호출해 MariaDB 로그를 게시할 수 있습니다.

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`

- `restore-db-instance-to-point-in-time`

다음 옵션으로 AWS CLI 명령 중 하나를 실행합니다.

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

실행하는 AWS CLI 명령에 따라 다른 옵션이 필요할 수 있습니다.

Example

다음 예에서는 CloudWatch Logs에 로그 파일을 게시하도록 기존 MariaDB DB 인스턴스를 수정합니다. `--cloudwatch-logs-export-configuration` 값은 JSON 객체입니다. 이 객체에 대한 키는 `EnableLogTypes`이며, 해당 값은 `audit`, `error`, `general` 및 `slowquery`의 조합을 사용하는 문자열의 배열입니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes": \
    ["audit", "error", "general", "slowquery"]}'
```

Windows의 경우:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes": \
    ["audit", "error", "general", "slowquery"]}'
```

Example

다음 명령은 CloudWatch Logs에 로그 파일을 게시하도록 MariaDB DB 인스턴스를 만듭니다. `--enable-cloudwatch-logs-exports` 값은 문자열의 JSON 배열입니다. 문자열은 `audit`, `error`, `general` 및 `slowquery`의 조합일 수 있습니다.

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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

RDS API를 사용하여 MariaDB 로그를 게시할 수 있습니다. 다음 파라미터로 [ModifyDBInstance](#) 작업을 호출할 수 있습니다.

- [DBInstanceIdentifier](#)
- [CloudwatchLogsExportConfiguration](#)

Note

[CloudwatchLogsExportConfiguration](#) 파라미터에 대한 변경 사항은 항상 DB 인스턴스에 즉시 적용됩니다. 따라서 [ApplyImmediately](#) 파라미터는 지정해도 아무런 효과가 없습니다.

또 다음 RDS API 작업을 호출해 MariaDB 로그를 게시할 수 있습니다.

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

다음 파라미터로 RDS API 작업 중 하나를 실행합니다.

- [DBInstanceIdentifier](#)
- [EnableCloudwatchLogsExports](#)
- [Engine](#)
- [DBInstanceClass](#)

실행하는 AWS CLI 명령에 따라 다른 파라미터가 필요할 수 있습니다.

로그 파일 크기

MariaDB 느린 쿼리 로그, 오류 로그 및 일반 로그 파일 크기는 DB 인스턴스에 대해 할당된 스토리지 공간의 2% 이하로 제한됩니다. 이 임계값을 유지하기 위해 로그는 매시간 자동으로 순환되면서 24시간 이상 지난 로그 파일은 제거됩니다. 오래된 로그 파일을 제거한 후 로그 파일의 총 크기가 임계값을 초과하는 경우에는 로그 파일의 전체 크기가 임계값 이하로 작아질 때까지 가장 큰 로그 파일부터 차례대로 삭제됩니다.

테이블 기반 MariaDB 로그 관리

DB 파라미터 그룹을 만들고 `log_output` 서버 파라미터를 TABLE로 설정하여 일반 및 느린 쿼리 로그를 DB 인스턴스 상의 테이블로 전송할 수 있습니다. 그러면 일반 쿼리는 `mysql.general_log` 테이블에, 느린 쿼리는 `mysql.slow_log` 테이블에 로그가 기록됩니다. 이들 테이블을 쿼리하여 로그 정보에 액세스할 수 있습니다. 이 로깅을 활성화하면 데이터베이스에 기록되는 데이터의 양이 증가하여 성능이 저하될 수 있습니다.

일반 로그와 느린 쿼리 로그는 모두 기본적으로 비활성화됩니다. 테이블에 대한 로깅을 활성화하려면 `general_log` 및 `slow_query_log` 서버 파라미터도 1로 설정해야 합니다.

관련 파라미터를 0으로 설정하여 각 로깅 활동을 해제할 때까지 로그 테이블은 계속 커집니다. 흔히 대량의 데이터가 시간 경과에 따라 누적되어 할당된 스토리지 공간 중 상당한 비율을 사용할 수 있습니다. Amazon RDS에서는 로그 테이블을 자를 수 없지만 테이블의 콘텐츠를 이동할 수 있습니다. 테이블을 순환하면 그 내

용이 백업 테이블에 저장된 다음 빈 로그 테이블이 새로 생성됩니다. 다음 명령줄 프로시저로 로그 테이블을 수동으로 순환시킬 수 있으며, 이때 명령 프롬프트는 PROMPT>로 표시됩니다.

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

오래된 데이터를 완전히 제거하고 디스크 공간을 회수하려면 알맞은 프로시저를 두 번 연속으로 호출합니다.

이진 로깅 형식

Amazon RDS의 MariaDB는 행 기반, 설명문 기반 및 혼합 바이너리 로깅 형식을 지원합니다. 기본 이진 로깅 형식은 혼합입니다. 다양한 MariaDB 이진 로그 형식에 관한 세부 정보는 MariaDB 설명서에서 [이진 로깅 형식](#) 단원을 참조하십시오.

복제를 사용하려는 경우 바이너리 로깅 형식은 원본에 기록되고 복제 대상으로 전송되는 데이터 변경 내용의 레코드를 확인하므로 중요합니다. 복제와 관련된 다양한 바이너리 로깅 형식의 장/단점에 대한 자세한 내용은 MySQL 설명서의 [Advantages and Disadvantages of Statement-Based and Row-Based Replication](#)을 참조하십시오.

Important

이진 로깅 형식을 행 기반으로 설정하면 이진 로그 파일이 매우 커질 수 있습니다. 큰 이진 로그 파일은 DB 인스턴스에 사용할 수 있는 스토리지의 양을 줄이므로, DB 인스턴스의 복원 작업 수행에 필요한 시간이 늘어날 수 있습니다.

설명문 기반 복제는 원본 DB 인스턴스와 읽기 전용 복제본 간의 불일치를 초래할 수 있습니다. 자세한 내용은 MariaDB 설명서의 [Unsafe Statements for Statement-based Replication](#)을 참조하십시오.

MariaDB 이진 로깅 형식을 설정하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 파라미터 그룹을 선택합니다.
3. 수정할 DB 인스턴스에 사용되는 파라미터 그룹을 선택합니다.

기본 파라미터 그룹을 수정할 수 없습니다. DB 인스턴스에서 기본 파라미터 그룹을 사용 중인 경우 새 파라미터 그룹을 생성하여 DB 인스턴스와 연결합니다.

DB 파라미터 그룹에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

4. 파라미터 그룹 작업에서 편집을 선택합니다.
5. binlog_format 파라미터를 선택한 바이너리 로깅 형식(ROW, STATEMENT 또는 MIXED)으로 설정합니다.
6. 변경 내용 저장을 선택하여 업데이트를 DB 파라미터 그룹에 저장합니다.

MariaDB 이진 로그 액세스

mysqlbinlog 유ти리티를 사용하여 MariaDB DB 인스턴스로부터 텍스트 형식의 이진 로그를 다운로드할 수 있습니다. 이진 로그는 로컬 컴퓨터로 다운로드됩니다. mysqlbinlog 유ти리티 사용에 대한 자세한 내용은 MariaDB 설명서에서 [mysqlbinlog 사용](#)을 참조하십시오.

Amazon RDS 인스턴스에 대해 mysqlbinlog 유ти리티를 실행하려면 다음 옵션을 사용합니다.

- --read-from-remote-server 옵션을 지정합니다.
- --host: 인스턴스의 앤드포인트에서 DNS 이름을 지정합니다.

- **--port:** 인스턴스에서 사용되는 포트를 지정합니다.
- **--user:** 복제 슬레이브 권한이 부여된 MariaDB 사용자를 지정합니다.
- **--password:** 사용자의 암호를 지정하거나, 유ти리티에서 암호 입력을 요구하는 메시지가 표시되도록 암호 값을 생략합니다.
- **--result-file:** 출력을 수신할 로컬 파일을 지정합니다.
- 하나 이상의 이진 로그 파일의 이름을 지정합니다. 사용 가능한 로그의 목록을 획득하려면 SQL 명령 SHOW BINARY LOGS를 사용합니다.

mysqlbinlog 옵션에 대한 자세한 내용은 MariaDB 설명서에서 [mysqlbinlog 옵션](#)을 참조하십시오.

다음은 그 한 예입니다.

Linux, OS X, Unix의 경우:

```
mysqlbinlog \
    --read-from-remote-server \
    --host=mariadbinstance1.1234abcd.region.rds.amazonaws.com \
    --port=3306 \
    --user ReplUser \
    --password <password> \
    --result-file=/tmp/binlog.txt
```

Windows의 경우:

```
mysqlbinlog ^
    --read-from-remote-server ^
    --host=mariadbinstance1.1234abcd.region.rds.amazonaws.com ^
    --port=3306 ^
    --user ReplUser ^
    --password <password> ^
    --result-file=/tmp/binlog.txt
```

Amazon RDS는 보통은 최대한 빨리 이진 로그를 제거하지만, mysqlbinlog가 액세스할 수 있도록 인스턴스에서 이진 파일을 여전히 사용할 수 있어야 합니다. RDS가 이진 파일을 보존할 시간을 지정하려면 `mysql.rds_set_configuration` 저장 프로시저를 사용하고 로그를 다운로드하기에 충분한 시간으로 기간을 지정합니다. 보존 기간을 설정한 후, DB 인스턴스의 스토리지 사용량을 모니터링하여 보존된 이진 로그가 너무 많은 스토리지를 차지하지 않도록 합니다.

다음 예제에서는 보존 기간을 1일로 설정합니다.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

현재 설정을 표시하려면 `mysql.rds_show_configuration` 저장 프로시저를 사용합니다.

```
call mysql.rds_show_configuration;
```

이진 로그 주석

MariaDB DB 인스턴스에서, `Annotate_rows` 이벤트를 사용하여 행 이벤트로 인해 발생한 SQL 쿼리의 녹사본으로 해당 행 이벤트에 주석을 달 수 있습니다. 이 접근 방식은 MySQL 5.6 이상 DB 인스턴스에서 `binlog_rows_query_log_events` 파라미터를 활성화하는 것과 비슷한 기능을 제공합니다.

사용자 지정 파라미터 그룹을 생성하고 `binlog_annotation_row_events` 파라미터를 1로 설정하여 이진 로그 주석을 전역에서 활성화할 수 있습니다. 또한 `SET SESSION binlog_annotation_row_events = 1`

호출로 세션 수준에서 주석을 활성화할 수도 있습니다. 슬레이브 인스턴스에서 이진 로깅이 활성화되어 있다면 `replicate_annotation_row_events`를 사용하여 이진 로그 주석을 슬레이브 인스턴스로 복제합니다. 이러한 설정을 사용하는 데는 특별한 권한이 필요하지 않습니다.

다음 예는 MariaDB의 행 기반 트랜잭션입니다. 트랜잭션 격리 수준을 읽기 커밋으로 설정하면 행 기반 로깅 사용이 트리거됩니다.

```
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;
```

주석이 없을 경우, 트랜잭션의 이진 로그 항목은 다음과 같습니다.

```
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/*!*/;
```

다음 문은 이 동일한 트랜잭션에 대해 세션 수준의 주석을 활성화하고, 트랜잭션을 커밋한 후 다시 비활성화 합니다.

```
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;
```

주석이 있는 경우, 트랜잭션의 이진 로그 항목은 다음과 같습니다.

```
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 데이터베이스 로그 파일

Amazon RDS 콘솔 또는 API를 사용하여 Microsoft SQL Server 오류 로그, 에이전트 로그, 추적 파일 및 덤프 파일에 액세스할 수 있습니다. 파일 기반 데이터베이스 로그 보기, 다운로드 및 조사 방법에 대한 자세한 내용은 [Amazon RDS 데이터베이스 로그 파일 \(p. 306\)](#) 단원을 참조하십시오.

보존 일정

로그 파일은 날마다, 그리고 DB 인스턴스가 다시 시작될 때마다 교환됩니다. 다음은 Amazon RDS의 Microsoft SQL Server 로그에 대한 보존 일정입니다.

로그 유형	보존 일정
오류 로그	최대 30개의 오류 로그가 보존됩니다. Amazon RDS는 7일이 경과한 오류 로그를 삭제할 수 있습니다.
에이전트 로그	최대 10개의 에이전트 로그가 보존됩니다. Amazon RDS는 7일이 경과한 에이전트 로그를 삭제할 수 있습니다.
추적 파일	추적 파일은 DB 인스턴스의 추적 파일 보존 기간에 따라 보존됩니다. 추적 파일의 기본 보존 기간은 7일입니다. DB 인스턴스의 추적 파일 보존 기간을 수정하려면 추적 및 덤프 파일의 보존 기간 설정 (p. 567) 단원을 참조하십시오.
덤프 파일	덤프 파일은 DB 인스턴스의 덤프 파일 보존 기간에 따라 보존됩니다. 덤프 파일의 기본 보존 기간은 7일입니다. DB 인스턴스의 덤프 파일 보존 기간을 수정하려면 추적 및 덤프 파일의 보존 기간 설정 (p. 567) 단원을 참조하십시오.

rds_read_error_log 절차를 사용하여 SQL Server 오류 로그 보기

Amazon RDS 저장 프로시저 `rds_read_error_log`를 사용하여 오류 로그 및 에이전트 로그를 볼 수 있습니다. 자세한 내용은 [rds_read_error_log 프로시저 사용 \(p. 566\)](#) 단원을 참조하십시오.

관련 주제

- [SQL Server 에이전트의 사용 \(p. 564\)](#)
- [Microsoft SQL Server 로그 작업 \(p. 566\)](#)
- [추적 및 덤프 파일 작업 \(p. 566\)](#)

MySQL 데이터베이스 로그 파일

MySQL 오류 로그, 느린 쿼리 로그 및 일반 로그를 모니터링할 수 있습니다. MySQL 오류 로그는 기본적으로 생성됩니다. DB 파라미터 그룹에서 파라미터를 설정하여 느린 쿼리와 일반 로그를 생성할 수 있습니다. Amazon RDS는 모든 MySQL 로그 파일을 교체하며 각 유형에 대한 간격은 다음에 제공됩니다.

Amazon RDS 콘솔, Amazon RDS API, AWS CLI 또는 AWS SDK를 통해 MySQL 로그를 직접 모니터링할 수 있습니다. 또한, 주 데이터베이스에 있는 데이터베이스 테이블로 로그를 전송하고 그 테이블을 쿼리하여

MySQL 로그에 액세스할 수 있습니다. mysqlbinlog 유ти리티를 사용하여 이진 로그를 다운로드할 수 있습니다.

파일 기반 데이터베이스 로그 보기, 다운로드 및 조사 방법에 대한 자세한 내용은 [Amazon RDS 데이터베이스 로그 파일 \(p. 306\)](#) 단원을 참조하십시오.

MySQL 오류 로그 액세스

MySQL 오류 로그는 mysql-error.log 파일에 기록됩니다. Amazon RDS 콘솔을 사용하거나 Amazon RDS API, Amazon RDS CLI 또는 AWS SDK를 사용하여 로그를 검색하면 mysql-error.log를 볼 수 있습니다. mysql-error.log는 5분마다 플러시되며 해당 콘텐츠가 mysql-error-running.log에 추가됩니다. 그런 다음, mysql-error-running.log 파일은 1시간마다 교체되고 마지막 24시간 일 동안 생성된 시 간별 파일이 보존됩니다. 보존 기간은 Amazon RDS와 Aurora 간에 다릅니다.

각 로그 파일이 생성된 시간(UTC)이 파일 이름에 추가됩니다. 로그 파일에는 타임스탬프도 포함되어 있어, 로그 항목이 작성된 시간을 확인하는 데 도움이 됩니다.

MySQL에서는 시작, 종료 및 오류 발생 시에만 오류 로그에 데이터가 기록됩니다. DB 인스턴스는 오류 로그에 새 항목이 기록되지 않는 상태로 몇 시간이나 며칠씩 작동할 수 있습니다. 최근 항목이 보이지 않으면 이는 서버에서 로그에 입력될 만한 오류가 발생하지 않았기 때문입니다.

MySQL 느린 쿼리 및 일반 로그 액세스

DB 파라미터 그룹에서 파라미터를 설정하면 MySQL 느린 쿼리 로그와 일반 로그가 파일이나 데이터베이스 테이블에 기록될 수 있습니다. DB 파라미터 그룹의 생성 및 변경에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오. 이런 파라미터를 설정해야 Amazon RDS 콘솔에서 느린 쿼리 로그 또는 일반 로그를 볼 수 있습니다. 아니면 Amazon RDS API, Amazon RDS CLI 또는 AWS SDK를 사용하여 볼 수 있습니다.

이 목록에 있는 파라미터를 사용하여 MySQL 로깅을 제어할 수 있습니다.

- **slow_query_log**: 느린 쿼리 로그를 만들려면 1로 설정합니다. 기본값은 0입니다.
- **general_log**: 일반 로그를 만들려면 1로 설정합니다. 기본값은 0입니다.
- **long_query_time**: 빠르게 실행되는 쿼리가 느린 쿼리 로그에 기록되지 않도록 하려면 로그에 기록할 쿼리의 최단 실행 시간 값(초)을 지정하십시오. 기본값은 10초이고, 최소값은 0초입니다. log_output = FILE인 경우에는 마이크로초 단위까지 부동 소수점 값을 지정할 수 있습니다. log_output = TABLE인 경우에는 초 단위로 정수 값을 지정해야 합니다. 실행 시간이 long_query_time 값을 초과하는 쿼리만 로그에 기 록됩니다. 예를 들어 long_query_time을 0.1로 설정하면 100밀리초 미만의 시간 동안 작동하는 쿼리가 로그에 기록되지 않습니다.
- **log_queries_not_using_indexes**: 인덱스를 사용하지 않는 모든 쿼리를 느린 쿼리 로그에 기록하려 면 1로 설정합니다. 기본값은 0입니다. 인덱스를 사용하지 않는 쿼리는 실행 시간이 long_query_time 파라미터의 값보다 짧아도 로그에 기록됩니다.
- **log_output option**: log_output 파라미터에 대해 다음 옵션 중 하나를 지정할 수 있습니다.
 - TABLE(기본값)- mysql.general_log 테이블에는 일반 쿼리를, mysql.slow_log 테이블에는 느린 쿼리를 씁니다.
 - FILE- 파일 시스템에 일반 쿼리 로그와 느린 쿼리 로그를 모두 씁니다. 로그 파일은 매시간 순환됩니다.
 - NONE- 로깅을 비활성화합니다.

로깅을 사용하는 경우, Amazon RDS는 테이블 로그를 순환하거나 로그 파일을 정기적으로 삭제합니다. 이러한 예방 조치를 취하면 데이터베이스 사용에 방해가 되거나 성능에 영향을 미치는 큰 로그 파일이 생성될 가능성을 줄일 수 있습니다. FILE 및 TABLE 로깅 접근 방식 교체 및 삭제는 다음과 같습니다.

- FILE 로깅을 사용하는 경우, 로그 파일은 매시간 검사되며 24시간 이상 지난 로그 파일은 삭제됩니다. 경우에 따라 삭제 후 나머지 로그 파일의 총 크기가 DB 인스턴스에 할당된 공간 중 2%의 임계값을 초과할 수 있습니다. 이러한 경우 로그 파일의 전체 크기가 임계값 이하로 작아질 때까지 가장 큰 로그 파일부터 차례 대로 삭제됩니다.

- TABLE 로깅이 활성화된 경우에 따라 로그 테이블이 24시간마다 순환됩니다. 테이블 로그에서 사용되는 공간이 할당된 스토리지 공간 중 20% 이상을 차지하거나 모든 로그의 총 크기가 10GB 이상일 경우 이 순환이 발생합니다. DB 인스턴스에 대해 사용된 공간의 양이 DB 인스턴스의 할당된 스토리지 공간 중 90% 이상을 차지할 경우, 로그 순환을 위한 임계값은 줄어듭니다. 테이블 로그에서 사용되는 공간이 할당된 스토리지 공간 중 10% 이상을 차지하거나 모든 로그의 총 크기가 5GB 이상일 경우, 로그 테이블은 순환됩니다. 공간 확보를 위해 로그 테이블이 순환되면 알림을 받으려면 `low_free_storage` 이벤트를 구독할 수 있습니다. 자세한 내용은 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오.

로그 테이블이 순환되면 현재 로그 테이블은 백업 로그 테이블에 복사되며 현재 로그 테이블의 해당 항목들은 제거됩니다. 백업 로그 테이블이 이미 존재할 경우, 현재 로그 테이블이 백업으로 복사되기 전에 백업 로그 테이블이 삭제됩니다. 필요하다면 백업 로그 테이블을 쿼리할 수 있습니다. `mysql.general_log` 테이블에 대한 백업 로그 테이블 이름은 `mysql.general_log_backup`으로 지정됩니다. `mysql.slow_log` 테이블에 대한 백업 로그 테이블 이름은 `mysql.slow_log_backup`으로 지정됩니다.

`mysql.rds_rotate_general_log` 절차를 호출하면 `mysql.general_log` 테이블을 순환할 수 있습니다. `mysql.rds_rotate_slow_log` 절차를 호출하면 `mysql.slow_log` 테이블을 순환할 수 있습니다.

데이터베이스 버전 업그레이드가 진행되는 동안 테이블 로그가 순환됩니다.

Amazon RDS 콘솔, Amazon RDS API, Amazon RDS CLI 또는 AWS SDK에서 로그를 사용하여 작업하려면 `log_output` 파라미터를 FILE로 설정합니다. MySQL 오류 로그와 같이, 이런 로그 파일은 매시간 순환됩니다. 이전 24시간 동안 생성된 로그 파일이 보존됩니다. 보존 기간은 Amazon RDS와 Aurora 간에 다릅니다.

느린 쿼리 및 일반 로그에 대한 자세한 내용은 MySQL 문서에서 다음 항목을 참조하십시오.

- [The Slow Query Log](#)
- [The General Query Log](#)

MySQL 감사 로그에 대한 액세스

감사 로그에 액세스하려면 DB 인스턴스가 `MARIADB_AUDIT_PLUGIN` 옵션과 함께 사용자 지정 옵션 그룹을 사용해야 합니다. 자세한 내용은 [MariaDB 감사 플러그인 지원 \(p. 664\)](#) 단원을 참조하십시오.

CloudWatch Logs에 MySQL 로그 게시

Amazon CloudWatch Logs의 로그 그룹에 로그 데이터를 게시하도록 Amazon RDS MySQL DB 인스턴스를 구성할 수 있습니다. CloudWatch Logs를 통해 로그 데이터에 대한 실시간 분석을 수행할 수 있고, CloudWatch를 사용하여 경보를 만들고 지표를 볼 수 있습니다. CloudWatch Logs를 사용하여 내구성이 뛰어난 스토리지에 로그 레코드를 저장할 수 있습니다.

Amazon RDS는 각 MySQL 데이터베이스 로그를 로그 그룹에 개별적인 데이터베이스 스트림으로 게시합니다. 예를 들어 느린 쿼리 로그를 포함하도록 내보내기 함수를 구성하면 느린 쿼리 데이터가 `/aws/rds/instance/my_instance/slowquery` 로그 그룹의 느린 쿼리 로그 스트림에 저장됩니다.

오류 로그는 기본적으로 활성화됩니다. 다음 표에는 기타 MySQL 로그의 요구 사항이 요약되어 있습니다.

로그	요구 사항
감사 로그	DB 인스턴스는 <code>MARIADB_AUDIT_PLUGIN</code> 옵션과 함께 사용자 지정 옵션 그룹을 사용해야 합니다.
일반 로그	DB 인스턴스는 파라미터 설정 <code>general_log = 1</code> 과 함께 사용자 지정 파라미터 그룹을 사용하여 일반 로그를 활성화해야 합니다.

로그	요구 사항
느린 쿼리 로그	DB 인스턴스는 파라미터 설정 <code>slow_query_log</code> = 1과 함께 사용자 지정 파라미터 그룹을 사용하여 느린 쿼리 로그를 활성화해야 합니다.
로그 출력	DB 인스턴스는 파라미터 설정 <code>log_output = FILE</code> 과 함께 사용자 지정 파라미터 그룹을 사용하여 로그를 파일 시스템에 쓰고 CloudWatch Logs에 게시해야 합니다.

Note

CloudWatch Logs에 로그 파일 게시는 MySQL 버전 5.6, 5.7, 8.0에서만 지원됩니다.

AWS Management 콘솔

콘솔을 사용하여 MySQL 로그를 CloudWatch Logs에 게시하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 변경하려는 DB 인스턴스를 선택합니다.
3. [Modify]를 선택합니다.
4. 로그 내보내기 섹션에서 CloudWatch Logs에 게시하기 시작할 로그를 선택합니다.
5. [Continue]를 선택한 후, 요약 페이지에서 [Modify DB Instance]를 선택합니다.

AWS CLI

AWS CLI를 사용하여 MySQL 로그를 게시할 수 있습니다. 다음 파라미터로 `modify-db-instance` 명령을 호출할 수 있습니다.

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

`--cloudwatch-logs-export-configuration` 옵션에 대한 변경 사항은 항상 DB 인스턴스에 즉시 적용됩니다. 따라서 `--apply-immediately` 및 `--no-apply-immediately` 옵션은 지정해도 아무런 효과가 없습니다.

또 다음 AWS CLI 명령을 호출해 MySQL 로그를 게시할 수 있습니다.

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

다음 옵션으로 AWS CLI 명령 중 하나를 실행합니다.

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`

- `--engine`

실행하는 AWS CLI 명령에 따라 다른 옵션이 필요할 수 있습니다.

Example

다음 예에서는 CloudWatch Logs에 로그 파일을 게시하도록 기존 MySQL DB 인스턴스를 수정합니다. `--cloudwatch-logs-export-configuration` 값은 JSON 객체입니다. 이 객체에 대한 키는 `EnableLogTypes`이며, 해당 값은 `audit`, `error`, `general` 및 `slowquery`의 조합을 사용하는 문자열의 배열입니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes": \
    ["audit", "error", "general", "slowquery"]}'
```

Windows의 경우:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes": \
    ["audit", "error", "general", "slowquery"]}'
```

Example

다음 예에서는 CloudWatch Logs에 로그 파일을 게시하도록 MySQL DB 인스턴스를 생성합니다. `--enable-cloudwatch-logs-exports` 값은 문자열의 JSON 배열입니다. 문자열은 `audit`, `error`, `general` 및 `slowquery`의 조합일 수 있습니다.

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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

RDS API를 사용하여 MySQL 로그를 게시할 수 있습니다. 다음 파라미터로 [ModifyDBInstance](#) 작업을 호출할 수 있습니다.

- `DBInstanceIdentifier`

- CloudwatchLogsExportConfiguration

Note

CloudwatchLogsExportConfiguration 파라미터에 대한 변경 사항은 항상 DB 인스턴스에 즉시 적용됩니다. 따라서 ApplyImmediately 파라미터는 지정해도 아무런 효과가 없습니다.

또 다음 RDS API 작업을 호출해 MySQL 로그를 게시할 수 있습니다.

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

다음 파라미터로 RDS API 작업 중 하나를 실행합니다.

- DBInstanceIdentifier
- EnableCloudwatchLogsExports
- Engine
- DBInstanceClass

실행하는 AWS CLI 명령에 따라 다른 파라미터가 필요할 수 있습니다.

로그 파일 크기

MySQL 느린 쿼리 로그, 오류 로그 및 일반 로그 파일 크기는 DB 인스턴스에 대해 할당된 스토리지 공간의 2% 이하로 제한됩니다. 이 임계값을 유지하기 위해 로그는 매시간 자동으로 순환되면서 24시간 이상 지난 로그 파일은 제거됩니다. 오래된 로그 파일을 제거한 후 로그 파일의 총 크기가 임계값을 초과하는 경우에는 로그 파일의 전체 크기가 임계값 이하로 작아질 때까지 가장 큰 로그 파일부터 차례대로 삭제됩니다.

MySQL의 경우 다시 실행 로그에 기록되는 BLOB에 대한 크기 제한이 있습니다. 이러한 제한을 감안하려면 MySQL DB 인스턴스에 대한 innodb_log_file_size 파라미터가 테이블에서 발견된 BLOB 데이터의 최대 크기 및 동일한 테이블에서의 다른 가변 길이 필드(VARCHAR, VARBINARY, TEXT)의 길이 보다 10배 커야 합니다. 파라미터 값 설정 방법에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오. 다시 실행 로그 BLOB 크기 제한에 대한 자세한 내용은 [MySQL 5.6.20의 변경 내용](#) 단원을 참조하십시오.

테이블 기반 MySQL 로그 관리

DB 파라미터 그룹을 만들고 log_output 서버 파라미터를 TABLE로 설정하여 일반 및 느린 쿼리 로그를 DB 인스턴스 상의 테이블로 전송할 수 있습니다. 그러면 일반 쿼리는 mysql.general_log 테이블에, 느린 쿼리는 mysql.slow_log 테이블에 로그가 기록됩니다. 이들 테이블을 쿼리하여 로그 정보에 액세스할 수 있습니다. 이 로깅을 활성화하면 데이터베이스에 기록되는 데이터의 양이 증가하여 성능이 저하될 수 있습니다.

일반 로그와 느린 쿼리 로그는 모두 기본적으로 비활성화됩니다. 테이블에 대한 로깅을 활성화하려면 general_log 및 slow_query_log 서버 파라미터도 1로 설정해야 합니다.

관련 파라미터를 0으로 설정하여 각 로깅 활동을 해제할 때까지 로그 테이블은 계속 커집니다. 흔히 대량의 데이터가 시간 경과에 따라 누적되어 할당된 스토리지 공간 중 상당한 비율을 사용할 수 있습니다. Amazon RDS에서는 로그 테이블을 자를 수 없지만 테이블의 콘텐츠를 이동할 수 있습니다. 테이블을 순환하면 그 내용이 백업 테이블에 저장된 다음 빈 로그 테이블이 새로 생성됩니다. 다음 명령줄 프로시저로 로그 테이블을 수동으로 순환시킬 수 있으며, 이때 명령 프롬프트는 PROMPT>로 표시됩니다.

```
PROMPT> CALL mysql.rds_rotate_slow_log;
```

```
PROMPT> CALL mysql.rds_rotate_general_log;
```

오래된 데이터를 완전히 제거하고 디스크 공간을 회수하려면 알맞은 프로시저를 두 번 연속으로 호출합니다.

이진 로깅 형식

Amazon RDS에서 MySQL은 MySQL 버전 5.6 이상에 대해 행 기반, 설명문 기반 및 혼합 바이너리 로깅 형식을 지원합니다. 기본 바이너리 로깅 형식은 혼합입니다. MySQL 버전 5.1 및 5.5를 실행하는 DB 인스턴스의 경우 혼합 이진 로깅만 지원됩니다. 다양한 MySQL 바이너리 로그 형식에 대한 자세한 내용은 MySQL 설명서에서 [바이너리 로깅 형식](#)을 참조하십시오.

복제를 사용하려는 경우 바이너리 로깅 형식은 원본에 기록되고 복제 대상으로 전송되는 데이터 변경 내용의 레코드를 확인하므로 중요합니다. 복제와 관련된 다양한 바이너리 로깅 형식의 장/단점에 대한 자세한 내용은 MySQL 설명서의 [Advantages and Disadvantages of Statement-Based and Row-Based Replication](#)을 참조하십시오.

Important

이진 로깅 형식을 행 기반으로 설정하면 이진 로그 파일이 매우 커질 수 있습니다. 큰 이진 로그 파일은 DB 인스턴스에 사용할 수 있는 스토리지의 양을 줄이므로, DB 인스턴스의 복원 작업 수행에 필요한 시간이 늘어날 수 있습니다.

설명문 기반 복제는 원본 DB 인스턴스와 읽기 전용 복제본 간의 불일치를 초래할 수 있습니다. 자세한 내용은 MySQL 설명서의 [Determination of Safe and Unsafe Statements in Binary Logging](#)을 참조하십시오.

MySQL 이진 로깅 형식을 설정하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 파라미터 그룹을 선택합니다.
3. 수정할 DB 인스턴스에 사용되는 파라미터 그룹을 선택합니다.

기본 파라미터 그룹을 수정할 수 없습니다. DB 인스턴스에서 기본 파라미터 그룹을 사용 중인 경우 새 파라미터 그룹을 생성하여 DB 인스턴스와 연결합니다.

DB 파라미터 그룹에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

4. 파라미터 그룹 작업에서 편집을 선택합니다.
5. binlog_format 파라미터를 선택한 바이너리 로깅 형식(ROW, STATEMENT 또는 MIXED)으로 설정합니다.
6. 변경 내용 저장을 선택하여 업데이트를 DB 파라미터 그룹에 저장합니다.

Important

default.mysql5.6, default.mysql5.7, default.mysql8.0 DB 파라미터 그룹을 변경하면 그 파라미터 그룹을 사용하는 모든 MySQL 버전 DB 인스턴스에 변경 내용이 적용됩니다. 한 AWS 리전에서 다양한 MySQL 5.6, 5.7, 8.0 DB 인스턴스에 대해 각기 다른 이진 로깅 형식을 지정하려면 자체 DB 파라미터 그룹을 만들어야 합니다. 이 파라미터 그룹은 서로 다른 로깅 형식을 식별하고 그 DB 파라미터 그룹을 의도한 DB 인스턴스에 할당합니다.

MySQL 이진 로그 액세스

mysqlbinlog 유ти리티를 사용하여 MySQL 5.6 이상을 실행하는 Amazon RDS 인스턴스로부터 이진 로그를 다운로드하거나 스트리밍할 수 있습니다. 이진 로그를 로컬 컴퓨터로 다운로드하면 mysql 유ти리티를 사용하여 로그를 재생하는 것과 같은 작업을 수행할 수 있습니다. Mysqlbinlog 유ти리티 사용에 대한 자세한 내용은 [Using mysqlbinlog to Back Up Binary Log Files](#)를 참조하십시오.

Amazon RDS 인스턴스에 대해 mysqlbinlog 유ти리티를 실행하려면 다음 옵션을 사용합니다.

- `--read-from-remote-server` 옵션을 지정합니다.
- `--host`: 인스턴스의 앤드포인트에서 DNS 이름을 지정합니다.
- `--port`: 인스턴스에서 사용되는 포트를 지정합니다.
- `--user`: 복제 슬레이브 권한이 부여된 MySQL 사용자를 지정합니다.
- `--password`: 사용자의 암호를 지정하거나, 유ти리티에서 암호 입력을 요구하는 메시지가 표시되도록 암호 값을 생략합니다.
- 파일이 이진 형식으로 다운로드되도록 하려면 `--raw` 옵션을 지정합니다.
- `--result-file`: 원시 출력을 수신할 로컬 파일을 지정합니다.
- 하나 이상의 이진 로그 파일의 이름을 지정합니다. 사용 가능한 로그의 목록을 획득하려면 SQL 명령 `SHOW BINARY LOGS`를 사용합니다.
- 이진 로그 파일을 스트리밍하려면 `--stop-never` 옵션을 지정합니다.

`Mysqlbinlog` 옵션에 대한 자세한 내용은 [mysqlbinlog - Utility for Processing Binary Log Files](#) 단원을 참조하십시오.

예를 들어 다음을 참조하십시오.

Linux, OS X, Unix의 경우:

```
mysqlbinlog \
  --read-from-remote-server \
  --host=MySQL56Instance1.cg034hpkmmt.region.rds.amazonaws.com \
  --port=3306 \
  --user ReplUser \
  --password \
  --raw \
  --result-file=/tmp/ \
  binlog.00098
```

Windows의 경우:

```
mysqlbinlog ^
  --read-from-remote-server ^
  --host=MySQL56Instance1.cg034hpkmmt.region.rds.amazonaws.com ^
  --port=3306 ^
  --user ReplUser ^
  --password ^
  --raw ^
  --result-file=/tmp/ ^
  binlog.00098
```

Amazon RDS는 보통은 최대한 빨리 이진 로그를 제거하지만, `mysqlbinlog`가 액세스할 수 있도록 인스턴스에서 이진 파일을 여전히 사용할 수 있어야 합니다. RDS가 이진 파일을 보존할 시간을 지정하려면 `mysql.rds_set_configuration` 저장 프로시저를 사용하고 로그를 다운로드하기에 충분한 시간으로 기간을 지정합니다. 보존 기간을 설정한 후, DB 인스턴스의 스토리지 사용량을 모니터링하여 보존된 이진 로그가 너무 많은 스토리지를 차지하지 않도록 합니다.

Note

`mysql.rds_set_configuration` 저장 프로시저는 MySQL 5.6 이상 버전에서만 사용할 수 있습니다.

다음 예제에서는 보존 기간을 1일로 설정합니다.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

현재 설정을 표시하려면 `mysql.rds_show_configuration` 저장 프로시저를 사용합니다.

```
call mysql.rds_show_configuration;
```

Oracle 데이터베이스 로그 파일

Amazon RDS 콘솔 또는 API를 사용하여 Oracle 알림 로그, 감사 파일 및 추적 파일에 액세스할 수 있습니다. 파일 기반 데이터베이스 로그 보기, 다운로드 및 조사 방법에 대한 자세한 내용은 [Amazon RDS 데이터베이스 로그 파일 \(p. 306\)](#) 단원을 참조하십시오.

제공되는 Oracle 감사 파일은 표준 Oracle 감사 파일입니다. Amazon RDS는 Oracle 세분화된 감사(FGA) 기능을 지원합니다. 하지만 로그 액세스는 `SYS.FGA_LOG$` 테이블에 저장되고 `DBA_FGA_AUDIT_TRAIL` 보기를 통해 액세스 가능한 FGA 이벤트에 대한 액세스를 제공하지 않습니다.

DB 인스턴스에 사용 가능한 Oracle 로그 파일을 나열하는 `DescribeDBLogFiles` API 작업에서는 `MaxRecords` 파라미터를 무시하고 최대 1,000개의 레코드를 반환합니다.

보존 일정

로그 파일이 매우 커질 경우 Oracle 데이터베이스 엔진이 커진 파일들을 순환시킬 수 있습니다. 감사 또는 추적 파일을 보존하려면 해당 파일을 다운로드해야 합니다. 파일을 로컬 위치에 저장하면 Amazon RDS 스토리지 비용이 절감되고 더 많은 공간을 데이터에 저장에 사용할 수 있습니다.

다음은 Amazon RDS의 Oracle 알림 로그, 감사 파일 및 추적 파일에 대한 보존 일정입니다.

로그 유형	보존 일정
알림 로그	텍스트 알림 로그는 매일 교체되며 Amazon RDS에서 30일간 보존하도록 관리합니다. XML 알림 로그는 7일 이상 보관됩니다. <code>ALERTLOG</code> 보기 사용하여 이 로그에 액세스할 수 있습니다.
감사 파일	감사 파일의 기본 보존 기간은 7일입니다. Amazon RDS는 7일이 경과한 감사 파일을 삭제할 수 있습니다.
추적 파일	추적 파일의 기본 보존 기간은 7일입니다. Amazon RDS는 7일이 경과한 추적 파일을 삭제할 수 있습니다.
리스너 로그	리스너 로그의 기본 보존 기간은 7일입니다. Amazon RDS는 7일이 경과한 리스너 로그를 삭제할 수 있습니다.

Note

감사 파일과 추적 파일의 보존 구성은 동일합니다.

온라인 로그 파일 전환

Amazon RDS 절차 `rdsadmin.rdsadmin_util.switch_logfile`을 사용하여 온라인 로그 파일을 전환할 수 있습니다. 자세한 내용은 [온라인 로그 파일 전환 \(p. 853\)](#) 단원을 참조하십시오.

아카이빙된 다시 실행 로그 검색

보관된 다시 실행 로그를 보존할 수 있습니다. 자세한 내용은 [보관된 다시 실행 로그 보존 \(p. 856\)](#) 단원을 참조하십시오.

Oracle 추적 파일을 사용한 작업

추적 파일을 만들고 새로 고치고 액세스하고 삭제하기 위한 Amazon RDS 절차의 설명을 확인할 수 있습니다.

파일 나열

`background_dump_dest` 경로 내에 있는 임의의 파일에 대한 액세스를 허용하기 위해 두 절차 중 하나를 사용할 수 있습니다. 첫 번째 절차는 현재 `background_dump_dest`에 있는 모든 파일의 목록을 포함한 보기를 새로 고칩니다.

```
exec rdsadmin.manage_tracefiles.refresh_tracefile_listing;
```

보기가 새로 고쳐지면 다음 보기를 사용하여 결과에 액세스합니다.

```
rdsadmin.tracefile_listing
```

이전 프로세스를 대체하는 프로세스는 `FROM table`을 사용하여 테이블과 같은 형식으로 되어 있는 테이블 데이터가 아닌 데이터를 스트리밍하여 데이터베이스 딕터리 내용을 나열하는 것입니다.

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP'));
```

다음 쿼리는 로그 파일의 텍스트를 표시합니다.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','alert_xxx.log'));
```

추적 파일 생성 및 세션 추적

`alter session`에 대한 제한이 없으므로, Oracle에서 추적 파일을 생성하는 다양한 표준 메서드를 Amazon RDS DB 인스턴스에도 그대로 사용할 수 있습니다. 더 높은 액세스 권한이 필요한 추적 파일에 대해서는 다음 프로시저가 제공됩니다.

Oracle 메서드	Amazon RDS 메서드
<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>

여러 표준 메서드를 사용하여 Amazon RDS의 Oracle DB 인스턴스에 연결된 개별 세션을 추적할 수 있습니다. 세션 추적 기능을 활성화하기 위해 `DBMS_SESSION` 및 `DBMS_MONITOR` 패키지 등 Oracle에서 제공한 PL/SQL 패키지의 하위 프로그램을 실행할 수 있습니다. 자세한 내용은 Oracle 문서의 [세션 추적 기능 활성화 단원](#)을 참조하십시오.

추적 파일 검색

Amazon RDS에서 관리되는 외부 테이블의 표준 SQL 쿼리를 사용하여 `background_dump_dest`에 있는 추적 파일을 검색할 수 있습니다. 이 메서드를 사용하려면 특정 추적 파일에 대한 이 테이블의 위치를 설정하는 프로시저를 실행해야 합니다.

예를 들어 이전에 언급한 `rdsadmin.tracefile_listing` 보기와 함께 시스템 상의 모든 추적 파일을 나열할 수 있습니다. 그러면 다음 프로시저를 사용하여 의도한 추적 파일을 가리키도록 `tracefile_table` 보기와 설정할 수 있습니다.

```
exec
rdsadmin.manage_tracefiles.set_tracefile_table_location('CUST01_ora_3260_SYSTEMSTATE.trc');
```

다음 예제에서는 제공된 파일로 위치가 설정된 현재 스키마에서 외부 테이블을 만듭니다. SQL 쿼리를 사용하여 콘텐츠를 로컬 파일로 가져올 수 있습니다.

```
# eg: send the contents of the tracefile to a local file:
sqlplus user/password@TNS alias << EOF > /tmp/tracefile.txt
select * from tracefile_table;
EOF
```

추적 파일 제거

추적 파일이 누적되어 디스크 공간이 소모될 수 있습니다. Amazon RDS는 기본적으로 추적 파일을 제거하고, 7일 이상 지난 로그 파일도 제거합니다. `show_configuration` 절차를 사용하여 추적 파일 보존 기간을 보고 설정할 수 있습니다. 구성 결과를 볼 수 있도록 `SET SERVEROUTPUT ON` 명령을 실행해야 합니다.

다음 예제에서는 현재 추적 파일 보존 기간을 표시한 다음, 새 추적 파일 보존 기간을 설정합니다.

```
# 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.
```

주기적인 제거 프로세스 외에, `background_dump_dest`에서 파일을 수동으로 제거할 수 있습니다. 다음 예제에서는 5분 이상 지난 모든 파일을 제거하는 방법을 보여줍니다.

```
exec rdsadmin.manage_tracefiles.purge_tracefiles(5);
```

(.trc와 같은 파일 확장명을 포함하지 않고) 특정 패턴과 일치하는 모든 파일을 제거할 수도 있습니다. 다음 예제에서는 "SCHPOC1_ora_5935"로 시작하는 모든 파일을 제거하는 방법을 보여줍니다.

```
exec rdsadmin.manage_tracefiles.purge_tracefiles('SCHPOC1_ora_5935');
```

Amazon CloudWatch Logs에 Oracle 로그 게시

Amazon CloudWatch Logs의 로그 그룹에 로그 데이터를 게시하도록 Amazon RDS Oracle DB 인스턴스를 구성할 수 있습니다. CloudWatch Logs를 통해 로그 데이터에 대한 분석을 수행할 수 있고, CloudWatch를 사

용하여 경보를 만들고 측정치를 볼 수 있습니다. CloudWatch Logs를 사용하여 내구성이 뛰어난 스토리지에 로그 레코드를 저장할 수 있습니다.

Amazon RDS는 각 Oracle 데이터베이스 로그를 로그 그룹에 개별적인 데이터베이스 스트림으로 게시합니다. 예를 들어 감사 로그를 포함하도록 내보내기 함수를 구성하면 감사 데이터가 /aws/rds/instance/my_instance/audit 로그 그룹의 감사 로그 스트림에 저장됩니다.

콘솔

AWS Management 콘솔에서 CloudWatch Logs에 Oracle DB 로그를 게시하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 변경하려는 DB 인스턴스를 선택합니다.
3. [Modify]를 선택합니다.
4. 로그 내보내기 섹션에서 CloudWatch Logs에 게시하기 시작할 로그를 선택합니다.
5. [Continue]를 선택한 후, 요약 페이지에서 [Modify DB Instance]를 선택합니다.

AWS CLI

Oracle 로그를 게시하기 위해 다음 파라미터와 함께 `modify-db-instance` 명령을 사용할 수 있습니다.

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

`--cloudwatch-logs-export-configuration` 옵션에 대한 변경 사항은 항상 DB 인스턴스에 즉시 적용됩니다. 따라서 `--apply-immediately` 및 `--no-apply-immediately` 옵션은 지정해도 아무런 효과가 없습니다.

또한 다음 명령을 사용하여 Oracle 로그를 게시할 수 있습니다.

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Example

다음 예에서는 CloudWatch Logs 게시가 활성화된 Oracle DB 인스턴스를 생성합니다. `--enable-cloudwatch-logs-exports` 값은 문자열의 JSON 배열입니다. 문자열은 `alert`, `audit`, `listener` 및 `trace`의 조합일 수 있습니다.

Linux, OS X, Unix의 경우:

```
aws rds create-db-instance \
--db-instance-identifier mydbinstance \
--enable-cloudwatch-logs-exports '["trace","audit","alert","listener"]' \
--db-instance-class db.m1.small \
--engine oracle-se1
```

Windows의 경우:

```
aws rds create-db-instance ^
```

```
--db-instance-identifier mydbinstance ^
--enable-cloudwatch-logs-exports '['trace","audit","alert","listener"]' ^
--db-instance-class db.m1.small ^
--engine oracle-se1
```

Example

다음 예에서는 CloudWatch Logs에 로그 파일을 게시하도록 기존 Oracle DB 인스턴스를 수정합니다. --cloudwatch-logs-export-configuration 값은 JSON 객체입니다. 이 객체에 대한 키는 EnableLogTypes이며, 해당 값은 alert, audit, listener 및 trace의 조합을 사용하는 문자열의 배열입니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
  --db-instance-identifier mydbinstance \
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":'
  ["trace","alert","audit","listener"]}'
```

Windows의 경우:

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":'
  ["trace","alert","audit","listener"]}'
```

Example

다음 예에서는 CloudWatch Logs에 감사 및 리스너 로그 파일 게시를 비활성화하도록 기존 Oracle DB 인스턴스를 수정합니다. --cloudwatch-logs-export-configuration 값은 JSON 객체입니다. 이 객체에 대한 키는 DisableLogTypes이며, 해당 값은 alert, audit, listener 및 trace의 조합을 사용하는 문자열의 배열입니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
  --db-instance-identifier mydbinstance \
  --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit","listener"]}'
```

Windows의 경우:

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit","listener"]}'
```

RDS API

RDS API를 사용하여 Oracle DB 로그를 게시할 수 있습니다. 다음 파라미터로 [ModifyDBInstance](#) 작업을 호출할 수 있습니다.

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

Note

`CloudwatchLogsExportConfiguration` 파라미터에 대한 변경 사항은 항상 DB 인스턴스에 즉시 적용됩니다. 따라서 `ApplyImmediately` 파라미터는 지정해도 아무런 효과가 없습니다.

또한 다음 RDS API 작업을 호출하여 Oracle 로그를 게시할 수도 있습니다.

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`

다음 파라미터로 RDS API 작업 중 하나를 실행합니다.

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

실행하는 RDS 명령에 따라 다른 파라미터가 필요할 수 있습니다.

알림 로그 및 Listener 로그에 액세스하기 위한 이전의 메서드

Amazon RDS 콘솔을 사용하여 알림 로그를 볼 수 있습니다. 알림 로그는 다음 SQL 문을 사용하여 액세스할 수 있습니다.

```
select message_text from alertlog;
```

리스너 로그에 액세스하려면 다음 SQL 문을 사용합니다.

```
select message_text from listenerlog;
```

Note

Oracle은 알림 및 리스너 로그가 Amazon RDS 보기에서 이들 로그를 사용할 수 없게 되는 시점인 10MB를 초과할 때 이들 로그를 순환시킵니다.

PostgreSQL 데이터베이스 로그 파일

RDS PostgreSQL에서는 쿼리 로그 및 오류 로그를 생성합니다. `auto-vacuum information` 및 `rds_admin actions`를 오류 로그에 기록합니다. PostgreSQL은 로그 연결, 연결 해제 및 체크포인트를 오류 로그에 기록합니다. 자세한 내용은 PostgreSQL 문서에서 [Error Reporting and Logging](#)을 참조하십시오.

DB 인스턴스와 연결된 DB 파라미터 그룹에 있는 `rds.log_retention_period` 파라미터를 사용하여 시스템 로그의 보존 기간을 설정할 수 있습니다. 이 파라미터의 단위는 분입니다. 예를 들어 1440으로 설정하면 로그가 하루 동안 보관됩니다. 기본값은 4320(3일)입니다. 최대값은 10080(7일)입니다. 인스턴스에는 보존되는 로그 파일을 포함하기에 충분하게 할당된 스토리지가 있어야 합니다.

DB 인스턴스와 연결된 DB 파라미터 그룹에서 `log_statement` 및 `log_min_duration_statement`의 두 파라미터를 설정하여 PostgreSQL DB 인스턴스에 대한 쿼리 로깅을 활성화할 수 있습니다.

`log_statement` 파라미터는 어떤 SQL 문이 로그에 기록되는지 제어합니다. 이 파라미터를 `all`로 설정하여 DB 인스턴스에서 문제를 디버깅할 때 모든 문을 로그에 기록하는 것이 좋습니다. 기본값은 `none`입니다. 또는 이 값을 `ddl`로 설정하여 모든 데이터 정의 언어(DDL) 문(CREATE, ALTER, DROP 등)을 로깅하거나

mod로 설정하여 모든 DDL 및 데이터 수정 언어(DML) 문(INSERT, UPDATE, DELETE 등)을 로깅할 수 있습니다.

`log_min_duration_statement` 파라미터는 로그에 기록할 문의 한계를 밀리초 단위로 설정합니다. 이 파라미터 설정보다 오래 실행되는 모든 SQL 문이 로그에 기록됩니다. 이 파라미터는 기본적으로 비활성화되어 있고 -1로 설정됩니다. 이 파라미터를 활성화하면 최적화되지 않은 쿼리를 찾는데 도움이 될 수 있습니다.

처음으로 DB 파라미터 그룹에서 파라미터를 설정하고 그 파라미터 그룹을 DB 인스턴스와 연결해보는 사용자라면, [DB 파라미터 그룹 작업 \(p. 161\)](#)을(를) 참조하십시오.

다음 단계에서는 쿼리 로깅을 설정하는 방법을 보여줍니다.

1. `log_statement` 파라미터를 `all`로 설정합니다. 다음 예제에서는 `postgres.log` 파일에 기록되는 정보를 보여줍니다.

```
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"
```

쿼리를 실행할 때 `postgres.log` 파일에 추가 정보가 기록됩니다. 다음 예제에서는 쿼리 후 파일에 기록되는 정보의 유형을 보여줍니다.

```
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. `log_min_duration_statement` 파라미터를 설정합니다. 다음 예제에서는 이 파라미터가 1로 설정되어 있을 때 `postgres.log` 파일에 기록되는 정보를 보여줍니다.

```
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"
```

지속 시간 파라미터 설정을 조과하는 쿼리를 실행할 때 `postgres.log` 파일에 추가 정보가 기록됩니다. 다음 예제에서는 쿼리 후 파일에 기록되는 정보의 유형을 보여줍니다.

```
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.reltablespace
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
```

CloudWatch Logs에 PostgreSQL 로그 게시

Amazon CloudWatch Logs의 로그 그룹에 로그 데이터를 게시하도록 PostgreSQL용 Amazon RDS DB 인스턴스를 구성할 수 있습니다. CloudWatch Logs를 통해 로그 데이터에 대한 실시간 분석을 수행할 수 있고, CloudWatch를 사용하여 경보를 만들고 지표를 볼 수 있습니다. CloudWatch Logs를 사용하여 내구성이 뛰어난 스토리지에 로그 레코드를 저장할 수 있습니다.

Note

로그 파일을 CloudWatch Logs에 게시하는 기능은 PostgreSQL 버전 9.6.6 이상과 10.4 이상에만 지원됩니다.

PostgreSQL용 Amazon RDS의 CloudWatch Logs에 게시할 수 있는 로그 유형은 다음과 같습니다.

- Postgresql 로그
- 업그레이드 로그

구성을 완료하고 나면 Amazon RDS에서 CloudWatch 로그 그룹 내 로그 스트림에 로그 이벤트를 게시합니다. 예를 들면 PostgreSQL 로그 데이터가 로그 그룹 /aws/rds/instance/*my_instance*/postgresql 안에 저장됩니다. Amazon CloudWatch Logs를 보려면 <https://console.aws.amazon.com/cloudwatch/>를 엽니다.

AWS Management 콘솔

콘솔을 사용하여 PostgreSQL 로그를 CloudWatch Logs에 게시하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 수정할 DB 인스턴스를 선택한 후, 수정을 선택합니다.
4. Log exports(로그 내보내기) 섹션에서 CloudWatch Logs에 게시하기 시작할 로그를 선택합니다.
5. [Continue]를 선택한 후, 요약 페이지에서 [Modify DB Instance]를 선택합니다.

AWS CLI

AWS CLI를 사용하여 PostgreSQL 로그를 게시할 수 있습니다. 다음 파라미터로 `modify-db-instance` 명령을 호출할 수 있습니다.

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

Note

`--cloudwatch-logs-export-configuration` 옵션에 대한 변경 사항은 항상 DB 인스턴스에 즉시 적용됩니다. 따라서 `--apply-immediately` 및 `--no-apply-immediately` 옵션은 지정해도 아무런 효과가 없습니다.

또한 다음 AWS CLI 명령을 호출하여 PostgreSQL 로그를 게시할 수도 있습니다.

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

다음 옵션으로 AWS CLI 명령 중 하나를 실행합니다.

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

실행하는 AWS CLI 명령에 따라 다른 옵션이 필요할 수 있습니다.

로그를 CloudWatch Logs에 게시하도록 인스턴스 설정

Example

다음 예에서는 CloudWatch Logs에 로그 파일을 게시하도록 기존 PostgreSQL DB 인스턴스를 수정합니다. `--cloudwatch-logs-export-configuration` 값은 JSON 객체입니다. 이 객체에 대한 키는 `EnableLogTypes`이며, 해당 값은 `postgresql` 및 `upgrade`의 조합을 사용하는 문자열의 배열입니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
  --db-instance-identifier mydbinstance \
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":["postgresql", "upgrade"]}'
```

Windows의 경우:

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --cloudwatch-logs-export-configuration '{"EnableLogTypes":["postgresql", "upgrade"]}'
```

Example

로그를 CloudWatch Logs에 게시할 인스턴스 생성

다음 예에서는 CloudWatch Logs에 로그 파일을 게시하도록 PostgreSQL DB 인스턴스를 생성합니다. `--enable-cloudwatch-logs-exports` 값은 문자열의 JSON 배열입니다. 문자열은 `postgresql` 및 `upgrade`의 조합일 수 있습니다.

Linux, OS X, Unix의 경우:

```
aws rds create-db-instance \
  --db-instance-identifier mydbinstance \
  --enable-cloudwatch-logs-exports '[{"postgreSQL": "upgrade"}]' \
  --db-instance-class db.m4.large \
  --engine postgres
```

Windows의 경우:

```
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

RDS API를 사용하여 PostgreSQL 로그를 게시할 수 있습니다. 다음 파라미터로 [ModifyDBInstance](#) 작업을 호출할 수 있습니다.

- [DBInstanceIdentifier](#)
- [CloudwatchLogsExportConfiguration](#)

Note

[CloudwatchLogsExportConfiguration](#) 파라미터에 대한 변경 사항은 항상 DB 인스턴스에 즉시 적용됩니다. 따라서 [ApplyImmediately](#) 파라미터는 지정해도 아무런 효과가 없습니다.

또한 다음 RDS API 작업을 호출해 PostgreSQL 로그를 게시할 수 있습니다.

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

다음 파라미터로 RDS API 작업 중 하나를 실행합니다.

- [DBInstanceIdentifier](#)
- [EnableCloudwatchLogsExports](#)
- [Engine](#)
- [DBInstanceClass](#)

실행하는 작업에 따라 다른 파라미터가 필요할 수 있습니다.

AWS CloudTrail을 사용하여 Amazon RDS API 호출 로깅

Amazon RDS는 Amazon RDS의 사용자, 역할 또는 AWS 서비스가 수행한 작업에 대한 레코드를 제공하는 서비스인 AWS CloudTrail과 통합됩니다. CloudTrail은 Amazon RDS 콘솔의 호출 및 Amazon RDS API 코드 호출 등 Amazon RDS에 대한 모든 API 호출을 이벤트로 캡처합니다. 추적을 생성하면 Amazon RDS에 대한 이벤트를 포함해 CloudTrail 이벤트를 Amazon S3 버킷에 지속적으로 제공할 수 있습니다. 추적을 구성하지 않은 경우 Event history(이벤트 기록)에서 CloudTrail 콘솔의 최신 이벤트를 볼 수도 있습니다. CloudTrail에서 수집하는 정보를 사용하여 Amazon RDS에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 [AWS CloudTrail User Guide](#)를 참조하십시오.

CloudTrail의 Amazon RDS 정보

CloudTrail은 계정 생성 시 AWS 계정에서 활성화됩니다. Amazon RDS에서 활동이 수행되면 해당 활동은 이벤트 기록에서 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤

트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 [CloudTrail 이벤트 기록에서 이벤트 보기](#)를 참조하십시오.

Amazon RDS 이벤트를 비롯하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 생성하십시오. 추적은 CloudTrail이 Amazon S3 버킷으로 로그 파일을 전송할 수 있도록 합니다. 콘솔에서 추적을 생성하면 기본적으로 모든 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또는 CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 처리하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음을 참조하십시오.

- [추적 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기](#)

모든 Amazon RDS 작업이 CloudTrail에서 로깅되고 [Amazon RDS API Reference](#)에 문서화됩니다. 예를 들어, `CreateDBInstance`, `ModifyDBInstance` 및 `CreateDBParameterGroup` 작업을 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 자격 증명 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 IAM 사용자 자격 증명으로 했는지 여부
- 역할 또는 연합된 사용자에 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 [CloudTrail userIdentity 요소](#)를 참조하십시오.

Amazon RDS 로그 파일 항목 이해

추적은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 제공할 수 있도록 해 주는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함됩니다. 이벤트는 어떤 소스로부터의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 호출의 주문 스택 추적이 아니기 때문에 특정 순서로 표시되지 않습니다.

다음은 `CreateDBInstance` 작업을 보여 주는 CloudTrail 로그 항목이 나타낸 예제입니다.

```
{  
    "eventVersion": "1.04",  
    "userIdentity": {  
        "type": "IAMUser",  
        "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": "72.21.198.65",  
    "userAgent": "aws-cli/1.15.42 Python/3.6.1 Darwin/17.7.0 botocore/1.10.42",  
    "requestParameters": {  
        "enableCloudwatchLogsExports": [  
            "audit",  
            "error",  
            "general",  
            "replica",  
            "statement",  
            "transaction"  
        ]  
    }  
}
```

```
        "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"
}
```

Amazon RDS의 보안 구성

DB 인스턴스에서 Amazon RDS 리소스 및 데이터베이스에 대한 액세스를 관리할 수 있습니다. 액세스에 사용하는 방법은 사용자가 Amazon RDS를 사용하여 수행해야 하는 작업 유형에 따라 다릅니다.

- 네트워크 액세스 제어를 최대한 강화하려면 VPC(Amazon Virtual Private Cloud)에서 DB 인스턴스를 실행합니다. VPC에서 DB 인스턴스를 생성하는 방법에 대한 자세한 내용은 [Amazon Virtual Private Cloud\(VPC\)와 함께 Amazon RDS 사용](#)을 참조하십시오.
- RDS 리소스를 관리할 수 있는 사용자를 결정하는 권한을 배정하려면 AWS Identity and Access Management(IAM) 정책을 사용합니다. 예를 들면, IAM을 사용하여 DB 인스턴스, 태그 리소스를 생성, 설명, 수정, 삭제하거나 보안 그룹을 수정할 수 있는 사용자를 결정할 수 있습니다.
- 보안 그룹을 사용하여 어떤 IP 주소 또는 Amazon EC2 인스턴스가 DB 인스턴스에 있는 데이터베이스에 연결할 수 있는지 제어합니다. DB 인스턴스를 처음 생성하면, DB 인스턴스 방화벽에서 연결된 보안 그룹에서 지정한 규칙 이외의 데이터베이스 액세스를 차단합니다.
- MySQL, MariaDB, PostgreSQL, Oracle 또는 Microsoft SQL Server 데이터베이스 엔진을 실행하는 DB 인스턴스에 Secure Socket Layer(SSL) 연결을 사용합니다. DB 인스턴스에서 SSL 사용에 대한 자세한 내용은 [SSL을 사용하여 DB 인스턴스에 대한 연결 암호화](#) (p. 388) 단원을 참조하십시오.
- RDS 암호화를 사용하여 휴면 상태의 RDS 인스턴스와 스냅샷의 보안을 유지합니다. RDS 암호화는 RDS 인스턴스를 호스팅하는 서버의 데이터를 업계 표준 AES-256 암호화 알고리즘을 사용하여 암호화합니다. 자세한 내용은 [Amazon RDS 리소스 암호화](#) (p. 385) 단원을 참조하십시오.
- Oracle DB 인스턴스와 함께 네트워크 암호화 및 Transparent Data Encryption을 사용합니다. 자세한 내용은 [Oracle 기본 네트워크 암호화](#) (p. 807) 및 [Oracle Transparent Data Encryption](#) (p. 827) 단원을 참조하십시오.
- 데이터베이스가 로컬 네트워크에 있을 때처럼 DB 엔진의 보안 기능을 사용하여 DB 인스턴스에 있는 데이터베이스에 누가 로그인할 수 있는지 제어합니다.

Note

사용 사례에 따라 보안을 구성하기만 하면 됩니다. 백업 생성, 마스터와 읽기 전용 복제본 간 복제 또는 기타 프로세스와 같이 Amazon RDS가 관리하는 프로세스에 대한 보안 액세스는 구성하지 않아도 됩니다.

Amazon RDS 리소스를 비롯해 DB 인스턴스의 데이터베이스에 대한 액세스 관리는 아래 주제를 참조하십시오.

주제

- [인증 및 액세스 제어](#) (p. 339)
- [Amazon RDS 리소스 암호화](#) (p. 385)
- [SSL을 사용하여 DB 인스턴스에 대한 연결 암호화](#) (p. 388)
- [보안 그룹을 통한 액세스 제어](#) (p. 390)
- [마스터 사용자 계정 권한](#) (p. 402)
- [Amazon RDS에 서비스 연결 역할 사용](#) (p. 403)
- [Amazon Virtual Private Cloud\(VPC\) 및 Amazon RDS](#) (p. 406)

인증 및 액세스 제어

Amazon RDS에 액세스하려면 AWS가 요청을 인증하는 데 사용할 수 있는 자격 증명이 필요합니다. 이러한 자격 증명에는 AWS 리소스(예: Amazon RDS DB 인스턴스)에 액세스할 수 있는 권한이 있어야 합니다. 다

음 섹션에서는 리소스에 액세스할 수 있는지 대상을 제어하여 리소스를 보호할 수 있도록 [AWS Identity and Access Management\(IAM\)](#) 및 Amazon RDS를 사용하는 방법에 대한 세부 정보를 제공합니다.

- [인증 \(p. 340\)](#)
- [액세스 제어 \(p. 341\)](#)

인증

다음과 같은 자격 증명 유형으로 AWS에 액세스할 수 있습니다.

- AWS 계정 루트 사용자 – AWS 계정을 처음 생성할 때는 해당 계정의 모든 AWS 서비스와 리소스에 대해 완전한 액세스 권한이 있는 SSO(single sign-in) 자격 증명으로 시작합니다. 이 자격 증명은 AWS 계정 루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 관리 작업이라 할지라도 일상적인 작업에 루트 사용자를 사용하지 마십시오. 대신, [IAM 사용자를 처음 생성할 때만 루트 사용자를 사용하는 모범 사례](#)를 준수하십시오. 그런 다음 루트 사용자를 안전하게 보관해 두고 몇 가지 계정 및 서비스 관리 작업을 수행할 때만 자격 증명을 사용합니다.
- IAM 사용자 – [IAM 사용자는](#) 특정 사용자 지정 권한(예: Amazon RDS에서 a DB instance을 만들 권한)이 있는 AWS 계정 내의 자격 증명입니다. IAM 사용자 이름과 암호를 사용하여 [AWS Management 콘솔](#), [AWS 토큰 포럼](#) 또는 [AWS Support Center](#)와 같은 보안 AWS 웹 페이지에 로그인할 수 있습니다.

사용자 이름과 암호 외에도 각 사용자에 대해 [액세스 키](#)를 생성할 수 있습니다. [여러 SDK 중 하나](#)를 통해 또는 [AWS Command Line Interface\(CLI\)](#)를 사용하여 AWS 서비스에 프로그래밍 방식으로 액세스할 때 이러한 키를 사용할 수 있습니다. SDK 및 CLI 도구는 액세스 키를 사용하여 암호화 방식으로 요청에 서명합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. Amazon RDS supports는 인바운드 API 요청을 인증하기 위한 프로토콜인 서명 버전 4를 지원합니다. 요청 인증에 대한 자세한 내용은 AWS General Reference의 [서명 버전 4 서명 프로세스](#)를 참조하십시오.

- IAM 역할 - [IAM 역할은](#) 특정 권한을 가진 계정에 생성할 수 있는 IAM 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. IAM 역할을 사용하면 AWS 서비스 및 리소스에 액세스하는 데 사용할 수 있는 임시 액세스 키를 얻을 수 있습니다. 임시 자격 증명이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.
 - **연합된 사용자 액세스** – IAM 사용자를 만드는 대신 AWS Directory Service의 기준 사용자 자격 증명, 엔터프라이즈 사용자 디렉터리 또는 웹 자격 증명 공급자를 사용할 수 있습니다. 이러한 사용자를 연합된 사용자라고 합니다. [자격 증명 공급자](#)를 통해 액세스를 요청하면 AWS가 연합된 사용자에게 역할을 할당합니다. 연합된 사용자에 대한 자세한 내용은 IAM 사용 설명서의 [연합된 사용자 및 역할](#)을 참조하십시오.
- AWS 서비스 액세스 – 계정의 IAM 역할을 사용하여 AWS 서비스에 계정의 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 예를 들어 Amazon Redshift에서 사용자 대신 Amazon S3 버킷에 액세스하도록 허용하는 역할을 만든 후 그 버킷으로부터 데이터를 Amazon Redshift 클러스터로 로드할 수 있습니다. 자세한 내용은 [IAM 사용 설명서](#)의 Creating a Role to Delegate Permissions to an AWS Service 단원을 참조하십시오.
- Amazon EC2에서 실행하는 애플리케이션 – EC2 인스턴스에서 실행되고 AWS API 요청을 하는 애플리케이션의 임시 자격 증명을 IAM 역할을 사용하여 관리할 수 있습니다. 이것은 EC2 인스턴스 내에 액세스 키를 저장하는 경우에 바람직한 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 그 역할을 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 만들어야 합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 자격 증명을 사용할 수 있습니다.

명을 얻을 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 대한 부여하기](#) 섹션을 참조하십시오.

액세스 제어

요청을 인증하는 데 유용한 자격 증명이 있더라도 권한이 없다면 Amazon RDS 리소스를 생성하거나 액세스 할 수 없습니다. 예를 들어, Amazon RDS DB 인스턴스를 추가하거나, DB 스냅샷을 생성하거나, 이벤트 구독을 추가하려면 권한이 있어야 합니다.

다음 섹션에서는 Amazon RDS에 대한 권한을 관리하는 방법에 대해 설명합니다. 먼저 개요를 읽어 보면 도움이 됩니다.

- [Amazon RDS 리소스에 대한 액세스 권한 관리 개요](#) (p. 341)
- [Amazon RDS에 대한 자격 증명 기반 정책\(IAM 정책\) 사용](#) (p. 344)

Amazon RDS 리소스에 대한 액세스 권한 관리 개요

모든 AWS 리소스는 AWS 계정의 소유이고, 리소스 생성 또는 액세스 권한은 권한 정책에 따라 결정됩니다. 계정 관리자는 IAM 자격 증명(즉, 사용자, 그룹, 역할)에 권한 정책을 연결할 수 있고, 일부 서비스(예: AWS Lambda)에서는 리소스에 대한 권한 정책 연결도 지원합니다.

Note

계정 관리자 또는 관리자 사용자는 관리자 권한이 있는 사용자입니다. 자세한 내용은 IAM 사용 설명서에서 [IAM 모범 사례](#)를 참조하십시오.

권한을 부여하려면 권한을 부여 받을 사용자, 권한 대상이 되는 리소스, 해당 리소스에 허용되는 특정 작업을 결정합니다.

주제

- [Amazon RDS 리소스와 작업](#) (p. 341)
- [리소스 소유권 이해](#) (p. 342)
- [리소스 액세스 관리](#) (p. 342)
- [정책 요소 지정: 작업, 효과, 리소스, 보안 주체](#) (p. 344)
- [정책에서 조건 지정](#) (p. 344)

Amazon RDS 리소스와 작업

Amazon RDS에서 기본 리소스는 DB 인스턴스입니다. Amazon RDS는 기본 리소스와 함께 사용할 수 있는 다른 리소스(예: DB 스냅샷, 파라미터 그룹, 이벤트 구독 등)를 지원합니다. 이러한 리소스를 가리켜 하위 리소스라 합니다.

다음 표에 나와 있는 것처럼 이러한 리소스와 하위 리소스에는 고유한 Amazon 리소스 이름(ARN)이 연결되어 있습니다.

리소스 유형	ARN 형식
DB 클러스터	<code>arn:aws:rds:<i>region</i>:<i>account-id</i>:cluster:<i>db-cluster-name</i></code>
DB 클러스터 파라미터 그룹	<code>arn:aws:rds:<i>region</i>:<i>account-id</i>:cluster-pg:<i>cluster-parameter-group-name</i></code>

리소스 유형	ARN 형식
DB 클러스터 스냅샷	<code>arn:aws:rds:region:account-id:cluster-snapshot:cluster-snapshot-name</code>
DB 인스턴스	<code>arn:aws:rds:region:account-id:db:db-instance-name</code>
DB 옵션 그룹	<code>arn:aws:rds:region:account-id:og:option-group-name</code>
DB 파라미터 그룹	<code>arn:aws:rds:region:account-id:pg:parameter-group-name</code>
DB 스냅샷	<code>arn:aws:rds:region:account-id:snapshot:snapshot-name</code>
DB 보안 그룹	<code>arn:aws:rds:region:account-id:secgrp:security-group-name</code>
DB 서브넷 그룹	<code>arn:aws:rds:region:account-id:subgrp:subnet-group-name</code>
이벤트 구독	<code>arn:aws:rds:region:account-id:es:subscription-name</code>
읽기 전용 복제본	<code>arn:aws:rds:region:account-id:db:db-instance-name</code>
예약 DB 인스턴스	<code>arn:aws:rds:region:account-id:ri:reserved-db-instance-name</code>

Amazon RDS(은)는 Amazon RDS 리소스를 처리하기 위한 작업을 제공합니다. 사용 가능한 작업 목록은 [작업 단원](#)을 참조하십시오.

리소스 소유권 이해

리소스 소유자는 리소스를 만든 AWS 계정입니다. 즉, 리소스 소유자는 리소스를 생성하는 요청을 인증하는 보안 주체 엔터티(루트 계정, IAM 사용자 또는 IAM 역할)의 AWS 계정입니다. 다음 예제에서는 이 계정의 작동 방식을 설명합니다.

- AWS 계정의 루트 계정 자격 증명을 사용하여 DB 인스턴스와 같은 RDS 리소스를 만들면 AWS 계정은 RDS 리소스의 소유자가 됩니다.
- AWS 계정에서 IAM 사용자를 만들고 RDS 리소스를 생성할 수 있는 권한을 해당 사용자에게 부여하면 해당 사용자는 RDS 리소스를 생성할 수 있습니다. 하지만 해당 사용자가 속한 AWS 계정이 RDS 리소스를 소유합니다.
- AWS 계정에서 RDS 리소스를 생성할 권한이 있는 IAM 역할을 만드는 경우 해당 역할을 담당할 수 있는 사람은 누구나 RDS 리소스를 만들 수 있습니다. 이 경우 역할이 속한 AWS 계정이 RDS 리소스를 소유합니다.

리소스 액세스 관리

권한 정책은 누가 무엇에 액세스 할 수 있는지를 나타냅니다. 다음 단원에서는 권한 정책을 만드는 데 사용 가능한 옵션에 대해 설명합니다.

Note

이 단원에서는 Amazon RDS의 맥락에서 IAM을 사용하는 방법에 대해 설명하며, IAM 서비스에 대한 자세한 내용은 다루지 않습니다. 전체 IAM 설명서는 IAM 사용 설명서의 [IAM이란 무엇인가?](#)를 참조하십시오. IAM 정책 구문과 설명에 대한 자세한 내용은 IAM 사용 설명서의 [AWS IAM 정책 참조](#)를 참조하십시오.

IAM 자격 증명에 연결된 정책을 자격 증명 기반 정책(IAM 정책)이라고 하고, 리소스에 연결된 정책을 리소스 기반 정책이라고 합니다. Amazon RDS의 경우에는 자격 증명 기반 정책(IAM 정책)만 지원합니다.

주제

- 자격 증명 기반 정책(IAM 정책) (p. 343)
- 리소스 기반 정책 (p. 344)

자격 증명 기반 정책(IAM 정책)

정책을 IAM 자격 증명에 연결할 수 있습니다. 예를 들면,

- 계정 내 사용자 또는 그룹에 관한 정책 연결 – 계정 관리자는 특정 사용자에 연결된 권한 정책을 사용하여 해당 사용자에게 Amazon RDS 리소스(예: DB 인스턴스) 생성 권한을 부여할 수 있습니다.
- 역할에 관한 정책 연결(교차 계정 권한 부여) – 자격 증명 기반 권한 정책을 IAM 역할에 연결하여 교차 계정 권한을 부여할 수 있습니다. 예를 들어, 계정 A의 관리자는 다음과 같이 다른 AWS 계정(예: 계정 B) 또는 AWS 서비스에 교차 계정 권한을 부여할 역할을 생성할 수 있습니다.
 1. 계정 A 관리자는 IAM 역할을 생성하고 계정 A의 리소스에 대한 권한을 부여하는 역할에 관한 정책을 연결합니다.
 2. 계정 A 관리자는 계정 B를 역할을 수임할 보안 주체로 식별하는 역할에 신뢰 정책을 연결합니다.
 3. 계정 B 관리자는 계정 B의 사용자에게 역할을 수임할 권한을 위임할 수 있습니다. 그러면 계정 B의 사용자가 계정 A에서 리소스를 생성하거나 액세스할 수 있습니다. AWS 서비스에 역할 수임 권한을 부여할 경우 신뢰 정책의 보안 주체가 AWS 서비스 보안 주체이기도 합니다.

IAM을 사용하여 권한을 위임하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [액세스 관리를 참조하십시오.](#)

다음은 ID가 123456789012인 사용자에게 AWS 계정에 대한 DB 인스턴스를 생성하도록 허용하는 정책 예제입니다. 이 정책에 따라 새 DB 인스턴스의 이름은 test로 시작해야 합니다. 또한 새 DB 인스턴스는 MySQL 데이터베이스 엔진 및 db.t2.micro DB 인스턴스 클래스를 사용해야 합니다. 또한 새로운 DB 인스턴스는 default로 시작하는 옵션 그룹과 DB 파라미터 그룹을, 그리고 default 서브넷 그룹을 사용해야 합니다.

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

Amazon RDS에서 자격 증명 기반 정책을 사용하는 방법에 대한 자세한 내용은 [Amazon RDS에 대한 자격 증명 기반 정책\(IAM 정책\) 사용 \(p. 344\)](#) 단원을 참조하십시오. 사용자, 그룹, 역할 및 권한에 대한 자세한 내용은 IAM 사용 설명서의 [자격 증명\(사용자, 그룹 및 역할\)](#)을 참조하십시오.

리소스 기반 정책

Amazon S3와 같은 다른 서비스도 리소스 기반 권한 정책을 지원합니다. 예를 들어, 정책을 S3 버킷에 연결하여 해당 버킷에 대한 액세스 권한을 관리할 수 있습니다. Amazon RDS는 리소스 기반 정책을 지원하지 않습니다.

정책 요소 지정: 작업, 효과, 리소스, 보안 주체

각 Amazon RDS 리소스([Amazon RDS 리소스와 작업 \(p. 341\)](#) 참조)에 대해 서비스는 API 작업을 정의합니다([작업](#) 참조). 이러한 API 작업에 대한 권한을 부여하기 위해 Amazon RDS에서는 정책에서 지정할 수 있는 작업을 정의합니다. API 작업을 실시하려면 둘 이상의 작업에 대한 권한이 필요할 수 있습니다.

다음은 기본 정책 요소입니다.

- 리소스 – 정책에서 Amazon 리소스 이름(ARN)을 사용하여 정책을 적용할 리소스를 식별합니다. 자세한 내용은 [Amazon RDS 리소스와 작업 \(p. 341\)](#) 단원을 참조하십시오.
- 작업 – 작업 키워드를 사용하여 허용 또는 거부할 리소스 작업을 식별합니다. 예를 들어, `rds:DescribeDBInstances` 권한은 사용자에게 Amazon RDS `DescribeDBInstances` 작업 수행 권한을 허용합니다.
- 결과 – 사용자가 특정 작업을 요청하는 경우의 결과를 지정합니다. 이는 허용 또는 거부 중에 하나가 될 수 있습니다—. 명시적으로 리소스에 대한 액세스 권한을 부여(허용)하지 않는 경우, 액세스는 목시적으로 거부됩니다. 다른 정책에서는 액세스 권한을 부여하더라도 리소스에 대한 액세스를 명시적으로 거부하여 사용자가 해당 리소스에 액세스하지 못하게 할 수도 있습니다.
- 보안 주체 – 자격 증명 기반 정책(IAM 정책)에서 정책이 연결되는 사용자는 암시적인 보안 주체입니다. 리소스 기반 정책의 경우 사용자, 계정, 서비스 또는 권한의 수신자인 기타 개체를 지정합니다(리소스 기반 정책에만 해당). Amazon RDS의 경우 리소스 기반 정책을 지원하지 않습니다.

IAM 정책 구문과 설명에 대한 자세한 내용은 IAM 사용 설명서의 [AWS IAM 정책 참조](#)를 참조하십시오.

모든 Amazon RDS API 작업과 해당 작업이 적용되는 리소스를 보여주는 표는 [Amazon RDS API 권한: 작업, 리소스 및 조건 참조 \(p. 348\)](#) 단원을 참조하십시오.

IAM 정책은 IAM 정책 시뮬레이터로 테스트할 수 있습니다. 이 시뮬레이터는 Amazon RDS 작업을 포함하여 각 AWS 작업에 필요한 리소스 및 파라미터 목록을 자동으로 제공합니다. 또한 IAM 정책은 사용자가 지정하는 작업마다 필요한 권한을 결정합니다. IAM 정책 시뮬레이터에 대한 자세한 내용은 IAM 사용 설명서에서 [IAM 정책 시뮬레이터로 IAM 정책 테스트하기](#) 단원을 참조하십시오.

정책에서 조건 지정

권한을 부여할 때 액세스 정책 언어를 사용하여 조건이 적용되는 조건을 지정할 수 있습니다. 예를 들어, 특정 날짜 이후에만 정책을 적용할 수 있습니다. 정책 언어에서의 조건 지정에 관한 자세한 내용은 IAM 사용 설명서의 [조건](#) 단원을 참조하십시오.

조건을 표시하려면 미리 정의된 조건 키를 사용합니다. AWS 차원 조건 키와 RDS 지정 키를 적절하게 사용할 수 있습니다. AWS 차원 키의 전체 목록은 IAM 사용 설명서의 [사용 가능한 조건 키](#)를 참조하십시오. RDS 지정 키에 대한 전체 목록은 [IAM 정책 조건을 사용하여 세부적인 액세스 제어 구현 \(p. 363\)](#) 단원을 참조하십시오.

Amazon RDS에 대한 자격 증명 기반 정책(IAM 정책) 사용

이 항목에서는 계정 관리자가 IAM 자격 증명(사용자, 그룹, 역할)에 권한 정책을 연결할 수 있는 자격 증명 기반 정책의 예를 제공합니다.

Important

Amazon RDS 리소스에 대한 액세스 관리를 위해 제공되는 기본 개념과 옵션 설명에 대한 소개 주제 부분을 우선 읽어 보는 것이 좋습니다. 자세한 내용은 [Amazon RDS 리소스에 대한 액세스 권한 관리 개요 \(p. 341\)](#) 단원을 참조하십시오.

이 주제의 섹션에서는 다음 내용을 학습합니다.

- [Amazon RDS 콘솔 사용에 필요한 권한 \(p. 346\)](#)
- [Amazon RDS에 대한 AWS 관리형\(미리 정의된\) 정책 \(p. 346\)](#)
- [고객 관리형 정책 예 \(p. 346\)](#)

다음은 IAM 정책 예제입니다.

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

정책은 다음 IAM 사용자 권한을 지정하는 단일 명령문을 포함합니다.

- 정책은 IAM 사용자가 [CreateDBInstance API](#) 작업을 사용하여 DB 인스턴스를 생성할 수 있도록 허용합니다. 이는 [create-db-instance](#) AWS CLI 명령과 AWS Management 콘솔에도 적용됩니다.
- **Resource** 요소는 사용자가 리소스 위치에서 또는 리소스를 사용하여 작업을 수행할 수 있도록 지정합니다. 리소스는 Amazon 리소스 이름(ARN)을 사용하여 지정합니다. 이 ARN에는 리소스가 속하는 서비스 이름(rds), AWS 리전(*)는 위의 예제에서 사용하는 모든 리전을 의미함), 사용자 계정 번호(위의 예제에서 사용하는 사용자 ID는 123456789012임), 그리고 리소스 유형이 포함됩니다. ARN 생성에 대한 자세한 내용은 [Amazon RDS의 Amazon 리소스 이름\(ARN\)을 사용한 작업 \(p. 172\)](#) 단원을 참조하십시오.

위의 예제에서 **Resource** 요소는 사용자 리소스에 대해 다음과 같은 정책 제약 조건을 지정합니다.

- 새 DB 인스턴스에 대한 DB 인스턴스 식별자는 test로 시작해야 합니다(예: testCustomerData1, test-region2-data).
- 새로운 DB 인스턴스의 옵션 그룹은 default로 시작해야 합니다.
- 새로운 DB 인스턴스의 DB 파라미터 그룹은 default로 시작해야 합니다.
- 새로운 DB 인스턴스의 서브넷 그룹은 default 서브넷 그룹이 되어야 합니다.
- **Condition** 요소는 DB 엔진은 MySQL이 되고, DB 인스턴스 클래스는 db.t2.micro가 되도록 지정합니다. **Condition** 요소는 정책 적용 시 조건을 지정합니다. 그 밖에도 **Condition** 요소를 사용하여 다른 권

한이나 제한을 추가할 수 있습니다. 조건 지정에 대한 자세한 내용은 [IAM 정책 조건을 사용하여 세부적인 액세스 제어 구현 \(p. 363\)](#) 단원을 참조하십시오.

자격 증명 기반 정책에서는 권한을 가질 보안 주체를 지정하지 않으므로 이 정책은 Principal 요소를 지정하지 않습니다. 정책을 사용자에게 연결할 경우 사용자는 암시적인 보안 주체입니다. IAM 역할에 권한 정책을 연결할 경우 역할의 신뢰 정책에 식별된 보안 주체는 권한을 가집니다.

모든 Amazon RDS API 작업과 해당 작업이 적용되는 리소스를 보여주는 표는 [Amazon RDS API 권한: 작업, 리소스 및 조건 참조 \(p. 348\)](#) 단원을 참조하십시오.

Amazon RDS 콘솔 사용에 필요한 권한

Amazon RDS 콘솔에서 작업하려면 최소한의 권한이 사용자에게 필요합니다. 이러한 권한이 있어야만 사용자가 자신의 AWS 계정에서 사용할 Amazon RDS 리소스를 설명하고, Amazon EC2 보안 및 네트워크 정보 등 다른 관련 정보를 입력할 수 있습니다.

최소 필수 권한보다 더 제한적인 IAM 정책을 만들면 콘솔은 해당 IAM 정책에 연결된 사용자에 대해 의도대로 작동하지 않습니다. 이 사용자가 Amazon RDS 콘솔을 사용할 수 있도록 하려면 [AmazonRDSReadOnlyAccess 관리형 정책을 사용자에게 연결합니다\(Amazon RDS에 대한 AWS 관리형 \(미리 정의된\) 정책 \(p. 346\) 참조\)](#).

AWS CLI 또는 Amazon RDS API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요가 없습니다.

Amazon RDS에 대한 AWS 관리형(미리 정의된) 정책

AWS는 AWS에서 생성하고 관리하는 독립형 IAM 정책을 제공하여 많은 일반 사용 사례를 처리합니다. 관리형 정책은 사용자가 필요한 권한을 조사할 필요가 없도록 일반 사용 사례에 필요한 권한을 부여합니다. 자세한 내용은 IAM 사용 설명서의 [AWS 관리형 정책](#) 단원을 참조하십시오.

계정의 사용자에게 연결할 수 있는 다음 AWS 관리형 정책은 Amazon RDS에 대해 고유합니다.

- [AmazonRDSReadOnlyAccess](#) – 루트 AWS 계정의 모든 Amazon RDS 리소스에 대한 읽기 전용 액세스를 부여합니다.
- [AmazonRDSFullAccess](#) – 루트 AWS 계정의 모든 Amazon RDS 리소스에 대한 전체 액세스를 부여합니다.

사용자에게 필요한 Amazon RDS API 작업 및 리소스에 액세스하도록 허용하는 사용자 지정 IAM 정책을 생성할 수도 있습니다. 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 사용자 지정 정책을 연결할 수 있습니다.

고객 관리형 정책 예

이 단원에서는 다양한 Amazon RDS 작업에 대한 권한을 부여하는 사용자 정책의 예를 제공합니다. 이러한 정책은 RDS API 작업, AWS SDK 또는 를 사용하는 경우에 유효합니다. 콘솔을 사용하는 경우 [Amazon RDS 콘솔 사용에 필요한 권한 \(p. 346\)](#)의 설명과 같이 콘솔에 특정한 추가 권한을 부여해야 합니다.

Note

모든 예제에서는 미국 서부(오리건) 리전(us-west-2)을 사용하며 가상의 계정 ID를 포함합니다.

예제

- [예제 1: 사용자가 모든 RDS 리소스에서 Describe 작업을 수행할 수 있도록 허용 \(p. 347\)](#)
- [예제 2: 사용자가 지정된 DB 파라미터 및 보안 그룹을 사용하는 DB 인스턴스를 생성할 수 있도록 허용 \(p. 347\)](#)
- [예제 3: 사용자의 DB 인스턴스 삭제 방지 \(p. 347\)](#)

예제 1: 사용자가 모든 RDS 리소스에서 Describe 작업을 수행할 수 있도록 허용

다음 권한 정책은 사용자에게 `Describe`로 시작하는 모든 작업을 실행할 수 있는 권한을 부여합니다. 이러한 작업은 DB 인스턴스와 같은 RDS 리소스에 대한 정보를 보여 줍니다. `Resource` 요소에 와일드카드 문자 (*)가 있으면 계정이 소유한 모든 Amazon RDS 리소스에 대해 작업이 허용됩니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowRDSDescribe",  
            "Effect": "Allow",  
            "Action": "rds:Describe*",  
            "Resource": "*"  
        }  
    ]  
}
```

예제 2: 사용자가 지정된 DB 파라미터 및 보안 그룹을 사용하는 DB 인스턴스를 생성할 수 있도록 허용

다음 권한 정책은 사용자가 `mysql-production` DB 파라미터 그룹 및 `db-production` DB 보안 그룹을 사용해야 하는 DB 인스턴스만 생성할 수 있도록 허용하는 권한을 부여합니다.

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

예제 3: 사용자의 DB 인스턴스 삭제 방지

다음 권한 정책은 사용자의 특정 DB 인스턴스 삭제를 방지하는 권한을 부여합니다. 예를 들어, 관리자가 아닌 모든 사용자에 대해 프로덕션 인스턴스를 삭제할 수 있는 권한을 거부해야 할 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "DenyDelete1",  
            "Effect": "Deny",  
            "Action": "rds>DeleteDBInstance",  
            "Resource": "arn:aws:rds:us-west-2:123456789012:db:my-mysql-instance"  
        }  
    ]  
}
```

Amazon RDS API 권한: 작업, 리소스 및 조건 참조

IAM 자격 증명에 연결할 수 있는 액세스 제어 (p. 341) 및 쓰기 권한 정책(자격 증명 기반 정책)을 설정할 때 다음 표를 참조로 사용할 수 있습니다.

표를 보면 Amazon RDS API 작업이 각각 나열되어 있습니다. 실행 권한을 부여할 수 있는 대상 작업과 권한을 부여할 수 있는 대상 AWS 리소스, 그리고 세부적인 액세스 제어를 위해 추가할 수 있는 조건 키가 포함되어 있습니다. 정책의 Action 필드에서 작업을 지정하고, Resource 필드에서 리소스 값을 지정하고, Condition 필드에서 조건을 지정합니다. 조건에 대한 자세한 내용은 [IAM 정책 조건을 사용하여 세부적인 액세스 제어 구현 \(p. 363\)](#) 단원을 참조하십시오.

Amazon RDS 정책에서 AWS 차원 조건 키를 사용하여 조건을 표시할 수 있습니다. AWS 차원 키의 전체 목록은 IAM 사용 설명서의 [사용할 수 있는 키](#) 단원을 참조하십시오.

IAM 정책은 IAM 정책 시뮬레이터로 테스트할 수 있습니다. 이 시뮬레이터는 Amazon RDS 작업을 포함하여 각 AWS 작업에 필요한 리소스 및 파라미터 목록을 자동으로 제공합니다. 또한 IAM 정책은 사용자가 지정하는 작업마다 필요한 권한을 결정합니다. IAM 정책 시뮬레이터에 대한 자세한 내용은 IAM 사용 설명서에서 [IAM 정책 시뮬레이터로 IAM 정책 테스트하기](#) 단원을 참조하십시오.

Note

작업을 지정하려면 rds: 접두사 다음에 API 작업 이름을 사용합니다(예: rds:CreateDBInstance).

다음 표에는 RDS API 작업을 비롯한 관련 작업과 리소스, 그리고 조건 키가 나와 있습니다.

주제

- [리소스 수준 권한을 지원하는 Amazon RDS 작업 \(p. 348\)](#)
- [리소스 수준 권한을 지원하지 않는 Amazon RDS 작업 \(p. 363\)](#)

리소스 수준 권한을 지원하는 Amazon RDS 작업

리소스 수준 권한은 사용자가 작업을 수행할 수 있는 리소스를 지정할 수 있는 기능입니다. Amazon RDS는 리소스 수준 권한을 부분적으로 지원합니다. 즉, 필요 조건을 지정하거나 사용 가능한 특정 리소스를 지정하여 사용자가 특정 Amazon RDS 작업을 사용할 수 있는지 여부를 제어할 수 있습니다. 예를 들어 특정 DB 인스턴스로 제한된 변경 권한을 사용자에게 부여할 수 있습니다.

다음 표에는 RDS API 작업을 비롯한 관련 작업과 리소스, 그리고 조건 키가 나와 있습니다.

RDS API 연산 및 작업	리소스	조건 키
AddRoleToDBCluster	DB 클러스터	rds:cluster-tag
rds:AddRoleToDBCluster	arn:aws:rds:region:account-id:cluster:db-cluster-name	
	IAM 역할 arn:aws:iam::account-id:role/ <i>role-name</i>	—
AddSourceIdentifierToSubscription	아이피티 구독	rds:es-tag
rds:AddSourceIdentifier	arn:aws:rds:region:account-id:es:subscription-name	
AddTagsToResource	DB 인스턴스	rds:db-tag

RDS API 연산 및 작업	리소스	조건 키
rds:AddTagsToResource	arn:aws:rds: region:account-id:db:db-instance-name	rds:req-tag
	DB 클러스터 arn:aws:rds: region:account-id:cluster:db-cluster-name	rds:cluster-tag rds:req-tag
	DB 옵션 그룹 arn:aws:rds: region:account-id:og:option-group-name	rds:og-tag rds:req-tag
	DB 파라미터 그룹 arn:aws:rds: region:account-id:pg:parameter-group-name	rds:pg-tag rds:req-tag
	DB 클러스터 파라미터 그룹 arn:aws:rds: region:account-id:cluster-pg:cluster-parameter-group-name	rds:cluster-pg-tag rds:req-tag
	DB 보안 그룹 arn:aws:rds: region:account-id:secgrp:security-group-name	rds:secgrp-tag rds:req-tag
	DB 서브넷 그룹 arn:aws:rds: region:account-id:subgrp:subnet-group-name	rds:subgrp-tag rds:req-tag
	DB 스냅샷 arn:aws:rds: region:account-id:snapshot:snapshot-name	rds:snapshot-tag rds:req-tag
	DB 클러스터 스냅샷 arn:aws:rds: region:account-id:cluster-snapshot:cluster-snapshot-name	rds:cluster-snapshot-tag rds:req-tag
	이벤트 구독 arn:aws:rds: region:account-id:es:subscription-name	rds:es-tag rds:req-tag
	예약 DB 인스턴스 arn:aws:rds: region:account-id:ri:reserved-db-instance-name	rds:ri-tag rds:req-tag
ApplyPendingMaintenanceAction	DB 인스턴스 arn:aws:rds: region:account-id:db:db-instance-name	rds:db-tag

RDS API 연산 및 작업	리소스	조건 키
AuthorizeDBSecurityGroup	DB 보안 그룹 rds:AuthorizeDBSecurityGroup arn:aws:rds:region:account-id:secgrp:security-group-name	rds:secgrp-tag
BacktrackDBCluster	DB 클러스터 rds:BacktrackDBCluster arn:aws:rds:region:account-id:cluster:db-cluster-name	rds:cluster-tag
CopyDBClusterSnapshot	DB 클러스터 스냅샷 rds:CopyDBClusterSnapshot arn:aws:rds:region:account-id:cluster-snapshot:cluster-snapshot-name	rds:cluster-snapshot-tag
CopyDBParameterGroup	DB 파라미터 그룹 rds:CopyDBParameterGroup arn:aws:rds:region:account-id:pg:parameter-group-name	rds:pg-tag
CopyDBSnapshot	DB 스냅샷 rds:CopyDBSnapshot arn:aws:rds:region:account-id:snapshot:snapshot-name	rds:snapshot-tag
CopyOptionGroup	DB 옵션 그룹 rds:CopyOptionGroup arn:aws:rds:region:account-id:og:option-group-name	rds:og-tag
CreateDBCluster	DB 클러스터 rds:CreateDBCluster arn:aws:rds:region:account-id:cluster:db-cluster-name	rds:DatabaseEngine rds:DatabaseName rds:req-tag
	DB 옵션 그룹 arn:aws:rds:region:account-id:og:option-group-name	rds:og-tag
	DB 클러스터 파라미터 그룹 arn:aws:rds:region:account-id:cluster-pg:cluster-parameter-group-name	rds:cluster-pg-tag
	DB 서브넷 그룹 arn:aws:rds:region:account-id:subgrp:subnet-group-name	rds:subgrp-tag
CreateDBClusterEndpoint	DB 클러스터 rds:CreateDBClusterEndpoint arn:aws:rds:region:account-id:cluster:db-cluster-name	rds:cluster-tag

RDS API 연산 및 작업	리소스	조건 키
	DB 클러스터 엔드포인트 arn:aws:rds: region:account-id:cluster-endpoint:db-cluster-endpoint-identifier	rds:endpointType
CreateDBClusterParameter	DB 클러스터 파라미터 그룹 rds:CreateDBClusterParameter parameters:group:region:account-id:cluster-pg:cluster-parameter-group-name	rds:req-tag
CreateDBClusterSnapshot	DB 클러스터 rds:CreateDBClusterSnapshot snapshots:rds:region:account-id:cluster:db-cluster-name	rds:cluster-tag
	DB 클러스터 스냅샷 arn:aws:rds: region:account-id:cluster-snapshot:cluster-snapshot-name	rds:req-tag
CreateDBInstance	DB 인스턴스 rds:CreateDBInstance arn:aws:rds: region:account-id:db:db-instance-name	rds:DatabaseClass rds:DatabaseEngine rds:DatabaseName rds:MultiAz rds:Piops rds:StorageEncrypted rds:StorageSize rds:Vpc rds:req-tag
	DB 옵션 그룹 arn:aws:rds: region:account-id:og:option-group-name	rds:og-tag
	DB 파라미터 그룹 arn:aws:rds: region:account-id:pg:parameter-group-name	rds:pg-tag
	DB 보안 그룹 arn:aws:rds: region:account-id:secgrp:security-group-name	rds:secgrp-tag

RDS API 연산 및 작업	리소스	조건 키
	DB 서브넷 그룹 arn:aws:rds: region:account-id:subgrp:subnet-group-name	rds:subgrp-tag
	DB 클러스터 arn:aws:rds: region:account-id:cluster:db-cluster-name	rds:cluster-tag
CreateDBInstanceReadReplica	DB 인스턴스 rds:CreateDBInstanceReadReplica: region:account-id:db:db-instance-name	rds:DatabaseClass rds:Piops rds:req-tag
	DB 옵션 그룹 arn:aws:rds: region:account-id:og:option-group-name	rds:og-tag
	DB 서브넷 그룹 arn:aws:rds: region:account-id:subgrp:subnet-group-name	rds:subgrp-tag
CreateDBParameterGroup	DB 파라미터 그룹 rds:CreateDBParameterGroup: region:account-id:pg:parameter-group-name	rds:req-tag
CreateDBSecurityGroup	DB 보안 그룹 rds:CreateDBSecurityGroup: region:account-id:secgrp:security-group-name	rds:req-tag
CreateDBSnapshot	DB 인스턴스 rds:CreateDBSnapshot: arn:aws:rds:region:account-id:db:db-instance-name	rds:db-tag
	DB 스냅샷 arn:aws:rds: region:account-id:snapshot:snapshot-name	rds:req-tag
CreateDBSubnetGroup	DB 서브넷 그룹 rds:CreateDBSubnetGroup: region:account-id:subgrp:subnet-group-name	rds:req-tag
CreateEventSubscription	이벤트 구독 rds:CreateEventSubscription: region:account-id:es:subscription-name	rds:req-tag

RDS API 연산 및 작업	리소스	조건 키
CreateOptionGroup	DB 옵션 그룹	rds:req-tag
rds:CreateOptionGroup	arn:aws:rds:region:account-id:og:option-group-name	
DeleteDBCluster	DB 클러스터	rds:cluster-tag
	arn:aws:rds:region:account-id:cluster:db-cluster-name	
	DB 클러스터 스냅샷 arn:aws:rds:region:account-id:cluster-snapshot:cluster-snapshot-name	rds:cluster-snapshot-tag
DeleteDBClusterEndpoint	DB 클러스터 엔드포인트	
rds:DeleteDBClusterEndpoint	Endpoint:arn:aws:rds:region:account-id:cluster-endpoint:db-cluster-endpoint-identifier	
DeleteDBClusterParameter	DB 클러스터 파라미터 그룹	rds:cluster-pg-tag
rds:DeleteDBClusterParameter	ParameterGroup:arn:aws:rds:region:account-id:cluster-pg:cluster-parameter-group-name	
DeleteDBClusterSnapshot	DB 클러스터 스냅샷	rds:cluster-snapshot-tag
rds:DeleteDBClusterSnapshot	Snapshot:arn:aws:rds:region:account-id:cluster-snapshot:cluster-snapshot-name	
DeleteDBInstance	DB 인스턴스	rds:db-tag
rds:DeleteDBInstance	arn:aws:rds:region:account-id:db:db-instance-name	
DeleteDBParameterGroup	DB 파라미터 그룹	rds:pg-tag
rds:DeleteDBParameterGroup	Group:arn:aws:rds:region:account-id:pg:parameter-group-name	
DeleteDBSecurityGroup	DB 보안 그룹	rds:secgrp-tag
rds:DeleteDBSecurityGroup	Group:arn:aws:rds:region:account-id:secgrp:security-group-name	
DeleteDBSnapshot	DB 스냅샷	rds:snapshot-tag
rds:DeleteDBSnapshot	arn:aws:rds:region:account-id:snapshot:snapshot-name	
DeleteDBSubnetGroup	DB 서브넷 그룹	rds:subgrp-tag
rds:DeleteDBSubnetGroup	Group:arn:aws:rds:region:account-id:subgrp:subnet-group-name	

RDS API 연산 및 작업	리소스	조건 키
DeleteEventSubscription	이벤트 구독 <code>rds:DeleteEventSubscription</code> <code>arn:aws:rds:region:account-id:es:subscription-name</code>	<code>rds:es-tag</code>
DeleteOptionGroup	DB 옵션 그룹 <code>rds:DeleteOptionGroup</code> <code>arn:aws:rds:region:account-id:og:option-group-name</code>	<code>rds:og-tag</code>
DescribeDBClusterEndpoints	DB 클러스터 <code>rds:DescribeDBClusterEndpoints</code> <code>arn:aws:rds:region:account-id:cluster:db-cluster-name</code>	<code>rds:cluster-tag</code>
	DB 클러스터 엔드포인트 <code>arn:aws:rds:region:account-id:cluster-endpoint:db-cluster-endpoint-identifier</code>	
DescribeDBClusterParameterGroups	DB 클러스터 파라미터 그룹 <code>rds:DescribeDBClusterParameterGroups</code> <code>arn:aws:rds:region:account-id:cluster-pg:cluster-parameter-group-name</code>	<code>rds:cluster-pg-tag</code>
DescribeDBClusterParameters	DB 클러스터 파라미터 그룹 <code>rds:DescribeDBClusterParameters</code> <code>arn:aws:rds:region:account-id:cluster-pg:cluster-parameter-group-name</code>	<code>rds:cluster-pg-tag</code>
DescribeDBClusters	DB 클러스터 <code>rds:DescribeDBClusters</code> <code>arn:aws:rds:region:account-id:cluster:db-cluster-instance-name</code>	<code>rds:cluster-tag</code>
DescribeDBClusterSnapshots	DB 클러스터 스냅샷 <code>rds:DescribeDBClusterSnapshots</code> <code>arn:aws:rds:region:account-id:cluster-snapshot:cluster-snapshot-name</code>	<code>rds:cluster-snapshot-tag</code>
DescribeDBEngineVersions	DB 파라미터 그룹 <code>rds:DescribeDBEngineVersions</code> <code>arn:aws:rds:region:account-id:pg:parameter-group-name</code>	<code>rds:pg-tag</code>
DescribeDBLogFile	DB 인스턴스 <code>rds:DescribeDBLogFile</code> <code>arn:aws:rds:region:account-id:db:db-instance-name</code>	<code>rds:db-tag</code>

RDS API 연산 및 작업	리소스	조건 키
DescribeDBParameterGroup	DB 파라미터 그룹 rds:DescribeDBParameterGroups aws:rds:region:account-id:pg:parameter-group-name	rds:pg-tag
DescribeDBParameters	DB 파라미터 그룹 rds:DescribeDBParameters aws:rds:region:account-id:pg:parameter-group-name	rds:pg-tag
DescribeDBSecurityGroups	DB 보안 그룹 rds:DescribeDBSecurityGroups aws:rds:region:account-id:secgrp:security-group-name	rds:secgrp-tag
DescribeDBSnapshotAttributes	DB 스냅샷 rds:DescribeDBSnapshotAttributes aws:rds:region:account-id:snapshot:snapshot-name	rds:snapshot-tag
DescribeDBSnapshots	DB 인스턴스 rds:DescribeDBSnapshots aws:rds:region:account-id:db:db-instance-name	rds:db-tag
	DB 스냅샷 arn:aws:rds:region:account-id:snapshot:snapshot-name	rds:snapshot-tag
DescribeDBSubnetGroups	DB 서브넷 그룹 rds:DescribeDBSubnetGroups aws:rds:region:account-id:subgrp:subnet-group-name	rds:subgrp-tag
DescribeEventSubscriptions	이벤트 구독 rds:DescribeEventSubscriptions aws:rds:region:account-id:es:subscription-name	rds:es-tag
DescribeOptionGroups	DB 옵션 그룹 rds:DescribeOptionGroups aws:rds:region:account-id:og:option-group-name	rds:og-tag

RDS API 연산 및 작업	리소스	조건 키
DescribePendingMaintenanceActions	DB 인스턴스 rds:DescribePendingMaintenanceActions region:account-id:db:db-instance-name	rds:DatabaseClass rds:DatabaseEngine rds:DatabaseName rds:MultiAz rds:Piops rds:StorageSize rds:Vpc rds:db-tag
DescribeReservedDBInstancesOfferings	예약 DB 인스턴스 rds:DescribeReservedDBInstancesOfferings region:account-id:ri:reserved-db-instance-name	rds:DatabaseClass rds:MultiAz rds:ri-tag
DescribeReservedDBInstances	DB 인스턴스 rds:DescribeReservedDBInstances region:account-id:db:db-instance-name	rds:DatabaseClass rds:MultiAz
DownloadDBLogFilePortion	DB 인스턴스 rds:DownloadDBLogFilePortion region:account-id:db:db-instance-name	rds:db-tag
FailoverDBCluster	DB 클러스터 rds:FailoverDBCluster arn:aws:rds:region:account-id:cluster:db-cluster-instance-name	rds:cluster-tag
ListTagsForResource	DB 인스턴스 rds>ListTagsForResource arn:aws:rds:region:account-id:db:db-instance-name	rds:db-tag
	DB 클러스터 arn:aws:rds:region:account-id:cluster:db-cluster-name	rds:cluster-tag
	DB 옵션 그룹 arn:aws:rds:region:account-id:og:option-group-name	rds:og-tag
	DB 파라미터 그룹 arn:aws:rds:region:account-id:pg:parameter-group-name	rds:pg-tag

RDS API 연산 및 작업	리소스	조건 키
	DB 보안 그룹 <code>arn:aws:rds:region:account-id:secgrp:security-group-name</code>	<code>rds:secgrp-tag</code>
	DB 클러스터 파라미터 그룹 <code>arn:aws:rds:region:account-id:cluster-pg:cluster-parameter-group-name</code>	<code>rds:cluster-pg-tag</code>
	DB 서브넷 그룹 <code>arn:aws:rds:region:account-id:subgrp:subnet-group-name</code>	<code>rds:subgrp-tag</code>
	DB 스냅샷 <code>arn:aws:rds:region:account-id:snapshot:snapshot-name</code>	<code>rds:snapshot-tag</code>
	DB 클러스터 스냅샷 <code>arn:aws:rds:region:account-id:cluster-snapshot:cluster-snapshot-name</code>	<code>rds:cluster-snapshot-tag</code>
	이벤트 구독 <code>arn:aws:rds:region:account-id:es:subscription-name</code>	<code>rds:es-tag</code>
ModifyDBCluster <code>rds:ModifyDBCluster</code>	예약 DB 인스턴스 <code>arn:aws:rds:region:account-id:ri:reserved-db-instance-name</code>	<code>rds:ri-tag</code>
	DB 클러스터 <code>arn:aws:rds:region:account-id:cluster:db-cluster-name</code>	<code>rds:cluster-tag</code>
	DB 옵션 그룹 <code>arn:aws:rds:region:account-id:og:option-group-name</code>	<code>rds:og-tag</code>
	DB 클러스터 파라미터 그룹 <code>arn:aws:rds:region:account-id:cluster-pg:cluster-parameter-group-name</code>	<code>rds:cluster-pg-tag</code>
ModifyDBClusterEndpoint <code>rds:ModifyDBClusterEndpoint</code>	DB 클러스터 엔드포인트 <code>arn:aws:rds:region:account-id:cluster-endpoint:db-cluster-endpoint-identifier</code>	<code>rds:endpointType</code>

RDS API 연산 및 작업	리소스	조건 키
ModifyDBClusterParameter	DB 클러스터 파라미터 그룹 <code>rds:ModifyDBClusterParameterGroup:region:account-id:cluster-pg:cluster-parameter-group-name</code>	<code>rds:cluster-pg-tag</code>
ModifyDBClusterSnapshot	DB 클러스터 스냅샷 <code>rds:ModifyDBClusterSnapshotAttribute:region:account-id:cluster-snapshot:cluster-snapshot-name</code>	<code>rds:cluster-snapshot-tag</code>
ModifyDBInstance	DB 인스턴스 <code>rds:ModifyDBInstance:arn:aws:rds:region:account-id:db:db-instance-name</code>	<code>rds:DatabaseClass</code> <code>rds:MultiAz</code> <code>rds:Piops</code> <code>rds:StorageSize</code> <code>rds:Vpc</code> <code>rds:db-tag</code>
	DB 옵션 그룹 <code>arn:aws:rds:region:account-id:og:option-group-name</code>	<code>rds:og-tag</code>
	DB 파라미터 그룹 <code>arn:aws:rds:region:account-id:pg:parameter-group-name</code>	<code>rds:pg-tag</code>
	DB 보안 그룹 <code>arn:aws:rds:region:account-id:secgrp:security-group-name</code>	<code>rds:secgrp-tag</code>
ModifyDBParameterGroup	DB 파라미터 그룹 <code>rds:ModifyDBParameterGroup:groups:rds:region:account-id:pg:parameter-group-name</code>	<code>rds:pg-tag</code>
ModifyDBSnapshotAttribute	DB 스냅샷 <code>rds:ModifyDBSnapshotAttribute:attribute:ards:region:account-id:snapshot:snapshot-name</code>	<code>rds:snapshot-tag</code>
ModifyDBSubnetGroup	DB 서브넷 그룹 <code>rds:ModifyDBSubnetGroup:group:aws:rds:region:account-id:subgrp:subnet-group-name</code>	<code>rds:subgrp-tag</code>
ModifyEventSubscription	이벤트 구독 <code>rds:ModifyEventSubscription:subscriptions:rds:region:account-id:es:subscription-name</code>	<code>rds:es-tag</code>

RDS API 연산 및 작업	리소스	조건 키
ModifyOptionGroup	DB 옵션 그룹 <code>rds:ModifyOptionGroup</code> <code>arn:aws:rds:region:account-id:og:option-group-name</code>	<code>rds:og-tag</code>
PromoteReadReplica	DB 인스턴스 <code>rds:PromoteReadReplica</code> <code>arn:aws:rds:region:account-id:db:db-instance-name</code>	<code>rds:db-tag</code>
PromoteReadReplicaDBCluster	DB 클러스터 <code>rds:PromoteReadReplicaDBCluster</code> <code>arn:aws:rds:region:account-id:cluster:db-cluster-name</code>	
RebootDBInstance	DB 인스턴스 <code>rds:RebootDBInstance</code> <code>arn:aws:rds:region:account-id:db:db-instance-name</code>	<code>rds:db-tag</code>
RemoveSourceIdentifierFromSubscription	이벤트 구독 <code>rds:RemoveSourceIdentifierFromSubscription</code> <code>arn:aws:rds:region:account-id:es:subscription-name</code>	<code>rds:es-tag</code>
RemoveTagsFromResource	DB 인스턴스 <code>rds:RemoveTagsFromResource</code> <code>arn:aws:rds:region:account-id:db:db-instance-name</code>	<code>rds:db-tag</code>
	DB 클러스터 <code>arn:aws:rds:region:account-id:cluster:db-cluster-name</code>	<code>rds:cluster-tag</code>
	DB 옵션 그룹 <code>arn:aws:rds:region:account-id:og:option-group-name</code>	<code>rds:og-tag</code>
DB Parameter Group	DB 파라미터 그룹 <code>arn:aws:rds:region:account-id:pg:parameter-group-name</code>	<code>rds:pg-tag</code>
	DB 클러스터 파라미터 그룹 <code>arn:aws:rds:region:account-id:cluster-pg:cluster-parameter-group-name</code>	<code>rds:cluster-pg-tag</code>
DB Security Group	DB 보안 그룹 <code>arn:aws:rds:region:account-id:secgrp:security-group-name</code>	<code>rds:secgrp-tag</code>

RDS API 연산 및 작업	리소스	조건 키
	DB 서브넷 그룹 arn:aws:rds: region:account-id:subgrp:subnet-group-name	rds:subgrp-tag rds:req-tag
	DB 스냅샷 arn:aws:rds: region:account-id:snapshot:snapshot-name	rds:snapshot-tag rds:req-tag
	DB 클러스터 스냅샷 arn:aws:rds: region:account-id:cluster-snapshot:cluster-snapshot-name	rds:cluster-snapshot-tag rds:req-tag
	이벤트 구독 arn:aws:rds: region:account-id:es:subscription-name	rds:es-tag rds:req-tag
	예약 DB 인스턴스 arn:aws:rds: region:account-id:ri:reserved-db-instance-name	rds:ri-tag rds:req-tag
ResetDBClusterParameter	DB 클러스터 파라미터 그룹 rds:ResetDBClusterParameter ParameterGroup:region:account-id:cluster-pg:cluster-parameter-group-name	rds:cluster-pg-tag
ResetDBParameterGroup	DB 파라미터 그룹 rds:ResetDBParameterGroup aws:rds:region:account-id:pg:parameter-group-name	rds:pg-tag
RestoreDBClusterFromS3	DB 클러스터 rds:RestoreDBClusterFromS3 aws:rds:region:account-id:cluster:db-cluster-instance-name	rds:DatabaseEngine rds:DatabaseName rds:req-tag
	DB 클러스터 파라미터 그룹 arn:aws:rds: region:account-id:cluster-pg:cluster-parameter-group-name	rds:cluster-pg-tag
	DB 옵션 그룹 arn:aws:rds: region:account-id:og:option-group-name	rds:og-tag
	DB 서브넷 그룹 arn:aws:rds: region:account-id:subgrp:subnet-group-name	rds:subgrp-tag

RDS API 연산 및 작업	리소스	조건 키
RestoreDBClusterFromSnapshot	DB 클러스터	rds:DatabaseEngine
	rds:RestoreDBClusterFromSnapshot arn:rds:region:account-id:cluster:db-cluster-instance-name	rds:DatabaseName
	DB 옵션 그룹 arn:aws:rds: region:account-id:og:option-group-name	rds:req-tag
RestoreDBClusterToPointInTime	DB 클러스터 스냅샷 arn:aws:rds: region:account-id:cluster-snapshot:cluster-snapshot-name	rds:cluster-snapshot-tag
	DB 클러스터	rds:req-tag
	rds:RestoreDBClusterToPointInTime arn:rds:region:account-id:cluster:db-cluster-instance-name	
RestoreDBInstanceFromDBSnapshot	DB 옵션 그룹 arn:aws:rds: region:account-id:og:option-group-name	rds:og-tag
	DB 서브넷 그룹 arn:aws:rds: region:account-id:subgrp:subnet-group-name	rds:subgrp-tag
	DB 인스턴스	rds:DatabaseClass
RestoreDBInstanceFromDBSnapshot	rds:RestoreDBInstanceFromDBSnapshot arn:rds:region:account-id:db:db-instance-name	rds:DatabaseEngine
	rds:DatabaseName	rds:DatabaseName
	rds:MultiAZ	rds:MultiAZ
RestoreDBInstanceFromDBSnapshot	rds:Piops	rds:Piops
	rds:Vpc	rds:Vpc
	rds:req-tag	rds:req-tag
RestoreDBInstanceFromDBSnapshot	DB 옵션 그룹 arn:aws:rds: region:account-id:og:option-group-name	rds:og-tag
	DB 스냅샷 arn:aws:rds: region:account-id:snapshot:snapshot-name	rds:snapshot-tag

RDS API 연산 및 작업	리소스	조건 키
	DB 서브넷 그룹 arn:aws:rds: <i>region:account-id:subgrp:subnet-group-name</i>	rds:subgrp-tag
RestoreDBInstanceToPointInTime	DB 인스턴스 rds:RestoreDBInstanceToPointInTime <i>region:account-id:db:db-instance-name</i>	rds:DatabaseClass rds:DatabaseEngine rds:DatabaseName rds:MultiAz rds:Piops rds:Vpc rds:req-tag
	DB 옵션 그룹 arn:aws:rds: <i>region:account-id:og:option-group-name</i>	rds:og-tag
	DB 스냅샷 arn:aws:rds: <i>region:account-id:snapshot:snapshot-name</i>	rds:snapshot-tag
	DB 서브넷 그룹 arn:aws:rds: <i>region:account-id:subgrp:subnet-group-name</i>	rds:subgrp-tag
RevokeDBSecurityGroupIngress	DB 보안 그룹 rds:RevokeDBSecurityGroupIngress <i>region:account-id:secgrp:security-group-name</i>	rds:secgrp-tag
StartDBInstance	DB 인스턴스 rds:StartDBInstance <i>region:account-id:db:db-instance-name</i>	rds:DatabaseClass rds:DatabaseEngine rds:DatabaseName rds:MultiAz rds:Piops rds:Vpc rds:db-tag

RDS API 연산 및 작업	리소스	조건 키
StopDBInstance <code>rds:StopDBInstance</code>	DB 인스턴스 <code>arn:aws:rds:<i>region:account-id:db:db-instance-name</i></code>	<code>rds:DatabaseClass</code> <code>rds:DatabaseEngine</code> <code>rds:DatabaseName</code> <code>rds:MultiAZ</code> <code>rds:Piops</code> <code>rds:Vpc</code> <code>rds:db-tag</code>

리소스 수준 권한을 지원하지 않는 Amazon RDS 작업

작업 사용 권한의 부여 여부를 결정하는 IAM 정책에서는 모든 Amazon RDS 작업을 사용할 수 있습니다. 하지만 모든 작업이 리소스 수준 권한을 지원하는 것은 아닙니다. 여기에서 리소스 수준 권한이란 작업이 가능한 리소스를 지정할 수 있는 권한을 말합니다. 현재 다음 API 작업은 리소스 수준 권한을 지원하지 않습니다. 따라서 IAM 정책에서 이러한 작업을 사용하려면 Resource 요소에 * 와일드카드를 사용하여 해당 작업 리소스를 모두 사용할 수 있는 권한을 사용자에게 부여해야 합니다.

- `rds:DescribeAccountAttributes`
- `rds:DescribeCertificates`
- `rds:DescribeDBClusterSnapshots`
- `rds:DescribeDBInstances`
- `rds:DescribeEngineDefaultClusterParameters`
- `rds:DescribeEngineDefaultParameters`
- `rds:DescribeEventCategories`
- `rds:DescribeEvents`
- `rds:DescribeOptionGroupOptions`
- `rds:DescribeOrderableDBInstanceOptions`
- `rds:DownloadCompleteDBLogFile`
- `rds:PurchaseReservedDBInstancesOffering`

IAM 정책 조건을 사용하여 세부적인 액세스 제어 구현

Amazon RDS에서 권한을 부여할 때 권한 정책이 적용되는 방식을 결정하는 조건을 지정할 수 있습니다.

개요

Amazon RDS에서는 IAM 정책을 사용하여 권한을 부여할 때 조건을 지정하는 옵션이 있습니다(액세스 제어 (p. 341) 참조). 예를 들면,

- 사용자가 특정 데이터베이스 엔진을 지정하는 경우에만 DB 인스턴스를 만들 수 있도록 허용합니다.
- 사용자가 특정 태그 이름 및 태그 값으로 태그 지정된 RDS 리소스를 수정할 수 있도록 허용합니다.

Amazon RDS에 대한 IAM 정책에서는 다음 두 가지 방법으로 조건을 지정할 수 있습니다.

- 조건 키 사용 (p. 364)
- 사용자 지정 태그 사용 (p. 366)

조건 지정: 조건 키 사용

AWS는 액세스 제어를 위해 IAM을 지원하는 모든 AWS 서비스에 대해 사전 정의된 조건 키(AWS 차원의 조건 키) 집합을 제공합니다. 예를 들어, 작업을 요청할 때 특정 AWS ID를 요구하려면 `aws:userid` 조건 키를 사용할 수 있습니다. 자세한 내용과 AWS 차원의 조건 키 목록은 IAM 사용 설명서의 [사용 가능한 조건 키를 참조하십시오.](#)

Note

조건 키 이름은 대소문자를 구분하지 않습니다. 조건 키 값의 대소문자 구분은 사용하는 조건 연산자에 따라 다릅니다. 자세한 내용은 IAM 사용 설명서에서 [IAM JSON 정책 요소: 조건 연산자](#)를 참조하십시오.

또한 Amazon RDS는 IAM 권한 정책의 `Condition` 요소에 포함시킬 수 있는 고유의 조건 키도 제공합니다. 다음 표는 RDS 리소스에 적용되는 RDS 조건 키를 보여 줍니다.

RDS 조건 키	설명	값 유형
<code>rds:DatabaseClass</code>	DB 인스턴스 클래스의 유형입니다.	문자열
<code>rds:DatabaseEngine</code>	MySQL과 같은 데이터베이스 엔진입니다.	문자열
<code>rds:DatabaseName</code>	DB 인스턴스에 있는 데이터베이스의 사용자 정의 이름입니다.	문자열
<code>rds:MultiAz</code>	DB 인스턴스를 여러 가용 영역에서 실행할지 여부를 지정하는 값입니다. DB 인스턴스가 다중 AZ를 사용하고 있음을 나타내려면 <code>true</code> 를 지정합니다.	부울
<code>rds:Piops</code>	인스턴스가 지원하는 프로비저닝된 IOPS(PIOPS)의 개수가 포함된 값입니다. PIOPS가 활성화되어 있지 않은 DB 인스턴스를 나타내려면 0을 지정합니다.	정수
<code>rds:StorageSize</code>	스토리지 볼륨 크기(GiB)입니다.	정수
<code>rds:Vpc</code>	DB 인스턴스를 Amazon Virtual Private Cloud(Amazon VPC)에서 실행할지 여부를 지정하는 값입니다. DB 인스턴스가 Amazon VPC에서 실행됨을 나타내려면 <code>true</code> 를 지정합니다.	부울
<code>rds:req-tag</code>	리소스에 태그 지정하는 데 사용할 수 있는 태그 키와 값 집합을 제한하는 값입니다.	문자열

예를 들어, 다음 `Condition` 요소는 조건 키를 사용하여 MySQL 데이터베이스 엔진을 지정합니다. `rds:CreateDBInstance` 작업에 대한 권한을 허용하는 IAM 정책에 이를 적용하여 사용자가 MySQL 데이터베이스 엔진에서 DB 인스턴스만 생성하도록 설정할 수 있습니다. 이 조건을 사용하는 IAM 정책의 예는 [정책 예: 조건 키 사용 \(p. 364\)](#) 단원을 참조하십시오.

```
"Condition": { "StringEquals": { "rds:DatabaseEngine": "mysql" } }
```

모든 RDS 조건 키 식별자와 이 식별자가 적용되는 RDS 작업 및 리소스의 목록은 [Amazon RDS API 권한: 작업, 리소스 및 조건 참조 \(p. 348\)](#) 단원을 참조하십시오.

정책 예: 조건 키 사용

다음은 Amazon RDS IAM 권한 정책에서 조건 키를 사용할 수 있는 방법의 예입니다.

예제 1: 특정 DB 엔진을 사용하고 MultiAZ가 아닌 DB 인스턴스를 생성할 수 있는 권한 부여

다음 정책은 RDS 조건 키를 사용하며, 사용자가 MySQL 데이터베이스 엔진을 사용하는 DB 인스턴스만 생성할 수 있도록 허용하며, MultiAZ를 사용하지 않습니다. Condition 요소는 데이터베이스 엔진이 MySQL이라는 요구 사항을 나타냅니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowMySQLCreate",  
            "Effect": "Allow",  
            "Action": "rds:CreateDBInstance",  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "rds:DatabaseEngine": "mysql"  
                },  
                "Bool": {  
                    "rds:MultiAz": false  
                }  
            }  
        }  
    ]  
}
```

예제 2: 특정 DB 인스턴스 클래스에 대한 DB 인스턴스를 만들고 프로비저닝된 IOPS를 사용하는 DB 인스턴스를 만들 수 있는 권한을 명시적으로 거부

다음 정책은 가장 크고 가장 비싼 인스턴스인 DB 인스턴스 클래스 `r3.8xlarge` 및 `m4.10xlarge`를 사용하는 DB 인스턴스를 만들 수 있는 권한을 명시적으로 거부합니다. 또한 이 정책은 추가 비용이 발생하는 프로비저닝된 IOPS를 사용하는 DB 인스턴스를 사용자가 생성하지 못하도록 합니다.

명시적으로 거부하는 권한은 이미 부여된 다른 모든 권한에 우선합니다. 따라서 부여하지 않으려는 권한을 자격 증명이 우연히 획득하지 않도록 할 수 있습니다.

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

```
        }
    ]
}
```

예제 3: 리소스에 태그 지정하는 데 사용할 수 있는 태그 키와 값 집합 제한

다음 정책은 RDS 조건 키를 사용하고 키가 stage인 태그를 값이 test, qa, production인 리소스에 추가할 수 있도록 허용합니다.

```
{
  {
    "Version" : "2012-10-17",
    "Statement" : [
      {
        "Effect" : "Allow",
        "Action" : [ "rds:AddTagsToResource", "rds:RemoveTagsFromResource" ],
        "Resource" : "*",
        "Condition" : { "streq" : { "rds:req-tag/stage" : [ "test", "qa", "production" ] } }
      }
    ]
  }
}
```

조건 지정: 사용자 지정 태그 사용

RDS에서는 사용자 지정 태그를 사용하여 IAM 정책에서 조건을 지정할 수 있습니다.

예를 들어, beta, staging, production 등의 값과 함께 environment라는 태그를 DB 인스턴스에 추가하는 경우, environment 태그 값을 기반으로 특정 사용자를 DB 인스턴스로 제한하는 정책을 만들 수 있습니다.

Note

사용자 지정 태그 식별자는 대/소문자를 구분합니다.

다음 표에는 Condition 요소에서 사용할 수 있는 RDS 태그 식별자가 나와 있습니다.

RDS 태그 식별자	적용 대상
db-tag	읽기 전용 복제본을 포함하는 DB 인스턴스입니다
snapshot-tag	DB 스냅샷
ri-tag	예약 DB 인스턴스
secgrp-tag	DB 보안 그룹
og-tag	DB 옵션 그룹
pg-tag	DB 파라미터 그룹
subgrp-tag	DB 서브넷 그룹
es-tag	이벤트 구독
cluster-tag	DB 클러스터
cluster-pg-tag	DB 클러스터 파라미터 그룹

RDS 태그 식별자	적용 대상
cluster-snapshot-tag	DB 클러스터 스냅샷

사용자 지정 태그 조건의 구문은 다음과 같습니다.

```
"Condition": {"StringEquals": {"rds:rds-tag-identifier/tag-name": ["value"]}}
```

예를 들어, 다음 Condition 요소는 태그 이름이 environment이고 태그 값이 production인 DB 인스턴스에 적용됩니다.

```
"Condition": {"StringEquals": {"rds:db-tag/environment": ["production"]}}
```

태그 생성에 대한 자세한 내용은 [Amazon RDS 리소스에 태그 지정 \(p. 131\)](#) 단원을 참조하십시오.

Important

태깅을 사용하여 RDS 리소스에 대한 액세스를 관리하는 경우 RDS 리소스의 태그에 대한 액세스의 보안을 유지하는 것이 좋습니다. AddTagsToResource 및 RemoveTagsFromResource 작업에 대한 정책을 생성하여 태그에 대한 액세스를 관리할 수 있습니다. 예를 들어, 다음 정책은 모든 리소스에 대해 태그를 추가하거나 제거할 수 있는 사용자의 권한을 거부합니다. 그런 다음 특정 사용자가 태그를 추가하거나 제거할 수 있도록 허용하기 위한 정책을 생성할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyTagUpdates",
            "Effect": "Deny",
            "Action": [
                "rds:AddTagsToResource",
                "rds:RemoveTagsFromResource"
            ],
            "Resource": "*"
        }
    ]
}
```

모든 조건 키 값과 이 값이 적용되는 RDS 작업 및 리소스의 목록은 [Amazon RDS API 권한: 작업, 리소스 및 조건 참조 \(p. 348\)](#) 단원을 참조하십시오.

정책 예: 사용자 지정 태그 사용

다음은 Amazon RDS IAM 권한 정책에서 사용자 지정 태그를 사용할 수 있는 방법의 예입니다. Amazon RDS 리소스에 태그를 추가하는 방법에 대한 자세한 내용은 [Amazon RDS의 Amazon 리소스 이름\(ARN\)을 사용한 작업 \(p. 172\)](#)을 참조하십시오.

Note

모든 예는 us-west-2 리전을 사용하며 가상의 계정 ID를 포함합니다.

예제 1: 두 개의 값을 갖는 특정 태그를 사용하는 리소스 작업에 대한 권한 부여

다음 정책은 development 또는 test로 설정된 stage 태그가 있는 인스턴스에서 ModifyDBInstance 및 CreateDBSnapshot API를 수행할 수 있는 권한을 허용합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDevTestCreate",
            "Effect": "Allow",
            "Action": [
                "rds:ModifyDBInstance",
                "rds>CreateDBSnapshot"
            ],
            "Resource": [
                "arn:aws:rds:::db:/*",
                "arn:aws:rds:::db::snapshot:/*"
            ],
            "Condition": {
                "StringEquals": {"rds:stage": "development", "rds:stage": "test"}
            }
        }
    ]
}
```

```
"Effect": "Allow",
"Action": [
    "rds:ModifyDBInstance",
    "rds>CreateDBSnapshot"
],
"Resource": "*",
"Condition": {
    "StringEquals": {
        "rds:db-tag/stage": [
            "development",
            "test"
        ]
    }
}
}
```

예제 2: 지정된 DB 파라미터 그룹을 사용하는 DB 인스턴스를 만들 수 있는 권한을 명시적으로 거부

다음 정책은 특정 태그 값이 있는 DB 파라미터 그룹을 사용하는 DB 인스턴스를 만들 수 있는 권한을 명시적으로 거부합니다. DB 인스턴스를 생성할 때 특정 고객 생성 DB 파라미터 그룹을 사용해야 할 경우 이 정책을 적용할 수 있습니다. Deny를 사용하는 정책은 더 광범위한 정책에서 부여한 액세스 권한을 제한하기 위해 가장 자주 사용됩니다.

명시적으로 거부하는 권한은 이미 부여된 다른 모든 권한에 우선합니다. 따라서 부여하지 않으려는 권한을 자격 증명이 우연히 획득하지 않도록 할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyProductionCreate",
            "Effect": "Deny",
            "Action": "rds>CreateDBInstance",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "rds:pg-tag/usage": "prod"
                }
            }
        }
    ]
}
```

예제 3: 인스턴스 이름에 사용자 이름이 접두사로 붙은 DB 인스턴스 작업에 대한 권한 부여

다음 정책은 DB 인스턴스 이름에 사용자 이름이 접두사로 붙어 있고 devo와 동일한 stage라는 태그가 있거나 stage라는 태그가 없는 DB 인스턴스에서 API(AddTagsToResource 또는 RemoveTagsFromResource 제외)를 호출할 수 있는 권한을 허용합니다.

정책의 Resource 줄은 Amazon 리소스 이름(ARN)을 기준으로 리소스를 식별합니다. Amazon RDS 리소스에서 ARN을 사용하는 방법에 대한 자세한 내용은 [Amazon RDS의 Amazon 리소스 이름\(ARN\)을 사용한 작업 \(p. 172\)](#)을 참조하십시오.

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

MySQL 및 PostgreSQL을 위한 IAM 데이터베이스 인증

AWS Identity and Access Management(IAM) 데이터베이스 인증을 사용하여 DB 인스턴스에 인증할 수 있습니다. IAM 데이터베이스 인증은 MySQL 및 PostgreSQL에서 작동합니다. 이러한 인증 방식은 DB 인스턴스에 연결할 때 암호를 사용할 필요 없습니다. 대신에 인증 토큰을 사용합니다.

인증 토큰이란 요청이 있을 때 Amazon RDS가 생성하는 고유 문자열입니다. 인증 토큰은 AWS 서명 버전 4를 통해 생성됩니다. 각 토큰의 수명은 15분입니다. 인증을 외부에서 사용해 관리하기 때문에 사용자 자격 증명을 데이터베이스에 저장할 필요도 없습니다. 또한 표준 데이터베이스 인증 방식도 사용 가능합니다.

IAM 데이터베이스 인증은 다음과 같은 이점이 있습니다.

- 데이터베이스를 오가는 네트워크 트래픽은 SSL(Secure Sockets Layer)을 통해 암호화됩니다.
- 데이터베이스 리소스에 대한 액세스는 DB 인스턴스에서 개별적으로 관리할 필요 없이 IAM을 통해 중앙에서 관리할 수 있습니다.
- Amazon EC2에서 실행되는 애플리케이션의 경우, 암호가 아닌 EC2 인스턴스용 프로파일 자격 증명을 사용해 데이터베이스에 액세스하기 때문에 보안을 더욱 강화하는 효과가 있습니다.

주제

- [IAM 데이터베이스 인증 방식의 가용성 \(p. 369\)](#)
- [IAM 데이터베이스 인증에 대한 MySQL 한도 \(p. 370\)](#)
- [IAM 데이터베이스 인증에 대한 PostgreSQL 한도 \(p. 370\)](#)
- [IAM 데이터베이스 인증의 활성화 및 비활성화 \(p. 370\)](#)
- [IAM 데이터베이스 액세스를 위한 IAM 정책 생성 및 사용 \(p. 372\)](#)
- [IAM 인증을 사용하여 데이터베이스 계정 생성 \(p. 374\)](#)
- [IAM 인증을 사용하여 DB 인스턴스에 연결 \(p. 375\)](#)

IAM 데이터베이스 인증 방식의 가용성

IAM 데이터베이스 인증 방식은 다음과 같은 데이터베이스 엔진 및 인스턴스 클래스에서 사용할 수 있습니다.

- MySQL 5.6, 마이너 버전 5.6.34 이상. db.t2.small과 db.m1.small을 제외한 모든 DB 인스턴스 클래스가 지원됩니다.
- MySQL 5.7, 마이너 버전 5.7.16 이상. db.t2.small과 db.m1.small을 제외한 모든 DB 인스턴스 클래스가 지원됩니다.
- PostgreSQL 버전 10.6 이상, 버전 9.6.11 이상 및 버전 9.5.15 이상.

Note

IAM 데이터베이스 인증은 MySQL 5.5 또는 MySQL 8.0에서 지원되지 않습니다.

IAM 데이터베이스 인증에 대한 MySQL 한도

MySQL에 IAM 데이터베이스 인증을 사용할 경우 새로운 연결 수는 초당 최대 256개로 제한됩니다. db.t2.micro 인스턴스 클래스를 사용하는 경우의 연결 한도는 초당 10개입니다.

Amazon RDS에서 작동하는 데이터베이스 엔진은 초당 인증 횟수에 제한이 없습니다. 하지만 IAM 데이터베이스 인증 방식을 사용할 때는 애플리케이션이 인증 토큰을 생성해야 합니다. 이렇게 생성된 토큰은 애플리케이션이 DB 인스턴스에 연결하는 데 사용됩니다. 초당 허용되는 새 연결의 최대 수를 초과하면 IAM 데이터베이스 인증에 오버헤드가 추가로 발생하여 연결 병목 현상이 발생할 수 있습니다. 추가 오버헤드로 인해 기존 연결까지 끊어질 수도 있습니다. MySQL용 최대 총 연결에 대한 자세한 내용은 [최대 MySQL 연결 \(p. 599\)](#) 단원을 참조하십시오.

MySQL 엔진을 사용할 경우 다음을 따르십시오.

- IAM 데이터베이스 인증 방식은 개인이 일시적으로 데이터베이스에 액세스하기 위한 메커니즘으로 사용하십시오.
- 재시도가 용이한 워크로드에서만 IAM 데이터베이스 인증 방식을 사용하십시오.
- 애플리케이션에 초당 256개 이상의 새 연결이 필요한 경우에는 IAM 데이터베이스 인증 방식을 사용하지 마십시오.

IAM 데이터베이스 인증에 대한 PostgreSQL 한도

PostgreSQL과 함께 IAM 데이터베이스 인증을 사용하는 경우 다음 제한 사항을 적어둡니다.

- 데이터베이스 인스턴스에 대한 초당 최대 연결 수는 인스턴스 유형 및 워크로드에 따라 제한할 수 있습니다.

IAM 데이터베이스 인증의 활성화 및 비활성화

DB 인스턴스에서는 기본적으로 IAM 데이터베이스 인증이 비활성화되어 있습니다. IAM 데이터베이스 인증은 AWS Management 콘솔, AWS CLI 또는 API를 사용하여 활성화하거나 다시 비활성화 할 수 있습니다.

PostgreSQL DB 인스턴스에 대한 IAM 인증을 사용하려면 SSL 값이 1이어야 합니다. SSL 값이 0인 경우 PostgreSQL DB 인스턴스에 대해 IAM 인증을 활성화할 수 없습니다. PostgreSQL DB 인스턴스에 대해 IAM 인증을 활성화할 경우 SSL 값을 0으로 변경할 수 없습니다.

AWS Management 콘솔

콘솔을 사용하여 IAM 인증을 통해 새로운 DB 인스턴스를 생성하려면 [MySQL 데이터베이스 엔진 기반 DB 인스턴스 생성 \(p. 587\)](#) 또는 [PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스 만들기 \(p. 947\)](#) 단원을 참조하십시오.

각 생성 워크플로우에는 고급 설정 구성 페이지가 있으며, 여기서 IAM DB 인증을 활성화할 수 있습니다. 이 페이지의 [Database Options] 섹션에 있는 [Enable IAM DB Authentication]에서 [Yes]를 선택합니다.

기존 DB 인스턴스에서 IAM 인증을 활성화하거나 비활성화하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 수정하려는 DB 인스턴스를 선택합니다.
4. [Modify]를 선택합니다.
5. 데이터베이스 옵션 섹션의 IAM DB 인증에서 IAM DB 인증 또는 비활성화를 선택한 다음 계속을 선택합니다.
6. 변경 사항을 즉시 적용하려면 즉시 적용을 선택합니다.
7. DB 인스턴스 수정을 선택합니다.

DB 인스턴스를 복원하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 복원할 스냅샷을 선택한 다음 작업에서 스냅샷 복원을 선택합니다.
4. 설정 섹션에서 DB 인스턴스 ID에 DB 인스턴스의 ID를 입력합니다.
5. [Database options] 섹션의 [IAM DB authentication]에서 [Enable IAM DB authorization] 또는 [Disable]을 선택합니다.
6. [Restore DB Instance]를 선택합니다.

AWS CLI

AWS CLI를 사용하여 새로운 DB 인스턴스를 IAM 인증 방식으로 생성하려면 `create-db-instance` 명령을 사용하십시오. 다음 예제와 같이 `--enable-iam-database-authentication` 옵션을 지정합니다.

```
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
```

기존 DB 클러스터를 업데이트하여 IAM 인증을 사용하게 하거나 사용하지 않게 하려면 AWS CLI 명령 `modify-db-instance`를 사용합니다. 상황에 따라 `--enable-iam-database-authentication` 또는 `--no-enable-iam-database-authentication` 옵션을 지정합니다.

기본적으로 Amazon RDS는 다음 유지 관리 기간에 수정 작업을 수행합니다. 이러한 기본 설정을 무시하고 IAM DB 인증을 최대한 빠르게 활성화하려면 `--apply-immediately` 파라미터를 사용합니다.

다음은 기존 DB 인스턴스일 때 IAM 인증을 바로 활성화하는 방법을 설명한 예제입니다.

```
aws rds modify-db-instance \
  --db-instance-identifier mydbinstance \
  --apply-immediately \
  --enable-iam-database-authentication
```

DB 인스턴스를 복원하는 경우에는 다음 AWS CLI 명령 중 하나를 사용하십시오.

- `restore-db-instance-to-point-in-time`
- `restore-db-instance-from-db-snapshot`

IAM 데이터베이스 인증은 기본적으로 원본 스냅샷으로 기본 설정됩니다. 이 설정을 변경하려면 상황에 따라 `--enable-iam-database-authentication` 또는 `--no-enable-iam-database-authentication` 옵션을 설정합니다.

RDS API

API를 사용하여 새로운 DB 인스턴스를 IAM 인증 방식으로 생성하려면 API 작업 `CreateDBInstance`를 사용하십시오. `EnableIAMDatabaseAuthentication` 파라미터를 `true`로 설정합니다.

기존 DB 인스턴스를 업데이트하여 IAM 인증을 사용하게 하거나 사용하지 않게 하려면 API 작업 `ModifyDBInstance`를 사용합니다. `EnableIAMDatabaseAuthentication` 파라미터를 `true`로 설정하여 IAM 인증을 활성화하거나, `false`로 설정하여 비활성화합니다.

DB 인스턴스를 복원하는 경우에는 다음 API 작업 중 하나를 사용하십시오.

- [RestoreDBInstanceToPointInTime](#)
- [RestoreDBInstanceFromDBSnapshot](#)

IAM 데이터베이스 인증은 기본적으로 원본 스냅샷으로 기본 설정됩니다. 이 설정을 변경하려면 `EnableIAMDatabaseAuthentication` 파라미터를 `true`로 설정하여 IAM 인증을 활성화하거나, 혹은 `false`로 설정하여 비활성화합니다.

IAM 데이터베이스 액세스를 위한 IAM 정책 생성 및 사용

IAM 사용자 또는 역할이 DB 인스턴스에 연결할 수 있도록 허용하려면 IAM 정책을 생성해야 합니다. 그런 다음 정책을 IAM 사용자 또는 역할에 연결해야 합니다.

Note

IAM 정책에 대한 자세한 정보는 [인증 및 액세스 제어 \(p. 339\)](#) 단원을 참조하십시오.

다음은 IAM 사용자가 IAM 데이터베이스 인증 방식을 사용해 DB 인스턴스에 연결할 수 있도록 허용하는 정책 예제입니다.

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

Note

`rds-db:` 접두사를 `rds:`로 시작하는 다른 Amazon RDS 작업 접두사와 혼동하지 마십시오. `rds-db:` 접두사와 `rds-db:connect` 작업은 IAM 데이터베이스 인증 전용입니다. 다른 컨텍스트에서는 유효하지 않습니다.

현재 IAM 콘솔에는 `rds-db:connect` 작업이 포함된 정책에 대한 오류가 표시됩니다. 이 오류는 무시할 수 있습니다.

위의 예제 정책에는 다음 요소와 함께 단일 문이 포함되어 있습니다.

- `Effect - Allow`를 지정하여 DB 인스턴스에 대한 액세스를 부여합니다. 액세스를 명시적으로 허용하지 않으면 액세스가 기본적으로 거부됩니다.
- `Action - rds-db:connect`를 지정하여 DB 인스턴스에 대한 연결을 허용합니다.
- `Resource -` 하나의 DB 인스턴스의 한 데이터베이스 계정을 기술하는 Amazon 리소스 이름(ARN)을 지정합니다. ARN 형식은 다음과 같습니다.

```
arn:aws:rds-db:region:account-id:dbuser:DbiResourceId/db-user-name
```

이 형식에서 다음 항목을 교체합니다.

- **region**은 DB 인스턴스 AWS 리전입니다. 정책 예제에서 사용되는 AWS 리전은 **us-east-2**입니다.
- **account-id**은 DB 인스턴스의 AWS 계정 번호입니다. 정책 예제에서 사용되는 계정 번호는 **1234567890**입니다.
- **DbiResourceId**는 DB 인스턴스의 식별자입니다. 이 식별자는 AWS 리전에 고유하며, 절대로 바뀌지 않습니다. 정책 예제에서 사용되는 식별자는 **db-ABCDEFGHIJKLM01234**입니다.

Amazon RDS용 AWS Management 콘솔에서 DB 인스턴스 리소스 ID를 찾으려면 DB 인스턴스를 선택하여 세부 정보를 확인하십시오. 그런 다음 구성 탭을 선택합니다. 그러면 리소스 ID가 구성 섹션에 표시됩니다.

그 밖에 다음과 같이 AWS CLI 명령을 사용하여 현재 AWS 리전에 속한 모든 DB 인스턴스의 식별자와 리소스 ID 목록을 조회하는 방법도 있습니다.

```
aws rds describe-db-instances --query "DBInstances[*].  
[DBInstanceIdentifier,DbiResourceId]"
```

- **db-user-name**은 IAM 인증과 연결할 데이터베이스 계정 이름입니다. 정책 예제에서 사용되는 데이터베이스 계정은 **db_user**입니다.

다른 ARN을 구성하여 다양한 액세스 패턴을 지원할 수 있습니다. 다음 정책에서는 DB 인스턴스에서 서로 다른 데이터베이스 계정 2개에 대한 액세스를 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "rds-db:connect"  
            ],  
            "Resource": [  
                "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKLM01234/jane_doe",  
                "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKLM01234/mary_roe"  
            ]  
        }  
    ]  
}
```

다음 정책에서는 "*" 문자를 사용하여 특정 AWS 계정과 AWS 리전의 모든 DB 인스턴스 및 데이터베이스 계정을 일치시킵니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "rds-db:connect"  
            ],  
            "Resource": [  
                "arn:aws:rds-db:us-east-2:1234567890:dbuser:/*"  
            ]  
        }  
    ]  
}
```

다음 정책은 특정 AWS 계정과 AWS 리전의 모든 DB 인스턴스를 일치시킵니다. 하지만 정책에 따라 jane_doe 데이터베이스 계정을 가지고 있는 DB 인스턴스에게만 액세스 권한이 부여됩니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "rds-db:connect"  
            ],  
            "Resource": [  
                "arn:aws:rds-db:us-east-2:123456789012:dbuser:*/jane_doe"  
            ]  
        }  
    ]  
}
```

IAM 사용자 또는 역할은 데이터베이스 사용자가 액세스할 수 있는 데이터베이스에만 액세스할 수 있습니다. 예를 들어 DB 인스턴스에 이름이 dev인 데이터베이스와 test인 데이터베이스가 있다고 가정하겠습니다. 이 때 데이터베이스 사용자인 jane_doe가 dev에 대한 액세스 권한만 가지고 있다면 사용자 jane_doe와 함께 해당 DB 인스턴스에 액세스하는 IAM 사용자 또는 역할도 dev 액세스 권한만 갖게 됩니다. 이러한 액세스 제한은 테이블, 뷰 등 다른 데이터베이스 객체에 대해서도 똑같이 적용됩니다.

IAM 정책과 IAM 사용자 또는 역할의 연결

데이터베이스 인증을 위한 IAM 정책을 생성하였으면 이제 정책을 IAM 사용자 또는 역할에 연결해야 합니다. 이번 주제에 대한 자습서는 IAM 사용자 안내서에서 [첫 번째 고객 관리형 정책 만들기 및 연결](#) 단원을 참조하십시오.

자습서를 읽어보면 이번 단원에서 소개하는 정책 예제 중 한 가지를 출발점으로 자신만의 요건에 따라 지정하여 사용할 수 있습니다. 자습서를 끝까지 따르다 보면 연결된 정책을 통해 rds-db:connect 작업이 가능한 IAM 사용자를 얻게 될 것입니다.

Note

여러 IAM 사용자 또는 역할을 동일한 데이터베이스 사용자 계정에 매핑할 수 있습니다. 예를 들어 IAM 정책이 다음과 같은 리소스 ARN을 지정하였다고 가정하겠습니다.

```
arn:aws:rds-db:us-east-2:123456789012:dbuser:db-12ABC34DEFG5HIJ6KLMNOP78QR/jane_doe
```

IAM 사용자 Jane, Bob 및 Diego에게 정책을 연결하면 각 사용자는 jane_doe 데이터베이스 계정을 사용하여 지정된 DB 인스턴스에 연결할 수 있습니다.

IAM 인증을 사용하여 데이터베이스 계정 생성

IAM 데이터베이스 인증 방식에서는 데이터베이스 암호를 사용자 계정에 할당할 필요 없습니다. 데이터베이스 계정에 매핑되어 있는 IAM 사용자를 제거할 경우에는 DROP USER 문으로 데이터베이스 계정도 제거해야 합니다.

PostgreSQL에서 IAM 인증 사용

PostgreSQL에 IAM 인증을 사용하려면 DB 인스턴스에 연결한 후 데이터베이스 사용자를 만들고 다음 예제와 같이 사용자에게 rds_iam 역할을 부여합니다.

```
CREATE USER db_userx WITH LOGIN;  
GRANT rds_iam TO db_userx;
```

MySQL에서 IAM 인증 사용

MySQL에서는 AWS에서 제공하는 플러그인인 `AWSAuthenticationPlugin`에서 인증을 처리합니다. 이 플러그인은 IAM과 연동하여 IAM 사용자를 인증합니다. 다음 예제와 같이, DB 인스턴스에 연결한 후 `CREATE USER` 문을 실행합니다.

```
CREATE USER jane_doe IDENTIFIED WITH AWSAuthenticationPlugin AS 'RDS';
```

`IDENTIFIED WITH` 절은 MySQL이 `AWSAuthenticationPlugin`을 사용하여 데이터베이스 계정 (`jane_doe`)을 인증할 수 있도록 허용하는 데 필요합니다. `AS 'RDS'` 절은 인증 방법을 나타냅니다. 지정한 데이터베이스 계정과 IAM 사용자나 역할의 이름이 동일해야 합니다. 이 예에서 데이터베이스 계정과 IAM 사용자, 역할의 이름은 모두 `jane_doe`로 동일해야 합니다.

Note

다음과 같은 메시지가 표시되면 현재 DB 인스턴스에서 AWS 제공 플러그인을 사용할 수 없는 것입니다.

`ERROR 1524 (HY000): Plugin 'AWSAuthenticationPlugin' is not loaded`
위와 같은 오류 문제를 해결하려면 지원되는 구성 사용하고 있는지, 그리고 DB 인스턴스에서 IAM 데이터베이스 인증이 활성화되어 있는지 확인하십시오. 자세한 내용은 [IAM 데이터베이스 인증 방식의 가용성 \(p. 369\)](#) 및 [IAM 데이터베이스 인증의 활성화 및 비활성화 \(p. 370\)](#) 단원을 참조하십시오.

`AWSAuthenticationPlugin`을 사용하여 계정을 생성한 이후 계정 관리 방법은 다른 데이터베이스 계정과 동일합니다. 예를 들어 `GRANT` 및 `REVOKE` 문으로 계정 권한을 수정하거나, 혹은 `ALTER USER` 문으로 여러 가지 계정 속성을 변경할 수 있습니다.

IAM 인증을 사용하여 DB 인스턴스에 연결

IAM 데이터베이스 인증 방식에서는 DB 인스턴스에 연결할 때 인증 토큰을 사용합니다. 인증 토큰이란 암호 대신 사용하는 문자열을 말합니다. 인증 토큰은 생성 후 15분 동안 유효하며 이 시간이 지나면 만료됩니다. 만료된 토큰을 사용하여 연결하려고 하면 연결 요청이 거부됩니다.

모든 인증 토큰은 AWS 서명 버전 4를 사용하여 유효한 서명이 있어야 합니다. (자세한 내용은 AWS General Reference의 [서명 버전 4 서명 프로세스](#) 참조) AWS CLI와 AWS SDK for Java는 생성되는 각 토큰에 자동으로 설명할 수 있습니다.

AWS Lambda 같은 다른 AWS 서비스에서 Amazon RDS에 연결할 때 인증 토큰을 사용할 수 있습니다. 토큰을 사용하면 코드에 암호를 넣지 않아도 됩니다. 그 밖에 AWS SDK for Java를 사용하여 인증 토큰을 프로그래밍 방식으로 생성하고 프로그래밍 방식으로 서명하는 방법도 있습니다.

IAM 인증 토큰에 서명까지 마쳤으면 이제 Amazon RDS DB에 연결할 수 있습니다. 다음 단원에서는 명령줄 도구 또는 AWS SDK for Java를 사용하여 연결하는 방법에 대해 알아보겠습니다.

자세한 내용은 [Use IAM authentication to connect with SQL Workbench/J to Amazon Aurora MySQL or Amazon RDS for MySQL](#) 단원을 참조하십시오.

주제

- [명령줄: AWS CLI 및 mysql 클라이언트에서 DB 인스턴스에 연결 \(p. 376\)](#)
- [명령줄 :AWS CLI 및 psql 클라이언트에서 DB 인스턴스에 연결 \(p. 377\)](#)
- [AWS SDK for Java를 사용하여 DB 인스턴스에 연결 \(p. 378\)](#)

명령줄: AWS CLI 및 mysql 클라이언트에서 DB 인스턴스에 연결

아래 설명과 같이 AWS CLI 및 mysql 명령줄 도구를 사용하여 Amazon RDS DB 인스턴스에 연결할 수 있습니다.

주제

- [IAM 인증 토큰 생성 \(p. 376\)](#)
- [DB 인스턴스에 연결 \(p. 376\)](#)

IAM 인증 토큰 생성

다음은 AWS CLI를 사용하여 서명이 되어 있는 인증 토큰을 생성하는 방법을 설명한 예제입니다.

```
aws rds generate-db-auth-token \
  --hostname rdsmysql.cdgmuqiadpid.us-west-2.rds.amazonaws.com \
  --port 3306 \
  --region us-west-2 \
  --username jane_doe
```

위의 예제에서 각 파라미터는 다음과 같습니다.

- **--hostname** – 액세스할 DB 인스턴스의 호스트 이름입니다.
- **--port** – DB 인스턴스에 연결할 때 사용할 포트 이름입니다.
- **--region** – DB 인스턴스가 실행되는 AWS 리전입니다.
- **--username** – 액세스할 데이터베이스 계정입니다.

토큰에서 가장 앞의 일부 문자는 다음과 같은 모습입니다.

```
rdsmysql.cdgmuqiadpid.us-west-2.rds.amazonaws.com:3306/?Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

DB 인스턴스에 연결

일반적인 연결 형식은 다음과 같습니다.

```
mysql --host=hostName --port=portNumber --ssl-ca=[full path]rds-combined-ca-bundle.pem --enable-cleartext-plugin --user=userName --password=authToken
```

파라미터는 다음과 같습니다.

- **--host** – 액세스할 DB 인스턴스의 호스트 이름입니다.
- **--port** – DB 인스턴스에 연결할 때 사용할 포트 이름입니다.
- **--ssl-ca** – 퍼블릭 키를 포함하는 SSL 인증서 파일입니다. 자세한 내용은 [SSL을 사용하여 DB 인스턴스에 대한 연결 암호화 \(p. 388\)](#) 단원을 참조하십시오.
- **--enable-cleartext-plugin** – 현재 연결에서 AWSAuthenticationPlugin을 사용하도록 지정하는 값입니다.
- **--user** – 액세스할 데이터베이스 계정입니다.
- **--password** – 서명이 되어 있는 IAM 인증 토큰입니다.

인증 토큰은 수백 자의 문자로 구성됩니다. 그렇기 때문에 명령줄에서는 다루지 불편할 수도 있습니다. 이러한 문제를 해결하기 위해 토큰을 환경 변수로 저장한 후 연결할 때 이 변수를 사용하는 것도 한 가지 방법입니다. 다음은 이러한 문제 해결 방법을 설명한 예제입니다.

```
RDSHOST="rdsmysql.cdgmuqiadpid.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
```

AWSAuthenticationPlugin을 사용하여 연결할 때는 SSL을 통해 보안을 유지합니다. 이러한 보안 여부를 확인하려면 mysql> 명령 프롬프트에 다음과 같이 입력합니다.

```
show status like 'Ssl%';
```

그러면 출력 시 다음과 같이 자세하게 표시됩니다.

```
+-----+-----+
| Variable_name | Value
+-----+-----+
| ...          | ...
| Ssl_cipher    | AES256-SHA
|
| ...          | ...
| Ssl_version   | TLSv1.1
|
| ...          | ...
+-----+
```

명령줄 :AWS CLI 및 psql 클라이언트에서 DB 인스턴스에 연결

아래 설명과 같이 AWS CLI 및 psql 명령줄 도구를 사용하여 PostgreSQL용 Amazon RDS DB 인스턴스에 연결할 수 있습니다.

주제

- [IAM 인증 토큰 생성 \(p. 377\)](#)
- [Amazon RDS PostgreSQL 인스턴스에 연결 \(p. 378\)](#)

IAM 인증 토큰 생성

인증 토큰은 수백 자의 문자로 구성되므로 명령줄에서는 다루기 불편할 수 있습니다. 이러한 문제를 해결하기 위해 토큰을 환경 변수로 저장한 후 연결할 때 이 변수를 사용하는 것도 한 가지 방법입니다. 다음 예제는 AWS CLI에서 generated-db-auth-token 명령을 사용하여 서명된 인증 토큰을 받고 이를 PGASSWORD 환경 변수에 저장하는 방법을 보여 줍니다.

```
export RDSHOST="rdspostgres.cdgmuqiadpid.us-west-2.rds.amazonaws.com"
export PGPASSWORD=$(aws rds generate-db-auth-token --hostname $RDSHOST --port 5432 --
region us-west-2 --username jane_doe )"
```

예제에서 generate-db-auth-token 명령에 대한 파라미터는 다음과 같습니다.

- --hostname – 액세스할 DB 인스턴스의 호스트 이름입니다.
- --port – DB 인스턴스에 연결할 때 사용할 포트 이름입니다.
- --region – DB 인스턴스가 실행되는 AWS 리전입니다.
- --username – 액세스할 데이터베이스 계정입니다.

생성된 토큰에서 처음 몇 글자는 다음과 같은 모습입니다.

```
rdspostgres.cdgmuqiadpid.us-west-2.rds.amazonaws.com:5432/?  
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

Amazon RDS PostgreSQL 인스턴스에 연결

psql을 사용한 일반적인 연결 형식은 다음과 같습니다.

```
psql "host=hostName port=portNumber sslmode=verify-full sslrootcert=certificateFile  
dbname=DBName user=userName"
```

파라미터는 다음과 같습니다.

- host – 액세스할 DB 인스턴스의 호스트 이름입니다.
- port – DB 인스턴스에 연결할 때 사용할 포트 이름입니다.
- sslmode – 사용할 SSL 모드입니다. sslmode=verify-full을 사용하면 SSL 연결에서 SSL 인증서의 엔드포인트와 비교하여 DB 인스턴스 엔드포인트를 확인합니다.
- sslrootcert – 퍼블릭 키를 포함하는 SSL 인증서 파일입니다. 자세한 내용은 [PostgreSQL DB 인스턴스와 함께 SSL 사용을 참조하십시오.](#)
- dbname – 액세스할 데이터베이스 계정입니다.
- user – 액세스할 데이터베이스 계정입니다.

다음 예제는 명령을 사용하여 연결하는 것을 보여 줍니다. 이 예제는 이전 단원에서 토큰이 생성될 때 설정된 환경 변수를 사용합니다.

```
psql "host=$RDSHOST port=5432 sslmode=verify-full sslrootcert=/sample_dir/rds-combined-ca-  
bundle.pem dbname=DBName user=jane_doe"
```

AWS SDK for Java를 사용하여 DB 인스턴스에 연결

아래 설명과 같이 AWS SDK for Java를 사용하여 명령줄에서 Amazon RDS DB 인스턴스로 연결할 수 있습니다.

주제

- [IAM 인증 토큰 생성 \(p. 378\)](#)
- [IAM 인증 토큰 수동 구성 \(p. 379\)](#)
- [DB 인스턴스에 연결 \(p. 382\)](#)

IAM 인증 토큰 생성

AWS SDK for Java로 프로그램을 개발할 때는 `RdsIamAuthTokenGenerator` 클래스를 사용하여 서명된 인증 토큰을 가져올 수 있습니다. 이 클래스를 사용하려면 AWS 자격 증명을 입력해야 합니다. 이렇게 하려면 `DefaultAWSCredentialsProviderChain` 클래스의 인스턴스를 생성합니다. `DefaultAWSCredentialsProviderChain`은 [기본 자격 증명 공급자 체인](#)에서 찾은 첫 번째 AWS 액세스 키와 보안 키를 사용합니다. AWS 액세스 키에 대한 자세한 내용은 [IAM 사용자를 위한 액세스 키 관리](#) 단원을 참조하십시오.

`RdsIamAuthTokenGenerator` 인스턴스를 생성한 후에는 `getAuthToken` 메서드를 호출하여 서명된 토큰을 가져올 수 있습니다. 이때 AWS 리전, 호스트 이름, 포트 이름 및 사용자 이름을 입력합니다. 다음은 각 정보의 입력 방법을 설명한 코드 예제입니다.

```
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.cdgmuiadpid.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;
    }
}
```

IAM 인증 토큰 수동 구성

Java에서 인증 토큰을 가장 쉽게 생성할 수 있는 방법은 `RdsIamAuthTokenGenerator`를 사용하는 것입니다. 이 클래스는 인증 토큰을 생성한 후 AWS 서명 버전 4를 사용해 서명까지 마칩니다. 자세한 내용은 AWS General Reference의 [서명 버전 4 서명 프로세스](#) 단원을 참조하십시오.

그 밖에 다음 코드 예제와 같이 인증 토큰을 수동으로 구성하여 서명하는 방법도 있습니다.

```
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.cdgmuiadpid.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'HHmmssZ").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,
newSigningKey(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
    //should be in format YYYYMMDDTHHMMSSZ
    public static String createCanonicalString(String user, String accessKey, String date,
String dateTimeStamp, 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", dateTimeStamp);
        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.toHex(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 dateTime, String canonicalRequest,
String accessKey, String date, String region) throws Exception {
    String credentialScope = date + "/" + region + "/" + serviceName + "/aws4_request";
    return algorithm + '\n' + dateTime + '\n' + credentialScope + '\n' +
BinaryUtils.toHex(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);
    }
}
}
```

DB 인스턴스에 연결

다음은 인증 토큰을 생성한 다음 이 토큰을 사용하여 MySQL을 실행하는 인스턴스에 연결하는 방법을 보여주는 코드 예제입니다.

이 코드 예제를 실행하려면 [AWS SDK for Java](#)가 필요하며 이것은 AWS에서 받을 수 있습니다. 또한 다음이 필요합니다.

- MySQL Connector/J. 이 코드 예제는 `mysql-connector-java-5.1.33-bin.jar`을 사용하여 테스트되었습니다.
- 한 AWS 리전에 고유한 Amazon RDS 중간 인증서입니다. (자세한 내용은 [SSL을 사용하여 DB 인스턴스에 대한 연결 암호화 \(p. 388\)](#) 단원을 참조하십시오.) 예제가 실행되면 클래스 로더가 쉽게 찾을 수 있도록 이 Java 코드 예제와 동일한 디렉터리에서 인증서를 찾기 시작합니다.
- 필요하다면 다음 변수 값을 변경합니다.
 - `RDS_INSTANCE_HOSTNAME` – 액세스할 DB 인스턴스의 호스트 이름입니다.
 - `RDS_INSTANCE_PORT` – PostgreSQL DB 인스턴스에 연결할 때 사용할 포트 이름입니다.
 - `REGION_NAME` – DB 인스턴스가 실행되는 AWS 리전입니다.
 - `DB_USER` – 액세스할 데이터베이스 계정입니다.
 - `SSL_CERTIFICATE` – 한 AWS 리전에 특정한 Amazon RDS용 SSL 인증서입니다. AWS 리전 인증서를 다운로드하는 방법은 [중간 인증서 \(p. 389\)](#) 단원을 참조하십시오. SSL 인증서는 예제 실행 시 클래스 로더가 인증서를 찾을 수 있도록 이 Java 프로그램 파일과 동일한 디렉터리에 설치합니다.

다음은 [기본 자격 증명 공급자 체인](#)에서 AWS 자격 증명을 가져오는 코드 예제입니다.

```
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.cdgmuqiadpid.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-2015-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, setMySqlConnectionProperties());
    }

    /**
     * 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 setMySqlConnectionProperties() {
        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:
     * btusi123.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=AKIAPFXHGVDI5RNFO4AQ%2F20171003%2Fcn-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");
}

}
```

Amazon RDS 리소스 암호화

Amazon RDS DB 인스턴스에 대해 암호화 옵션을 활성화하여 휴면 상태의 Amazon RDS DB 인스턴스와 스냅샷을 암호화할 수 있습니다. 휴면 상태에서 암호화된 데이터에는 DB 인스턴스에 대한 기본 스토리지, 자동 백업, 읽기 전용 복제본, 스냅샷이 포함됩니다.

Amazon RDS 암호화된 DB 인스턴스는 Amazon RDS DB 인스턴스를 호스팅하는 서버의 데이터를 업계 표준 AES-256 암호화 알고리즘을 사용하여 암호화합니다. 데이터가 암호화되면 Amazon RDS가 성능에 미치는 영향을 최소화한 상태에서 데이터 액세스 및 암호 해독의 인증을 투명하게 처리합니다. 암호화를 사용하도록 데이터베이스 클라이언트 애플리케이션을 수정하지 않아도 됩니다.

Note

교차 리전 읽기 전용 복제본이 있는 암호화 및 비암호화 DB 인스턴스의 경우, 소스 및 읽기 전용 복제본 간에 전송 중인 데이터는 암호화됩니다.

주제

- [Amazon RDS 리소스 암호화 개요 \(p. 386\)](#)
- [DB 인스턴스에 대해 Amazon RDS 암호화 활성화 \(p. 386\)](#)
- [Amazon RDS 암호화 가용성 \(p. 387\)](#)
- [Amazon RDS 암호화 키 관리 \(p. 388\)](#)
- [Amazon RDS 암호화된 DB 인스턴스의 제한 \(p. 388\)](#)

Amazon RDS 리소스 암호화 개요

Amazon RDS 암호화된 DB 인스턴스는 기본 스토리지에 대한 무단 액세스로부터 데이터의 보안을 유지해 추가 계층의 데이터 보호를 제공합니다. 클라우드에 배포된 애플리케이션의 데이터 보호를 강화하고 휴면 상태의 데이터 암호화를 위한 규정 준수 요구 사항을 만족하기 위해 Amazon RDS 암호화를 사용할 수 있습니다.

또한 Amazon RDS에서는 TDE(Transparent Data Encryption)를 사용하여 Oracle 또는 SQL Server DB 인스턴스 암호화를 지원합니다. TDE를 저장 데이터 암호화와 함께 사용하면 데이터베이스의 성능에 약간의 영향을 미칠 수 있지만, TDE를 저장 데이터 암호화와 함께 사용할 수 있습니다. 암호화 방법별로 다른 키를 관리해야 합니다. TDE에 대한 자세한 내용은 [Oracle Transparent Data Encryption \(p. 827\)](#) 또는 [Microsoft SQL Server Transparent Data Encryption\(TDE\) 지원 \(p. 547\)](#) 단원을 참조하십시오.

Amazon RDS 리소스의 암호화 및 암호 해독에 사용되는 키를 관리하려면 [AWS Key Management Service\(AWS KMS\)](#)를 사용합니다. AWS KMS는 클라우드에 맞게 확장된 키 관리 시스템을 제공하기 위해 안전하고 가용성이 높은 하드웨어 및 소프트웨어를 결합합니다. AWS KMS를 사용하면 암호화 키를 생성하고 이 키를 사용할 수 있는 방법을 제어하는 정책을 정의할 수 있습니다. AWS KMS는 CloudTrail를 지원하므로 키가 적절하게 사용되고 있는지 확인하기 위해 키 사용을 감사할 수 있습니다. AWS KMS 키는 Amazon RDS 및 지원되는 AWS 서비스(예: Amazon Simple Storage Service(Amazon S3), Amazon Elastic Block Store(Amazon EBS), Amazon Redshift)와 함께 사용할 수 있습니다. AWS KMS를 지원하는 서비스 목록을 보려면 AWS Key Management Service 개발자 안내서의 [지원되는 서비스](#)로 이동하십시오.

Amazon RDS 암호화된 DB 인스턴스의 경우 모든 로그, 백업 및 스냅샷이 암호화됩니다. 동일한 리전에 있는 경우, Amazon RDS 암호화된 인스턴스의 읽기 전용 복제본 또한 마스터 인스턴스와 동일한 키를 사용하여 암호화됩니다. 마스터와 읽기 전용 복제본이 서로 다른 리전에 있을 경우 해당 리전의 암호화 키를 사용하여 암호화하십시오.

DB 인스턴스에 대해 Amazon RDS 암호화 활성화

새로운 DB 인스턴스에 대해 암호화를 활성화하려면 Amazon RDS 콘솔에서 암호화 활성화를 선택합니다. DB 인스턴스 생성에 대한 자세한 내용은 다음 항목 중 하나를 참조하십시오.

- [MySQL 데이터베이스 엔진 기반 DB 인스턴스 생성 \(p. 587\)](#)
- [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#)
- [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 생성 \(p. 494\)](#)
- [PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스 만들기 \(p. 947\)](#)
- [MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 437\)](#)

`rds-create-db-instance` AWS CLI 명령을 사용하여 암호화된 RDS DB 인스턴스를 생성할 경우 `--storage-encrypted` 파라미터를 `true`로 설정하십시오. [CreateDBInstance](#) API 작업을 사용할 경우 `StorageEncrypted` 파라미터를 `true`로 설정하십시오.

암호화된 DB 인스턴스를 생성할 때 암호화 키에 대한 AWS KMS 키 식별자를 제공할 수도 있습니다. AWS KMS 키 식별자를 지정하지 않으면 Amazon RDS는 새 DB 인스턴스에 대해 기본 암호화 키를 사용합니다. AWS KMS는 AWS 계정의 Amazon RDS에 대한 기본 암호화 키를 생성합니다. AWS 계정에는 AWS 리전마다 다른 기본 암호화 키가 있습니다.

암호화된 DB 인스턴스를 생성한 후에는 해당 인스턴스의 암호화 키를 변경할 수 없습니다. 따라서 암호화된 DB 인스턴스를 생성하기 전에 암호화 키 요구 사항을 결정해야 합니다.

AWS CLI `create-db-instance` 명령을 사용하여 암호화된 RDS DB 인스턴스를 생성할 경우 DB 인스턴스의 AWS KMS 암호화 키에 대한 Amazon 리소스 이름(ARN)으로 `--kms-key-id` 파라미터를 설정하십시오. Amazon RDS API `CreateDBInstance` 작업을 사용할 경우 DB 인스턴스의 AWS KMS 키에 대한 ARN으로 `KmsKeyId` 파라미터를 설정하십시오.

다른 계정에 있는 키의 ARN을 사용하여 RDS DB 인스턴스를 암호화할 수 있습니다. 혹은 새 DB 인스턴스를 암호화하는 데 사용된 AWS KMS 암호화 키를 소유한 동일한 AWS 계정으로 DB 인스턴스를 생성할 수도 있습니다. 이 경우에는 전달하는 AWS KMS 키 ID가 키의 ARN이 아닌 AWS KMS 키 별칭일 수도 있습니다.

Important

Amazon RDS에서 더 이상 DB 인스턴스에 대한 암호화 키에 액세스할 수 없는 경우(예: 키에 대한 RDS 액세스 권한이 취소된 경우) 암호화된 DB 인스턴스가 터미널 상태로 전환됩니다. 이러한 경우에는 백업 파일에서만 DB 인스턴스를 복원할 수 있습니다. 데이터베이스에서 암호화된 데이터가 손실되지 않도록 보호하려면 암호화된 DB 인스턴스에 대해 항상 백업을 활성화하는 것이 좋습니다.

Amazon RDS 암호화 가용성

Amazon RDS 암호화는 현재 모든 데이터베이스 엔진과 스토리지 형식에 사용할 수 있습니다. 중국(베이징) 리전에서는 현재 Amazon RDS 암호화를 사용할 수 없습니다.

Amazon RDS 암호화는 대부분의 DB 인스턴스 클래스에서 사용 가능합니다. 다음 테이블에는 Amazon RDS 암호화를 지원하지 않는 DB 인스턴스 클래스가 열거되어 있습니다.

인스턴스 유형	인스턴스 클래스
범용(M1)	db.m1.small db.m1.medium db.m1.large db.m1.xlarge
메모리 최적화(M2)	db.m2.xlarge db.m2.2xlarge db.m2.4xlarge
확장 가능(T2)	db.t2.micro

Note

SQL Server Express Edition을 실행하는 DB 인스턴스에는 비활성화되어 있는 암호화를 사용할 수 없습니다.

Amazon RDS 암호화 키 관리

IAM 콘솔에서 [AWS Key Management Service\(AWS KMS\)](#)를 사용하여 Amazon RDS 암호화된 DB 인스턴스에 사용되는 키를 관리할 수 있습니다. 키를 완벽하게 제어하기 위해서는 고객 관리 키를 생성해야 합니다.

AWS KMS가 프로비저닝하는 기본 키는 삭제, 취소 또는 교체가 불가능합니다. 스냅샷을 공유한 AWS 계정의 기본 AWS KMS 암호화 키를 사용하여 암호화된 스냅샷은 공유할 수 없습니다.

[AWS CloudTrail](#)을 사용하여 고객 관리 키로 수행한 모든 작업의 감사 로그를 볼 수 있습니다.

Important

암호화된 DB 인스턴스에 대해 키를 비활성화할 경우 해당 DB 인스턴스에서 읽거나 쓸 수 없습니다. Amazon RDS에서 Amazon RDS가 액세스할 수 없는 키로 암호화된 DB 인스턴스가 발견될 경우, Amazon RDS는 DB 인스턴스를 터미널 상태로 전환합니다. 이러한 상태에서는 DB 인스턴스를 더 이상 사용하지 못하기 때문에 데이터베이스의 현재 상태를 복구할 수 없습니다. DB 인스턴스를 복원하려면 Amazon RDS의 암호화 키에 대한 액세스 권한을 다시 활성화한 후 백업에서 DB 인스턴스를 복원해야 합니다.

Amazon RDS 암호화된 DB 인스턴스의 제한

Amazon RDS 암호화된 DB 인스턴스에는 다음과 같은 제한이 있습니다.

- Amazon RDS DB 인스턴스에 대한 암호화는 암호화를 생성할 때에만 활성화할 수 있으며 DB 인스턴스가 생성된 후에는 불가능합니다.

다만 암호화되지 않은 DB 스냅샷의 사본을 암호화할 수 있기 때문에 암호화되지 않은 DB 인스턴스에 실질적으로 암호화를 추가할 수 있습니다. 즉, DB 인스턴스의 스냅샷을 만든 다음 해당 스냅샷의 암호화된 사본을 만들 수 있습니다. 그런 다음 암호화된 스냅샷에서 DB 인스턴스를 복구할 수 있고, 원본 DB 인스턴스의 암호화된 사본이 생깁니다. 자세한 내용은 [스냅샷 복사 \(p. 207\)](#) 단원을 참조하십시오.

- 암호화된 DB 인스턴스는 암호화를 비활성화하도록 수정할 수 없습니다.
- 암호화되지 않은 DB 인스턴스의 암호화된 읽기 전용 복제본이나 암호화된 DB 인스턴스의 암호화되지 않은 읽기 전용 복제본은 보유할 수 없습니다.
- 암호화된 읽기 전용 복제본은 소스 DB 인스턴스와 동일한 키를 사용해 암호화해야 합니다.
- 암호화된 백업 또는 스냅샷을 암호화된 DB 인스턴스로 복원할 수 없습니다.
- 암호화된 스냅샷을 한 리전에서 다른 리전으로 복사하려면 대상 리전의 KMS 키 식별자를 지정해야 합니다. 왜냐하면 KMS 암호화 키는 이 키를 생성한 리전에 고유한 것이기 때문입니다.

소스 스냅샷은 복사 프로세스 전체에서 암호화를 유지합니다. AWS Key Management Service(KMS)는 봉투 암호화를 이용해 복사 프로세스가 진행되는 동안 데이터를 보호합니다. 봉투 암호화에 대한 자세한 내용은 [봉투 암호화](#)를 참조하십시오.

SSL을 사용하여 DB 인스턴스에 대한 연결 암호화

애플리케이션에서 SSL을 사용하여 MySQL, MariaDB, SQL Server, Oracle 또는 PostgreSQL을 실행하는 DB 인스턴스에 대한 연결을 암호화할 수 있습니다. 각 DB 엔진에는 SSL을 구현하기 위한 고유한 프로세스가 있습니다. DB 인스턴스에 대해 SSL을 구현하는 방법을 알아보려면 DB 엔진에 해당하는 아래 링크를 사용하십시오.

- [MariaDB DB 인스턴스와 함께 SSL 사용 \(p. 433\)](#)
- [Microsoft SQL Server DB 인스턴스와 함께 SSL 사용 \(p. 541\)](#)
- [MySQL DB 인스턴스와 함께 SSL 사용 \(p. 583\)](#)

- Oracle DB 인스턴스에 SSL 사용 (p. 708)
- PostgreSQL DB 인스턴스와 함께 SSL 사용 (p. 1059)

<https://s3.amazonaws.com/rds-downloads/rds-ca-2015-root.pem>에서 모든 리전에 적용되는 루트 인증서를 다운로드할 수 있습니다. 이 인증서는 신뢰할 수 있는 루트 개체이므로 대부분의 경우에 작동하지만 애플리케이션에서 인증서 체인을 허용하지 않을 경우 실패할 수 있습니다. 애플리케이션에서 인증서 체인을 허용하지 않는 경우 이 단원의 뒤에 나오는 중간 인증서 목록에서 AWS 리전별 인증서를 다운로드하십시오. <https://s3-us-gov-west-1.amazonaws.com/rds-downloads/rds-GovCloud-Root-CA-2017.pem>에서 AWS GovCloud 리전의 루트 인증서를 다운로드할 수 있습니다.

Note

모든 인증서는 SSL 연결을 통한 다운로드에만 사용 가능합니다.

중간 인증서와 루트 인증서를 모두 포함하는 인증서 번들은 <https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem>에서 다운로드할 수 있습니다.

AWS GovCloud 리전의 중간 인증서와 루트 인증서를 모두 포함하는 인증서 번들은 <https://s3-us-gov-west-1.amazonaws.com/rds-downloads/rds-combined-ca-us-gov-bundle.pem>에서 다운로드할 수 있습니다.

애플리케이션이 Microsoft Windows 플랫폼에 있고 PKCS7 파일이 필요한 경우 <https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.p7b>에서 중간 인증서와 루트 인증서를 모두 포함하는 PKCS7 인증서 번들을 다운로드할 수 있습니다.

중간 인증서

자신의 리전에 연결하려면 중간 인증서를 사용해야 할 수도 있습니다. 예를 들어 SSL을 사용하여 AWS GovCloud(미국 서부) 리전에 연결하려면 중간 인증서를 사용해야 합니다. 특정 AWS 리전에 대한 중간 인증서가 필요한 경우 다음 목록에서 인증서를 다운로드합니다.

[아시아 태평양\(홍콩\)](#)

[아시아 태평양\(뭄바이\)](#)

[아시아 태평양\(도쿄\)](#)

[아시아 태평양\(서울\)](#)

[아시아 태평양\(오사카-로컬\)](#)

[아시아 태평양\(싱가포르\)](#)

[아시아 태평양\(시드니\)](#)

[캐나다\(중부\)](#)

[중국\(베이징\)](#)

[중국\(닝샤\)](#)

[EU\(프랑크푸르트\)](#)

[EU\(아일랜드\)](#)

[EU\(런던\)](#)

[EU\(파리\)](#)

[EU\(스톡홀름\)](#)

[남아메리카\(상파울루\)](#)

[미국 동부\(버지니아 북부\)](#)

[미국 동부\(오하이오\)](#)

[미국 서부\(캘리포니아 북부 지역\)](#)

[미국 서부\(오레곤\)](#)

[AWS GovCloud\(US 동부\)\(CA-2017\)](#)

[AWS GovCloud\(US 서부\)\(CA-2017\)](#)

[AWS GovCloud\(US 서부\)\(CA-2012\)](#)

보안 그룹을 통한 액세스 제어

보안 그룹은 DB 인스턴스에서 송수신되는 트래픽에 대한 액세스를 제어합니다. DB 보안 그룹, VPC 보안 그룹, Amazon EC2 보안 그룹이라는 세 가지 유형의 보안 그룹이 Amazon RDS에서 사용됩니다. 쉽게 말해서 각 보안 그룹은 다음과 같은 역할을 합니다.

- DB 보안 그룹은 VPC에 속하지 않는 EC2-Classic DB 인스턴스에 대한 액세스를 제어합니다.
- VPC 보안 그룹은 VPC에 속하는 DB 인스턴스 및 EC2 인스턴스에 대한 액세스를 제어합니다.
- EC2 보안 그룹은 EC2 인스턴스에 대한 액세스를 제어합니다.

기본적으로 DB 인스턴스에 대한 네트워크 액세스는 해제되어 있습니다. IP 주소 범위, 포트 또는 EC2 보안 그룹에서 액세스를 허용하는 보안 그룹의 규칙을 지정할 수 있습니다. 수신 규칙이 설정되면 동일한 규칙이 해당 보안 그룹과 연결된 모든 DB 인스턴스에 적용됩니다. 보안 그룹에서 최대 20개의 규칙을 지정할 수 있습니다.

DB 보안 그룹

DB 보안 그룹은 VPC에 있지 않고 EC2-Classic 플랫폼에 있는 DB 인스턴스에서 사용됩니다. 각 DB 보안 그룹 규칙을 설정하면 특정 소스가 해당 DB 보안 그룹과 연결되어 있는 DB 인스턴스에 액세스할 수 있습니다. 소스는 주소 범위(예: 203.0.113.0/24) 또는 EC2 보안 그룹일 수 있습니다. EC2 보안 그룹을 소스로 지정하면 해당 EC2 보안 그룹을 사용하는 모든 EC2 인스턴스에서 수신 트래픽이 허용됩니다. DB 보안 그룹 규칙은 인바운드 트래픽에만 적용되며, 아웃바운드 트래픽은 현재 DB 인스턴스에서 허용되지 않습니다.

DB 보안 그룹 규칙을 생성할 때는 대상 포트 번호를 지정하지 않아도 됩니다. DB 보안 그룹에 규칙을 정의 할 때는 항상 DB 인스턴스로 정의되는 포트 번호가 대상 포트 번호로 사용되기 때문입니다. DB 보안 그룹은 Amazon RDS API 작업 또는 AWS Management 콘솔의 Amazon RDS 페이지를 사용하여 생성할 수 있습니다.

DB 보안 그룹으로 작업하는 방법에 대한 자세한 내용은 [DB 보안 그룹 작업\(EC2-Classic 플랫폼\) \(p. 394\)](#) 단원을 참조하십시오.

VPC 보안 그룹

각 VPC 보안 그룹 규칙을 설정하면 특정 소스가 해당 VPC 보안 그룹과 연결되어 있는 VPC의 DB 인스턴스에 액세스할 수 있습니다. 소스는 주소 범위(예: 203.0.113.0/24) 또는 다른 VPC 보안 그룹일 수 있습니다. VPC 보안 그룹을 소스로 지정하면 소스 VPC 보안 그룹을 사용하는 모든 인스턴스(일반적으로 애플리케이션 서버)에서 수신 트래픽이 허용됩니다. VPC 보안 그룹에 인바운드와 아웃바운드 트래픽을 모두 제어하는 규칙이 있을 수 있지만, 아웃바운드 트래픽 규칙은 일반적으로 DB 인스턴스에 적용되지 않습니다. 아웃바운드

드 트래픽 규칙은 DB 인스턴스가 클라이언트로 작동하는 경우에만 적용됩니다. 예를 들면 아웃바운드 트래픽 규칙은 아웃바운드 데이터베이스 링크가 있는 Oracle DB 인스턴스에 적용됩니다. [Amazon EC2 API](#)를 사용하거나, 혹은 VPC 콘솔에서 보안 그룹 옵션을 선택하여 VPC 보안 그룹을 생성해야 합니다.

VPC의 인스턴스에 대한 액세스를 허용하는 VPC 보안 그룹과 관련된 규칙을 생성할 때 그 규칙이 액세스를 허용하는 주소들의 각 범위에 대해 포트를 지정해야 합니다. 예를 들어, VPC의 인스턴스에 대한 SSH 액세스를 활성화하고 싶다면 지정된 주소 범위와 관련해 TCP 포트 22에 대한 액세스를 허용하는 규칙을 생성해야 합니다.

VPC의 서로 다른 인스턴스에게 서로 다른 포트에 대한 액세스를 허용하는 여러 개의 VPC 보안 그룹을 구성할 수 있습니다. 예를 들어 VPC의 웹 서버에 대해서는 TCP 포트 80으로 액세스가 가능하도록 VPC 보안 그룹을 생성합니다. 그런 다음 VPC의 RDS MySQL DB 인스턴스에 대해서는 TCP 포트 3306으로 액세스할 수 있도록 다른 VPC 보안 그룹을 생성하면 됩니다.

VPC 보안 그룹에 관한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 [보안 그룹](#)을 참조하십시오.

DB 보안 그룹 및 VPC 보안 그룹

다음 표는 DB 보안 그룹과 VPC 보안 그룹의 주된 차이를 보여 줍니다.

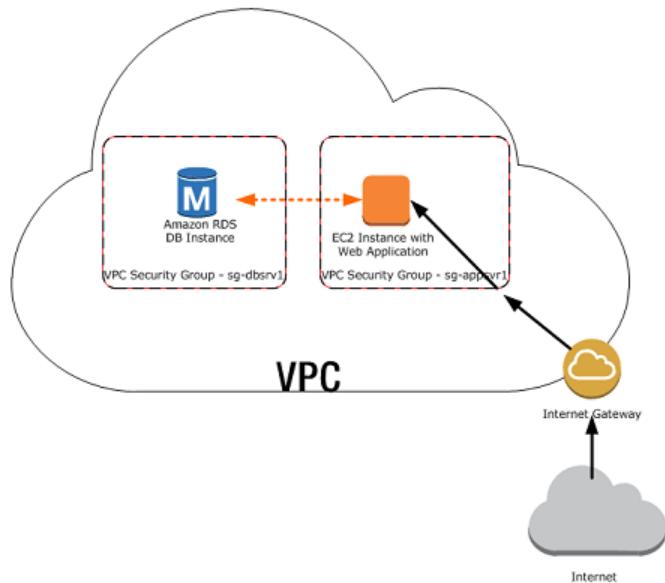
DB 보안 그룹	[VPC Security Group]
VPC 외부의 DB 인스턴스에 대한 액세스를 제어합니다.	VPC에 있는 DB 인스턴스에 대한 액세스를 제어합니다.
Amazon RDS API 작업 또는 AWS Management 콘솔의 Amazon RDS 페이지를 사용하여 그룹과 규칙을 생성하고 관리합니다.	Amazon EC2 API 작업 또는 AWS Management 콘솔의 Amazon VPC 페이지를 사용하여 그룹과 규칙을 생성하고 관리합니다.
그룹에 규칙을 추가하면 포트 번호나 프로토콜을 지정하지 않아도 됩니다.	규칙을 그룹에 추가할 때 프로토콜을 TCP로 지정합니다. 또한 그룹 구성원으로 추가할 목적으로 DB 인스턴스(또는 옵션)를 생성할 때 사용했던 것과 동일하게 포트 번호를 지정합니다.
그룹이 AWS 계정 또는 기타 계정의 EC2 보안 그룹에서 액세스를 허용합니다.	그룹이 VPC에 있는 다른 VPC 보안 그룹에서만 액세스를 허용합니다.

보안 그룹 시나리오

VPC에서 RDS 인스턴스를 사용하는 일반적인 사례는 동일한 VPC의 Amazon EC2 인스턴스에서 실행 중이며 VPC 외부의 클라이언트 애플리케이션에서 액세스한 애플리케이션을 사용하여 데이터를 공유하는 것입니다. 이러한 시나리오에서는 RDS 및 VPC 페이지를 사용하거나, 혹은 RDS 및 EC2 API 작업을 사용하여 필요한 인스턴스와 보안 그룹을 생성합니다.

1. VPC 보안 그룹(예: sg-appsrv1)을 생성하고 클라이언트 애플리케이션의 IP 주소를 소스로 사용하는 인바운드 규칙을 생성합니다. 이 보안 그룹에서는 클라이언트 애플리케이션이 이 보안 그룹을 사용하는 VPC의 EC2 인스턴스에 연결할 수 있습니다.
2. 애플리케이션에 대한 EC2 인스턴스를 생성하고 이전 단계에서 생성한 VPC 보안 그룹(sg-appsrv1)에 EC2 인스턴스를 추가합니다. VPC의 EC2 인스턴스는 DB 인스턴스와 VPC 보안 그룹을 공유합니다.
3. 두 번째 VPC 보안 그룹(예: sg-dbsrv1)을 생성하고 1단계에서 만든 VPC 보안 그룹(sg-appsrv1)을 스스로 지정해 새 규칙을 생성합니다.
4. 새 DB 인스턴스를 생성하고 이전 단계에서 생성한 VPC 보안 그룹(sg-dbsrv1)에 DB 인스턴스를 추가합니다. 인스턴스를 생성할 때 3단계에서 생성한 VPC 보안 그룹(sg-dbsrv1) 규칙에 대해 지정한 것과 동일한 보안 포트 번호를 사용하십시오.

다음 다이어그램은 이 시나리오를 보여 줍니다.



VPC 사용에 대한 자세한 내용은 [Amazon Virtual Private Cloud\(VPC\)](#) 및 [Amazon RDS \(p. 406\)](#) 단원을 참조하십시오.

VPC 보안 그룹 생성

VPC 콘솔을 사용하여 DB 인스턴스에 대한 VPC 보안 그룹을 생성할 수 있습니다. 보안 그룹 생성에 대한 자세한 내용은 [보안 그룹을 생성하여 VPC 내부의 DB 인스턴스에 대한 액세스를 제공 \(p. 8\)](#) 및 Amazon Virtual Private Cloud 사용 설명서의 [보안 그룹을 참조하십시오.](#)

보안 그룹과 DB 인스턴스의 연결

보안 그룹은 RDS 콘솔의 수정 옵션을 사용하거나, `ModifyDBInstance` Amazon RDS API 또는 `modify-db-instance` AWS CLI 명령을 사용해 DB 인스턴스와 연결할 수 있습니다.

DB 인스턴스 수정에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#) 단원을 참조하십시오. DB 인스턴스를 DB 스냅샷에서 복원할 때 보안 그룹에서 고려해야 할 사항은 [보안 그룹 고려 사항 \(p. 205\)](#) 단원을 참조하십시오.

DB VPC 보안 그룹 삭제

DB VPC 보안 그룹은 보안 정보를 VPC 보안 그룹과 동기화하는 RDS 메커니즘입니다. 그러나 이제 RDS가 VPC 보안 그룹 정보를 직접 사용하도록 업데이트되었으므로 이 동기화는 더 이상 필요하지 않습니다.

Note

DB VPC 보안 그룹은 사용되지 않으며 DB 보안 그룹, VPC 보안 그룹 및 EC2 보안 그룹과 다릅니다.

현재 사용 중인 DB VPC 보안 그룹을 삭제하는 것이 좋습니다. DB VPC 보안 그룹을 삭제하지 않을 경우, RDS DB 인스턴스와 관련된 예상치 못한 동작으로 인해 DB 인스턴스 액세스가 끊어지는 심각한 상황까지 발생할 수 있습니다. DB 인스턴스, 옵션 그룹 등으로 업데이트하는 작업은 예상치 못한 동작으로 이어집니다. 이러한 업데이트는 RDS에서 DB VPC 보안 그룹이 VPC 보안 그룹과 다시 동기화되는 결과를 초래합니다. 이러한 재동기화로 인해 보안 정보를 오랜 시간이 지나 잘못된 보안 정보로 덮어쓰게 됩니다. 그 결과 RDS DB 인스턴스 액세스에 심각한 영향을 줄 수 있습니다.

DB VPC 보안 그룹이 있는지 확인하는 방법은 무엇입니까?

DB VPC 보안 그룹은 더 이상 사용되지 않기 때문에 RDS 콘솔에 표시되지 않습니다. 그러나 [describe-db-security-groups](#) AWS CLI 명령 또는 [DescribeDBSecurityGroups](#) API 작업을 호출하여 DB VPC 보안 그룹이 있는지를 확인할 수 있습니다.

이러한 경우에는 출력 형식으로 지정된 JSON과 함께 [describe-db-security-groups](#) AWS CLI 명령을 호출할 수 있습니다. 그러면 다음 예제에 표시된 것처럼 보안 그룹에 대한 두 번째 출력 줄에서 VPC 식별자로 DB VPC 보안 그룹을 식별할 수 있습니다.

```
{  
    "DBSecurityGroups": [  
        {  
            "VpcId": "vpc-abcd1234",  
            "DBSecurityGroupDescription": "default:vpc-abcd1234",  
            "IPRanges": [  
                {  
                    "Status": "authorized",  
                    "CIDRIP": "xxx.xxx.xxx.xxxx/n"  
                },  
                {  
                    "Status": "authorized",  
                    "CIDRIP": "xxx.xxx.xxx.xxxx/n "  
                }  
            ],  
            "OwnerId": "123456789012",  
            "EC2SecurityGroups": [],  
            "DBSecurityGroupName": "default:vpc-abcd1234"  
        }  
    ]  
}
```

[DescribeDBSecurityGroups](#) API 작업을 실행할 경우 다음 예제와 같이 <VpcId> 응답 요소를 사용하여 DB VPC 보안 그룹을 식별할 수 있습니다.

```
<DBSecurityGroup>  
    <EC2SecurityGroups/>  
    <DBSecurityGroupDescription>default:vpc-abcd1234</DBSecurityGroupDescription>  
    <IPRanges>  
        <IPRange>  

```

DB VPC 보안 그룹을 어떻게 삭제합니까?

DB VPC 보안 그룹은 RDS 콘솔에 표시되지 않으므로 [delete-db-security-group](#) AWS CLI 명령 또는 [DeleteDBSecurityGroup](#) API 작업을 호출하여 DB VPC 보안 그룹을 삭제해야 합니다.

DB VPC 보안 그룹을 삭제한 후에도 VPC의 DB 인스턴스는 해당 VPC의 VPC 보안 그룹에 의해 계속해서 보호됩니다. 삭제된 DB VPC 보안 그룹은 VPC 보안 그룹 정보의 복사본에 불과합니다.

AWS CloudFormation 템플릿 검토

AWS CloudFormation 템플릿의 이전 버전에 DB VPC 보안 그룹을 만들라는 지침이 포함되어 있을 수 있습니다. DB VPC 보안 그룹은 아직 완전히 폐기되지 않았으므로 계속해서 만들어질 수 있습니다. 또한 DB 인스턴스에 보안 설정을 프로비저닝하는 데 사용하는 템플릿에서도 DB VPC 보안 그룹을 만들지 않도록 하십시오. 다음 예제에 표시된 것처럼 EC2VpcId를 가진 RDS DBSecurityGroup을 만드는 AWS CloudFormation 템플릿을 사용하지 마십시오.

```
{  
    "DbSecurityByEC2SecurityGroup" : {  
        "Type" : "AWS::RDS::DBSecurityGroup",  
        "Properties" : {  
            "GroupDescription" : "Ingress for Amazon EC2 security group",  
            "EC2VpcId" : "MyVPC",  
            "DBSecurityGroupIngress" : [ {  
                "EC2SecurityGroupId" : "sg-b0ff1111",  
                "EC2SecurityGroupOwnerId" : "111122223333"  
            }, {  
                "EC2SecurityGroupId" : "sg-ffd722222",  
                "EC2SecurityGroupOwnerId" : "111122223333"  
            } ]  
        }  
    }  
}
```

대신 다음 예제와 같이 VPC 보안 그룹을 사용하여 VPC의 RDS DB 인스턴스에 대한 보안 정보를 추가하십시오.

```
{  
    "DBInstance" : {  
        "Type": "AWS::RDS::DBInstance",  
        "Properties": {  
            "DBName" : { "Ref" : "DBName" },  
            "Engine" : "MySQL",  
            "MultiAZ" : { "Ref": "MultiAZDatabase" },  
            "MasterUsername" : { "Ref" : "<master_username>" },  
            "DBInstanceClass" : { "Ref" : "DBCClass" },  
            "AllocatedStorage" : { "Ref" : "DBAllocatedStorage" },  
            "MasterUserPassword": { "Ref" : "<master_password>" },  
            "VPCSecurityGroups" : [ { "Fn::GetAtt": [ "VPCSecurityGroup", "GroupId" ] } ]  
        }  
    }  
}
```

DB 보안 그룹 작업(EC2-Classic 플랫폼)

기본적으로 DB 인스턴스에 대한 네트워크 액세스는 해제되어 있습니다. IP 주소 범위, 포트 또는 EC2 보안 그룹에서 액세스를 허용하는 보안 그룹의 규칙을 지정할 수 있습니다. 수신 규칙이 설정되면 동일한 규칙이 해당 보안 그룹과 연결된 모든 DB 인스턴스에 적용됩니다. 보안 그룹에서 최대 20개의 규칙을 지정할 수 있습니다.

Amazon RDS는 서로 다른 두 가지 유형의 보안 그룹을 지원합니다. 사용하는 보안 그룹은 어떤 플랫폼에 있는지에 따라 결정됩니다.

- VPC 보안 그룹 – EC2-VPC 플랫폼일 때
- DB 보안 그룹 – EC2-Classic 플랫폼일 때

다음 중 하나의 경우에 해당된다면 EC2-VPC 플랫폼일 가능성이 높기 때문에 VPC 보안 그룹을 사용해야 합니다.

- 신규 Amazon RDS 고객인 경우
- 이전에 DB 인스턴스를 한 번도 생성하지 않은 경우
- 이전에 사용하지 않았던 AWS 리전에 DB 인스턴스를 생성하는 경우

그 외에 EC2-Classic 플랫폼인 경우에는 DB 보안 그룹을 사용하여 Amazon RDS DB 인스턴스에 대한 액세스를 관리해야 합니다. DB 보안 그룹과 VPC 보안 그룹의 차이에 대한 자세한 정보는 [보안 그룹을 통한 액세스 제어 \(p. 390\)](#) 단원을 참조하십시오.

Note

플랫폼 결정에 대한 자세한 정보는 [EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 \(p. 407\)](#) 단원을 참조하십시오.

EC2-VPC 플랫폼일 때는 DB 보안 그룹이 아닌 VPC 보안 그룹을 사용해야 합니다. VPC 사용에 대한 자세한 정보는 [Amazon Virtual Private Cloud\(VPC\) 및 Amazon RDS \(p. 406\)](#) 단원을 참조하십시오.

주제

- [DB 보안 그룹 생성 \(p. 395\)](#)
- [사용 가능한 DB 보안 그룹 나열 \(p. 396\)](#)
- [DB 보안 그룹 보기 \(p. 397\)](#)
- [DB 보안 그룹과 DB 인스턴스의 연결 \(p. 398\)](#)
- [특정 IP 범위에서 DB 보안 그룹에 대한 네트워크 액세스 승인 \(p. 398\)](#)
- [Amazon EC2 인스턴스에서 DB 인스턴스에 대한 네트워크 액세스 승인 \(p. 399\)](#)
- [IP 범위에서 DB 인스턴스에 대한 네트워크 액세스 취소 \(p. 401\)](#)

DB 보안 그룹 생성

DB 보안 그룹을 생성하려면 이름과 설명을 제공해야 합니다.

AWS Management 콘솔

DB 보안 그룹을 생성하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 화면 왼쪽의 탐색 창에서 [Security Groups]를 선택합니다.

Note

EC2-VPC 플랫폼에 있는 경우 [Security Groups] 옵션이 탐색 창에 표시되지 않습니다. 이 경우 DB 보안 그룹이 아닌 VPC 보안 그룹을 사용해야 합니다. VPC 사용에 대한 자세한 정보는 [Amazon Virtual Private Cloud\(VPC\) 및 Amazon RDS \(p. 406\)](#) 단원을 참조하십시오.

3. [Create DB Security Group]를 선택합니다.
4. Name 및 Description 텍스트 상자에 새 DB 보안 그룹의 이름과 설명을 입력합니다. 보안 그룹 이름은 공백을 포함할 수 없으며 숫자로 시작할 수 없습니다.
5. [Yes, Create]를 선택합니다.

DB 보안 그룹이 생성됩니다.

새로 만든 DB 보안 그룹은 기본적으로 DB 인스턴스에 대한 액세스를 제공하지 않습니다. IP 주소 범위나 DB 인스턴스에 액세스할 수 있는 Amazon EC2 보안 그룹을 지정해야 합니다. DB 보안 그룹에 대한 Amazon EC2 보안 그룹이나 IP 주소를 지정하려면 [특정 IP 범위에서 DB 보안 그룹에 대한 네트워크 액세스 승인 \(p. 398\)](#) 주제를 참조하십시오.

CLI

DB 보안 그룹을 생성하려면 AWS CLI 명령 `create-db-security-group`을 사용하십시오.

Example

Linux, OS X, Unix의 경우:

```
aws rds create-db-security-group \
--db-security-group-name mydbsecuritygroup \
--db-security-group-description "My new security group"
```

Windows의 경우:

```
aws rds create-db-security-group ^
--db-security-group-name mydbsecuritygroup ^
--db-security-group-description "My new security group"
```

새로 만든 DB 보안 그룹은 기본적으로 DB 인스턴스에 대한 액세스를 제공하지 않습니다. IP 주소 범위나 DB 인스턴스에 액세스할 수 있는 Amazon EC2 보안 그룹을 지정해야 합니다. DB 보안 그룹에 대한 Amazon EC2 보안 그룹이나 IP 주소를 지정하려면 [특정 IP 범위에서 DB 보안 그룹에 대한 네트워크 액세스 승인 \(p. 398\)](#) 주제를 참조하십시오.

API

DB 보안 그룹을 만들려면 Amazon RDS 함수인 `CreateDBSecurityGroup`을 다음 파라미터와 함께 호출합니다.

- `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=<AWS Access Key ID>
&Signature=<Signature>
```

새로 만든 DB 보안 그룹은 기본적으로 DB 인스턴스에 대한 액세스를 제공하지 않습니다. IP 주소 범위나 DB 인스턴스에 액세스할 수 있는 Amazon EC2 보안 그룹을 지정해야 합니다. DB 보안 그룹에 대한 Amazon EC2 보안 그룹이나 IP 주소를 지정하려면 [특정 IP 범위에서 DB 보안 그룹에 대한 네트워크 액세스 승인 \(p. 398\)](#) 주제를 참조하십시오.

사용 가능한 DB 보안 그룹 나열

AWS 계정에 대해 생성된 DB 보안 그룹을 나열할 수 있습니다.

AWS Management 콘솔

AWS 계정에 사용할 수 있는 모든 DB 보안 그룹을 나열하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.

- 화면 왼쪽의 탐색 창에서 [Security Groups]를 선택합니다.

사용 가능한 DB 보안 그룹이 [DB Security Groups] 목록에 나타납니다.

CLI

AWS 계정에 사용 가능한 모든 DB 보안 그룹을 나열하려면 AWS CLI 명령 `describe-db-security-groups`를 파라미터 없이 사용하십시오.

Example

```
aws rds describe-db-security-groups
```

API

AWS 계정에 사용할 수 있는 모든 DB 보안 그룹을 나열하려면 `DescribeDBSecurityGroups`를 파라미터 없이 호출합니다.

Example

```
https://rds.amazonaws.com/  
    ?Action=DescribeDBSecurityGroups  
    &MaxRecords=100  
    &Version=2009-10-16  
    &SignatureVersion=2  
    &SignatureMethod=HmacSHA256  
    &AWSAccessKeyId=<AWS Access Key ID>  
    &Signature=<Signature>
```

DB 보안 그룹 보기

DB 보안 그룹에 대한 자세한 정보를 보고 승인되지 않은 IP 범위를 확인할 수 있습니다.

AWS Management 콘솔

특정 DB 보안 그룹의 속성을 보려면

- <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
- 화면 왼쪽의 탐색 창에서 [Security Groups]를 선택합니다.
- 보려는 DB 보안 그룹에 대한 세부 정보 아이콘을 선택합니다. DB 보안 그룹에 대한 세부 정보가 표시됩니다.

CLI

특정 DB 보안 그룹의 속성을 보려면 AWS CLI `describe-db-security-groups`를 사용하십시오. 보려는 DB 보안 그룹을 지정합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds describe-db-security-groups \  
    --db-security-group-name mydbsecuritygroup
```

Windows의 경우:

```
aws rds describe-db-security-groups ^
--db-security-group-name mydbsecuritygroup
```

API

특정 DB 보안 그룹의 속성을 보려면 [DescribeDBSecurityGroups](#)를 다음 파라미터와 함께 호출합니다.

- 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=<AWS Access Key ID>
&Signature=<Signature>
```

DB 보안 그룹과 DB 인스턴스의 연결

DB 보안 그룹은 RDS 콘솔의 Modify(수정) 옵션을 사용하거나, [ModifyDBInstance](#) Amazon RDS API 또는 AWS CLI `modify-db-instance` 명령을 사용하여 DB 인스턴스와 연결할 수 있습니다.

DB 인스턴스 수정에 대한 자세한 정보는 [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#) 단원을 참조하십시오.

특정 IP 범위에서 DB 보안 그룹에 대한 네트워크 액세스 승인

기본적으로 DB 인스턴스에 대한 네트워크 액세스는 해제되어 있습니다. VPC 내부에 없는 DB 인스턴스에 액세스해야 하는 경우 특정 EC2 보안 그룹이나 CIDR IP 범위에서 액세스할 수 있게 DB 보안 그룹에 대한 액세스 규칙을 설정해야 합니다. 그런 다음 DB 인스턴스를 해당 DB 보안 그룹과 연결해야 합니다. 이 프로세스를 수신이라고 합니다. DB 보안 그룹에 대해 수신이 구성되면 동일한 수신 규칙이 해당 DB 보안 그룹과 연결된 모든 DB 인스턴스에 적용됩니다.

Warning

방화벽 뒤의 DB 인스턴스에 연결하려는 경우 네트워크 관리자와 논의하여 사용해야 하는 IP 주소를 확인합니다.

다음 예에서는 CIDR IP 범위에 대한 수신 규칙으로 DB 보안 그룹을 구성합니다.

AWS Management 콘솔

CIDR IP 범위에 대한 수신 규칙으로 DB 보안 그룹을 구성하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 콘솔 왼쪽의 탐색 창에서 [Security Groups]를 선택합니다.
3. 승인하려는 DB 보안 그룹에 대한 세부 정보 아이콘을 선택합니다.
4. 보안 그룹에 대한 세부 정보 페이지의 [Connection Type] 드롭다운 목록에서 [CIDR/IP]를 선택하고, 이 DB 보안 그룹에 추가하려는 수신 규칙에 대한 CIDR 범위를 [CIDR] 텍스트 상자에 입력한 후 [Authorize]를 선택합니다.

Tip

AWS Management 콘솔에 CIDR 텍스트 필드 아래의 연결을 바탕으로 CIDR IP가 표시됩니다.
방화벽 뒤에서 DB 인스턴스에 액세스하지 않을 경우 이 CIDR IP를 사용할 수 있습니다.

- 수신 규칙의 상태는 사용자가 수정한 DB 보안 그룹과 연결된 모든 DB 인스턴스에 새 수신 규칙이 적용될 때까지 [authorizing]이 됩니다. 수신 규칙을 성공적으로 적용한 후에는 상태가 [authorized]로 변경됩니다.

CLI

CIDR IP 범위에 대한 수신 규칙으로 DB 보안 그룹을 구성하려면 AWS CLI 명령 [authorize-db-security-group-ingress](#)를 사용하십시오.

Example

Linux, OS X, Unix의 경우:

```
aws rds authorize-db-security-group-ingress \
--db-security-group-name mydbsecuritygroup \
--cidrip 192.168.1.10/27
```

Windows의 경우:

```
aws rds authorize-db-security-group-ingress ^
--db-security-group-name mydbsecuritygroup ^
--cidrip 192.168.1.10/27
```

다음과 비슷한 출력이 생성되어야 합니다.

```
SECURITYGROUP mydbsecuritygroup My new DBSecurityGroup
IP-RANGE 192.168.1.10/27 authorizing
```

API

CIDR IP 범위에 대한 수신 규칙으로 DB 보안 그룹을 구성하려면 Amazon RDS API [AuthorizeDBSecurityGroupIngress](#)를 다음 파라미터와 함께 호출합니다.

- 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=<AWS Access Key ID>
&Signature=<Signature>
```

Amazon EC2 인스턴스에서 DB 인스턴스에 대한 네트워크 액세스 승인

Amazon EC2 인스턴스에서 DB 인스턴스에 액세스하려는 경우 먼저 EC2 인스턴스와 DB 인스턴스가 VPC에 있는지 확인해야 합니다. 기본 VPC를 사용하는 경우 EC2 인스턴스에서 액세스할 DB 인스턴스를 생성하고 수정할 때 EC2 인스턴스에 사용한 동일한 EC2 및 VPC 보안 그룹을 배정할 수 있습니다.

DB 인스턴스와 EC2 인스턴스가 VPC에 없는 경우 Amazon EC2 인스턴스의 트래픽을 허용하는 수신 규칙으로 DB 인스턴스의 보안 그룹을 구성해야 합니다. DB 인스턴스에 대한 DB 보안 그룹에 EC2 인스턴스에 대한 보안 그룹을 추가하면 됩니다. 이 예에서는 DB 보안 그룹에 Amazon EC2 보안 그룹에 대한 수신 규칙을 추가합니다.

Important

- DB 보안 그룹에 Amazon EC2 보안 그룹에 대한 수신 규칙을 추가하면 해당 Amazon EC2 보안 그룹과 연결된 Amazon EC2 인스턴스만 DB 인스턴스에 액세스할 수 있습니다.
- 사용 중인 DB 인스턴스와 다른 AWS 리전에 있는 Amazon EC2 보안 그룹은 승인할 수 없습니다. IP 범위를 승인하거나, 다른 AWS 리전의 IP 주소를 참조하는 같은 AWS 리전의 보안 그룹을 지정할 수 있습니다. IP 범위를 지정할 경우 Amazon EC2 인스턴스의 프라이빗 IP 주소를 사용하는 것이 좋습니다. 그러면 Amazon EC2 인스턴스에서 Amazon RDS DB 인스턴스로 더 직접적인 네트워크 라우팅이 제공되며 Amazon 네트워크 외부로 전송되는 데이터에 대한 네트워크 요금이 발생하지 않습니다.

AWS Management 콘솔

DB 보안 그룹에 EC2 보안 그룹에 추가하려면

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 액세스를 허용하려는 DB 보안 그룹에 대한 세부 정보 아이콘을 선택합니다.
4. 보안 그룹에 대한 세부 정보 페이지의 Connection Type(연결 유형)에서 EC2 Security Group(EC2 보안 그룹)을 선택하고, 사용하려는 Amazon EC2 보안 그룹을 선택합니다. 그런 다음 [Authorize]를 선택합니다.
5. 수신 규칙의 상태는 사용자가 수정한 DB 보안 그룹과 연결된 모든 DB 인스턴스에 새 수신 규칙이 적용될 때까지 [authorizing]이 됩니다. 수신 규칙을 성공적으로 적용한 후에는 상태가 [authorized]로 변경됩니다.

CLI

Amazon EC2 보안 그룹에 대한 액세스를 허용하려면 AWS CLI 명령 [authorize-db-security-group-ingress](#)를 사용합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds authorize-db-security-group-ingress \
--db-security-group-name default \
--ec2-security-group-name myec2group \
--ec2-security-group-owner-id 987654321021
```

Windows의 경우:

```
aws rds authorize-db-security-group-ingress ^
--db-security-group-name default ^
--ec2-security-group-name myec2group ^
--ec2-security-group-owner-id 987654321021
```

다음과 비슷한 출력이 생성되어야 합니다.

SECGROUP	Name	Description
SECGROUP	default	default
EC2-SECGROUP	myec2group	987654321021 authorizing

API

Amazon EC2 보안 그룹에 대한 네트워크 액세스를 승인하려면 해당 Amazon RDS API 함수인 https://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_AuthorizeDBSecurityGroupIngress.html를 다음 파라미터와 함께 호출합니다.

- 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=<AWS Access Key ID>
&Signature=<Signature>
```

IP 범위에서 DB 인스턴스에 대한 네트워크 액세스 취소

연결된 CIDR IP 수신 규칙을 취소하여 CIDR IP 범위에서 DB 보안 그룹에 속하는 DB 인스턴스에 대한 네트워크 액세스를 쉽게 취소할 수 있습니다.

이 예제에서는 DB 보안 그룹에서 CIDR IP 범위에 대한 수신 규칙을 취소합니다.

AWS Management 콘솔

DB 보안 그룹에서 CIDR IP 범위에 대한 수신 규칙을 취소하는 방법

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 취소하려는 수신 규칙이 있는 DB 보안 그룹에 대한 세부 정보 아이콘을 선택합니다.
4. 보안 그룹에 대한 세부 정보 페이지에서 취소하려는 수신 규칙 옆의 [Remove]를 선택합니다.
5. 수정한 DB 보안 그룹과 연결된 모든 DB 인스턴스에서 수신 규칙이 제거될 때까지 수신 규칙의 상태는 [revoking]입니다. 수신 규칙 제거에 성공하면 DB 보안 그룹에서도 해당 수신 규칙이 제거됩니다.

CLI

DB 보안 그룹에서 CIDR IP 범위에 대한 수신 규칙을 취소하려면 AWS CLI 명령 [revoke-db-security-group-ingress](#)를 사용합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds revoke-db-security-group-ingress \
--db-security-group-name mydbsecuritygroup \
--cidrip 192.168.1.1/27
```

Windows의 경우:

```
aws rds revoke-db-security-group-ingress ^
--db-security-group-name mydbsecuritygroup ^
--cidrip 192.168.1.1/27
```

다음과 비슷한 출력이 생성되어야 합니다.

```
SECGROUP mydbsecuritygroup My new DBSecurityGroup
IP-RANGE 192.168.1.1/27 revoking
```

API

DB 보안 그룹에서 CIDR IP 범위에 대한 수신 규칙을 취소하려면 Amazon RDS API 작업 https://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_RevokeDBSecurityGroupIngress.html를 다음 파라미터와 함께 호출합니다.

- 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=<AWS Access Key ID>
&Signature=<Signature>
```

마스터 사용자 계정 권한

새로운 DB 인스턴스를 생성할 때 사용되는 기본 마스터 사용자는 해당 DB 인스턴스에 대한 특정 권한을 갖습니다. 다음 표에는 마스터 사용자가 각 데이터베이스 엔진에 대해 갖는 권한이 나와 있습니다.

Important

애플리케이션에서 직접 마스터 사용자를 사용하지 않는 것이 좋습니다. 대신에 애플리케이션에 필요한 최소 권한으로 생성한 데이터베이스 사용자를 사용하는 모범 사례를 준수하십시오.

Note

마스터 사용자의 권한을 실수로 삭제한 경우, DB 인스턴스를 수정하고 새 마스터 사용자 암호를 설정하여 복원할 수 있습니다. DB 인스턴스 수정에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#) 단원을 참조하십시오.

데이터베이스 엔진	시스템 권한	역할
MySQL 및 MariaDB	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES, INDEX, ALTER, SHOW DATABASES, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION	—

데이터베이스 엔진	시스템 권한	역할
	CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER ON *.* WITH GRANT OPTION, REPLICATION SLAVE(Amazon RDS MySQL 버전 5.6, 5.7 및 8.0, Amazon RDS MariaDB에만 적용)	
PostgreSQL	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	RDS_SUPERUSER
Oracle	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	AQ_ADMINISTRATOR_ROLE, AQ_USER_ROLE, CONNECT, CTXAPP, DBA, EXECUTE_CATALOG_ROLE, RECOVERY_CATALOG_OWNER, RESOURCE, SELECT_CATALOG_ROLE
Microsoft SQL Server	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(데이터베이스 수준 역할) PROCESSADMIN(서버 수준 역할) SETUPADMIN(서버 수준 역할) SQLAgentUserRole(서버 수준 역할)

Amazon RDS에 서비스 연결 역할 사용

Amazon RDS의 경우 AWS Identity and Access Management(IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 Amazon RDS에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 Amazon RDS에서 사전 정의하며 서비스에서 다른 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 포함합니다.

서비스 연결 역할을 통해 Amazon RDS 사용이 쉬워지는데 필요한 권한을 수동으로 추가할 필요가 없기 때문입니다. Amazon RDS에서 서비스 연결 역할 권한을 정의하므로, 달리 정의되지 않은 한 Amazon RDS에서만 해당 역할을 맡을 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며, 이 권한 정책은 다른 IAM 개체에 연결할 수 없습니다.

먼저 역할의 관련 리소스를 삭제해야만 역할을 삭제할 수 있습니다. 이렇게 하면 리소스에 대한 액세스 권한을 부주의로 삭제할 수 없기 때문에 Amazon RDS 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 기타 서비스에 대한 자세한 정보는 [IAM으로 작업하는 AWS 서비스](#)를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾으십시오. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

Amazon RDS에 대한 서비스 연결 역할 권한

Amazon RDS에서는 AWSServiceRoleForRDS – to allow Amazon RDS to call AWS services on behalf of your DB instances 서비스 연결 역할을 사용합니다.

AWSServiceRoleForRDS 서비스 연결 역할은 역할을 위임하기 위해 다음 서비스를 신뢰합니다.

- rds.amazonaws.com

역할 권한 정책은 Amazon RDS가 지정된 리소스에서 다음 작업을 완료하도록 허용합니다.

- ec2에 대한 작업:
 - AssignPrivateIpAddresses
 - AuthorizeSecurityGroupIngress
 - CreateNetworkInterface
 - CreateSecurityGroup
 - DeleteNetworkInterface
 - DeleteSecurityGroup
 - DescribeAvailabilityZones
 - DescribeInternetGateways
 - DescribeSecurityGroups
 - DescribeSubnets
 - DescribeVpcAttribute
 - DescribeVpcs
 - ModifyNetworkInterfaceAttribute
 - RevokeSecurityGroupIngress
 - UnassignPrivateIpAddresses
- sns에 대한 작업:
 - ListTopic
 - Publish
- cloudwatch에 대한 작업:
 - PutMetricData
 - GetMetricData
 - CreateLogStream
 - PullLogEvents
 - DescribeLogStreams
 - CreateLogGroup

Note

IAM 개체(사용자, 그룹, 역할 등)가 서비스 연결 역할을 작성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 다음 오류 메시지가 표시되는 경우:

리소스를 만들 수 없습니다. 서비스 연결 역할을 생성할 권한이 있는지 확인하십시오. 그렇지 않은 경우 기다렸다가 나중에 다시 시도하십시오.

다음 권한이 활성화되어 있는지 확인하십시오.

```
{  
    "Action": "iam:CreateServiceLinkedRole",  
    "Effect": "Allow",  
}
```

API 버전 2014-10-31

```
"Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/AWSServiceRoleForRDS",
"Condition": {
    "StringLike": {
        "iam:AWSServiceName": "rds.amazonaws.com"
    }
}
```

자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 권한](#) 단원을 참조하십시오.

Amazon RDS에 대한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. create a DB instance 시 Amazon RDS에서 서비스 연결 역할을 생성합니다.

Important

Amazon RDS 서비스가 서비스 연결 역할을 지원하기 시작한 December 1, 2017 이전에 이 서비스를 사용 중이었다면 Amazon RDS에서 사용자 계정에 AWSServiceRoleForRDS 역할을 이미 생성했습니다. 자세한 내용은 [내 IAM 계정에 표시되는 새 역할](#)을 참조하십시오.

이 서비스 연결 역할을 삭제한 다음 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. create a DB instance 시 Amazon RDS에서 서비스 연결 역할을 다시 생성합니다.

Amazon RDS에 대한 서비스 연결 역할 편집

Amazon RDS에서는 AWSServiceRoleForRDS 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 연결 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 그러나 IAM를 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [서비스에 연결 역할 편집](#)을 참조하십시오.

Amazon RDS에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제할 것을 권합니다. 따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 개체가 없도록 합니다. 그러나 서비스 연결 역할을 삭제하려면 먼저 모든 DB 인스턴스를 삭제해야 합니다.

서비스 연결 역할 정리

IAM을 사용하여 서비스 연결 역할을 삭제하기 전에 먼저 역할에 활성 세션이 있는지 확인하고 역할에서 사용되는 리소스를 모두 제거해야 합니다.

IAM 콘솔에서 서비스 연결 역할에 활성 세션이 있는지 확인하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택합니다. 그런 다음 AWSServiceRoleForRDS 역할의 이름(확인란 아님)을 선택합니다.
3. 선택한 역할의 요약 페이지에서 Access Advisor(액세스 관리자) 탭을 선택합니다.
4. [Access Advisor] 탭에서 서비스 연결 역할의 최근 활동을 검토합니다.

Note

Amazon RDS에서 AWSServiceRoleForRDS 역할을 사용하는지 잘 모를 경우에는 역할을 삭제할 수 있습니다. 서비스에서 역할을 사용하는 경우에는 삭제가 되지 않으며, 역할이 사용되고 있는 AWS 리전을 볼 수 있습니다. 역할이 사용 중인 경우에는 세션이 종료될 때까지 기다렸다가 역할을 삭제해야 합니다. 서비스 연결 역할에 대한 세션은 취소할 수 없습니다.

AWS Service Role for RDS 역할을 제거하려면 먼저 모든 DB 인스턴스를 삭제해야 합니다.

모든 인스턴스 삭제

이러한 절차 중 하나에 따라 각 인스턴스를 삭제합니다.

인스턴스를 삭제하려면(콘솔)

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택합니다.
3. 삭제하려는 인스턴스를 선택합니다.
4. [Actions]에 대해 [Delete]를 선택합니다.
5. 최종 스냅샷 생성? 메시지가 표시되는 경우 예 또는 아니요를 선택합니다.
6. 이전 단계에서 예를 선택한 경우 최종 스냅샷 이름에 최종 스냅샷 이름을 입력합니다.
7. 삭제를 선택합니다.

인스턴스를 삭제하려면(CLI)

AWS CLI Command Reference의 [delete-db-instance](#) 참조.

인스턴스를 삭제하려면(API)

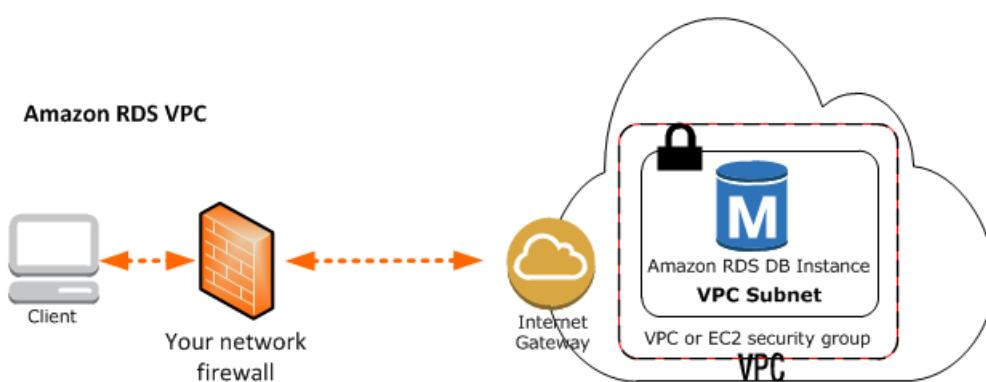
Amazon RDS API Reference의 [DeleteDBInstance](#) 참조.

IAM 콘솔, IAM CLI 또는 IAM API를 사용하여 AWS Service Role for RDS 서비스 연결 역할을 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [서비스에 연결 역할 삭제](#) 단원을 참조하십시오.

Amazon Virtual Private Cloud(VPC) 및 Amazon RDS

Amazon RDS DB 인스턴스를 호스팅하는 두 개의 Amazon Elastic Compute Cloud(EC2) 플랫폼이 있습니다. 이러한 플랫폼은 EC2-VPC와 EC2-Classic입니다. Amazon Virtual Private Cloud(Amazon VPC)에서 Amazon RDS DB 인스턴스와 같은 AWS 리소스를 Virtual Private Cloud(VPC)로 시작할 수 있습니다.

Amazon VPC를 사용하여 가상 네트워크 환경을 제어하는 경우, 자신의 IP 어드레스 범위를 선택하고 서브넷을 만들며 라우팅을 구성하고 제어 목록에 액세스할 수 있습니다. Amazon RDS의 기본 기능은 DB 인스턴스가 Amazon VPC에서 실행 중인지 여부와 상관없이 동일합니다. Amazon RDS는 백업, 소프트웨어 패치, 자동 오류 감지 및 복구를 관리합니다. DB 인스턴스를 Amazon VPC에서 실행하는 데는 추가 비용이 들지 않습니다.



EC2-VPC 플랫폼만 지원하는 계정에는 기본 VPC가 있습니다. 달리 지정하지 않는 한 새 DB 인스턴스는 모두 기본 VPC에서 생성됩니다. 신규 Amazon RDS 고객이거나, 전에 DB 인스턴스를 생성한 적이 전혀 없거나, 전에 사용한 적 없는 리전에서 DB 인스턴스를 생성하는 경우, 사용자는 틀림없이 EC2-VPC 플랫폼에 있고 기본 VPC가 있을 것입니다.

EC2-Classic 플랫폼 상의 일부 레거시 DB 인스턴스는 VPC에서 실행되지 않습니다. 레거시 EC2-Classic 플랫폼에는 기본 VPC가 없지만, 두 플랫폼 모두 마찬가지로 자체 VPC를 만들고 DB 인스턴스가 그 VPC에 위치하도록 지정할 수 있습니다.

주제

- [EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 \(p. 407\)](#)
- [VPC에서 DB 인스턴스에 액세스하는 시나리오 \(p. 408\)](#)
- [VPC에서 Amazon RDS DB 인스턴스를 사용한 작업 \(p. 415\)](#)
- [DB 인스턴스에 대한 VPC 업데이트 \(p. 420\)](#)
- [자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 \(p. 421\)](#)

이 문서에서는 Amazon RDS DB 인스턴스 관련 VPC 기능에 대해서만 설명합니다. Amazon VPC에 대한 자세한 내용은 [Amazon VPC 시작 안내서](#) 및 [Amazon VPC 사용 설명서](#)를 참조하십시오. 네트워크 주소 변환(NAT) 사용에 대한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 [NAT 게이트웨이](#)를 참조하십시오.

EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인

사용자의 AWS 계정과 사용자가 선택하는 리전이 두 RDS 플랫폼 EC2-Classic 또는 EC2-VPC 중 어디에 DB 인스턴스가 생성될지를 결정합니다. 플랫폼 유형에 따라 기본 VPC가 있는지 여부와 DB 인스턴스에 대한 액세스 권한을 제공할 때 사용하는 보안 그룹의 유형이 결정됩니다. 레거시 EC2-Classic 플랫폼은 Amazon RDS에서 사용하는 원래의 플랫폼입니다. 이 플랫폼에서 VPC를 사용하려면 Amazon VPC 콘솔 또는 Amazon VPC API를 사용하여 VPC를 생성해야 합니다. EC2-VPC 플랫폼을 지원하는 계정만 모든 DB 인스턴스가 생성되는 기본 VPC를 제공하며, EC2 또는 VPC 보안 그룹을 사용하여 DB 인스턴스에 대한 액세스 권한을 제공해야 합니다.

Note

신규 Amazon RDS 고객이거나, 전에는 DB 인스턴스를 생성한 적이 전혀 없거나, 전에 사용한 적 없는 리전에서 DB 인스턴스를 생성하는 경우, 사용자는 틀림없이 EC2-VPC 플랫폼에 있고 기본 VPC가 있을 것입니다.

RDS 콘솔 또는 EC2 콘솔에서 대시보드를 살펴보면 지정된 리전에서 AWS 계정이 어떤 플랫폼을 사용 중인지 알 수 있습니다. 신규 Amazon RDS 고객이거나, 이전에는 DB 인스턴스를 한 번도 만들어 본 적이 없거나, 과거에 사용한 적 없는 리전에서 DB 인스턴스를 만드는 것이라면, 콘솔 최초 실행 페이지로 리디렉션되어 아래 헤더 페이지는 보이지 않을 수 있습니다.

[Supported Platforms]가 VPC를 나타낼 경우 다음과 같이 현재 리전의 AWS 계정이 EC2-VPC 플랫폼을 사용 중이고 기본 VPC가 있는 것입니다. 기본 VPC의 이름은 지원되는 플랫폼 아래에 표시됩니다. EC2-VPC 플랫폼에서 생성된 DB 인스턴스에 대해 액세스 권한을 제공하려면 VPC 보안 그룹을 만들어야 합니다. VPC 보안 그룹 생성에 대한 자세한 내용은 [자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 \(p. 421\)](#) 단원을 참조하십시오.

You are using the following Amazon RDS resources in the US East (N. Virginia) region (used/quota)

DB Instances (4/40)
Allocated storage (80.00 GB/100.00 TB)
Click here to increase DB instances limit

Reserved instances (0/40)

Snapshots (132)
Manual (2/100)
Automated (22)

Recent events (28)

Event subscriptions (0/20)

Parameter groups (13)
Default (9)
Custom (4/100)

Option groups (9)
Default (7)
Custom (2/20)

Subnet groups (0/50)

Supported platforms VPC
Default network vpc-[redacted]

[Supported Platforms]가 EC2, VPC를 나타낼 경우 다음과 같이 현재 리전의 AWS 계정이 EC2-Classic 플랫폼을 사용 중이고 기본 VPC가 없는 것입니다. EC2-Classic 플랫폼에서 생성된 DB 인스턴스에 대해 액세스 권한을 제공하려면 DB 보안 그룹을 만들어야 합니다. DB 보안 그룹 생성에 대한 자세한 내용은 [DB 보안 그룹 생성 \(p. 395\)](#) 단원을 참조하십시오.

You are using the following Amazon RDS resources in the US West (N. California) region (used/quota)

DB Instances (1/40)
Allocated storage (100.00 GB/100.00 TB)
Click here to increase DB instances limit

Reserved instances (0/40)

Snapshots (30)
Manual (2/100)
Automated (9)

Recent events (4)

Event subscriptions (0/20)

DB Security groups (3/25)

Parameter groups (10)
Default (8)
Custom (2/100)

Option groups (12)
Default (7)
Custom (5/20)

Subnet groups (1/50)

Supported platforms EC2,VPC
Default network none

Note

- EC2-Classic 플랫폼에서 VPC를 만들 수 있지만 EC2-VPC 플랫폼을 지원하는 계정에서처럼 사용자를 위해 기본적으로 생성되지는 않습니다.
- 기존 DB 인스턴스를 VPC로 이동하는 데 관심이 있다면, AWS Management 콘솔을 사용해 손쉽게 이동할 수 있습니다. 자세한 내용은 [VPC에 있지 않은 DB 인스턴스를 VPC로 이동 \(p. 420\)](#) 단원을 참조하십시오.

VPC에서 DB 인스턴스에 액세스하는 시나리오

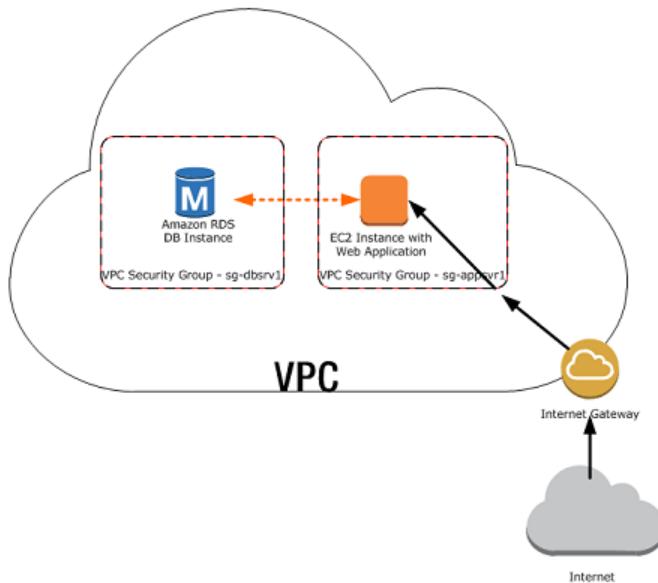
Amazon RDS에서는 VPC의 DB 인스턴스에 액세스하는 다음 시나리오를 지원합니다.

DB 인스턴스	액세스 수행자
VPC 내부	동일한 VPC에 있는 EC2 인스턴스 (p. 409)
	다른 VPC에 있는 EC2 인스턴스 (p. 410)
	VPC에 있지 않은 EC2 인스턴스 (p. 411)
	클라이언트 애플리케이션이 인터넷을 통해 (p. 411)
VPC 외부	VPC에 있는 EC2 인스턴스 (p. 412)
	VPC에 있지 않은 EC2 인스턴스 (p. 413)
	클라이언트 애플리케이션이 인터넷을 통해 (p. 414)

동일한 VPC에 있는 EC2 인스턴스가 VPC 내에 있는 DB 인스턴스에 액세스

VPC에서 RDS 인스턴스를 사용하는 일반적인 사례는 동일한 VPC의 EC2 인스턴스에서 실행 중인 애플리케이션 서버와 데이터를 공유하는 것입니다. 이 사용자 시나리오는 AWS Elastic Beanstalk을 사용해 동일한 VPC에서 EC2 인스턴스와 DB 인스턴스를 만드는 경우입니다.

다음 다이어그램은 이 시나리오를 보여 줍니다.



동일한 VPC에서 EC2 인스턴스와 DB 인스턴스 간 액세스를 관리하는 가장 간단한 방법은 다음과 같습니다.

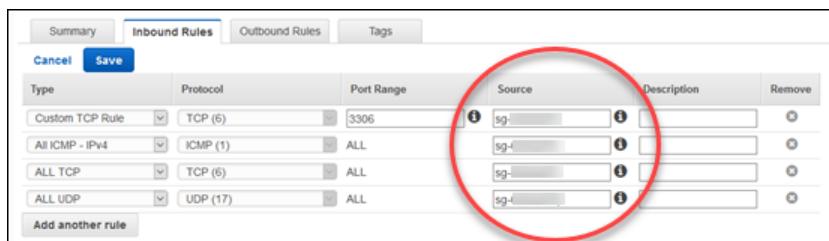
- DB 인스턴스가 포함될 VPC 보안 그룹을 만듭니다. 이 보안 그룹을 사용해 DB 인스턴스에 대한 액세스를 제한할 수 있습니다. 예를 들어, DB 인스턴스를 만들 때 할당한 포트와 개발 등을 위해 DB 인스턴스에 액세스할 때 이용할 IP 주소를 사용해 TCP 액세스를 허용하는 이 보안 그룹을 위한 사용자 지정 규칙을 만들 수 있습니다.
- EC2 인스턴스(웹 서버 및 클라이언트)가 포함될 VPC 보안 그룹을 만듭니다. 이 보안 그룹은 필요할 경우 VPC의 라우팅 테이블을 통한 EC2 인스턴스 액세스를 허용할 수 있습니다. 예를 들어, 이 보안 그룹에서 TCP가 포트 22를 통해 EC2 인스턴스에 액세스하도록 허용하는 규칙을 설정할 수 있습니다.
- DB 인스턴스에 대한 보안 그룹에서 EC2 인스턴스에 대해 생성한 보안 그룹으로부터의 연결을 허용하는 사용자 지정 규칙을 만듭니다. 그러면 보안 그룹의 모든 구성원이 DB 인스턴스에 액세스하도록 허용됩니다.

이 시나리오에 대해 퍼블릭 서브넷과 프라이빗 서브넷 모두에서 VPC를 생성하는 방법을 보여주는 자습서는 [자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 \(p. 421\)](#) 단원을 참조하십시오.

VPC 보안 그룹에서 다른 보안 그룹으로부터의 연결을 허용하는 규칙을 만들려면 다음을 수행합니다.

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/vpc>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 다른 보안 그룹의 구성원에 대한 액세스를 허용할 보안 그룹을 선택하거나 만듭니다. 위 시나리오에서는 DB 인스턴스에 대해 사용할 보안 그룹이 이에 해당합니다. 인바운드 규칙 탭을 선택한 후 규칙 편집을 선택합니다.

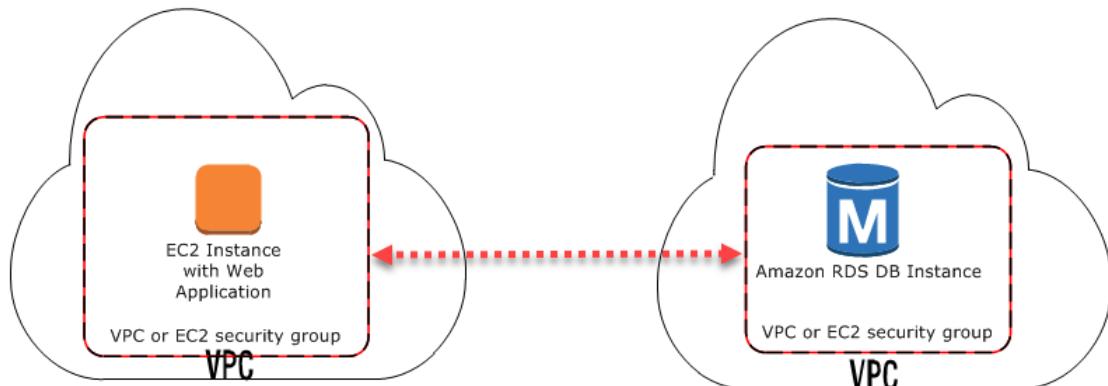
4. 인바운드 규칙 편집 페이지에서 규칙 추가를 선택합니다.
5. 유형에서 모든 ICMP 옵션 중 하나를 선택합니다. [Source] 상자에 보안 그룹 ID를 입력하기 시작합니다. 입력을 시작하면 보안 그룹 목록이 제공됩니다. 이 보안 그룹에서 보호 중인 리소스에 대한 액세스를 허용할 구성원이 포함된 보안 그룹을 선택합니다. 위 시나리오에서는 EC2 인스턴스에 대해 사용할 보안 그룹이 이에 해당합니다.
6. [Type]으로 [All TCP]를 지정하고 [Source] 상자에 사용자의 보안 그룹을 입력한 규칙을 만들어 TCP 프로토콜에 대해 단계를 반복합니다. UDP 프로토콜을 사용하려는 경우, [Type]으로 [All UDP]를 지정하고 [Source] 상자에 사용자의 보안 그룹을 입력하여 규칙을 만듭니다.
7. DB 인스턴스를 만들 때 사용한 포트(예: MySQL용 포트 3306)를 통한 액세스를 허용하는 사용자 지정 TCP 규칙을 만듭니다. [Source] 상자에 사용자의 보안 그룹 또는 IP 주소를 입력합니다.
8. 마치면 [Save]를 선택합니다.



VPC에 있는 DB 인스턴스에 다른 VPC에 있는 EC2 인스턴스가 액세스

DB 인스턴스가 액세스 시 사용할 EC2 인스턴스와 다른 VPC에 있는 경우 DB 인스턴스에 액세스하기 위해 VPC 피어링을 사용할 수 있습니다.

다음 다이어그램은 이 시나리오를 보여 줍니다.

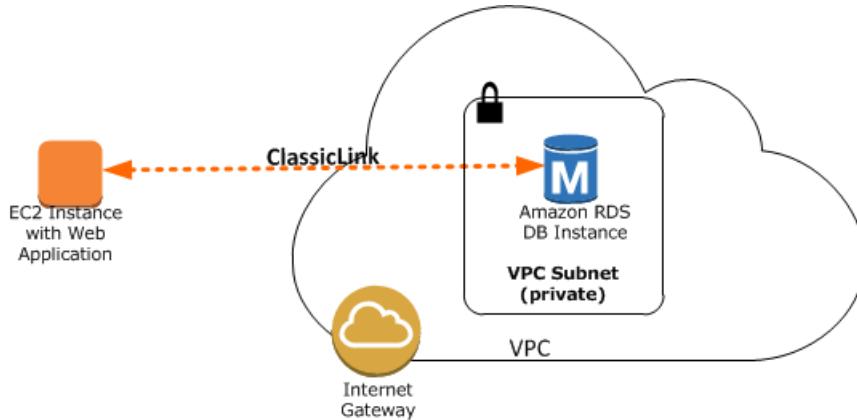


VPC 피어링 연결은 프라이빗 IP 주소를 사용하여 두 VPC 간에 트래픽을 라우팅할 수 있도록 하기 위한 두 VPC 사이의 네트워킹 연결입니다. 동일한 네트워크에 속하는 경우와 같이 VPC의 인스턴스가 서로 통신할 수 있습니다. 자체 VPC 간, 다른 AWS 계정에서 VPC를 사용하여 또는 다른 AWS 리전에서 VPC를 사용하여 VPC 피어링 연결을 만들 수 있습니다. VPC 피어링에 대한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 [VPC 피어링](#) 단원을 참조하십시오.

VPC에 있지 않은 EC2 인스턴스가 VPC에 있는 DB 인스턴스에 액세스

ClassicLink를 사용하여 VPC에 있는 Amazon RDS DB 인스턴스와 Amazon VPC에 있지 않은 EC2 인스턴스 간에 통신할 수 있습니다. ClassicLink를 사용하면 EC2 인스턴스상의 애플리케이션은 DB 인스턴스에 대한 RDS 엔드포인트를 사용하여 DB 인스턴스에 연결할 수 있습니다. ClassicLink는 무료로 사용할 수 있습니다.

다음 다이어그램은 이 시나리오를 보여 줍니다.



ClassicLink를 사용하여 IP 주소 범위를 정의하고 액세스 제어 목록(ACL)을 제어하여 네트워크 트래픽을 관리하는 논리적으로 분리되어 있는 데이터베이스에 EC2 인스턴스를 연결할 수 있습니다. VPC에서는 퍼블릭 IP 주소 또는 터널링을 사용하여 DB 인스턴스와 통신할 필요가 없습니다. 따라서 인스턴스 간 통신에서 처리량이 향상되고 연결 시 지연 시간이 짧아집니다.

VPC에 있는 DB 인스턴스와 VPC에 있지 않은 EC2 인스턴스 간에 ClassicLink를 활성화하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/vpc>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs(VPCs)를 선택합니다.
3. DB 인스턴스가 사용하는 VPC를 선택합니다.
4. [Actions]에서 [Enable ClassicLink]를 선택합니다. 확인 대화 상자에서 [Yes, Enable]을 선택합니다.
5. EC2 콘솔에서 VPC에 있는 DB 인스턴스에 연결하려는 EC2 인스턴스를 선택합니다.
6. [Actions]에서 [ClassicLink]를 선택한 후 [Link to VPC]를 선택합니다.
7. [Link to VPC] 페이지에서 사용하려는 보안 그룹을 선택한 후 [Link to VPC]를 선택합니다.

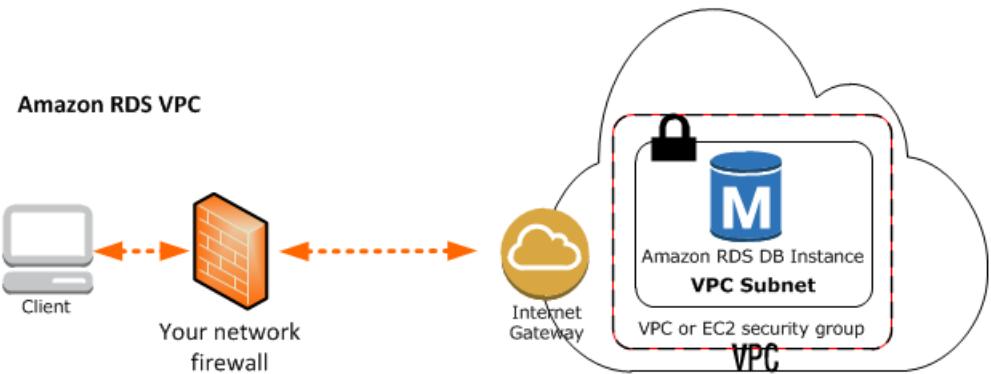
Note

ClassicLink 기능은 EC2-Classic을 지원하는 계정 및 리전의 콘솔에만 표시됩니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [ClassicLink](#) 단원을 참조하십시오.

클라이언트 애플리케이션이 인터넷을 통해 VPC에 있는 DB 인스턴스에 액세스

인터넷을 통해 클라이언트 애플리케이션에서 VPC에 있는 DB 인스턴스에 액세스하려면 단일 퍼블릭 서브넷만 있는 VPC와 인터넷을 통한 통신을 지원하는 인터넷 게이트웨이를 구성합니다.

다음 다이어그램은 이 시나리오를 보여 줍니다.



다음 구성을 권장합니다.

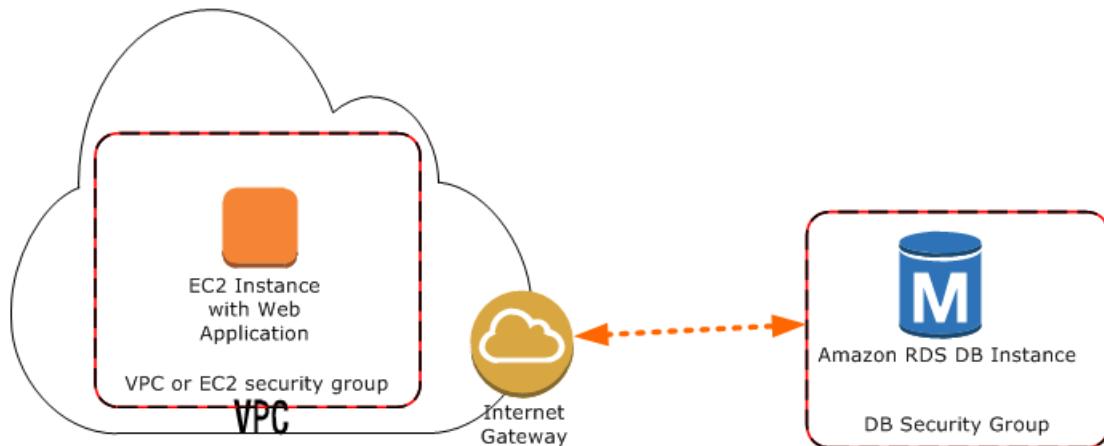
- 크기가 /16인 VPC(예: CIDR: 10.0.0.0/16). 이 크기는 65,536개의 프라이빗 IP 주소를 제공합니다.
- 크기가 /24인 서브넷(예: CIDR: 10.0.0.0/24). 이 크기는 256개의 프라이빗 IP 주소를 제공합니다.
- VPC 및 서브넷과 연결된 Amazon RDS DB 인스턴스입니다. Amazon RDS는 서브넷 내의 IP 주소를 DB 인스턴스에 할당합니다.
- VPC를 인터넷 및 다른 AWS 제품과 연결하는 인터넷 게이트웨이.
- DB 인스턴스와 연결된 보안 그룹. 보안 그룹의 인바운드 규칙은 클라이언트 애플리케이션이 사용자의 DB 인스턴스에 액세스할 수 있도록 해줍니다.

VPC에서 DB 인스턴스 생성에 대한 자세한 내용은 [VPC에 DB 인스턴스 만들기 \(p. 417\)](#) 단원을 참조하십시오.

VPC에 있는 EC2 인스턴스가 VPC에 있지 않은 DB 인스턴스에 액세스

EC2 인스턴스가 VPC에 있고 RDS DB 인스턴스가 VPC에 있지 않은 경우에는 퍼블릭 인터넷을 통해 연결할 수 있습니다.

다음 다이어그램은 이 시나리오를 보여 줍니다.



Note

[VPC에 있지 않은 EC2 인스턴스가 VPC에 있는 DB 인스턴스에 액세스 \(p. 411\)](#)에 설명된 대로 ClassicLink는 이 시나리오에 사용할 수 없습니다.

퍼블릭 인터넷을 통해 DB 인스턴스 및 EC2 인스턴스를 연결하려면 다음을 수행하십시오.

- EC2 인스턴스가 VPC 내 퍼블릭 서브넷 안에 있는지 확인합니다.
- RDS DB 인스턴스가 공개적으로 액세스 가능하도록 표시되었는지 확인합니다.
- 이때 네트워크 ACL에 관한 주의 사항이 하나 있습니다. 네트워크 ACL은 전체 서브넷의 방화벽과 같습니다. 그러므로 해당 서브넷의 모든 인스턴스에 네트워크 ACL 규칙이 적용됩니다. 기본적으로 ACL은 모든 트래픽을 허용하므로, 추가 보안 계층으로 규칙을 추가하려는 경우가 아니라면 신경 쓸 필요가 없습니다. 반면, 보안 그룹은 개별 인스턴스와 연결되므로 보안 그룹 규칙을 신경 써야 합니다.
- RDS DB 인스턴스에 대한 DB 보안 그룹에 수신 규칙을 추가합니다.

수신 규칙은 네트워크 포트와 CIDR/IP 범위를 지정합니다. 예를 들어 포트 3306이 MySQL RDS DB 인스턴스에 연결하도록 허용하고 CIDR/IP 범위를 203.0.113.25/32로 지정하는 수신 규칙을 추가할 수 있습니다. 자세한 내용은 [특정 IP 범위에서 DB 보안 그룹에 대한 네트워크 액세스 승인 \(p. 398\)](#) 단원을 참조하십시오.

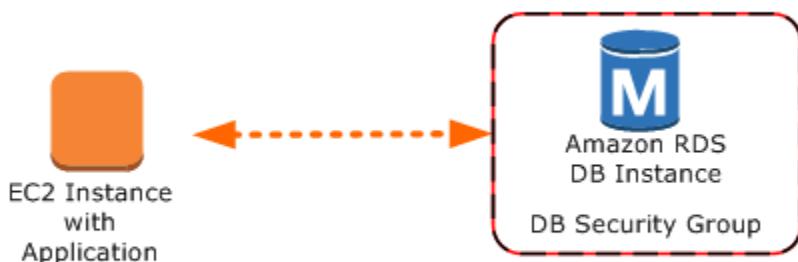
Note

기존 DB 인스턴스를 VPC로 이동하는 데 관심이 있다면, AWS Management 콘솔을 사용해 손쉽게 이동할 수 있습니다. 자세한 내용은 [VPC에 있지 않은 DB 인스턴스를 VPC로 이동 \(p. 420\)](#) 단원을 참조하십시오.

VPC에 있지 않은 EC2 인스턴스가 VPC에 있지 않은 DB 인스턴스에 액세스

DB 인스턴스와 EC2 인스턴스 상의 애플리케이션이 모두 VPC에 있지 않은 경우 엔드포인트 및 포트를 사용해 DB 인스턴스에 액세스할 수 있습니다.

다음 다이어그램은 이 시나리오를 보여 줍니다.



DB 인스턴스를 만들 때 지정한 포트로부터 액세스를 허용하는 DB 인스턴스 보안 그룹을 DB 인스턴스에 대해 만들어야 합니다. 예를 들어, Oracle DB 인스턴스에 액세스하기 위해 sqlplus와 함께 사용되는 이 연결 문자열과 비슷한 연결 문자열을 사용할 수 있습니다.

```
PROMPT>sqlplus 'mydbusr@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<endpoint>)
```

```
(PORT=<port number>))(CONNECT_DATA=(SID=<database name>)))'
```

자세한 내용은 다음 문서 단원을 참조하십시오.

데이터베이스 엔진	관련 문서
MariaDB	MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 (p. 445)
Microsoft SQL Server	Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 (p. 503)
MySQL	MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 (p. 595)
Oracle	Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 (p. 736)
PostgreSQL	PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 (p. 953)

Note

기존 DB 인스턴스를 VPC로 이동하는 데 관심이 있다면, AWS Management 콘솔을 사용해 손쉽게 이동할 수 있습니다. 자세한 내용은 [VPC에 있지 않은 DB 인스턴스를 VPC로 이동 \(p. 420\)](#) 단원을 참조하십시오.

클라이언트 애플리케이션이 인터넷을 통해 VPC에 있지 않은 DB 인스턴스에 액세스

신규 Amazon RDS 고객은 VPC에서만 DB 인스턴스를 만들 수 있습니다. 하지만 클라이언트 애플리케이션에서 인터넷을 통해 VPC에 있지 않은 기존 Amazon RDS DB 인스턴스에 연결해야 하는 경우도 있습니다.

다음 다이어그램은 이 시나리오를 보여 줍니다.



이 시나리오에서는 RDS DB 인스턴스에 대한 DB 보안 그룹이 클라이언트 애플리케이션이 연결하는데 필요한 수신 규칙을 포함하는지 확인해야 합니다. 수신 규칙은 네트워크 포트와 CIDR/IP 범위를 지정합니다. 예를 들어 포트 3306이 MySQL RDS DB 인스턴스에 연결하도록 허용하고 CIDR/IP 범위를 203.0.113.25/32로 지정하는 수신 규칙을 추가할 수 있습니다. 자세한 내용은 [특정 IP 범위에서 DB 보안 그룹에 대한 네트워크 액세스 승인 \(p. 398\)](#) 단원을 참조하십시오.

Warning

방화벽 뒤의 DB 인스턴스에 액세스하려는 경우 네트워크 관리자에게 문의하여 사용해야 하는 IP 주소를 확인합니다.

Note

기존 DB 인스턴스를 VPC로 이동하는 데 관심이 있다면, AWS Management 콘솔을 사용해 손쉽게 이동할 수 있습니다. 자세한 내용은 [VPC에 있지 않은 DB 인스턴스를 VPC로 이동 \(p. 420\)](#) 단원을 참조하십시오.

VPC에서 Amazon RDS DB 인스턴스를 사용한 작업

레거시 DB 인스턴스를 사용하지 않는 한, DB 인스턴스는 Virtual Private Cloud(VPC)에서 실행됩니다. 가상 사설 클라우드는 AWS 클라우드에 있는 다른 가상 네트워크와 논리적으로 격리되어 있는 가상 네트워크입니다. Amazon VPC에서는 Amazon RDS 또는 Amazon EC2 인스턴스와 같은 AWS 리소스를 VPC로 시작할 수 있습니다. VPC는 계정과 함께 제공되는 기본 VPC일 수도 있고, 사용자가 만들 수도 있습니다. 모든 VPC는 AWS 계정과 연결됩니다.

기본 VPC에는 VPC 내의 리소스를 격리하는 데 사용할 수 있는 3개의 서브넷이 있습니다. 또한 VPC 외부에서 VPC 내부의 리소스에 액세스할 수 있도록 하는 데 사용할 수 있는 인터넷 게이트웨이도 있습니다.

VPC 내부와 외부의 Amazon RDS DB 인스턴스를 포함하는 시나리오 목록은 [VPC에서 DB 인스턴스에 액세스하는 시나리오 \(p. 408\)](#) 단원을 참조하십시오.

일반 Amazon RDS 시나리오에서 사용할 수 있는 VPC를 생성하는 방법을 보여주는 자습서는 [자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 \(p. 421\)](#) 단원을 참조하십시오.

VPC 내부에서 Amazon RDS DB 인스턴스를 사용하여 작업을 수행하는 방법은 다음 단원을 참조하십시오.

주제

- [VPC에서 DB 인스턴스를 사용한 작업 \(p. 415\)](#)
- [DB 서브넷 그룹을 사용한 작업 \(p. 416\)](#)
- [VPC에 있는 DB 인스턴스를 인터넷에서 숨기기 \(p. 416\)](#)
- [VPC에 DB 인스턴스 만들기 \(p. 417\)](#)

VPC에서 DB 인스턴스를 사용한 작업

다음은 VPC에서 DB 인스턴스를 사용하여 작업할 때 유용한 몇 가지 팁입니다.

- VPC는 최소 2개 이상의 서브넷을 보유해야 합니다. 이러한 서브넷은 DB 인스턴스를 배포하고자 하는 리전에 있는 2개의 다른 가용 영역에 있어야 합니다. 서브넷은 사용자가 지정할 수 있고 보안 및 운영상의 필요를 바탕으로 인스턴스를 그룹화 할 수 있게 해주는 VPC IP 주소 범위의 한 부분입니다.
- VPC에서 DB 인스턴스에 공개적으로 액세스할 수 있도록 하려면 VPC 속성인 [DNS hostnames]와 [DNS resolution]을 활성화해야 합니다.
- VPC에는 사용자가 만드는 DB 서브넷 그룹이 있어야 합니다(자세한 내용은 다음 섹션 참조). 생성한 서브넷을 지정하여 DB 서브넷을 생성합니다. Amazon RDS는 해당 DB 서브넷 그룹과 기본 가용 영역을 사용하여 DB 인스턴스에 할당할 서브넷과 해당 서브넷 내의 IP 주소를 선택합니다.
- VPC에는 DB 인스턴스에 대한 액세스를 허용하는 VPC 보안 그룹이 있어야 합니다.
- 각 서브넷의 CIDR 블록은 장애 조치와 컴퓨팅 조정을 포함한 유지 관리 활동 중에 Amazon RDS가 사용 예비 IP 주소를 수용할 만큼 충분히 커야 합니다.
- VPC의 [instance tenancy] 속성 값은 [default] 또는 [dedicated] 중 하나일 수 있습니다. 모든 기본 VPC에서 인스턴스 테넌시 속성이 기본값으로 설정되어 있으며, 기본 VPC는 어떤 DB 인스턴스 클래스라도 지원할 수 있습니다.

인스턴스 테넌시 속성이 전용으로 설정되어 있는 전용 VPC에 DB 인스턴스를 두기로 선택할 경우, DB 인스턴스의 DB 인스턴스 클래스는 승인된 Amazon EC2 전용 인스턴스 유형 중 하나여야 합니다. 예를 들어 m3.medium EC2 전용 인스턴스는 db.m3.medium DB 인스턴스 클래스에 해당합니다. VPC의 인스턴스 테

년시에 대한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 [EC2 전용 인스턴스 사용](#)을 참조하십시오.

전용 인스턴스에 존재할 수 있는 인스턴스 유형에 대한 자세한 내용은 EC2 요금 페이지에서 [Amazon EC2 전용 인스턴스](#)를 참조하십시오.

- 옵션 그룹이 DB 인스턴스에 할당되면 VPC 또는 EC2-Classic(VPC 아님) 중에서 지원되는 DB 인스턴스 플랫폼에 연결됩니다. 또한 DB 인스턴스가 VPC에 속한 경우 해당 DB 인스턴스와 연결된 옵션 그룹은 바로 이 VPC에도 연결됩니다. 이 연결은 다른 VPC나 다른 플랫폼으로 DB 인스턴스를 복구하려고 해도 DB 인스턴스에 배정된 옵션 그룹은 사용할 수 없음을 의미합니다.
- DB 인스턴스를 다른 VPC나 다른 플랫폼으로 복구할 경우 기본 옵션 그룹을 DB 인스턴스에 할당하거나, 해당 VPC 또는 플랫폼에 연결된 옵션 그룹을 할당하거나, 새 옵션 그룹을 생성하여 DB 인스턴스에 할당해야 합니다. Oracle TDE와 같은 지속적 또는 영구적 옵션을 사용하여 DB 인스턴스를 다른 VPC로 복구할 때는 지속적 또는 영구적 옵션을 포함하는 새 옵션 그룹을 생성해야 합니다.

DB 서브넷 그룹을 사용한 작업

서브넷은 사용자가 보안 및 운영상의 필요를 바탕으로 리소스를 그룹화하기 위해 지정하는 VPC IP 주소 범위의 특정 부분입니다. DB 서브넷 그룹은 사용자가 VPC에서 만든 다음 DB 인스턴스에 대해 지정하는 서브넷(일반적으로 프라이빗)의 모음입니다. DB 서브넷 그룹을 통해 CLI 또는 API를 사용하여 DB 인스턴스를 만들 때 특정 VPC를 지정할 수 있고, 콘솔을 사용하면 사용하려는 VPC와 서브넷을 선택만 할 수 있습니다.

각 DB 서브넷 그룹은 지정된 리전에서 두 개 이상의 가용 영역에 서브넷이 있어야 합니다. VPC에서 DB 인스턴스를 생성할 때는 DB 서브넷 그룹을 선택해야 합니다. Amazon RDS는 해당 DB 서브넷 그룹과 기본 가용 영역을 사용하여 DB 인스턴스와 연결할 서브넷과 해당 서브넷 내의 IP 주소를 선택합니다. 다중 AZ 배포의 기본 DB 인스턴스에 오류가 있을 경우 Amazon RDS는 해당 스탠바이를 승격한 후 다른 가용 영역 중 하나에서 서브넷의 IP 주소를 사용하여 새 스탠바이를 만들 수 있습니다.

Amazon RDS가 VPC에 DB 인스턴스를 만들 때는 DB 서브넷 그룹에서 선택한 IP 주소를 사용하여 DB 인스턴스에 네트워크 인터페이스를 할당합니다. 하지만 기본 IP 주소가 장애 조치 중에 변경될 수 있기 때문에 DB 인스턴스에 연결할 때 반드시 DNS 이름을 사용하는 것이 좋습니다.

Note

VPC에서 실행하는 각각의 DB 인스턴스에 대해 Amazon RDS가 복구 작업에 사용할 수 있도록 DB 서브넷 그룹의 각 서브넷에 한 개 이상의 주소를 예약해야 합니다.

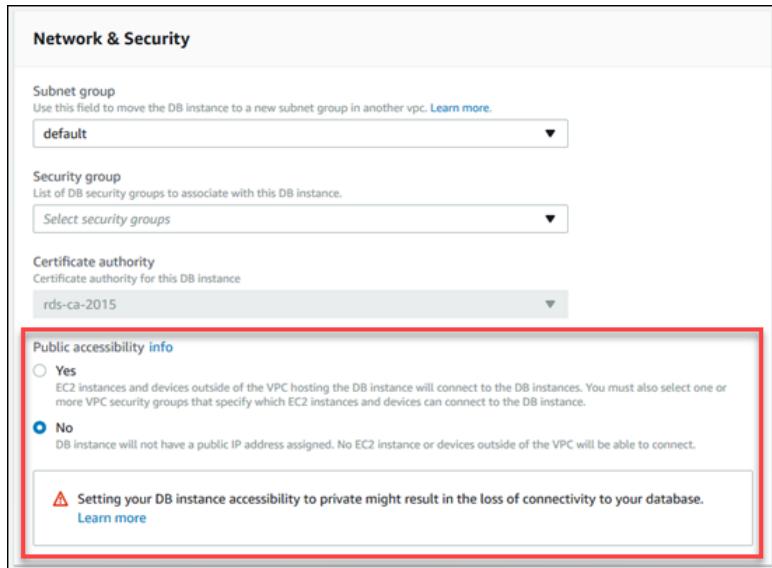
VPC에 있는 DB 인스턴스를 인터넷에서 숨기기

한 가지 일반적인 Amazon RDS 시나리오는 일반에 공개되어 있는 웹 애플리케이션을 포함한 EC2 인스턴스와 공개적으로 액세스할 수 없는 데이터베이스를 포함한 DB 인스턴스가 있는 VPC를 사용하는 경우입니다. 예를 들어, 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC를 생성할 수 있습니다. 웹 서버의 기능을 수행하는 Amazon EC2가 퍼블릭 서브넷에 배포될 수 있으며, Amazon RDS DB 인스턴스는 프라이빗 서브넷에 배포됩니다. 이런 배포 환경에서는 웹 서버만 DB 인스턴스에 액세스할 수 있습니다. 이 시나리오에 대한 그림은 [동일한 VPC에 있는 EC2 인스턴스가 VPC 내에 있는 DB 인스턴스에 액세스 \(p. 409\)](#) 단원을 참조하십시오.

VPC 내부에 있는 DB 인스턴스를 시작할 때 Public accessibility 파라미터를 사용하여 사용자가 만드는 DB 인스턴스가 퍼블릭 IP 주소로 확인되는 DNS를 가지도록 지정할 수 있습니다. 이 파라미터를 사용하여 DB 인스턴스에 대한 퍼블릭 액세스 여부를 지정할 수 있습니다. DB 인스턴스에 대한 액세스는 궁극적으로 인스턴스에서 사용하는 보안 그룹에 의해 제어되며, DB 인스턴스에 할당된 보안 그룹에서 퍼블릭 액세스를 허용하지 않으면 퍼블릭 액세스가 허용되지 않습니다.

Public accessibility 파라미터를 수정하여 퍼블릭 액세스 가능성을 켜거나 끄도록 DB 인스턴스를 수정할 수 있습니다. 이 파라미터는 다른 DB 인스턴스 파라미터와 같은 방법으로 수정합니다. 자세한 내용은 DB 엔진에 대한 수정을 참조하십시오.

다음 그림은 네트워크 보안 섹션의 퍼블릭 액세스 가능 옵션을 보여 줍니다.



VPC에 DB 인스턴스 만들기

다음 절차에 따라 VPC에서 DB 인스턴스를 생성할 수 있습니다. 계정에 기본 VPC가 있으면 VPC 및 DB 서브넷 그룹이 자동으로 생성된 것이므로 3단계부터 시작하면 됩니다. AWS 계정에 기본 VPC가 없거나 추가 VPC를 만들려면 새 VPC를 만들 수 있습니다.

기본 VPC가 있는지 모르겠으면 [EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 \(p. 407\)](#) 단원을 참조하십시오.

Note

VPC에서 DB 인스턴스에 공개적으로 액세스할 수 있도록 하려면 VPC 속성인 [DNS hostnames]와 [DNS resolution]을 활성화하여 VPC에 대한 DNS 정보를 업데이트해야 합니다. VPC 인스턴스에 대한 DNS 정보의 업데이트에 대한 자세한 내용은 [Updating DNS Support for Your VPC](#)를 참조하십시오.

다음 단계에 따라 VPC에서 DB 인스턴스를 만들 수 있습니다:

- 1 단계: VPC 생성 (p. 417)
- 2단계: VPC에 서브넷 추가 (p. 417)
- 3단계: DB 서브넷 그룹 만들기 (p. 418)
- 4단계: VPC 보안 그룹 만들기 (p. 419)
- 5단계: VPC에서 DB 인스턴스 만들기 (p. 419)

1 단계: VPC 생성

AWS 계정에 기본 VPC가 없거나 추가 VPC를 만들려면 지침에 따라 새 VPC를 만듭니다. Amazon RDS 문서의 [프라이빗 서브넷과 퍼블릭 서브넷을 포함하는 VPC 생성 \(p. 422\)](#) 또는 Amazon VPC 설명서의 [1단계: VPC 생성](#)을 참조하십시오.

2단계: VPC에 서브넷 추가

VPC를 만들었으면 두 개 이상의 가용 영역에 서브넷을 만들어야 합니다. DB 서브넷 그룹을 만들 때 이들 서브넷이 사용됩니다. 기본 VPC가 있는 경우에는 해당 리전의 각 가용 영역에 서브넷이 자동으로 생성됩니다.

VPC에서 서브넷을 생성하는 방법에 대한 지침은 Amazon RDS 설명서의 [프라이빗 서브넷과 퍼블릭 서브넷을 포함하는 VPC 생성 \(p. 422\)](#) 단원을 참조하십시오.

3단계: DB 서브넷 그룹 만들기

DB 서브넷 그룹은 사용자가 VPC에 대해 만든 다음 DB 인스턴스에 대해 지정하는 서브넷(일반적으로 프라이빗)의 모음입니다. DB 서브넷 그룹에서는 CLI 또는 API를 사용하여 DB 인스턴스를 생성할 때 특정 VPC를 지정할 수 있습니다. Amazon RDS 콘솔을 사용하는 경우 사용할 VPC와 서브넷만 선택할 수 있습니다. 각 DB 서브넷 그룹은 해당 리전에서 두 개 이상의 가용 영역에 한 개 이상의 서브넷이 있어야 합니다.

Note

DB 인스턴스에 공개적으로 액세스할 수 있도록 하려면 DB 서브넷 그룹의 서브넷에 인터넷 게이트웨이가 있어야 합니다. 서브넷용 인터넷 게이트웨이에 대한 자세한 내용은 Amazon VPC 설명서의 [인터넷 게이트웨이](#)를 참조하십시오.

VPC에서 DB 인스턴스를 생성할 때는 DB 서브넷 그룹을 선택해야 합니다. 그러면 Amazon RDS는 해당 DB 서브넷 그룹과 기본 가용 영역을 사용하여 서브넷과 해당 서브넷 내의 IP 주소를 선택합니다. Amazon RDS는 탄력적 네트워크 인터페이스를 생성하여 해당 IP 주소가 있는 DB 인스턴스에 연결합니다. 다중 AZ 배포의 경우, 한 리전에 있는 두 개 이상의 가용 영역에 대한 서브넷을 정의하면 Amazon RDS가 필요한 경우 다른 가용 영역에 새로운 예비 복제본을 만들 수 있습니다. 단일 AZ 배포의 경우도 어느 시점에 단일 AZ 배포를 다중 AZ 배포로 변환하려면 이 작업을 수행해야 합니다.

이 단계에서는 DB 서브넷 그룹을 만들고 VPC용으로 만든 서브넷을 추가합니다.

AWS Management 콘솔

DB 서브넷 그룹을 만드는 방법

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Subnet groups]를 선택합니다.
3. [Create DB Subnet Group]을 선택합니다.
4. [Name]에 DB 서브넷 그룹의 이름을 입력합니다.
5. [Description]에 DB 서브넷 그룹에 대한 설명을 입력합니다.
6. [VPC]에 대해 생성된 VPC를 선택합니다.
7. 서브넷 추가 섹션에서 이 VPC와 관련된 모든 서브넷 추가를 선택합니다.

Create DB subnet group

To create a new Subnet Group give it a name, description, and select an existing VPC below. Once you select an existing VPC, you will be able to add subnets related to that VPC.

Subnet group details

Name: mydbsubnetgroup

Description: My DB Subnet Group

VPC: tutorial-vpc (vpc-971c12ee)

Add subnets

Add subnet(s) to this subnet group. You may add subnets one at a time below or add all the subnets related to this VPC. You may make additions/edits after this group is created. A minimum of 2 subnets is required.

Add all the subnets related to this VPC

Availability zone: select an availability zone

Subnet: select a subnet

Subnets in this subnet group (4)

Availability zone	Subnet ID	CIDR block	Action
us-east-1a	subnet-d8c8e7f4	10.0.2.0/24	Remove
us-east-1f	subnet-718fdc7d	10.0.3.0/24	Remove
us-east-1a	subnet-cbc8e7e7	10.0.1.0/24	Remove
us-east-1a	subnet-0ccde220	10.0.0.0/24	Remove

Create

8. Create를 선택합니다.

새 DB 서브넷 그룹은 RDS 콘솔의 DB 서브넷 그룹 목록에 나타납니다. DB 서브넷 그룹을 클릭하면 창 하단에 있는 세부 정보 창에서 그룹과 연결된 모든 서브넷을 포함한 세부 정보를 확인할 수 있습니다.

4단계: VPC 보안 그룹 만들기

DB 인스턴스를 만들기 전에 DB 인스턴스와 연결할 VPC 보안 그룹을 만들어야 합니다. DB 인스턴스에 대한 보안 그룹을 생성하는 방법은 Amazon RDS 설명서의 [프라이빗 Amazon RDS DB 인스턴스에 대한 VPC 보안 그룹 생성 \(p. 424\)](#) 또는 Amazon VPC 설명서의 [VPC용 보안 그룹](#)을 참조하십시오.

5단계: VPC에서 DB 인스턴스 만들기

이 단계에서는 DB 인스턴스를 만들고 이전 단계에서 만든 VPC 이름, DB 서브넷 그룹 및 VPC 보안 그룹을 사용합니다.

Note

VPC에서 DB 인스턴스에 공개적으로 액세스할 수 있도록 하려면 VPC 속성인 [DNS hostnames]와 [DNS resolution]을 활성화해야 합니다. VPC 인스턴스에 대한 DNS 정보를 업데이트하는 방법에 대한 자세한 내용은 [Updating DNS Support for Your VPC](#)를 참조하십시오.

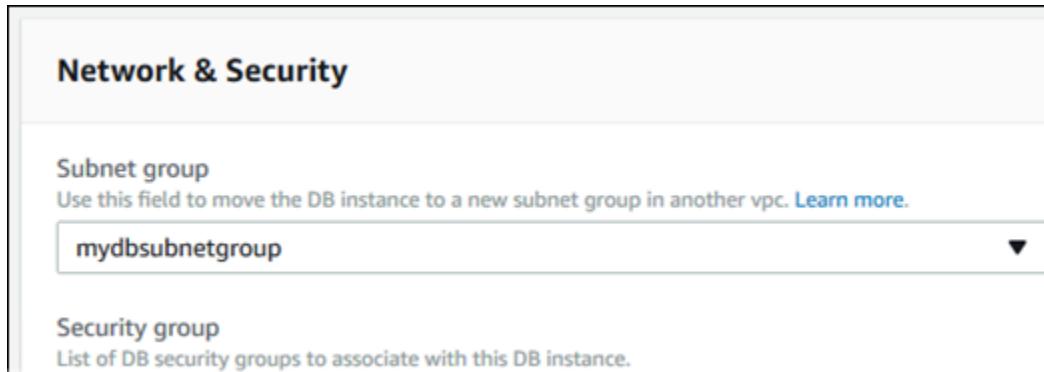
사용자의 DB 엔진을 위한 DB 인스턴스를 만드는 방법에 대한 자세한 내용은 해당 DB 엔진에 대해 설명하는 다음 주제를 참조하십시오. 각 엔진에 대해 네트워크 및 보안 섹션에 메시지가 표시되면, 이전 단계에서 만든 VPC 이름, DB 서브넷 그룹 및 VPC 보안 그룹을 입력합니다.

데이터베이스 엔진	관련 문서
MariaDB	MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 (p. 437)
Microsoft SQL Server	Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 생성 (p. 494)
MySQL	MySQL 데이터베이스 엔진 기반 DB 인스턴스 생성 (p. 587)
Oracle	Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 (p. 727)
PostgreSQL	PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스 만들기 (p. 947)

DB 인스턴스에 대한 VPC 업데이트

AWS Management 콘솔을 이용하여 DB 인스턴스를 손쉽게 다른 VPC로 이동할 수 있습니다.

사용자의 DB 엔진을 위한 DB 인스턴스를 수정하는 방법에 대한 자세한 내용은 다음 표에서 해당 DB 엔진에 대해 설명하는 주제를 참조하십시오. 수정 페이지의 Network & Security(네트워크 및 보안) 섹션에서 다음과 같이 Subnet group(서브넷 그룹)에 새 서브넷 그룹을 입력합니다. 새 서브넷 그룹은 새 VPC에 있는 서브넷 그룹이어야 합니다.



데이터베이스 엔진	관련 문서
MariaDB	MariaDB 데이터베이스 엔진 기반 DB 인스턴스의 변경 (p. 449)
Microsoft SQL Server	Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 (p. 509)
MySQL	MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경 (p. 600)
Oracle	Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 (p. 742)
PostgreSQL	PostgreSQL 데이터베이스 엔진을 실행 중인 DB 인스턴스 수정 (p. 956)

VPC에 있지 않은 DB 인스턴스를 VPC로 이동

EC2-Classic 플랫폼 상의 일부 레거시 DB 인스턴스는 VPC에서 실행되지 않습니다. DB 인스턴스가 VPC 내에 있지 않을 경우 AWS Management 콘솔을 사용하여 DB 인스턴스를 VPC 내로 손쉽게 이동할 수 있습니다. VPC에 있지 않은 DB 인스턴스를 VPC 내로 이동하려면 먼저 VPC를 만들어야 합니다.

다음 단계에 따라 DB 인스턴스에 대한 VPC를 만듭니다.

- 1 단계: VPC 생성 (p. 417)
- 2단계: VPC에 서브넷 추가 (p. 417)
- 3단계: DB 서브넷 그룹 만들기 (p. 418)
- 4단계: VPC 보안 그룹 만들기 (p. 419)

VPC를 만든 후 다음 단계에 따라 DB 인스턴스를 VPC 내로 이동합니다.

- DB 인스턴스에 대한 VPC 업데이트 (p. 420)

다음은 DB 인스턴스를 VPC로 이동하는 데 따른 몇 가지 제한 사항입니다.

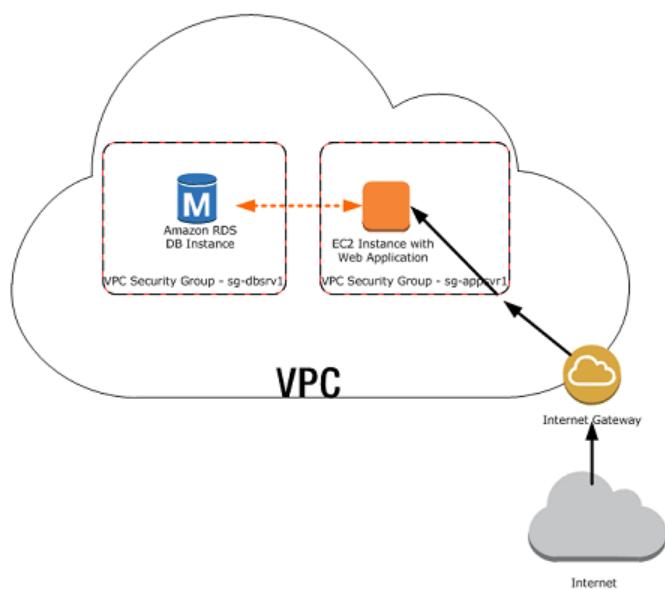
- VPC에 있지 않은 다른 AZ DB 인스턴스를 VPC로 이동하는 것은 현재 지원되지 않습니다.
- VPC에 있지 않은 읽기 전용 복제본을 포함한 DB 인스턴스를 VPC로 이동하는 것은 현재 지원되지 않습니다.

DB 인스턴스를 VPC로 이동하고 DB 인스턴스와 함께 사용자 지정 옵션 그룹을 사용 중인 경우에는 DB 인스턴스와 연결된 옵션 그룹을 변경해야 합니다. 옵션 그룹은 플랫폼마다 특정하며, VPC로 이동하는 것은 플랫폼의 변경에 해당합니다. 이런 경우에서 사용자 지정 옵션 그룹을 사용하려면 기본 VPC 옵션 그룹을 DB 인스턴스에 할당하거나, 이동하려는 대상 VPC에 있는 다른 DB 인스턴스에서 사용되는 옵션 그룹을 할당하거나, 새 옵션 그룹을 만들어 DB 인스턴스에 할당합니다. 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오.

자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성

일반 시나리오에서는 Amazon VPC에 동일한 VPC에서 실행 중인 웹 서버와 데이터를 공유하는 Amazon RDS DB 인스턴스가 포함됩니다. 이 자습서에서는 이 시나리오를 위한 VPC를 생성합니다.

다음 다이어그램은 이 시나리오를 보여 줍니다. 다른 시나리오에 대한 자세한 내용은 [VPC에서 DB 인스턴스에 액세스하는 시나리오 \(p. 408\)](#)을(를) 참조하십시오.



공용 인터넷이 아닌 웹 서버에서만 Amazon RDS DB 인스턴스를 사용할 수 있어야 하므로 퍼블릭 및 프라이빗 서브넷이 모두 있는 VPC를 생성합니다. 퍼블릭 서브넷에서 웹 서버를 호스팅하므로 웹 서버에서 퍼블릭 인터넷에 액세스할 수 있습니다. Amazon RDS DB 인스턴스는 프라이빗 서브넷에서 호스팅됩니다. 동일한 VPC에서 호스팅되므로 웹 서버에서는 Amazon RDS DB 인스턴스에 연결할 수 있지만, 퍼블릭 인터넷에서 Amazon RDS DB 인스턴스에 액세스할 수 없어 보다 강화된 보안이 가능합니다.

프라이빗 서브넷과 퍼블릭 서브넷을 포함하는 VPC 생성

다음은 퍼블릭 서브넷과 프라이빗 서브넷을 모두 포함하는 VPC를 생성하는 절차입니다.

VPC 및 서브넷을 생성하는 방법

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. AWS Management 콘솔의 오른쪽 상단 모서리에서 VPC를 생성할 리전을 선택합니다. 이 예에서는 미국 서부(오레곤) 리전을 사용합니다.
3. 왼쪽 상단 모서리에서 [VPC Dashboard]를 선택합니다. VPC 생성을 시작하려면 VPC 마법사 시작을 선택합니다.
4. [Step 1: Select a VPC Configuration] 페이지에서 [VPC with Public and Private Subnets]를 선택한 후 [Select]를 선택합니다.
5. [Step 2: VPC with Public and Private Subnets] 페이지에서 다음 값을 설정합니다.
 - 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
 - 가용 영역: us-west-2a
 - Public subnet name: Tutorial public
 - Private subnet's IPv4 CIDR: 10.0.1.0/24
 - 가용 영역: us-west-2a
 - Private subnet name: Tutorial Private 1
 - Instance type: t2.small

Important

콘솔에 [Instance type] 상자가 표시되지 않으면 [Use a NAT instance instead]를 클릭합니다. 이 링크는 오른쪽에 있습니다.

Note

t2.small 인스턴스 유형이 표시되지 않으면 다른 인스턴스 유형을 선택할 수 있습니다.

- Key pair name: No key pair
 - Service endpoints: Skip this field.
 - Enable DNS hostnames: Yes
 - Hardware tenancy: Default
6. 작업을 마쳤으면 [Create VPC]를 선택합니다.

추가 서브넷 생성

RDS DB 인스턴스를 VPC에서 사용하기 위한 Amazon RDS DB 서브넷 그룹을 만들려면 두 개의 프라이빗 서브넷 또는 두 개의 퍼블릭 서브넷이 있어야 합니다. 이 자습서의 RDS DB 인스턴스는 프라이빗이므로 두 번째 프라이빗 서브넷을 VPC에 추가합니다.

API 버전 2014-10-31

서브넷을 추가로 생성하는 방법

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 두 번째 프라이빗 서브넷을 VPC에 추가하려면 VPC 대시보드를 선택하고 서브넷을 선택한 다음 서브넷 생성을 선택합니다.
3. 서브넷 생성 페이지에서 다음 값을 설정합니다.
 - Name tag: Tutorial private 2
 - VPC: 이전 단계에서 생성한 VPC를 선택합니다(예: vpc-*identifier* (10.0.0.0/16) | tutorial-vpc).
 - 가용 영역: us-west-2b

Note

첫 번째 프라이빗 서브넷에 대해 선택한 것과 다른 가용 영역을 선택합니다.

- IPv4 CIDR block: 10.0.2.0/24
4. 작업을 마쳤으면 생성을 선택합니다. 그런 다음 확인 페이지에서 닫기를 선택합니다.
 5. 두 번째로 생성한 프라이빗 서브넷이 첫 번째 프라이빗 서브넷과 동일한 라우팅 테이블을 사용하는지 확인하려면 [VPC Dashboard]를 선택하고 [Subnets]를 선택한 다음 VPC에 대해 생성한 첫 번째 프라이빗 서브넷인 Tutorial private 1을 선택합니다.
 6. 서브넷 목록 아래에서 Route Table(라우팅 테이블) 탭을 선택하고 Route Table(라우팅 테이블)의 값(예: rtb-98b613fd)을 기록합니다.
 7. 서브넷 목록에서 첫 번째 프라이빗 서브넷 선택을 해제합니다.
 8. 서브넷 목록에서 두 번째 프라이빗 서브넷 Tutorial private 2를 선택하고 [Route Table] 탭을 선택합니다.
 9. 현재 라우팅 테이블과 첫 번째 프라이빗 서브넷에 사용한 라우팅 테이블이 동일하지 않은 경우 라우팅 테이블 연결 편집을 선택합니다. 라우팅 테이블 ID에서 앞서 기록한 라우팅 테이블(예: rtb-98b613fd)을 선택합니다. 그런 다음 선택 사항을 저장하기 위해 저장을 선택합니다.

퍼블릭 웹 서버에 대해 VPC 보안 그룹 생성

이제 퍼블릭 액세스를 위한 보안 그룹을 생성합니다. VPC의 퍼블릭 인스턴스에 연결하려면 인터넷으로부터의 트래픽 연결을 허용하는 VPC 보안 그룹에 인바운드 규칙을 추가해야 합니다.

VPC 보안 그룹의 생성 방법

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. VPC 대시보드, 보안 그룹, 보안 그룹 생성을 차례대로 선택합니다.
3. 보안 그룹 생성 페이지에서 다음 값을 설정합니다.
 - 보안 그룹 이름: tutorial-securitygroup
 - 설명: Tutorial Security Group
 - VPC: 앞에서 생성한 VPC를 선택합니다(예: vpc-*identifier* (10.0.0.0/16) | tutorial-vpc).
4. 보안 그룹을 생성하려면 생성을 선택합니다. 그런 다음 확인 페이지에서 닫기를 선택합니다.

이 자습서에서 나중에 필요하므로 보안 그룹 ID를 적어둡니다.

인바운드 규칙을 보안 그룹에 추가하려면

1. VPC의 인스턴스에 연결할 때 사용할 IP 주소를 측정합니다. 퍼블릭 IP 주소를 확인하려면 <https://checkip.amazonaws.com>의 서비스를 사용합니다. IP 주소의 예는 203.0.113.25/32입니다.

고정 IP 주소가 없는 방화벽 뒤나 ISP(인터넷 서비스 제공업체)를 통해 연결되어 있는 경우 클라이언트 컴퓨터가 사용하는 IP 주소의 범위를 찾아야 합니다.

Warning

0.0.0.0/0을 사용하면 모든 IP 주소에서 퍼블릭 인스턴스에 액세스할 수 있습니다. 이 방법은 테스트 환경에서 잠시 사용하는 것은 괜찮지만 프로덕션 환경에서는 안전하지 않습니다. 프로덕션에서는 특정 IP 주소나 주소 범위만 인스턴스에 액세스하도록 허용하십시오.

2. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
3. [VPC Dashboard], [Security Groups]와 이전 절차에서 생성한 tutorial-securitygroup 보안 그룹을 차례로 선택합니다.
4. 보안 그룹 목록 아래에서 인바운드 규칙 탭을 선택한 후, 규칙 편집을 선택합니다.
5. 인바운드 규칙 편집 페이지에서 규칙 추가를 선택합니다.
6. 새로운 인바운드 규칙으로 다음 값을 설정하여 SSH(Secure Shell)에서 EC2 인스턴스에 액세스하도록 허용합니다. 이렇게 하면 EC2 인스턴스에 연결하여 웹 서버 및 다른 유필리티를 설치하고 웹 서버의 콘텐츠를 업로드할 수 있습니다.
 - Type: SSH
 - [Source:] 1단계의 IP 주소 또는 범위(예: 203.0.113.25/32)
7. [Add another rule]을 선택합니다.
8. 새 인바운드 규칙에 다음 값을 설정하여 웹 서버에 대한 HTTP 액세스를 허용합니다.
 - Type: HTTP
 - 소스: 0.0.0.0/0.
9. 설정을 저장하려면 규칙 저장을 선택합니다. 그런 다음 확인 페이지에서 닫기를 선택합니다.

프라이빗 Amazon RDS DB 인스턴스에 대한 VPC 보안 그룹 생성

Amazon RDS DB 인스턴스를 프라이빗으로 유지하려면 프라이빗 액세스를 위한 보조 보안 그룹을 생성합니다. VPC의 프라이빗 인스턴스에 연결하려면 웹 서버로부터의 트래픽만을 허용하는 VPC 보안 그룹에 인바운드 규칙을 추가해야 합니다.

VPC 보안 그룹의 생성 방법

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. VPC 대시보드, 보안 그룹, 보안 그룹 생성을 차례대로 선택합니다.
3. 보안 그룹 생성 페이지에서 다음 값을 설정합니다.
 - 보안 그룹 이름: tutorial-db-securitygroup
 - 설명: Tutorial DB Instance Security Group
 - VPC: 앞에서 생성한 VPC를 선택합니다(예: vpc-**identifier** (10.0.0.0/16) | tutorial-vpc).
4. 보안 그룹을 생성하려면 생성을 선택합니다. 그런 다음 확인 페이지에서 닫기를 선택합니다.

인바운드 규칙을 보안 그룹에 추가하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. [VPC Dashboard], [Security Groups]와 이전 절차에서 생성한 tutorial-db-securitygroup 보안 그룹을 차례로 선택합니다.
3. 보안 그룹 목록 아래에서 인바운드 규칙 탭을 선택한 후, 규칙 편집을 선택합니다.
4. 인바운드 규칙 편집 페이지에서 규칙 추가를 선택합니다.

5. 새로운 인바운드 규칙으로 다음 값을 설정하여 EC2 인스턴스의 포트 3306에서 MySQL 트래픽을 허용합니다. 이렇게 하면 웹 서버에서 DB 인스턴스에 연결하여 웹 애플리케이션의 데이터를 데이터베이스에 저장하고 검색할 수 있습니다.
 - Type: MySQL/Aurora
 - Source: 이 자습서의 앞부분에서 만든 tutorial-securitygroup 보안 그룹의 식별자(예: sg-9edd5cfb)
6. 설정을 저장하려면 규칙 저장을 선택합니다. 그런 다음 확인 페이지에서 닫기를 선택합니다.

DB 서브넷 그룹 만들기

DB 서브넷 그룹은 사용자가 VPC에서 만든 다음 DB 인스턴스에 대해 지정하는 서브넷의 모음입니다. DB 서브넷 그룹에서 DB 인스턴스를 생성할 때 특정 VPC를 지정할 수 있습니다.

DB 서브넷 그룹을 만드는 방법

1. <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Subnet groups]를 선택합니다.
3. [Create DB Subnet Group]을 선택합니다.
4. [Create DB subnet group] 페이지에서 [Subnet group details]에 이들 값을 설정합니다.
 - 이름: tutorial-db-subnet-group
 - 설명: Tutorial DB Subnet Group
 - VPC: tutorial-vpc (`vpc-identifier`)
5. 서브넷 추가 섹션에서 이 VPC와 관련된 모든 서브넷 추가를 선택합니다.
6. Create를 선택합니다.

새 DB 서브넷 그룹은 RDS 콘솔의 DB 서브넷 그룹 목록에 나타납니다. DB 서브넷 그룹을 클릭하면 창 하단에 있는 세부 정보 창에서 그룹과 연결된 모든 서브넷을 포함한 세부 정보를 확인할 수 있습니다.

Amazon RDS MariaDB

Amazon RDS는 여러 MariaDB 버전을 실행하는 DB 인스턴스를 지원합니다. 다음 메이저 버전을 사용할 수 있습니다.

- MariaDB 10.3
- MariaDB 10.2
- MariaDB 10.1
- MariaDB 10.0

マイ너 버전 지원에 대한 자세한 내용은 [Amazon RDS MariaDB 버전 \(p. 428\)](#) 단원을 참조하십시오.

사용자는 먼저 Amazon RDS 관리 도구 또는 인터페이스를 사용하여 Amazon RDS MariaDB DB 인스턴스를 생성합니다. 그런 다음 Amazon RDS 도구를 사용하여 DB 인스턴스에 대한 관리 작업(예: DB 인스턴스 다시 구성 또는 크기 조정, DB 인스턴스에 대한 연결 인증, 백업 또는 스냅샷 생성 및 복원, 다중 AZ 보조 생성, 읽기 복제본 생성 및 DB 인스턴스의 성능 모니터링)을 수행할 수 있습니다. 표준 MariaDB 유ти리티 및 애플리케이션을 사용하여 DB 인스턴스에서 데이터를 저장하고 데이터에 액세스할 수 있습니다.

MariaDB는 모든 AWS 리전에서 이용할 수 있습니다. AWS 리전에 대한 자세한 내용은 [리전 및 사용 영역 \(p. 99\)](#) 단원을 참조하십시오.

MariaDB 데이터베이스용 Amazon RDS를 사용하여 HIPAA 인증 애플리케이션을 개발할 수 있습니다. 예를 들어 AWS와 체결하여 이행 중인 비즈니스 제휴 계약(AWS)에 따라 보호 대상 건강 정보(PHI)를 비롯한 의료 관련 정보를 저장할 수 있습니다. 자세한 내용은 [HIPAA 규정 준수](#) 단원을 참조하십시오. AWS 범위 내 서비스는 외부 감사 기관의 철저한 평가를 거쳐 인증, 규정 준수 증명 또는 운영 권한(ATO)을 받았습니다. 자세한 내용은 [규정 준수 프로그램 제공 AWS 범위 내 서비스](#)를 참조하십시오.

첫 번째 DB 인스턴스를 생성하기 전에 이 설명서의 설정 섹션에 나오는 단계를 완료해야 합니다. 자세한 내용은 [Amazon RDS 설정 \(p. 5\)](#) 단원을 참조하십시오.

Amazon RDS MariaDB를 위한 공통 관리 작업

다음은 MariaDB를 실행하는 Amazon RDS DB 인스턴스로 수행하는 일반적인 관리 작업과 각 작업에 해당하는 설명서 링크입니다.

작업 영역	관련 문서
인스턴스 클래스, 스토리지 및 PIOPS	DB 인스턴스 클래스 선택 (p. 77) Amazon RDS 스토리지 유형 (p. 101)
프로덕션 목적으로 DB 인스턴스를 만들 경우에는 Amazon RDS에서 인스턴스 클래스, 스토리지 유형 및 프로비저닝된 IOPS이 작동하는 방식을 이해해야 합니다.	
다중 AZ 배포	Amazon RDS를 위한 고가용성(다중 AZ) (p. 109)
Amazon Virtual Private Cloud(VPC)	EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 (p. 407)
AWS 계정에 기본 VPC가 있는 경우에는 DB 인스턴스가 기본 VPC 내부에 자동으로 생성됩니다. 계정에 기본 VPC가 없는데 VPC 안에 DB	

작업 영역	관련 문서
인스턴스를 만들려면 VPC와 서브넷 그룹을 만든 후 DB 인스턴스를 만들어야 합니다.	VPC에서 Amazon RDS DB 인스턴스를 사용한 작업 (p. 415)
보안 그룹 기본적으로, DB 인스턴스와 함께 인스턴스에 대한 액세스를 막는 방화벽도 생성됩니다. 따라서 DB 인스턴스에 액세스하기 위한 알맞은 IP 주소와 네트워크 구성으로 보안 그룹을 만들어야 합니다. 만들어야 할 보안 그룹은 DB 인스턴스가 어떤 Amazon EC2 플랫폼에 있고, Amazon EC2 인스턴스에서 DB 인스턴스에 액세스하는지 여부에 따라 결정됩니다. 일반적으로 DB 인스턴스가 EC2-Classic 플랫폼에 있으면 DB 보안 그룹을 생성해야 하고, DB 인스턴스가 EC2-VPC 플랫폼에 있으면 VPC 보안 그룹을 생성해야 합니다.	EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 (p. 407) 보안 그룹을 통한 액세스 제어 (p. 390)
파라미터 그룹	DB 파라미터 그룹 작업 (p. 161)
DB 인스턴스에 특정 데이터베이스 파라미터가 필요할 경우, 파라미터 그룹을 만든 후 DB 인스턴스를 만들어야 합니다.	
데이터 가져오기 및 내보내기 데이터를 가져오거나 내보내기 위한 절차를 수립합니다.	MariaDB DB 인스턴스로 데이터 가져오기 (p. 466)
복제 읽기 전용 복제본을 생성하여 기본 MariaDB DB 인스턴스에서 읽기 트래픽을 오프로드할 수 있습니다.	읽기 전용 복제본 작업 (p. 135)
DB 인스턴스에 연결 표준 SQL 클라이언트 애플리케이션을 사용하여 DB 인스턴스에 연결합니다.	MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 (p. 445)
백업 및 복원 DB 인스턴스를 생성할 때 자동 백업을 하도록 구성할 수 있습니다. 또한 전체 백업 파일(.bak 파일)을 사용하여 데이터베이스를 수동으로 백업 및 복원할 수도 있습니다.	백업 작업 (p. 196)
모니터링 Amazon CloudWatch RDS 측정치, 이벤트 및 향상된 모니터링 기능을 통해 RDS MariaDB DB 인스턴스를 모니터링합니다. RDS MariaDB DB 인스턴스에 대한 로그 파일을 봅니다.	DB 인스턴스 측정치 보기 (p. 236) Amazon RDS 이벤트 보기 (p. 305)
로그 파일 MariaDB DB 인스턴스의 로그 파일에 액세스할 수 있습니다.	Amazon RDS 데이터베이스 로그 파일 (p. 306) MariaDB 데이터베이스 로그 파일 (p. 310)

MariaDB를 실행하는 DB 인스턴스 작업을 위한 고급 관리 작업도 있습니다. 자세한 내용은 다음 설명서를 참조하십시오.

- [MariaDB에 대한 파라미터 \(p. 470\)](#)

- Amazon RDS SQL의 MariaDB 참조 (p. 474)

Amazon RDS MariaDB 버전

MariaDB의 경우, 버전 번호는 버전 X.Y.Z로 구성됩니다. Amazon RDS 용어에서, X.Y는 메이저 버전을 나타내고 Z는 마이너 버전 번호를 나타냅니다. Amazon RDS 구현을 위해서, 메이저 버전 번호가 변경될 경우(예: 버전 10.0에서 10.1으로) 이를 메이저 버전 변경으로 간주합니다. 단지 마이너 버전 번호가 변경된 경우(예: 버전 10.0.17에서 10.0.24으로)에는 마이너 버전 변경으로 간주합니다.

Amazon RDS에서는 현재 다음과 같은 MariaDB 버전을 지원합니다.

메이저 버전	마이너 버전
MariaDB 10.3	<ul style="list-style-type: none">• 10.3.13(모든 AWS 리전에서 지원됨)• 10.3.8(모든 AWS 리전에서 지원됨)
MariaDB 10.2	<ul style="list-style-type: none">• 10.2.21(모든 AWS 리전에서 지원됨)• 10.2.15(모든 AWS 리전에서 지원됨)• 10.2.12(모든 AWS 리전에서 지원됨)• 10.2.11(모든 AWS 리전에서 지원됨)
MariaDB 10.1	<ul style="list-style-type: none">• 10.1.34(모든 AWS 리전에서 지원됨)• 10.1.31(모든 AWS 리전에서 지원됨)• 10.1.26(모든 AWS 리전에서 지원됨)• 10.1.23(모든 AWS 리전에서 지원됨)• 10.1.19(모든 AWS 리전에서 지원됨)• 10.1.14(us-east-2를 제외한 모든 AWS 리전에서 지원됨)
MariaDB 10.0	<ul style="list-style-type: none">• 10.0.35(모든 AWS 리전에서 지원됨)• 10.0.34(모든 AWS 리전에서 지원됨)• 10.0.32(모든 AWS 리전에서 지원됨)• 10.0.31(모든 AWS 리전에서 지원됨)• 10.0.28(모든 AWS 리전에서 지원됨)• 10.0.24(모든 AWS 리전에서 지원됨)• 10.0.17(us-east-2, ca-central-1, eu-west-2를 제외한 모든 AWS 리전에서 지원됨)

새 DB 인스턴스를 생성할 때는 현재 지원되는 모든 MariaDB 버전을 지정할 수 있습니다. MariaDB 10.2와 같이 메이저 버전과 지정된 메이저 버전에 대해 지원되는 모든 마이너 버전을 지정할 수 있습니다. 버전이 지정되지 않은 경우 Amazon RDS는 지원되는 버전(보통 최신 버전)을 기본값으로 설정합니다. 메이저 버전이 지정되었지만 마이너 버전이 지정되지 않은 경우, Amazon RDS는 고객이 지정한 메이저 버전의 최근 릴리스를 기본값으로 설정합니다. 지원되는 버전 목록과 새로 만든 DB 인스턴스의 기본값을 보려면 [describe-db-engine-versions](#) AWS CLI 명령을 사용합니다.

MariaDB에 대한 Amazon RDS 운영 중단 정책에 대한 자세한 내용은 [Amazon RDS FAQ](#) 단원을 참조하십시오.

Amazon RDS에서의 버전 및 기능 지원

Amazon RDS에서의 MariaDB 10.3 지원

Amazon RDS는 다음과 같은 MariaDB 10.3 버전을 지원합니다.

- 10.3.8(모든 AWS 리전에서 지원됨)

Amazon RDS는 MariaDB 버전 10.3 이상을 실행하는 DB 인스턴스에서 다음과 같은 새로운 기능을 지원합니다.

- Oracle 호환성 – PL/SQL 호환성 파서, 시퀀스, UNION을 보완하는 INTERSECT 및 EXCEPT, 새 TYPE OF 및 ROW TYPE OF 선언 및 표시되지 않는 열
- 임시 데이터 처리 – 데이터베이스의 과거 및 현재 상태를 쿼리할 수 있는 시스템 버전 테이블.
- 유연성 – 사용자 정의 집계, 스토리지 독립적인 열 압축, 프록시가 클라이언트 IP 주소를 서버로 릴레이할 수 있도록 프록시 프로토콜 지원.
- 관리 효율성 – 즉각적인 ADD COLUMN 작업, 빠른 실패 데이터 정의 언어(DDL) 작업

MariaDB 10.3 전체 기능 목록과 설명서는 MariaDB 웹사이트에서 [Changes & Improvements in MariaDB 10.3](#)와 [Release Notes - MariaDB 10.3 Series](#)를 참조하십시오.

지원되지 않는 기능 목록은 [지원되지 않는 기능 \(p. 430\)](#) 단원을 참조하십시오.

Amazon RDS에서의 MariaDB 10.2 지원

Amazon RDS는 다음과 같은 MariaDB 10.2 버전을 지원합니다.

- 10.2.15(모든 AWS 리전에서 지원됨)
- 10.2.12(모든 AWS 리전에서 지원됨)
- 10.2.11(모든 AWS 리전에서 지원됨)

Amazon RDS는 MariaDB 버전 10.2 이상을 실행하는 DB 인스턴스에서 다음과 같은 새로운 기능을 지원합니다.

- ALTER USER
- 공통 테이블 표현식
- 이진 로그 크기 축소를 위한 이벤트 압축
- CREATE USER — 리소스 사용량 및 TLS/SSL 제한을 위한 새로운 옵션
- EXECUTE IMMEDIATE
- 플래시백
- InnoDB — XtraDB를 대신하는 기본 스토리지 엔진
- InnoDB — 버퍼풀 크기를 동적으로 설정
- JSON 함수
- 창 함수
- WITH

MariaDB 10.2 전체 기능 목록과 설명서는 MariaDB 웹사이트에서 [Changes & Improvements in MariaDB 10.2](#)와 [Release Notes - MariaDB 10.2 Series](#)를 참조하십시오.

지원되지 않는 기능 목록은 [지원되지 않는 기능 \(p. 430\)](#) 단원을 참조하십시오.

Amazon RDS에서의 MariaDB 10.1 지원

Amazon RDS는 다음과 같은 MariaDB 10.1 버전을 지원합니다.

- 10.1.34(모든 AWS 리전에서 지원됨)
- 10.1.31(모든 AWS 리전에서 지원됨)
- 10.1.26(모든 AWS 리전에서 지원됨)
- 10.1.23(모든 AWS 리전에서 지원됨)
- 10.1.19(모든 AWS 리전에서 지원됨)
- 10.1.14(us-east-2를 제외한 모든 AWS 리전에서 지원됨)

Amazon RDS는 MariaDB 버전 10.1 이상을 실행하는 DB 인스턴스에서 다음과 같은 새로운 기능을 지원합니다.

- 순서에 따른 낙관적 병렬 복제
- 페이지 압축
- XtraDB 데이터 스크러빙 및 조각 모음

MariaDB 10.1 전체 기능 목록과 설명서는 MariaDB 웹사이트에서 [Changes & Improvements in MariaDB 10.1](#)과 [Release Notes - MariaDB 10.1 Series](#)를 참조하십시오.

지원되지 않는 기능 목록은 [지원되지 않는 기능 \(p. 430\)](#) 단원을 참조하십시오.

Amazon RDS에서의 MariaDB 10.0 지원

Amazon RDS는 다음과 같은 MariaDB 10.0 버전을 지원합니다.

- 10.0.35(모든 AWS 리전에서 지원됨)
- 10.0.34(모든 AWS 리전에서 지원됨)
- 10.0.32(모든 AWS 리전에서 지원됨)
- 10.0.31(모든 AWS 리전에서 지원됨)
- 10.0.28(모든 AWS 리전에서 지원됨)
- 10.0.24(모든 AWS 리전에서 지원됨)
- 10.0.17(us-east-2, ca-central-1, eu-west-2를 제외한 모든 AWS 리전에서 지원됨)

MariaDB 10.0 전체 기능 목록과 설명서는 MariaDB 웹사이트에서 [Changes & Improvements in MariaDB 10.0](#)과 [Release Notes - MariaDB 10.0 Series](#)를 참조하십시오.

지원되지 않는 기능 목록은 [지원되지 않는 기능 \(p. 430\)](#) 단원을 참조하십시오.

지원되지 않는 기능

다음 MariaDB 기능은 Amazon RDS에서 지원되지 않습니다.

- 인증 플러그인 – GSSAPI
- 인증 플러그인 – Unix 소켓
- AWS 키 관리 암호화 플러그인
- 지연 복제
- XtraDB, InnoDB, Aria에 대한 기본 유형 시 MariaDB 암호화.

Amazon RDS 리소스 암호화 (p. 385)의 지침에 따라 MariaDB DB 인스턴스에 유 휴 시 암호화를 활성화할 수 있습니다.

- HandlerSocket
- JSON 테이블 유형
- MariaDB ColumnStore
- MariaDB Galera Cluster
- 멀티 소스 복제
- MyRocks 스토리지 엔진
- 암호 확인 플러그인, `simple_password_check` 및 `cracklib_password_check`
- 복제 필터
- 스파이더 스토리지 엔진
- Sphinx 스토리지 엔진
- TokuDB 스토리지 엔진
- 스토리지 엔진별 객체 속성(MariaDB 설명서의 [Engine-defined New Table/Field/Index Attributes](#)에서 설명).
- 테이블 및 tablespace 암호화

관리되는 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 셀 액세스를 제공하지 않으며, 고급 권한을 필요로 하는 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다. Amazon RDS는 표준 SQL 클라이언트 애플리케이션을 사용하여 DB 인스턴스의 데이터베이스에 대한 액세스를 지원합니다. Amazon RDS는 Telnet, Secure Shell (SSH), 또는 Windows 원격 데스크톱 연결을 사용하여 DB 인스턴스에 직접 호스트 액세스하는 것을 허용하지 않습니다.

Amazon RDS MariaDB에 대해 지원되는 스토리지 엔진

MariaDB는 다양한 기능으로 여러 스토리지 엔진을 지원하지만, 모든 기능이 복구와 데이터 내구성에 최적화되어 있지는 않습니다. InnoDB(버전 10.2 이상) 및 XtraDB(버전 10.0 및 10.1)는 Amazon RDS의 MariaDB DB 인스턴스에 대한 권장 및 지원되는 스토리지 엔진입니다. 특정 시점으로 복원 및 스냅샷 복원 기능 등 Amazon RDS 기능을 사용하려면 복구 가능 스토리지 엔진이 필요하며, 이러한 기능은 MariaDB 버전용 권장 스토리지 엔진에만 지원됩니다. Amazon RDS는 Aria도 지원합니다. 그러나 Aria를 사용하면 인스턴스 장애 발생 시 복구에 부정적인 영향을 미칠 수 있습니다. 단, MariaDB 10.1 또는 10.0에서 자리 데이터를 처리하기 위해 공간 인덱스를 사용해야 하는 경우에는 Aria를 사용해야 합니다. XtraDB에서는 공간 인덱스가 지원되지 않기 때문입니다. MariaDB 10.2 이상에서는 InnoDB 스토리지 엔진이 공간 인덱스를 지원합니다.

현재 다른 스토리지 엔진은 MariaDB용 Amazon RDS에서 지원되지 않습니다.

Amazon RDS MariaDB 보안

Amazon RDS MariaDB DB 인스턴스용 보안은 세 가지 수준에서 관리됩니다.

- AWS Identity and Access Management은 누가 DB 인스턴스에 대한 Amazon RDS 관리 작업을 수행할 수 있는지 제어합니다. IAM 자격 증명을 사용하여 AWS에 연결할 때, IAM 계정은 Amazon RDS 관리 작업을 수행하는 데 필요한 권한을 부여하는 IAM 정책을 보유하고 있어야 합니다. 자세한 내용은 [인증 및 액세스 제어 \(p. 339\)](#) 단원을 참조하십시오.
- DB 인스턴스를 생성할 때는 VPC 보안 그룹 또는 DB 보안 그룹을 사용하여 어떤 디바이스 및 Amazon EC2 인스턴스가 DB 인스턴스의 앤드포인트 및 포트에 대한 연결을 열 수 있는지를 제어합니다. 이러한

연결은 Secure Socket Layer(SSL)를 이용해 이루어집니다. 또한 회사의 방화벽 규칙을 통해 회사에서 실행하는 디바이스가 DB 인스턴스에 대한 연결을 열 수 있는지를 제어할 수 있습니다.

- 일단 MariaDB DB 인스턴스에 대한 연결이 활성화되면, 로그인 및 권한에 대한 인증은 MariaDB의 독립 실행형 인스턴스에서와 동일한 방식으로 적용됩니다. CREATE USER, RENAME USER, GRANT, REVOKE 및 SET PASSWORD 등의 명령은 독립 실행형 데이터베이스에서 작동하는 것과 마찬가지로 작동하며, 데이터베이스 스키마 테이블을 직접 수정할 때도 동일합니다.

Amazon RDS DB 인스턴스를 생성할 때 마스터 사용자는 다음과 같은 기본 권한을 갖습니다.

- 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

Amazon RDS MariaDB DB 인스턴스에서 이 권한은 제한됩니다. FLUSH LOGS 또는 FLUSH TABLES WITH READ LOCK 작업에는 액세스할 수 없습니다.

- replication client
- replication slave
- select
- show databases
- show view
- trigger
- update

권한에 대한 자세한 내용은 MariaDB 설명서에서 [User Account Management](#)를 참조하십시오.

Note

DB 인스턴스에서 마스터 사용자를 삭제할 수는 있지만, 마스터 사용자를 삭제하지 않을 것을 권장합니다. 마스터 사용자를 다시 생성하려면 `ModifyDBInstance` API 또는 `modify-db-instance` AWS 명령줄 도구를 사용하고 적절한 파라미터와 함께 새 마스터 사용자 암호를 지정합니다. 마스터 사용자가 인스턴스에 존재하지 않는 경우 마스터 사용자가 지정된 암호와 함께 생성됩니다.

각 DB 인스턴스에 관리 서비스를 제공하기 위해 DB 인스턴스가 생성될 때 `rdsadmin` 사용자가 만들어집니다. `rdsadmin` 계정을 삭제하려고 하거나, 계정 이름 또는 암호를 변경하려고 하거나, 계정 권한을 변경하고 하면 오류가 발생합니다.

DB 인스턴스의 관리를 허용하기 위해 표준 kill 및 kill_query 명령이 제한되었습니다. MariaDB 및 MySQL에서는 DB 인스턴스에서 사용자 세션 또는 쿼리를 종료할 수 있도록 Amazon RDS 명령 mysql.rds_kill, mysql.rds_kill_query 및 mysql.rds_kill_query_id가 제공됩니다.

MariaDB DB 인스턴스와 함께 SSL 사용

Amazon RDS는 MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스와의 Secure Sockets Layer(SSL) 연결을 지원합니다.

Amazon RDS가 SSL 인증서를 생성한 후 Amazon RDS가 인스턴스를 프로비저닝할 때 DB 인스턴스에 인증서를 설치합니다. 인증 기관이 서명하는 SSL 인증서에는 스포핑 공격으로부터 보호해주는 SSL 인증서를 위한 일반 이름(CN)으로 DB 인스턴스 앤드포인트가 포함되어 있습니다.

퍼블릭 키는 <https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem>에 저장되어 있습니다.

MariaDB는 다음과 같은 버전에서 보안 연결을 위해 yaSSL을 사용합니다.

- MariaDB 버전 10.1.26 및 10.1 이전 버전
- MariaDB 버전 10.0.32 및 10.0 이전 버전

MariaDB는 다음과 같은 버전에서 보안 연결을 위해 OpenSSL을 사용합니다.

- MariaDB 10.3 버전
- MariaDB 10.2 버전
- MariaDB 버전 10.1.31 및 10.1 이후 버전
- MariaDB 버전 10.0.34 및 10.0 이후 버전

MariaDB용 Amazon RDS는 TLS(전송 계층 보안) 버전 1.0, 1.1 및 1.2를 지원합니다. 다음 표는 MySQL 버전에 대한 TLS 지원을 보여줍니다.

MariaDB 버전	TLS 1.0	TLS 1.1	TLS 1.2
MariaDB 10.3	지원	지원	지원
MariaDB 10.2	지원	지원	지원
MariaDB 10.1	지원	10.1.31 및 10.1 이후 버전에 지원됨	10.1.31 및 10.1 이후 버전에 지원됨
MariaDB 10.0	지원	10.0.34 및 10.0 이후 버전에 지원됨	10.0.34 및 10.0 이후 버전에 지원됨

기본 mysql 클라이언트를 사용하여 연결을 암호화하려면 --ssl-ca 파라미터를 사용하여 mysql 클라이언트를 시작하고 다음 예제와 같이 퍼블릭 키를 참조하십시오.

다음 예제는 MariaDB 10.2 이후 버전의 경우 --ssl-ca 파라미터를 사용하여 클라이언트를 시작하는 방법을 보여줍니다.

```
mysql -h myinstance.c9akciq32.rds-us-east-1.amazonaws.com
--ssl-ca=[full path]rds-combined-ca-bundle.pem --ssl-mode=REQUIRED
```

다음 예제는 MariaDB 10.1 및 이전 버전의 경우 --ssl-ca 파라미터를 사용하여 클라이언트를 시작하는 방법을 보여줍니다.

```
mysql -h myinstance.c9akciq32.rds-us-east-1.amazonaws.com  
--ssl-ca=[full path]rds-combined-ca-bundle.pem --ssl-verify-server-cert
```

특정 사용자 계정에 대한 SSL 연결을 요구할 수 있습니다. 예를 들면 MariaDB 버전에 따라 다음 문 중 하나를 사용하여 사용자 계정 `encrypted_user`에 대한 SSL 연결을 요구할 수 있습니다.

MariaDB 10.2 및 이후 버전에서는 다음 문을 사용하십시오.

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

MariaDB 10.1 및 이전 버전에서는 다음 문을 사용하십시오.

```
GRANT USAGE ON *.* TO 'encrypted_user'@'%' REQUIRE SSL;
```

MariaDB와의 SSL 연결에 대한 자세한 내용은 MariaDB 설명서의 [SSL 개요](#) 단원을 참조하십시오.

캐시 워밍

InnoDB(버전 10.2 이상) 및 XtraDB(버전 10.0 및 10.1) 캐시 워밍은 DB 인스턴스가 종료될 때 버퍼 풀의 현재 상태를 저장한 다음 DB 인스턴스가 시작될 때 저장된 정보에서 버퍼 풀을 다시 로드하여 MariaDB DB 인스턴스의 성능 향상을 제공할 수 있습니다. 이렇게 하면 보통 데이터베이스 사용에서 "준비"까지의 버퍼 풀에 대한 필요를 무시하고, 대신 알려진 공용 쿼리에 대한 페이지와 함께 버퍼 풀을 미리 로드합니다. 캐시 워밍에 대한 자세한 내용은 MariaDB 설명서에서 [Dumping and restoring the buffer pool](#)을 참조하십시오.

캐시 워밍은 MariaDB 10.2 이상 DB 인스턴스에서 기본적으로 활성화됩니다. 캐시 워밍을 활성화하려면 DB 인스턴스의 파라미터 그룹에서 `innodb_buffer_pool_dump_at_shutdown` 및 `innodb_buffer_pool_load_at_startup` 파라미터를 1로 설정합니다. 파라미터 그룹에서 이들 파라미터 값을 변경하면 파라미터 그룹을 사용하는 모든 MariaDB DB 인스턴스가 영향을 받습니다. 특정 MariaDB DB 인스턴스에 대해 캐시 워밍을 활성화하려면, 이들 DB 인스턴스에 대한 새 파라미터 그룹을 생성해야 할 수도 있습니다. 파라미터 그룹에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

캐시 워밍은 주로 표준 스토리지를 사용하는 DB 인스턴스를 위해 성능 혜택을 제공합니다. PIOPS 스토리지 사용하는 경우에는 통상적으로 성능 혜택이 현저하지 않습니다.

Important

MariaDB DB 인스턴스가 정상적으로 종료되지 않는 경우(예: 장애 조치 도중), 버퍼 풀 상태가 디스크에 저장되지 않습니다. 이 경우 MariaDB는 DB 인스턴스가 다시 시작될 때 이용 가능한 모든 버퍼 풀 파일을 로드합니다. 어떤 손상도 발생하지 않지만, 복원된 버퍼 풀은 대부분의 경우 다시 시작하기 이전의 버퍼 풀 최신 상태를 반영하지 못할 수도 있습니다. 시작 시 캐시를 워밍하기 위해 버퍼 풀의 최신 상태를 이용할 수 있게 하려면, "요청 시" 버퍼 풀을 주기적으로 덤프하는 것이 좋습니다. 사용자가 요청 시 버퍼 풀을 덤프 또는 로드할 수 있습니다.

버퍼 풀을 자동으로 그리고 정기적으로 덤프하는 이벤트를 생성할 수 있습니다. 예를 들면, 다음 문은 매 시간마다 버퍼 풀을 덤프하는 이름이 `periodic_buffer_pool_dump`인 이벤트를 생성합니다.

```
CREATE EVENT periodic_buffer_pool_dump  
ON SCHEDULE EVERY 1 HOUR  
DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

자세한 내용은 MariaDB 설명서에서 [Events](#)를 참조하십시오.

요청 시 버퍼 풀 덤프 및 로딩

다음의 저장 프로시저를 사용해 요청 시 캐시를 저장하고 로드할 수 있습니다.

- 버퍼 풀의 현재 상태를 디스크에 덤프하려면 [mysql.rds_innodb_buffer_pool_dump_now \(p. 694\)](#) 저장 절차를 호출합니다.
- 디스크에서 저장된 버퍼 풀의 상태를 로드하려면 [mysql.rds_innodb_buffer_pool_load_now \(p. 695\)](#) 저장 절차를 호출합니다.
- 진행 중인 로드 작업을 취소하려면 [mysql.rds_innodb_buffer_pool_load_abort \(p. 695\)](#) 저장 절차를 호출합니다.

MariaDB 데이터베이스 파라미터

기본적으로, MariaDB DB 인스턴스는 MariaDB 데이터베이스에만 해당되는 DB 파라미터 그룹을 사용합니다. 이 파라미터 그룹에는 MySQL 데이터베이스 엔진용 Amazon RDS DB 파라미터 그룹의 파라미터 중 일부만 포함되어 있습니다. 또한 몇 개의 새로운 MariaDB 고유 파라미터도 포함되어 있습니다. Amazon RDS MariaDB DB 엔진에서 사용할 수 있는 파라미터에 대한 자세한 내용은 [MariaDB에 대한 파라미터 \(p. 470\)](#) 단원을 참조하십시오.

MariaDB를 위한 공통 DBA 작업

세션 또는 쿼리 종료, 복제 오류 넘어가기, 층들 복구 시간 개선을 위한 InnoDB(버전 10.2 이상) 및 XtraDB(버전 10.0 및 10.1) 테이블스페이스 작업 및 전역적 상태 이력 관리는 MariaDB DB 인스턴스에서 수행할 수 있는 공통 DBA 작업입니다. 이러한 작업을 [MySQL DB 인스턴스에 대한 공통 DBA 작업 \(p. 669\)](#) 단원에 설명된 대로 Amazon RDS MySQL DB 인스턴스에서 동일하게 처리할 수 있습니다. 여기의 층들 복구 지침은 MySQL InnoDB 엔진을 기준으로 하지만, InnoDB 또는 XtraDB를 실행하는 MariaDB 인스턴스에도 적용할 수 있습니다.

MariaDB DB 인스턴스의 현지 시간대

기본적으로 RDS MariaDB DB 인스턴스의 시간대는 협정 세계시(UTC)입니다. 대신 DB 인스턴스의 시간대를 애플리케이션의 현지 시간대로 설정할 수 있습니다.

DB 인스턴스의 현지 시간대를 설정하려면 DB 인스턴스의 파라미터 그룹에서 `time_zone` 파라미터를 이 섹션의 뒤에 나오는 지원되는 값 중 하나로 설정합니다. 파라미터 그룹에 대한 `time_zone` 파라미터를 설정하면 해당 파라미터 그룹을 사용 중인 모든 DB 인스턴스와 읽기 전용 복제본이 새로운 현지 시간대를 사용하도록 변경됩니다. 파라미터 그룹에서 파라미터를 설정하는 방법에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

현지 시간대를 설정하면 데이터베이스에 대한 모든 새 연결에 변경 사항이 반영됩니다. 현지 시간대를 변경할 때 데이터베이스에 대해 열린 연결이 있는 경우 연결을 닫고 새 연결을 열어야 현지 시간대 업데이트가 표시됩니다.

DB 인스턴스와 하나 이상의 읽기 전용 복제본에 대해 다른 현지 시간대를 설정할 수 있습니다. 이렇게 하려면 DB 인스턴스와 복제본에 대해 서로 다른 파라미터 그룹을 사용하고 각 파라미터 그룹에서 `time_zone` 파라미터를 다른 현지 시간대로 설정합니다.

리전 간 복제를 사용 중인 경우 복제 마스터 DB 인스턴스와 읽기 전용 복제본이 서로 다른 파라미터 그룹을 사용합니다. 파라미터 그룹은 리전에 고유합니다. 각 인스턴스에 대해 동일한 현지 시간대를 사용하려면 인

스턴스의 파라미터 그룹과 읽기 전용 복제본의 파라미터 그룹에서 `time_zone` 파라미터를 설정해야 합니다.

DB 스냅샷에서 DB 인스턴스를 복원할 경우 현지 시간대가 UTC로 설정됩니다. 복원이 완료된 후 시간대를 현지 시간대로 업데이트할 수 있습니다. DB 인스턴스를 특정 시점으로 복원할 경우 복원된 DB 인스턴스의 현지 시간대는 복원된 DB 인스턴스의 파라미터 그룹에서 설정한 시간대입니다.

현지 시간대를 다음 값 중 하나로 설정할 수 있습니다.

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

MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스 생성

Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. DB 인스턴스란 MariaDB 데이터베이스가 생성되는 곳을 말합니다.

Important

DB 인스턴스를 생성하거나 DB 인스턴스에 연결하려면 먼저 [Amazon RDS 설정 \(p. 5\)](#) 섹션의 작업을 완료해야 합니다.

사용자가 샘플 DB 인스턴스를 만들어 연결하는 절차를 실습하는 예제는 [MariaDB DB 인스턴스 생성 및 MariaDB DB 인스턴스의 데이터베이스에 연결 \(p. 10\)](#) 단원을 참조하십시오.

AWS Management 콘솔

MariaDB DB 인스턴스를 시작하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. AWS Management 콘솔의 오른쪽 위 모서리에서 DB 인스턴스를 만들 AWS 리전을 선택합니다.
3. 탐색 창에서 데이터베이스를 선택합니다.
탐색 창이 닫혀 있는 경우 왼쪽 상단의 메뉴 아이콘을 선택하여 여십시오.
4. 데이터베이스 생성을 선택하여 엔진 선택 페이지를 여십시오.

Select engine

Engine options

Amazon Aurora
Amazon Aurora

MySQL


MariaDB


PostgreSQL


Oracle


Microsoft SQL Server


MariaDB
MariaDB Community Edition is a MySQL-compatible database with strong support from the open source community, and extra features and performance optimizations.

- Supports database size up to 16 TB.
- Instances offer up to 32 vCPUs and 244 GiB Memory.
- Supports automated backup and point-in-time recovery.
- Supports cross-region read replicas.
- Supports global transaction ID (GTID) and thread pooling.
- Developed and supported by the MariaDB open source community.

Only enable options eligible for RDS Free Usage Tier [info](#)

[Cancel](#) **Next**

- [MariaDB]를 선택한 후 [Next]를 선택합니다.
- [Choose use case] 페이지에 생성하려는 DB 인스턴스를 프로덕션에 사용할지 묻는 메시지가 나타납니다. 사용할 계획이라면 [Production - MariaDB]를 선택합니다. 프로덕션 - MariaDB를 선택하면 다음 단계에서 다음 설정이 미리 선택됩니다.
 - 다중 AZ 장애 조치 옵션
 - 프로비저닝된 IOPS 스토리지 옵션
 - 삭제 방지 활성화 옵션

어떤 프로덕션 환경이든 이 기능을 권장합니다.

- 다음을 선택하여 계속 진행합니다. [Specify DB details] 페이지가 나타납니다.

DB 세부 정보 지정 페이지에서 DB 인스턴스 정보를 지정합니다. 각 설정에 대한 자세한 내용은 [MariaDB DB 인스턴스 설정 \(p. 442\)](#) 단원을 참조하십시오.

Specify DB details

Instance specifications

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

DB engine
MariaDB Community Edition

License model [Info](#)

DB engine version [Info](#)

DB instance class [Info](#)

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

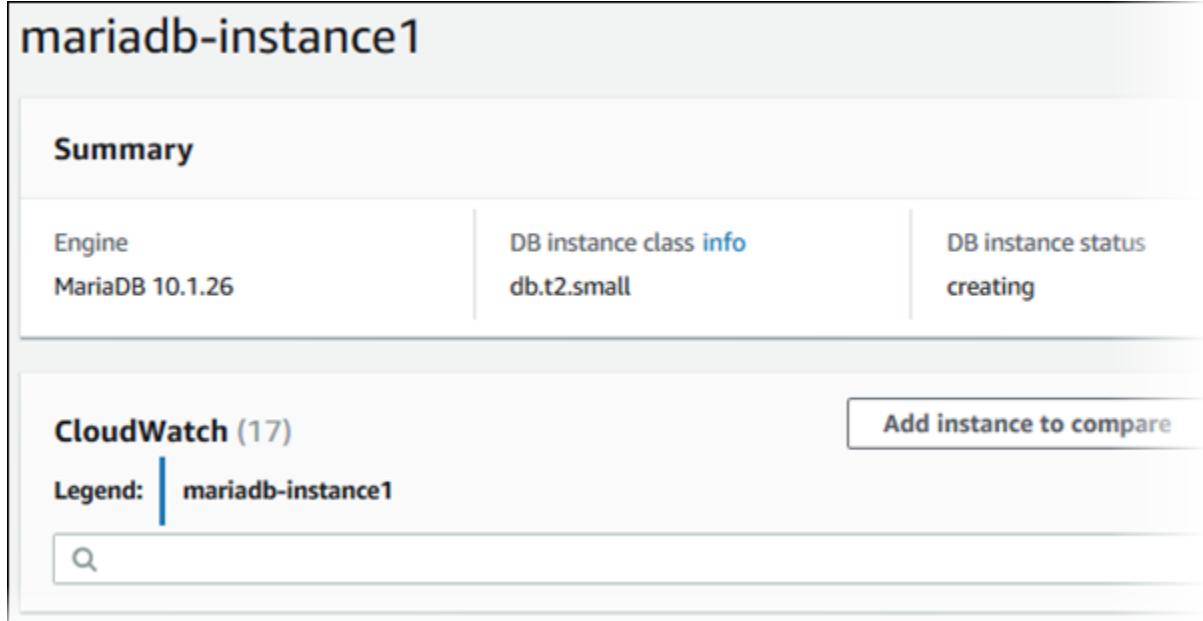
Storage type [Info](#)

8. 다음을 선택하여 계속 진행합니다.

Configure Advanced Settings(고급 설정 구성) 페이지에서 Amazon RDS가 DB 인스턴스를 시작하는 데 필요한 추가 정보를 입력합니다. 각 설정에 대한 자세한 내용은 [MySQL DB 인스턴스 설정 \(p. 591\)](#) 단원을 참조하십시오.

9. 데이터베이스 생성을 선택합니다.
10. 마지막 페이지에서 DB 인스턴스 세부 정보 보기를 선택합니다.

RDS 콘솔에 새 DB 인스턴스의 세부 정보가 표시됩니다. DB 인스턴스를 만들고 사용할 준비가 될 때까지 DB 인스턴스의 상태는 [creating]입니다. 상태가 available로 변경되면 DB 인스턴스에 연결할 수 있습니다. DB 인스턴스 클래스와 할당된 스토리지에 따라 새 인스턴스를 사용할 수 있을 때까지 몇 분 정도 걸릴 수 있습니다.



CLI

AWS CLI를 사용해 MariaDB DB 인스턴스를 생성하려면 다음 파라미터와 함께 [create-db-instance](#) 명령을 호출합니다. 각 설정에 대한 자세한 내용은 [MariaDB DB 인스턴스 설정 \(p. 442\)](#) 단원을 참조하십시오.

- `--db-instance-identifier`
- `--db-instance-class`
- `--db-security-groups`
- `--db-subnet-group`
- `--engine`
- `--master-user-name`
- `--master-user-password`
- `--allocated-storage`
- `--backup-retention-period`

Note

MariaDB의 특정 마이너 버전이 필요한 경우 `--engine-version` 파라미터를 포함합니다.

Example

다음 명령은 mydbinstance라는 MariaDB 인스턴스를 생성합니다.

Linux, OS X, Unix의 경우:

```
aws rds create-db-instance \
--db-instance-identifier mydbinstance \
--db-instance-class db.m4.xlarge \
--engine mariadb \
--allocated-storage 20 \
--master-username masteruser \
--master-user-password masteruserpassword \
```

```
--backup-retention-period 3
```

Windows의 경우:

```
aws rds create-db-instance ^
--db-instance-identifier mydbinstance ^
--db-instance-class db.m4.xlarge ^
--engine mariadb ^
--allocated-storage 20 ^
--master-username masteruser ^
--master-user-password masteruserpassword ^
--backup-retention-period 3
```

이 명령은 다음과 유사한 정보로 시작하는 출력을 생성해야 합니다.

```
DBINSTANCE 20 True 3 rds-ca-2015 False arn:aws:rds:us-east-1:1234567890:db:mydbinstance
db.m4.xlarge mydbinstance creating 0 **** mariadb 10.1.26
```

API

Amazon RDS API를 사용해 MariaDB DB 인스턴스를 생성하려면 다음 파라미터와 함께 [CreateDBInstance](#) 작업을 호출합니다. 각 설정에 대한 자세한 내용은 [MariaDB DB 인스턴스 설정 \(p. 442\)](#) 단원을 참조하십시오.

- AllocatedStorage
- BackupRetentionPeriod
- DBInstanceClass
- DBInstanceIdentifier
- DBSecurityGroups
- DBSubnetGroup
- Engine
- MasterUsername
- MasterUserPassword

Note

MariaDB의 특정 마이너 버전이 필요한 경우 EngineVersion 파라미터를 포함합니다.

Example

```
https://rds.us-west-2.amazonaws.com/
?Action=CreateDBInstance
&AllocatedStorage=20
&BackupRetentionPeriod=3
&DBInstanceClass=db.m4.xlarge
&DBInstanceIdentifier=mydbinstance
&DBName=mydatabase
&DBSecurityGroups.member.1=mysecuritygroup
&DBSubnetGroup=mydbsubnetgroup
&Engine=mariadb
&MasterUserPassword=masteruserpassword
&MasterUsername=masterawsuser
&Version=2014-10-31
&X-Amz-Algorithm=AWS4-HMAC-SHA256
```

```
&X-Amz-Credential=AKIADQKE4SARGYLE/20140213/us-west-2/rds/aws4_request
&X-Amz-Date=20140213T162136Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=8052a76dfb18469393c5f0182cdab0ebc224a9c7c5c949155376c1c250fc7ec3
```

MariaDB DB 인스턴스 설정

다음 표에는 Maria DB 인스턴스를 생성할 때 선택하는 설정에 대한 세부 정보가 나와 있습니다.

설정	설정 설명
할당된 스토리지	DB 인스턴스용으로 할당할 스토리지(단위: GB)입니다. 경우에 따라 DB 인스턴스에 대해 데이터베이스의 크기보다 많은 양의 스토리지를 할당하면 I/O 성능을 개선할 수 있습니다. 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.
마이너 버전 자동 업그레이드	기본 마이너 DB 엔진 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다. Amazon RDS는 유지 관리 기간에 마이너 버전 자동 업그레이드를 수행합니다.
[Availability zone]	DB 인스턴스의 가용 영역입니다. 가용 영역을 지정하지 않으면 기본값으로 No Preference를 사용합니다. 자세한 내용은 리전 및 가용 영역 (p. 99) 단원을 참조하십시오.
백업 보존 기간	DB 인스턴스의 자동 백업을 보존할 기간(단위: 일)입니다. 중요한 DB 인스턴스의 경우 이 값을 1 이상으로 설정해야 합니다. 자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.
백업 기간	Amazon RDS가 자동으로 DB 인스턴스를 백업하는 기간입니다. 데이터베이스를 백업할 특정 시간을 지정하지 않으려면 기본값으로 No Preference를 사용합니다. 자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.
스냅샷으로 태그 복사	스냅샷을 생성할 때 DB 인스턴스 태그를 DB 스냅샷에 복사하려면 이 옵션을 선택합니다. 자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.
데이터베이스 이름	DB 인스턴스에 있는 데이터베이스의 이름입니다. 데이터베이스 이름은 1~64자의 영숫자 문자로 구성되어야 합니다. 이름을 제공하지 않으면 생성 중인 DB 인스턴스에 Amazon RDS가 데이터베이스를 생성하지 않습니다. DB 인스턴스에서 추가 데이터베이스를 생성하려면, DB 인스턴스에 연결한 다음 SQL 명령 CREATE DATABASE를 사용하십시오. 자세한 내용은 MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 (p. 445) 단원을 참조하십시오.
데이터베이스 포트	DB 인스턴스에 액세스하는 데 사용할 포트입니다. MariaDB 설치의 기본값은 포트 3306로 설정됩니다. DB 보안 그룹을 DB

설정	설정 설명
	<p>인스턴스에 사용할 경우 DB 보안 그룹을 생성할 때 제공한 포트 값과 같아야 합니다.</p> <p>일부 기업에서는 방화벽이 기본 MariaDB 포트 연결을 차단하는 경우도 있습니다. 이처럼 기업 방화벽이 기본 포트를 차단할 경우 DB 인스턴스에 다른 포트를 선택해야 합니다.</p>
삭제 방지	<p>DB 인스턴스가 삭제되지 않도록 방지하려면, 삭제 방지를 활성화합니다. AWS Management 콘솔을 사용하여 프로덕션 DB 인스턴스를 생성할 경우 기본적으로 삭제 방지가 활성화됩니다. 자세한 내용은 DB 인스턴스 삭제 (p. 128) 단원을 참조하십시오.</p>
DB 엔진 버전	사용할 MariaDB 버전입니다.
DB 인스턴스 클래스	<p>DB 인스턴스에 대한 구성입니다.</p> <p>가능하면 일반 쿼리 작업 세트가 메모리에 상주할 수 있을 정도로 큰 인스턴스 클래스를 선택합니다. 작업 세트가 메모리에 상주할 경우 시스템의 디스크 쓰기가 불필요하여 성능이 향상됩니다.</p> <p>자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 단원을 참조하십시오.</p>
DB 인스턴스 식별자	DB 인스턴스의 이름입니다. DB 인스턴스 식별자는 최대 63자의 영문자 문자를 포함할 수 있으며 선택한 AWS 리전의 계정에 대해 고유해야 합니다. 예를 들어 <code>mariadb-instance1</code> 와 같이 선택한 AWS 리전을 포함하는 등의 방식으로 알아보기 쉬운 이름을 지정할 수 있습니다.
DB 파라미터 그룹	<p>DB 인스턴스의 파라미터 그룹입니다. 기본 파라미터 그룹을 사용하거나 사용자 지정 파라미터 그룹을 생성할 수 있습니다.</p> <p>자세한 내용은 DB 파라미터 그룹 작업 (p. 161) 단원을 참조하십시오.</p>
암호화	<p>이 DB 인스턴스에 대해 유튜 암호화를 활성화하기 위한 [Enable Encryption].</p> <p>자세한 내용은 Amazon RDS 리소스 암호화 (p. 385) 단원을 참조하십시오.</p>
확장 모니터링	<p>DB 인스턴스가 실행되는 운영 체제에 대한 실시간 측정치를 수집하려면 [Enable enhanced monitoring]을 선택합니다.</p> <p>자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.</p>
라이선스 모델	MariaDB에는 general-public-license MariaDB에 대한 일반 라이선스 계약이라는 하나의 라이선스 모델만 있습니다.
로그 내보내기	생성할 MariaDB 데이터베이스 로그 파일의 유형을 선택합니다. 자세한 내용은 MariaDB 데이터베이스 로그 파일 (p. 310) 단원을 참조하십시오.

설정	설정 설명
유지 관리 기간	<p>대기 중인 DB 인스턴스 설정 변경이 적용되기 위해 경과해야 하는 기간(30분)입니다. 이 시간이 중요하지 않은 경우 [No Preference]를 선택합니다.</p> <p>자세한 내용은 Amazon RDS 유지 관리 기간 (p. 116) 단원을 참조하십시오.</p>
Master username	<p>DB 인스턴스에 로그인할 때 사용하는 마스터 사용자 이름입니다.</p> <p>자세한 내용과 마스터 사용자의 기본 권한 목록은 Amazon RDS MariaDB 보안 (p. 431) 단원을 참조하십시오.</p>
마스터 암호	<p>마스터 사용자 계정의 암호입니다. 암호는 8~41자 인쇄 가능한 ASCII 문자를 포함해야 합니다(/, ", 공백 및 @ 제외).</p>
다중 AZ 배포	<p>장애 조치를 위해 다른 가용 영역에 DB 인스턴스의 대기 미러를 생성하기 위한 [Create replica in different zone]. 이때 고가용성을 유지하려면 프로덕션 워크로드를 위한 다중 AZ를 권장합니다. 개발 및 테스팅을 위해 No를 선택할 수 있습니다.</p> <p>자세한 내용은 Amazon RDS를 위한 고가용성(다중 AZ) (p. 109) 단원을 참조하십시오.</p>
옵션 그룹	<p>DB 인스턴스의 옵션 그룹입니다. 기본 옵션 그룹을 사용하거나 사용자 지정 옵션 그룹을 생성할 수 있습니다.</p> <p>자세한 내용은 옵션 그룹 작업 (p. 148) 단원을 참조하십시오.</p>
퍼블릭 액세스 가능성	<p>DB 인스턴스에 퍼블릭 IP 주소를 부여하려면 [Yes]를 선택합니다. 그러면 VPC 외부에서 DB 인스턴스에 액세스할 수 있습니다(DB 인스턴스도 VPC의 퍼블릭 서브넷에 있어야 함). VPC 내부에서만 DB 인스턴스에 액세스할 수 있게 하려면 [No]를 선택합니다.</p> <p>자세한 내용은 VPC에 있는 DB 인스턴스를 인터넷에서 숨기기 (p. 416) 단원을 참조하십시오.</p>
스토리지 유형	<p>DB 인스턴스의 스토리지 유형입니다.</p> <p>자세한 내용은 Amazon RDS 스토리지 유형 (p. 101) 단원을 참조하십시오.</p>
Subnet Group	<p>이 설정은 현재 사용 중인 플랫폼에 따라 다릅니다. AWS를 처음 사용하는 고객은 [default]를 선택합니다. 이 DB 서브넷 그룹은 계정에 대해 생성된 기본 DB 서브넷 그룹입니다. 이전 E2-Classic 플랫폼에서 DB 인스턴스를 만들어서 특정 VPC에 배포하려면 해당 VPC로 생성된 DB 서브넷 그룹을 선택합니다.</p>
Virtual Private Cloud(VPC)	<p>이 설정은 현재 사용 중인 플랫폼에 따라 다릅니다. AWS를 처음 사용하는 경우에는 기본 VPC를 선택합니다. 이전 E2-Classic 플랫폼에서 VPC를 사용하지 않는 DB 인스턴스를 만드는 경우에는 [Not in VPC]를 선택합니다.</p> <p>자세한 내용은 Amazon Virtual Private Cloud(VPC) 및 Amazon RDS (p. 406) 단원을 참조하십시오.</p>

설정	설정 설명
VPC 보안 그룹	AWS를 처음 사용하는 경우에는 [Create new VPC security group]을 선택합니다. 처음 사용하는 경우가 아니라면 [Select existing VPC security groups]를 선택하고 앞서 생성한 보안 그룹을 선택합니다. RDS 콘솔에서 새 VPC 보안 그룹 생성을 선택하는 경우 브라우저에서 검색된 IP 주소에서 DB 인스턴스에 액세스하도록 허용하는 발신 규칙을 사용하여 새 보안 그룹이 생성됩니다. 자세한 내용은 DB 보안 그룹 작업(EC2-Classic 플랫폼) (p. 394) 단원을 참조하십시오.

관련 주제

- 자습서: [Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 \(p. 421\)](#)
- [MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 \(p. 445\)](#)
- [MariaDB 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 449\)](#)
- [DB 인스턴스 삭제 \(p. 128\)](#)

MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결

Amazon RDS가 DB 인스턴스를 프로비저닝한 후에는 표준 MariaDB 클라이언트 애플리케이션 또는 유ти리티를 사용하여 인스턴스에 연결할 수 있습니다. 연결 문자열에는 호스트 파라미터로 DB 인스턴스 엔드포인트의 DNS 주소와 포트 파라미터로 DB 인스턴스 엔드포인트의 포트 번호를 지정합니다.

AWS Management 콘솔, AWS CLI [rds-describe-db-instances](#) 명령 또는 Amazon RDS API [DescribeDBInstances](#) 작업을 사용하여 엔드포인트를 포함한 Amazon RDS DB 인스턴스의 세부 정보를 나열할 수 있습니다.

AWS Management 콘솔에서 MariaDB 인스턴스의 엔드포인트를 찾으려면

1. RDS 콘솔을 연 다음 데이터베이스를 선택해 DB 인스턴스의 목록을 표시합니다.
2. 세부 정보를 표시하고자 하는 MariaDB DB 인스턴스 이름을 선택합니다.
3. 연결 탭에서 엔드포인트를 복사합니다. 또한 포트 번호를 적어둡니다. DB 인스턴스에 연결하려면 엔드포인트와 포트 번호가 모두 필요합니다.

Summary

DB Name: mymariadb

Role: Instance

CPU: [progress bar]

Current: [progress bar]

Connectivity Monitoring Logs & events Configuration Maintenance

Connectivity

Endpoint & port

Endpoint: mymariadb [redacted] rds.amazonaws.com

Port: 3306

예를 들어 앤드포인트 값이 mariadb-instance1.123456789012.us-east-1.rds.amazonaws.com:3306이라면 MariaDB 연결 문자열에 다음과 같이 값을 지정합니다.

- 호스트 또는 호스트 이름은 mariadb-instance1.123456789012.us-east-1.rds.amazonaws.com으로 지정합니다.
- 포트는 3306으로 지정합니다.

Amazon RDS MariaDB DB 인스턴스는 mysql 명령줄 유ти리티 같은 도구를 사용하여 연결할 수 있습니다. mysql 유ти리티 사용에 대한 자세한 내용은 MariaDB 설명서에서 [mysql 명령줄 클라이언트](#)를 참조하십시오. 연결에 사용할 수 있는 GUI 기반 애플리케이션 중 하나는 HeidiSQL입니다. 자세한 내용은 [HeidiSQL 다운로드](#) 페이지를 참조하십시오.

새로운 DB 인스턴스의 연결 오류가 발생하는 두 가지 공통 원인은 다음과 같습니다.

- 보안 그룹을 사용하여 DB 인스턴스를 만들었지만, 이 보안 그룹이 MariaDB 애플리케이션 또는 유ти리티를 실행 중인 디바이스나 Amazon EC2 인스턴스에서 연결할 수 있는 권한을 부여하지 않은 경우. DB 인스턴스가 Amazon VPC 내부에서 만들어진 경우에는 이 인스턴스의 VPC 보안 그룹이 연결 권한을 부여해야 합니다. DB 인스턴스가 VPC 외부에서 생성된 경우 이 인스턴스의 DB 보안 그룹에서 연결 권한을 부여해야 합니다.
- 포트 3306을 사용해 DB 인스턴스를 만들었는데 기업 방화벽 규칙에 따라 기업 네트워크의 디바이스에서 해당 포트에 연결하는 것이 차단된 경우. 이 오류를 수정하려면 인스턴스를 다른 포트로 다시 만들어야 합니다.

Amazon RDS MariaDB DB 인스턴스에 연결할 때는 SSL 암호화를 사용할 수 있습니다. 자세한 내용은 [MariaDB DB 인스턴스와 함께 SSL 사용 \(p. 433\)](#) 단원을 참조하십시오.

mysql 유ти리티를 통해 연결하기

mysql 유ти리티를 사용하여 DB 인스턴스에 연결하려면 클라이언트의 명령 프롬프트에 다음 명령을 입력하여 MariaDB DB 인스턴스에 있는 데이터베이스에 연결합니다. <endpoint>를 DB 인스턴스의 DNS 이름(엔드포인트)으로 대체하고, <mymasteruser>를 마스터 사용자 이름으로 대체한 후, 암호를 입력하라는 요청을 받으면 사용한 마스터 암호를 입력합니다.

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

사용자에 대한 암호를 입력하면 다음과 유사한 출력이 나타납니다.

```
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 >
```

SSL에 인스턴스 연결

Amazon RDS는 DB 인스턴스가 생성될 때 인스턴스의 SSL 인증서를 만듭니다. SSL 인증서 확인을 활성화하는 경우에는 SSL 인증서에 스푸핑 공격으로부터 보호해주는 SSL 인증서를 위한 일반 이름(CN)으로 DB 인스턴스 엔드포인트가 포함됩니다. 다음 단계에 따라 SSL을 사용하여 DB 인스턴스에 연결합니다.

mysql 유ти리티를 사용하여 SSL에 DB 인스턴스 연결하는 방법

- 여기에서 모든 리전에 적용되는 루트 인증서를 다운로드합니다.
- mysql 유ти리티에서 SSL이 포함된 DB 인스턴스에 연결하려면 명령 프롬프트에 다음 명령을 입력합니다. -h 파라미터에는 DB 인스턴스의 DNS 이름으로 대체합니다. --ssl-ca 파라미터에는 해당하는 SSL 인증서 파일 이름으로 대체합니다.

```
mysql -h mariadb-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=rds-
ca-2015-root.pem
```

- SSL 연결이 SSL 인증서의 엔드포인트와 비교하여 DB 인스턴스 엔드포인트를 검증할 수 있도록 --ssl-verify-server-cert 파라미터를 추가합니다. 예:

```
mysql -h mariadb-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=rds-ca-2015-root.pem --ssl-verify-server-cert
```

4. 입력 프롬프트가 표시되면 마스터 사용자 암호를 입력합니다.

다음과 유사한 출력 화면이 표시되어야 합니다.

```
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 >
```

최대 MariaDB 연결

Amazon RDS MariaDB DB 인스턴스에 허용되는 최대 연결 수는 DB 인스턴스의 DB 인스턴스 클래스에 사용 가능한 메모리의 양에 따라 결정됩니다. 사용 가능한 메모리가 많은 DB 인스턴스 클래스는 가능한 연결 수가 더 많아집니다. DB 인스턴스 클래스에 대한 자세한 내용은 다음([DB 인스턴스 클래스 선택 \(p. 77\)](#))을 (를) 참조하십시오.

DB 인스턴스의 연결 제한은 기본적으로 DB 인스턴스의 DB 인스턴스 클래스 최대값으로 설정됩니다. DB 인스턴스의 파라미터 그룹에서 `max_connections` 파라미터를 사용하여 동시 연결 수를 허용되는 최대 연결 수까지로 제한할 수 있습니다. 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

Amazon RDS MariaDB DB 인스턴스에 허용되는 최대 연결 수는 해당 DB 인스턴스에 대해 다음 쿼리를 실행하여 확인할 수 있습니다.

```
SELECT @@max_connections;
```

DB 인스턴스에 대해 다음 쿼리를 실행하여 Amazon RDS MariaDB DB 인스턴스에 대한 활성 연결 수를 확인할 수 있습니다.

```
SHOW STATUS WHERE `variable_name` = 'Threads_connected';
```

관련 주제

- [Amazon RDS DB 인스턴스 \(p. 76\)](#)
- [MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 437\)](#)
- [보안 그룹을 통한 액세스 제어 \(p. 390\)](#)
- [DB 인스턴스 삭제 \(p. 128\)](#)

MariaDB 데이터베이스 엔진 기반 DB 인스턴스의 변경

추가 스토리지를 더하거나 DB 인스턴스 클래스를 변경하는 것과 같은 작업을 완수하기 위해 DB 인스턴스의 설정을 변경할 수 있습니다. 이번 주제에서는 Amazon RDS MariaDB DB 인스턴스의 설정 변경과 MariaDB 인스턴스의 설정에 대해서 설명하겠습니다.

프로덕션 인스턴스를 수정하기 전에 테스트 인스턴스에서 변경 사항을 테스트하면 각 변경 사항이 미칠 영향을 완전히 이해하는 데 도움이 됩니다. 이는 특히 데이터베이스 버전을 업그레이드할 때 중요합니다.

DB 인스턴스 설정을 수정한 후 변경 사항을 즉시 적용할 수도 있고, DB 인스턴스에 대한 다음 유지 관리 기간에 적용할 수도 있습니다. 일부 변경 사항의 경우 DB 인스턴스가 재시작될 수 있습니다.

AWS Management 콘솔

MariaDB DB 인스턴스를 변경하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 변경하려는 DB 인스턴스를 선택합니다.
3. [Modify]를 선택합니다. [Modify DB instance] 페이지가 나타납니다.
4. 원하는 설정을 모두 변경합니다. 각 설정에 대한 자세한 내용은 [MariaDB DB 인스턴스 설정 \(p. 450\)](#) 단원을 참조하십시오.
5. 원하는 대로 모두 변경되었으면 [Continue]를 선택하고 수정 사항 요약을 확인합니다.
6. 변경 사항을 즉시 적용하려면 즉시 적용을 선택합니다. 일부의 경우 이 옵션을 선택하면 중단이 발생할 수 있습니다. 자세한 내용은 [즉시 적용 파라미터 사용 \(p. 112\)](#) 단원을 참조하십시오.
7. 확인 페이지에서 변경 내용을 검토합니다. 변경 내용이 정확할 경우 [Modify DB Instance]를 선택하여 변경 내용을 저장합니다.

그렇지 않으면 [Back]을 선택하여 변경 내용을 편집하거나 [Cancel]을 선택하여 변경 내용을 취소합니다.

CLI

AWS CLI를 사용해 MariaDB DB 인스턴스를 수정하려면 `modify-db-instance` 명령을 호출합니다. DB 인스턴스 식별자와 수정하려는 설정의 파라미터를 지정합니다. 각 파라미터에 대한 자세한 내용은 [MariaDB DB 인스턴스 설정 \(p. 450\)](#) 단원을 참조하십시오.

Example

다음 코드는 백업 보존 기간을 1주(7일)로 설정하여 `mydbinstance`를 수정합니다. 이 코드는 `--auto-minor-version-upgrade`를 사용하여 마이너 버전 자동 업그레이드를 활성화합니다. 마이너 버전 자동 업그레이드를 비활성화하려면 `--no-auto-minor-version-upgrade`를 사용합니다. 변경 사항은 `--no-apply-immediately`를 사용하여 다음 유지 관리 기간에 적용됩니다. 변경 사항을 바로 적용하려면 `--apply-immediately`를 사용합니다. 자세한 내용은 [즉시 적용 파라미터 사용 \(p. 112\)](#) 단원을 참조하십시오.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--backup-retention-period 7 \
--auto-minor-version-upgrade \
--no-apply-immediately
```

Windows의 경우:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--backup-retention-period 7 ^
--auto-minor-version-upgrade ^
--no-apply-immediately
```

API

Amazon RDS API를 사용하여 MariaDB 인스턴스를 수정하려면 [ModifyDBInstance](#) 작업을 호출합니다. DB 인스턴스 식별자와 수정하려는 설정의 파라미터를 지정합니다. 각 파라미터에 대한 자세한 내용은 [MariaDB DB 인스턴스 설정 \(p. 450\)](#) 단원을 참조하십시오.

Example

다음은 백업 보존 기간을 1주일(7일)로 설정하고 마이너 버전 자동 업그레이드를 활성화하여 mydbinstance를 수정하는 코드입니다. 이 변경 사항은 다음 유지 관리 기간에 적용됩니다.

```
https://rds.amazonaws.com/
?Action=ModifyDBInstance
&ApplyImmediately=false
&AutoMinorVersionUpgrade=true
&BackupRetentionPeriod=7
&DBInstanceIdentifier=mydbinstance
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-10-31
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20131016/us-west-1/rds/aws4_request
&X-Amz-Date=20131016T233051Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=087a8eb41cb1ab0fc9ec1575f23e73757ffc6a1e42d7d2b30b9cc0be988cff97
```

MariaDB DB 인스턴스 설정

다음 표에는 수정이 가능한 설정, 수정이 불가능한 설정, 변경 사항을 적용할 수 있는 시점, 변경 사항으로 인한 DB 인스턴스 가동 중지 여부에 대한 자세한 내용이 나와 있습니다.

설정	설정 설명	변경이 발생할 때	가동 중지 참고 사항
할당된 스토리지	DB 인스턴스에 할당할 스토리지(단위: GB). DB 인스턴스 상태가 스토리지 최적화이거나 지난 6시간 동안 DB 인스턴스에 할당된 스토리지가 수정된 경우 할당된 스토리지를 수정할 수 없습니다. 허용 가능한 최대 스토리지는 스토리지 유형에 따라 다릅니다. 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.	[Apply immediately]가 true로 설정되는 즉시 변경됩니다. [Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.	가동 중지 없음. 변경 도중 성능이 저하될 수 있습니다.
마이너 버전 자동 업그레이드	기본 마이너 DB 엔진 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전	-	-

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
	자동 업그레이드 사용을 선택합니다. Amazon RDS는 유지 관리 기간에 마이너 버전 자동 업그레이드를 수행합니다.		
백업 보존 기간	<p>자동 백업을 보존할 일수. 자동 백업을 비활성화하기 위해 백업 보존 기간을 0으로 설정합니다.</p> <p>자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되고 이 설정을 0이 아닌 값에서 다른 0이 아닌 값으로 변경하면 비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이때 적용되지 않을 경우, 다음 유지 관리 기간에 변경 사항이 적용됩니다.</p>	백업 보존 기간을 0에서 0이 아닌 값으로 또는 0이 아닌 값에서 0으로 변경할 경우 인스턴스가 종단됩니다.
백업 기간	<p>데이터베이스의 자동 백업이 실행되는 기간. 백업 기간은 국제 표준시(UTC)의 시작 시간과 시간 단위의 지속 기간으로 구성됩니다.</p> <p>자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.</p>	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다.	-
인증 기관	사용할 인증서.	-	-
스냅샷으로 태그 복사	<p>DB 인스턴스 태그가 있는 경우 이 옵션은 DB 스냅샷을 만들 때 태그를 복사합니다.</p> <p>자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-
데이터베이스 포트	<p>데이터베이스에 액세스하는 데 사용할 포트.</p> <p>포트 값은 DB 인스턴스의 옵션 그룹에서 옵션에 대해 지정한 포트 값과 일치하지 않아야 합니다.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	DB 인스턴스가 즉시 재부팅됩니다.
DB 엔진 버전	<p>사용할 MariaDB 데이터베이스 엔진의 버전입니다. 프로덕션 DB 인스턴스를 업그레이드 하려면 먼저 테스트 인스턴스에서 업그레이드 프로세스를 테스트하여 업그레이드 시간을 확인하고 애플리케이션을 검증하는 것이 좋습니다.</p> <p>자세한 내용은 MariaDB DB 엔진 업그레이드 (p. 455) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 종단됩니다.

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
DB 인스턴스 클래스	<p>사용할 DB 인스턴스 클래스. 자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다.
DB 인스턴스 식별자	<p>DB 인스턴스 식별자 이 값은 소문자 문자열로 저장됩니다.</p> <p>DB 인스턴스 이름 바꾸기의 영향에 대한 자세한 내용은 DB 인스턴스 이름 바꾸기 (p. 121) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다. DB 인스턴스가 재부팅됩니다.
DB 파라미터 그룹	<p>DB 인스턴스와 연결할 파라미터 그룹. 자세한 내용은 DB 파라미터 그룹 작업 (p. 161) 단원을 참조하십시오.</p>	파라미터 그룹 변경 사항은 즉시 적용됩니다.	<p>이 변경 도중 인스턴스가 중단되지 않습니다. 파라미터 그룹을 변경하는 경우 일부 파라미터에 대한 변경 내용은 재부팅 없이 DB 인스턴스에 즉시 적용됩니다. 다른 파라미터에 대한 변경 내용은 DB 인스턴스를 재부팅한 후에만 적용됩니다.</p> <p>자세한 내용은 DB 클러스터에서 DB 인스턴스 (p. 123) 단원을 참조하십시오.</p>
삭제 방지	DB 인스턴스가 삭제되지 않도록 방지하려면, 삭제 방지를 활성화합니다. 자세한 내용은 DB 인스턴스 삭제 (p. 128) 단원을 참조하십시오.	-	-
확장 모니터링	<p>DB 인스턴스가 실행되는 운영 체제에 대한 실시간 측정치 수집을 활성화하려면 [Enable enhanced monitoring]을 선택합니다.</p> <p>자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.</p>	-	-
로그 내보내기	<p>Amazon CloudWatch Logs에 게시할 MariaDB 데이터베이스 로그 파일의 유형을 선택합니다.</p> <p>자세한 내용은 MariaDB 데이터베이스 로그 파일 (p. 310) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
유지 관리 기간	<p>시스템 유지 관리를 실행하는 기간. 시스템 유지 관리는 업그레이드를 포함합니다(해당할 경우). 유지 관리 기간은 국제 표준시(UTC)의 시작 시간과 시간 단위의 지속 기간으로 구성됩니다.</p> <p>이 기간을 현재 시간으로 설정하려면 대기 중인 변경 사항이 모두 적용될 수 있도록 현재 시간과 기간 종료 시간 사이에 최소 30분 이상 필요합니다.</p> <p>자세한 내용은 Amazon RDS 유지 관리 기간 (p. 116) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	인스턴스가 종단될 수 있는 작업이 하나 이상 대기 중이고, 유지 관리 기간이 현재 시간을 포함하여 변경된 경우 대기 중인 작업들이 즉시 적용되고 인스턴스가 종단됩니다.
다중 AZ 배포	<p>다중 가용 영역에 DB 인스턴스를 배포하려면 [Yes], 그렇지 않으면 [No].</p> <p>자세한 내용은 리전 및 가용 영역 (p. 99) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	-
새 마스터 암호	마스터 사용자의 암호. 암호에는 8~41자의 영숫자 문자로 구성되어야 합니다.	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-
옵션 그룹	<p>DB 인스턴스와 연결할 옵션 그룹.</p> <p>자세한 내용은 옵션 그룹 작업 (p. 148) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	-
퍼블릭 액세스 가능성	<p>DB 인스턴스에 퍼블릭 IP 주소를 부여하려면(즉 VPC 외부에서 액세스할 수 있음) [Yes]를 선택합니다. 공개적으로 액세스가 가능하려면 DB 인스턴스도 VPC의 퍼블릭 서브넷에 있어야 합니다. VPC 내부에서만 DB 인스턴스에 액세스할 수 있게 하려면 [No].</p> <p>자세한 내용은 VPC에 있는 DB 인스턴스를 인터넷에서 습기기 (p. 416) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
보안 그룹	DB 인스턴스와 연결할 보안 그룹. 자세한 내용은 DB 보안 그룹 작업 (EC2-Classic 플랫폼) (p. 394) 단원을 참조하십시오.	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-
스토리지 유형	사용할 스토리지 유형. 자세한 내용은 Amazon RDS 스토리지 유형 (p. 101) 단원을 참조하십시오.	[Apply immediately]가 true로 설정되는 즉시 변경됩니다. [Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.	다음과 같이 변경하면 프로세스가 시작되는 동안 잠시 작동이 중단됩니다. 그 이후에 변경 사항이 적용되는 동안 데이터베이스를 정상적으로 사용 할 수 있습니다. <ul style="list-style-type: none"> • [General Purpose (SSD)] → [Magnetic]. • 범용(SSD)에서 프로비저닝된 IOPS(SSD)로. DB 인스턴스가 단일 AZ이고 사용자 지정 파라미터 그룹을 사용하는 경우에만 중단이 발생합니다. 다른 AZ DB 인스턴스가 중단되지는 않습니다. • [Magnetic] → [General Purpose (SSD)]. • [Magnetic] → [Provisioned IOPS (SSD)]. • [Provisioned IOPS (SSD)] → [Magnetic]. • 프로비저닝된 IOPS(SSD)에서 범용 (SSD)로. DB 인스턴스가 단일 AZ이고 사용자 지정 파라미터 그룹을 사용하는 경우에만 중단이 발생합니다. 다른 AZ DB 인스턴스가 중단되지는 않습니다.

설정	설정 설명	변경이 발생할 때	자동 종지 참고 사항
[Subnet Group]	DB 인스턴스에 대한 서브넷 그룹. 이 설정을 사용하여 DB 인스턴스를 다른 VPC로 이동할 수 있습니다. DB 인스턴스가 VPC 내에 있지 않을 경우 이 설정을 사용하여 DB 인스턴스를 VPC 내로 이동할 수 있습니다. 자세한 내용은 VPC에 있지 않은 DB 인스턴스를 VPC로 이동 (p. 420) 단원을 참조하십시오.	[Apply Immediately]가 true로 설정되는 즉시 변경됩니다. [Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.	이 변경 도중 인스턴스가 중단됩니다. DB 인스턴스가 재부팅됩니다.

관련 주제

- DB 클러스터에서 DB 인스턴스 (p. 123)
- MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 (p. 445)
- MariaDB DB 엔진 업그레이드 (p. 455)
- DB 인스턴스 삭제 (p. 128)

MariaDB DB 엔진 업그레이드

Amazon RDS에서 새 데이터베이스 엔진 버전을 지원하는 경우, DB 인스턴스를 새 버전으로 업그레이드할 수 있습니다. 메이저 버전 업그레이드와 마이너 버전 업그레이드라는 두 가지 업그레이드가 있습니다. DB 인스턴스를 수동으로 수정하여 메이저 버전 업그레이드를 수행해야 합니다. DB 인스턴스에서 마이너 버전 자동 업그레이드를 사용하도록 설정하면 마이너 버전 업그레이드가 자동으로 수행됩니다. 그 밖의 경우에는 DB 인스턴스를 수동으로 수정하여 마이너 버전 업그레이드를 수행해야 합니다.

MariaDB 지원 버전 및 버전 관리에 대한 자세한 내용은 [Amazon RDS MariaDB 버전 \(p. 428\)](#) 단원을 참조하십시오.

주제

- [업그레이드 개요 \(p. 455\)](#)
- [MariaDB DB 인스턴스 업그레이드 \(p. 456\)](#)

업그레이드 개요

메이저 버전 업그레이드에는 기존 애플리케이션과 호환되지 않는 데이터베이스 변경 사항이 포함될 수 있습니다. 결과적으로 Amazon RDS에서는 메이저 버전 업그레이드가 자동으로 적용되지 않으므로 DB 인스턴스를 수동으로 변경해야 합니다. 모든 업그레이드는 프로덕션 환경의 인스턴스에 적용하기 전에 반드시 철저하게 테스트해야 합니다.

사용자가 달리 지정하지 않는 한 Amazon RDS에서 새로운 MariaDB 마이너 버전을 지원할 때 DB 인스턴스가 새 버전으로 자동 업그레이드됩니다. 이 패치는 예약 유지보수 중에 발생합니다. 마이너 버전 자동 업그레이드를 끄도록 DB 인스턴스를 수정할 수 있습니다.

Amazon RDS는 업그레이드 프로세스 중에 DB 스냅샷을 2개 캡처합니다. 첫 번째 DB 스냅샷은 업그레이드 변경 이전 DB 인스턴스의 스냅샷입니다. 업그레이드가 데이터베이스에 맞지 않는 경우에는 이 스냅샷을 복구하여 이전 버전의 DB 인스턴스를 생성할 수 있습니다. 두 번째 DB 스냅샷은 업그레이드 완료와 함께 캡처됩니다.

Note

DB 인스턴스에 대한 백업 보존 기간을 0보다 큰 수로 설정하면 Amazon RDS는 DB 스냅샷만 캡처합니다. 백업 보존 기간을 변경하려면 [MariaDB 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 449\)](#) 단원을 참조하십시오.

업그레이드가 완료되면 이전 버전의 데이터베이스 엔진으로 되돌릴 수 없습니다. 이때 이전 버전으로 되돌리려면 첫 번째로 캡처한 DB 스냅샷을 복구하여 새로운 DB 인스턴스를 생성해야 합니다.

DB 인스턴스를 Amazon RDS가 지원하는 새 버전으로 업그레이드하는 시기는 사용자가 직접 관리합니다. 이러한 관리 수준은 특정 데이터베이스 버전과 호환성을 유지하거나 프로덕션 환경에 배포하기 전에 애플리케이션을 이용해 새 버전을 테스트하는 데 효과적입니다. 모든 준비를 마치면 일정에 가장 적합한 시기에 버전 업그레이드를 실행할 수 있습니다.

DB 인스턴스가 읽기 전용 복제본을 사용하는 경우 원본 인스턴스를 업그레이드하기 전에 읽기 전용 복제본 부터 모두 업그레이드해야 합니다.

DB 인스턴스를 다중 AZ 배포로 생성한 경우에는 기본 DB 인스턴스와 예비 DB 인스턴스가 모두 업그레이드 됩니다. 기본 DB 인스턴스와 예비 DB 인스턴스가 모두 동시에 업그레이드되므로 업그레이드가 끝날 때까지 작동 중단을 겪게 됩니다. 중단 시간은 데이터베이스 엔진, 엔진 버전 및 DB 인스턴스의 크기에 따라 달라집니다.

사용자 지정 파라미터 그룹을 사용 중이고 메이전 버전 업그레이드를 수행하는 경우 새 DB 엔진 버전에 대한 기본 파라미터 그룹을 지정하거나 새 DB 엔진 버전에 대한 사용자 지정 파라미터 그룹을 만들어야 합니다. 새 파라미터 그룹을 DB 인스턴스에 연결하려면 업그레이드가 완료된 후 고객이 데이터베이스 재부팅을 시작해야 합니다. 파라미터 그룹 변경 사항을 적용하기 위해 인스턴스를 재부팅해야 하는 경우, 인스턴스의 파라미터 그룹 상태가 pending-reboot로 표시됩니다. 인스턴스의 파라미터 그룹 상태는 AWS 콘솔에서 확인하거나 "describe" 호출(예: describe-db-instances)을 사용하여 확인할 수 있습니다.

MariaDB DB 인스턴스 업그레이드

MariaDB DB 인스턴스의 수동 또는 자동 업그레이드에 대한 자세한 내용은 [DB 인스턴스 엔진 버전 업그레이드 \(p. 118\)](#) 단원을 참조하십시오.

MySQL DB 스냅샷에서 MariaDB DB 인스턴스로 데이터 마이그레이션

AWS Management 콘솔, AWS CLI 또는 Amazon RDS API를 사용하여 MariaDB 10.1을 실행하는 새 DB 인스턴스로 Amazon RDS MySQL DB 스냅샷을 마이그레이션할 수 있습니다. MySQL 5.6을 실행하는 Amazon RDS DB 인스턴스에서 DB 스냅샷을 생성해야 합니다. RDS MySQL DB 스냅샷을 생성하는 방법은 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.

MySQL에서 MariaDB로 마이그레이션하면 MariaDB DB 인스턴스는 기본 DB 파라미터 그룹 및 옵션 그룹과 연동됩니다. DB 스냅샷을 복구한 후에는 새 DB 인스턴스에 대한 사용자 지정 DB 파라미터 그룹을 연동할 수 있습니다. 그러나 MariaDB 파라미터 그룹에는 구성 가능한 다양한 시스템 변수 집합이 있습니다. MySQL 및 MariaDB 시스템 변수의 차이에 대한 정보는 [System Variable Differences Between MariaDB 10.1 and MySQL 5.6](#)을 참조하십시오. DB 파라미터 그룹에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오. 옵션 그룹에 대한 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오.

MariaDB와 MySQL 간의 호환성 관련 문제

MariaDB와 MySQL 간의 비호환성 관련 문제로는 다음과 같은 것들이 있습니다.

- MySQL 5.5로 생성된 DB 스냅샷은 MariaDB 10.1로 마이그레이션할 수 없습니다.
- MySQL 5.7로 생성된 DB 스냅샷은 MariaDB로 마이그레이션할 수 없습니다.

- MySQL 8.0으로 생성된 DB 스냅샷은 MariaDB로 마이그레이션할 수 없습니다.
- 암호화된 DB 스냅샷을 마이그레이션할 수 없습니다.
- 원본 MySQL 데이터베이스에서 SHA256 암호 해시를 사용하는 경우, MariaDB 데이터베이스에 연결하려면 먼저 SHA256 해시 처리된 사용자 암호를 재설정해야 합니다. 다음 코드는 SHA256 해시 처리된 암호를 재설정하는 방법을 보여줍니다.

```
SET old_passwords = 0;
UPDATE mysql.user SET plugin = 'mysql_native_password',
Password = PASSWORD('new_password')
WHERE (User, Host) = ('master_user_name', %);
FLUSH PRIVILEGES;
```

- RDS 마스터 사용자 계정에서 SHA-256 암호 해시를 사용하는 경우, 이 암호는 RDS [modify-db-instance](#) AWS CLI 명령, [ModifyDBInstance](#) API 작업 또는 AWS Management 콘솔을 사용하여 재설정해야 합니다. MariaDB DB 인스턴스 수정에 대한 자세한 내용은 [MariaDB 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 449\)](#) 단원을 참조하십시오.
- MariaDB는 Memcached 플러그인을 지원하지 않지만, Memcached 플러그인이 사용하는 데이터는 InnoDB 테이블 형식으로 저장됩니다. MySQL DB 스냅샷을 마이그레이션한 후에는 SQL을 사용하여 Memcached 플러그인이 사용하는 데이터에 액세스할 수 있습니다. [innodb_memcache](#) 데이터베이스에 대한 자세한 내용은 [InnoDB memcached Plugin Internals](#)를 참조하십시오.

AWS Management 콘솔

MySQL DB 스냅샷에서 MariaDB DB 인스턴스로 마이그레이션하려면

- AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
- 탐색 창에서 [Snapshots]를 선택한 다음 마이그레이션하려는 MySQL DB 스냅샷을 선택합니다.
- 작업에서 스냅샷 마이그레이션을 선택합니다. [Migrate Database] 페이지가 표시됩니다.
- [Migrate to DB Engine]에서 [mariadb]를 선택합니다.
- [Migrate Database] 페이지에서 RDS가 MariaDB DB 인스턴스를 시작하는 데 필요한 추가 정보를 입력합니다.
 - DB 엔진 버전: 사용하려는 MariaDB 데이터베이스 엔진의 버전을 선택합니다. 자세한 내용은 [MariaDB DB 엔진 업그레이드 \(p. 455\)](#) 단원을 참조하십시오.
 - [DB Instance Class]: 데이터베이스에 필요한 스토리지와 용량을 지닌 DB 인스턴스 클래스를 선택합니다(예: db.r3.large). 빠르고 일관된 I/O 성능이 필요한 프로덕션 애플리케이션의 경우에는 프로비저닝된 IOPS 스토리지를 사용하는 것이 좋습니다. 자세한 내용은 [프로비저닝된 IOPS SSD 스토리지 \(p. 103\)](#) 단원을 참조하십시오. MariaDB 10.1은 이전 세대 DB 인스턴스 클래스를 지원하지 않습니다. 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.
 - [Multi-AZ Deployment]: 다중 가용 영역에 DB 인스턴스를 배포하려면 [Yes]를, 그렇지 않으면 [No]를 선택합니다. 자세한 내용은 [리전 및 가용 영역 \(p. 99\)](#) 단원을 참조하십시오.
 - [DB Snapshot ID]: DB 스냅샷 식별자의 이름을 입력합니다.

DB 스냅샷 식별자는 다음과 같은 제약 조건이 있습니다.

- 1~255자의 영숫자 문자 또는 하이픈으로 구성되어야 합니다.
- 반드시 글자이어야 합니다.
- 하이픈으로 끝나거나 하이픈이 2개 연속으로 이어져서는 안 됩니다.

공유된 수동 DB 스냅샷에서 복원하는 경우에는 DB 스냅샷 식별자가 공유된 DB 스냅샷의 Amazon 리소스 이름(ARN)이어야 합니다.

- DB 인스턴스 식별자: DB 인스턴스가 상주하는 AWS 리전의 계정에 고유한 DB 인스턴스 이름을 입력합니다. 이 식별자는 DB 인스턴스의 인스턴스에 대한 엔드포인트 주소에서 사용됩니다.

DB 인스턴스 식별자는 다음과 같은 제약 조건이 있습니다.

- 1~63자의 영숫자 문자 또는 하이픈으로 구성되어야 합니다.
- 첫 번째 문자는 글자이어야 합니다.
- 하이픈으로 끝나거나 하이픈이 2개 연속으로 이어져서는 안 됩니다.
- AWS 리전 내에서 AWS 계정의 모든 DB 인스턴스에 대해 고유한 것이어야 합니다.
- [Virtual Private Cloud(VPC)]: 기존 VPC가 있는 경우, VPC 식별자(예: vpc-a464d1c1)를 선택하여 그 VPC를 MariaDB DB 인스턴스에 사용할 수 있습니다. VPC에 대한 자세한 내용은 [Amazon Virtual Private Cloud\(VPC\)](#) 및 [Amazon RDS \(p. 406\)](#) 단원을 참조하십시오.

기존 VPC가 없다면 [Create a new VPC]를 선택하여 Amazon RDS에서 VPC를 새로 생성하도록 할 수 있습니다.

EC2 Classic Network에서는 MariaDB 인스턴스를 생성할 수 없습니다.

- [Subnet group]: 기존 서브넷 그룹이 있으면 해당 서브넷 그룹 식별자(예: gs-subnet-group1)를 선택하여 MariaDB DB 인스턴스에서 그 서브넷 그룹을 사용할 수 있습니다.

기존 서브넷 그룹이 없다면 [Create a new subnet group]을 선택하여 Amazon RDS에서 서브넷 그룹을 새로 생성하도록 할 수 있습니다.

- [Publicly accessibility]: DB 인스턴스의 인스턴스를 VPC에 있는 리소스에서만 액세스할 수 있도록 하려면 [No]를 선택합니다. 퍼블릭 네트워크에 있는 리소스가 DB 인스턴스의 인스턴스에 액세스할 수 있도록 하려면 [Yes]를 선택합니다. 기본값은 [Yes]입니다.
- [Availability zone]: MariaDB DB 인스턴스의 기본 인스턴스를 호스팅하려면 [Availability Zone]을 선택합니다. Amazon RDS가 Availability Zone(가용 영역)을 선택하도록 하려면 No Preference(기본 설정 없음)를 선택합니다.
- 데이터베이스 포트: DB 인스턴스의 인스턴스에 연결할 때 사용할 기본 포트를 입력합니다. 기본값은 3306입니다.

기업 방화벽 뒤에 있어서 MySQL 기본 포트인 3306 같은 기본 포트에 액세스하지 못할 수도 있습니다. 이런 경우에는 기업 방화벽이 허용하는 포트 값을 입력합니다.

- 옵션 그룹: DB 인스턴스와 연결할 옵션 그룹을 선택합니다. 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오.
- [Enable]: 새 MariaDB DB 인스턴스를 "유휴 상태에서" 암호화하려면 [Enable Encryption]을 선택합니다. Enable encryption(암호화 활성화)을 선택하면 Master key(마스터 키) 값으로 AWS KMS 암호화 키를 선택해야 합니다.
- 마이너 버전 자동 업그레이드: 기본 마이너 DB 엔진 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다. Amazon RDS는 유지 관리 기간에 마이너 버전 자동 업그레이드를 수행합니다. 마이너 버전 자동 업그레이드 옵션은 MariaDB DB 인스턴스의 경우, MySQL 엔진 마이너 버전으로 업그레이드할 때만 사용할 수 있습니다. 시스템 안정성 유지를 위한 정기 패치에는 적용되지 않습니다.

Migrate Database

Migrate this database to a new DB Engine by selecting your desired options for the migrated instance.

Instance specifications

Migrate to DB Engine
Name of the Database Engine
mariadb

DB Engine Version
Version Number or the Database Engine to be used for this instance
10.1.14 (default)

DB Instance Class
Contains the compute and memory capacity of the DB Instance.
db.m4.xlarge — 4 vCPU, 16 GiB RAM

Multi-AZ Deployment
Specifies if the DB Instance should have a standby deployed in another Availability Zone.

6. [Migrate]를 선택합니다.

CLI

MySQL DB 스냅샷에서 MariaDB DB 인스턴스로 데이터를 마이그레이션하려면 AWS CLI `restore-db-instance-from-db-snapshot` 명령을 다음 파라미터와 함께 사용합니다.

- `--db-instance-identifier` – DB 스냅샷에서 생성할 DB 인스턴스의 이름.
- `--db-snapshot-identifier` – 복구할 DB 스냅샷에 대한 식별자.
- `--engine` – 새 인스턴스에 사용할 데이터베이스 엔진.

Example

Linux, OS X, Unix의 경우:

```
aws rds restore-db-instance-from-db-snapshot \
--db-instance-identifier newmariadbinstance \
--db-snapshot-identifier mysqlsnapshot \
--engine mariadb
```

Windows의 경우:

```
aws rds restore-db-instance-from-db-snapshot \
--db-instance-identifier newmariadbinstance ^
--db-snapshot-identifier mysqlsnapshot ^
```

```
--engine mariadb
```

API

MySQL DB 스냅샷에서 MariaDB DB 인스턴스로 데이터를 마이그레이션하려면 Amazon RDS API 작업 [RestoreDBInstanceFromDBSnapshot](#)을 호출합니다.

Example

```
https://rds.us-west-2.amazonaws.com/
    ?Action=RestoreDBInstanceFromDBSnapshot
    &DBInstanceIdentifier= newmariadbinstance
    &DBSnapshotIdentifier= mysqlsnapshot
    &Engine= mariadb
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2013-09-09
    &X-Amz-Algorithm=AWS4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20140428/us-west-2/rds/aws4_request
    &X-Amz-Date=20140428T232655Z
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
    &X-Amz-Signature=78ac761e8c8f54a8c0727f4e67ad0a766fbb0024510b9aa34ea6d1f7df52fe92
```

MariaDB 복제 작업

일반적으로 읽기 전용 복제본을 사용하여 Amazon RDS DB 인스턴스 간 복제를 구성합니다. 읽기 전용 복제본에 대한 일반적인 정보는 [읽기 전용 복제본 작업 \(p. 135\)](#) 단원을 참조하십시오. Amazon RDS MariaDB의 읽기 전용 복제본 작업에 대한 자세한 내용은 [MariaDB 읽기 전용 복제본 작업 \(p. 461\)](#) 단원을 참조하십시오.

또한 MariaDB DB 인스턴스에 대해 바이너리 로그 좌표를 기반으로 복제를 구성할 수 있습니다. MariaDB 인스턴스의 경우에는 전역 트랜잭션 ID(GTID)를 기반으로 복제를 구성할 수도 있습니다. 그러면 충돌 안정성이 개선됩니다. 자세한 내용은 [Amazon RDS MariaDB DB 인스턴스로의 GTID 기반 복제 구성 \(p. 464\)](#) 단원을 참조하십시오.

다음은 Amazon RDS MariaDB에서 사용 가능한 다른 복제 옵션입니다.

- Amazon RDS MariaDB DB 인스턴스와 Amazon RDS 외부에 있는 MySQL 또는 MariaDB 인스턴스 간의 복제를 설정할 수 있습니다. 외부 소스를 사용하여 복제를 구성하는 방법에 대한 자세한 내용은 [Amazon RDS 외부에서 실행 중인 MySQL 또는 MariaDB 인스턴스를 사용한 복제 \(p. 655\)](#) 단원을 참조하십시오.
- Amazon RDS 외부에 있는 MySQL 또는 MariaDB 인스턴스에서 데이터베이스를 가져오거나 그런 인스턴스로 데이터베이스를 내보내도록 복제를 구성할 수 있습니다. 자세한 내용은 [자동 중지 시간을 단축하여 Amazon RDS MySQL MariaDB DB 인스턴스로 데이터 가져오기 \(p. 628\)](#) 및 [복제를 사용하여 MySQL DB 인스턴스에서 데이터 내보내기 \(p. 660\)](#) 단원을 참조하십시오.

이러한 복제 옵션의 경우 행 기반 복제, 문 기반 복제 또는 혼합 복제를 사용할 수 있습니다. 행 기반 복제는 SQL 문으로 인해 변경된 행만 복제합니다. 문 기반 복제는 전체 SQL 문을 복제합니다. 혼합 복제는 가능한 경우 문 기반 복제를 사용하지만, 문 기반 복제에 안전하지 않은 SQL 문이 실행될 경우 행 기반 복제로 전환합니다. 대부분의 경우 혼합 복제가 권장됩니다. DB 인스턴스의 이진 로그 형식은 복제가 행 기반인지, 문 기반인지, 혼합인지 결정합니다. 이진 로그 형식 설정에 대한 자세한 내용은 [이진 로깅 형식 \(p. 315\)](#) 단원을 참조하십시오.

주제

- [MariaDB 읽기 전용 복제본 작업 \(p. 461\)](#)
- [Amazon RDS MariaDB DB 인스턴스로의 GTID 기반 복제 구성 \(p. 464\)](#)

MariaDB 읽기 전용 복제본 작업

이 단원에는 Amazon RDS MariaDB의 읽기 전용 복제본 작업에 대한 자세한 정보가 포함되어 있습니다. 읽기 전용 복제본에 대한 일반적인 정보와 사용 지침은 [읽기 전용 복제본 작업 \(p. 135\)](#) 단원을 참조하십시오.

주제

- [MariaDB를 사용한 읽기 전용 복제본 구성 \(p. 461\)](#)
- [MariaDB를 사용한 읽기 전용 복제본 업데이트 \(p. 461\)](#)
- [MariaDB를 사용한 다중 AZ 읽기 전용 복제본 배포 \(p. 462\)](#)
- [MariaDB 읽기 전용 복제본 모니터링 \(p. 462\)](#)
- [MariaDB 읽기 전용 복제본을 사용한 복제 시작 및 중지 \(p. 462\)](#)
- [MariaDB를 사용한 읽기 전용 복제본 삭제 \(p. 462\)](#)
- [MariaDB 읽기 전용 복제본의 문제 해결 \(p. 463\)](#)

MariaDB를 사용한 읽기 전용 복제본 구성

MariaDB DB 인스턴스를 복제 원본으로 사용하려면 백업 보존 기간을 0이 아닌 다른 값으로 설정하여 원본 DB 인스턴스의 자동 백업을 활성화해야 합니다. 이러한 요건은 다른 읽기 전용 복제본의 원본 DB 인스턴스 인 읽기 전용 복제본에도 똑같이 적용됩니다.

DB 인스턴스 하나에서 최대 5개까지 읽기 전용 복제본을 생성할 수 있습니다. 효과적인 복제를 위해서는 읽기 전용 복제본도 각각 원본 DB 인스턴스와 동일한 용량의 컴퓨팅 파워와 스토리지 리소스를 가져야 합니다. 원본 DB 인스턴스를 확장하면 읽기 전용 복제본도 확장해야 합니다.

읽기 전용 복제본이 모든 버전의 MariaDB 기반인 경우에는 읽기 전용 복제본을 다른 읽기 전용 복제본의 원본 DB 인스턴스로 지정할 수 있습니다. 예를 들어 MyDBInstance에서 ReadReplica1을 생성한 후 다시 ReadReplica1에서 ReadReplica2를 생성할 수 있습니다. MyDBInstance의 업데이트는 ReadReplica1에도 복제되며, 다시 한 번 ReadReplica1에서 ReadReplica2로 복제됩니다. 하지만 복제 체인에 포함될 수 있는 인스턴스는 최대 4개로 제한됩니다. 예를 들어 MySourceDBInstance에서 ReadReplica1을 생성한 후 다시 ReadReplica1에서 ReadReplica2를 생성하고, ReadReplica2에서 ReadReplica3를 생성할 수 있지만 ReadReplica3에서 ReadReplica4를 생성할 수는 없습니다.

MariaDB 읽기 전용 복제본을 다른 읽기 전용 복제본으로 복제할 수 있도록 승격한 경우 이 읽기 전용 복제본은 활성화 상태를 유지합니다. MyDBInstance1이 MyDBInstance2로, 그리고 MyDBInstance2가 다시 MyDBInstance3으로 복제된다고 예를 들어보겠습니다. 이때 MyDBInstance2를 승격하는 경우 MyDBInstance1에서 MyDBInstance2로는 더 이상 복제가 이루어지지 않습니다. 하지만 MyDBInstance2는 여전히 MyDBInstance3으로 복제됩니다.

Amazon RDS MariaDB 읽기 전용 복제본에서 자동 백업을 활성화하려면 먼저 읽기 전용 복제본을 생성한 다음 자동 백업이 활성화되도록 읽기 전용 복제본을 변경해야 합니다.

읽기 전용 복제본이 원본 인스턴스의 한계치인 5개를 벗어나지 않는 경우에 한해 다수의 읽기 전용 복제본을 동시에 실행하거나, 동일한 원본 DB 인스턴스를 참조하는 작업은 생성 또는 삭제할 수도 있습니다.

MariaDB를 사용한 읽기 전용 복제본 업데이트

읽기 전용 복제본은 읽기 쿼리를 지원하도록 설계되었지만 때때로 업데이트가 필요할 수 있습니다. 예를 들어 특정 유형의 쿼리가 복제본에 액세스하는 속도를 높이기 위해 인덱스를 추가해야 할 경우가 있습니다. 업

데이터를 활성화하려면 읽기 전용 복제본의 DB 파라미터 그룹에서 `read_only` 파라미터를 0으로 설정합니다.

MariaDB를 사용한 다중 AZ 읽기 전용 복제본 배포

읽기 전용 복제본은 단일 AZ 또는 다중 AZ DB 인스턴스 배포를 통해서도 생성할 수 있습니다. 다중 AZ 배포는 중요 데이터의 내구성과 가용성을 개선하는 데 효과적이지만 읽기 전용 쿼리를 실행하는 데 보조로 사용 할 수는 없습니다. 그 대신 읽기 전용 쿼리 부하를 줄일 목적으로 트래픽이 많은 다중 AZ DB 인스턴스에서 읽기 전용 복제본을 생성할 수 있습니다. 다중 AZ 배포의 원본 인스턴스가 보조 인스턴스로 장애 조치된 경우에는 연결된 모든 읽기 전용 복제본이 자동으로 전환되어 보조(이제는 기본) 인스턴스를 복제 원본으로 사용합니다. 자세한 내용은 [Amazon RDS를 위한 고가용성\(다중 AZ\) \(p. 109\)](#) 단원을 참조하십시오.

읽기 전용 복제본을 Multi-AZ DB 인스턴스를 생성할 수 있습니다. Amazon RDS는 복제본에 대한 장애 조치 지원을 위해 다른 가용 영역에 복제본의 대기를 생성합니다. 읽기 전용 복제본을 다중 AZ DB 인스턴스로 생성하는 작업은 원본 데이터베이스가 다중 AZ DB 인스턴스인지 여부와는 독립적입니다.

MariaDB 읽기 전용 복제본 모니터링

MariaDB 읽기 전용 복제본의 경우 Amazon RDS ReplicaLag 지표를 보면서 Amazon CloudWatch의 복제 지연을 모니터링할 수 있습니다. ReplicaLag 지표는 `SHOW SLAVE STATUS` 명령의 `Seconds_Behind_Master` 필드의 값을 보고합니다.

이렇게 MariaDB에서 복제 지연이 발생하는 공통 원인은 다음과 같습니다.

- 네트워크 중단.
- 읽기 전용 복제본 관련 인덱스를 할당하여 테이블에 쓰기 작업 중일 때. 읽기 전용 복제본에서 `read_only` 파라미터가 0으로 설정되어 있지 않으면 복제가 중단될 수 있습니다.
- MyISAM과 같은 비트랜잭션 스토리지 엔진 사용. 복제는 MariaDB 10.2 이상의 InnoDB 스토리지 엔진 및 MariaDB 10.1 이전 XtraDB 스토리지 엔진에 대해서만 지원됩니다.

ReplicaLag 지표가 0에 도달하면 복제본이 원본 DB 인스턴스를 따라잡은 것입니다. ReplicaLag 지표가 -1을 반환하는 경우 복제가 현재 활성이 아닙니다. ReplicaLag = -1은 Seconds_Behind_Master = NULL과 동등합니다.

MariaDB 읽기 전용 복제본을 사용한 복제 시작 및 중지

Amazon RDS DB 인스턴스에서는 시스템에 저장된 프로시저인 [mysql.rds_stop_replication \(p. 691\)](#)과(와) [mysql.rds_start_replication \(p. 688\)](#)을 호출하여 복제 프로세스를 종료하거나 재시작할 수 있습니다. 대용량 인덱스를 생성하는 등 오랜 시간이 걸리는 작업에서 두 Amazon RDS 인스턴스를 서로 복제할 때도 이런 방법이 가능합니다. 또한 데이터베이스를 가져오거나 내보낼 때도 복제를 종료하거나 시작할 필요가 있습니다. 자세한 내용은 [자동 중지 시간을 단축하여 Amazon RDS MySQL MariaDB DB 인스턴스로 데이터 가져오기 \(p. 628\)](#) 및 [복제를 사용하여 MySQL DB 인스턴스에서 데이터 내보내기 \(p. 660\)](#) 단원을 참조하십시오.

수동으로, 혹은 복제 오류로 인해 연속하여 30일 이상 복제가 중단된 경우에는 Amazon RDS가 마스터 DB 인스턴스와 모든 읽기 전용 복제본 사이의 복제를 종료합니다. 이렇게 하는 이유는 마스터 DB 인스턴스에 대한 스토리지 요건 증가와 장애 조치의 장기화를 방지하기 위해서입니다. 이후에도 읽기 전용 복제본 DB 인스턴스는 여전히 사용할 수 있습니다. 하지만 읽기 전용 복제본에서 필요한 이진 로그가 복제 종료와 함께 마스터 DB 인스턴스에서 삭제되어 복제를 재개할 수 없습니다. 마스터 DB 인스턴스에서 복제를 재개하려면 새로운 읽기 전용 복제본을 생성해야 합니다.

MariaDB를 사용한 읽기 전용 복제본 삭제

읽기 전용 복제본을 삭제할 때는 DB 인스턴스를 삭제할 때와 똑같은 메커니즘으로 명시적이어야 합니다. 원본 DB 인스턴스만 삭제하고 복제본을 삭제하지 않으면 각 복제본은 독립 실행형 DB 인스턴스로 승격됩니다.

MariaDB 읽기 전용 복제본의 문제 해결

MariaDB의 복제본 기술은 비동기적입니다. 간혹 원본 DB 인스턴스에서 BinLogDiskUsage가 증가하면 읽기 전용 복제본의 ReplicaLag가 예상되는 이유도 비동기식이기 때문입니다. 예를 들어 원본 DB 인스턴스에 대해 대량의 쓰기 작업이 동시에 발생할 수 있습니다. 반대로 읽기 전용 복제본에 대한 쓰기 작업은 단일 I/O 스레드를 사용하기 때문에 연이어 차례로 발생합니다. 이로 인해 원본 인스턴스와 읽기 전용 복제본 사이에 지연 시간이 있기 마련입니다. 읽기 전용 복제본에 대한 자세한 내용은 MariaDB 설명서에서 [Replication Overview](#) 단원을 참조하십시오.

원본 DB 인스턴스와 뒤이어 일어나는 읽기 전용 복제본의 업데이트 간 지연 시간을 줄일 수 있는 방법이 몇 가지 있으며, 다음과 같습니다.

- 읽기 전용 복제본의 크기를 조정하여 원본 DB 인스턴스에 버금가는 스토리지 크기와 DB 인스턴스 클래스를 할당합니다.
- 원본 DB 인스턴스와 읽기 전용 복제본에 사용되는 DB 파라미터 그룹의 파라미터 설정이 서로 호환되는지 확인합니다. 자세한 정보와 예는 이번 섹션 후반의 `max_allowed_packet` 파라미터 관련 설명을 참조하십시오.

Amazon RDS는 읽기 전용 복제본의 복제 상태를 모니터링하고, 어떤 이유로든 복제가 중지되는 경우 읽기 전용 복제본 인스턴스의 Replication State 필드를 Error로 업데이트합니다. 읽기 전용 복제본에 대한 DML 쿼리가 원본 DB 인스턴스의 업데이트와 충돌하는 경우가 한 예가 될 수 있습니다.

MariaDB 앤진에서 발생하는 관련 오류에 대한 세부 정보는 Replication Error 필드에서 다시 확인할 수 있습니다. [RDS-EVENT-0045 \(p. 295\)](#), [RDS-EVENT-0046 \(p. 295\)](#) 및 [RDS-EVENT-0047 \(p. 294\)](#)을 포함하여 읽기 전용 복제본의 상태를 표시하는 이벤트도 생성됩니다. 이벤트와 이벤트 구독에 대한 자세한 내용은 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오. MariaDB 오류 메시지가 반환되는 경우 [MariaDB 오류 메시지 문서](#)에 설명되어 있는 오류를 검토하십시오.

복제 오류의 원인이 되는 공통적인 문제를 하나 꼽으라고 하면 읽기 전용 복제본의 `max_allowed_packet` 파라미터 값이 원본 DB 인스턴스의 `max_allowed_packet` 파라미터 값보다 작을 때입니다. `max_allowed_packet` 파라미터는 DB 파라미터 그룹에서 사용자가 직접 설정할 수 있는 파라미터로서 데 이터베이스에서 실행할 수 있는 DML 코드의 최대 크기를 지정하는 데 사용됩니다. 경우에 따라 원본 DB 인스턴스와 연결된 DB 파라미터 그룹에서 `max_allowed_packet` 파라미터 값이 원본의 읽기 전용 복제본과 연결된 DB 파라미터 그룹에서 `max_allowed_packet` 파라미터 값보다 작습니다. 이러한 경우에는 복제 프로세스에서 오류(패킷이 'max_allowed_packet' 바이트보다 큼)가 발생하여 복제가 중단될 수도 있습니다. 원본 및 읽기 전용 복제본이 같은 `max_allowed_packet` 파라미터 값을 가진 DB 파라미터 그룹을 사용하도록 하여 이 오류를 수정할 수 있습니다.

이밖에 복제 오류의 원인이 되는 공통적인 상황은 다음과 같습니다.

- 읽기 전용 복제본의 테이블에 쓰기 작업을 실행하는 경우. 읽기 전용 복제본에서 인덱스를 만들 때는 `read_only` 파라미터를 0으로 설정해야 합니다. 읽기 전용 복제본의 테이블에 데이터를 쓰면 복제가 중단될 수 있습니다.
- MyISAM 같은 비트랜잭션 스토리지 엔진을 사용할 때. 읽기 전용 복제본에는 트랜잭션 스토리지 엔진이 필요합니다. 복제는 MariaDB 10.2 이상의 InnoDB 스토리지 엔진 및 MariaDB 10.1 이전 XtraDB 스토리지 엔진에 대해서만 지원됩니다.
- `SYSDATE()`와 같이 안전하지 않은 비결정적 쿼리를 사용하는 경우. 자세한 내용은 [Determination of Safe and Unsafe Statements in Binary Logging](#)을 참조하십시오.

오류를 건너뛰어도 안전하다고 판단될 경우에는 [현재 복제 오류 넘어가기 \(p. 670\)](#) 섹션에 설명한 단계를 따르십시오. 그 밖에 읽기 전용 복제본을 삭제하고 엔드포인트가 이전 읽기 전용 복제본의 엔드포인트와 동일하게 유지되도록 같은 DB 인스턴스 식별자를 사용하여 인스턴스를 생성할 수도 있습니다. 복제 오류가 해결되면 Replication State가 replicating으로 변경됩니다.

MariaDB DB 인스턴스에서는 경우에 따라 읽기 전용 복제본을 보조 인스턴스로 전환하지 못할 수도 있습니다. 이는 일부 binlog 이벤트가 오류로 인해 플러시되지 않기 때문입니다. 이때는 수동으로 읽기 전용 복제본

을 삭제한 후 다시 생성해야 합니다. 다음 동적 변수 값을 설정하여 발생하는 이러한 가능성을 줄일 수 있습니다: `sync_binlog=1`, `innodb_flush_log_at_trx_commit=1` 및 `innodb_support_xa=1`. 단, 이 설정은 성능 감소의 원인이 될 수도 있으므로 변경 사항을 프로덕션 환경에 적용하기 전에 그 효과를 테스트하는 것이 좋습니다.

Amazon RDS MariaDB DB 인스턴스로의 GTID 기반 복제 구성

버전 10.0.24 이상을 실행하는 외부 MariaDB 인스턴스에서 Amazon RDS MariaDB DB 인스턴스로의 GTID 기반 복제를 설정할 수 있습니다. Amazon RDS에서 외부 복제 마스터와 복제본을 설정할 때 다음 지침을 준수해야 합니다.

- 사용자의 복제본인 Amazon RDS MariaDB DB 인스턴스에 대한 장애 조치 이벤트를 모니터링합니다. 장애 조치가 발생할 경우에는 사용자의 복제본인 DB 인스턴스가 다른 네트워크 주소를 가진 새 호스트에서 다시 생성될 수도 있습니다. 장애 조치 이벤트를 모니터링하는 자세한 방법은 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오.
- Binlog가 복제본에 적용되었음을 확인할 때까지는 마스터 인스턴스에서 binlog를 유지 관리합니다. 이렇게 유지 관리해야 오류 발생 시 마스터 인스턴스를 복원할 수 있습니다.
- Amazon RDS 상의 MariaDB DB 인스턴스에서 자동 백업을 활성화합니다. 자동 백업을 활성화하면 마스터 및 복제본을 다시 동기화할 필요가 있을 때 복제본을 특정 시점으로 복원할 수 있습니다. 백업과 특정 시점으로 복원에 대한 자세한 내용은 다음([Amazon RDS DB 인스턴스 백업 및 복원 \(p. 196\)](#))단원을 참조하십시오.

Note

Amazon RDS MariaDB DB 인스턴스에서 복제를 시작하는 데 필요한 권한은 제한되고 Amazon RDS 마스터 사용자는 사용할 수 없습니다. 따라서 Amazon RDS `mysql.rds_set_external_master_gtid (p. 474)` 및 `mysql.rds_start_replication (p. 688)` 명령을 사용하여 라이브 데이터베이스와 Amazon RDS MariaDB 데이터베이스 사이의 복제를 설정해야 합니다.

외부 마스터 인스턴스와 Amazon RDS의 MariaDB DB 인스턴스 간에 복제를 시작하려면 다음 절차를 수행하십시오.

복제를 시작하려면

- 원본 MariaDB 인스턴스를 읽기 전용으로 설정합니다.

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

- 외부 MariaDB 인스턴스의 현재 GTID를 가져옵니다. 이를 위해 mysql 또는 원하는 쿼리 편집기를 사용하여 `SELECT @@gtid_current_pos;`를 실행할 수 있습니다.

GTID는 <domain-id>-<server-id>-<sequence-id> 형식입니다. 일반적인 GTID는 **0-1234510749-1728**과 비슷한 형식입니다. GTID 및 그 구성 요소에 대한 자세한 내용은 MariaDB 설명서의 [전역 트랜잭션 ID](#)를 참조하십시오.

- `mysqldump`를 사용하여 외부 MariaDB 인스턴스에서 Amazon RDS MariaDB DB 인스턴스로 데이터베이스를 복사합니다. 매우 큰 데이터베이스의 경우, [자동 중지 시간을 단축하여 Amazon RDS MySQL MariaDB DB 인스턴스로 데이터 가져오기 \(p. 628\)](#)에서 이 절차를 사용하고 싶을 것입니다.

Note

-p 옵션과 입력한 암호 사이에 공백이 없어야 합니다.

Linux, OS X, Unix의 경우:

```
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>
```

Windows의 경우:

```
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>
```

Amazon RDS MariaDB DB 인스턴스에 연결하기 위해 호스트 이름, 사용자 이름, 포트 및 암호를 지정하려면 mysql 명령에서 --host, --user (-u), --port 및 -p 옵션을 사용합니다. 호스트 이름은 Amazon RDS MariaDB DB 인스턴스 엔드포인트의 DNS 이름입니다(예: myinstance.123456789012.us-east-1.rds.amazonaws.com). Amazon RDS Management Console의 인스턴스 세부 정보에서 엔드포인트 값을 찾을 수 있습니다.

4. 원본 MariaDB 인스턴스를 다시 쓰기 가능한 상태로 만듭니다.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

5. Amazon RDS Management Console에서 Amazon RDS MariaDB DB 인스턴스에 대한 VPC 보안 그룹에 외부 MariaDB 데이터베이스를 호스팅하는 서버의 IP 주소를 추가합니다. VPC 보안 그룹 수정에 관한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서에서 [VPC의 보안 그룹](#) 단원을 참조하십시오.

외부 MariaDB 인스턴스와 통신할 수 있도록, Amazon RDS MariaDB DB 인스턴스 IP 주소에서의 연결을 허용하도록 로컬 네트워크를 구성해야 할 수도 있습니다. Amazon RDS MariaDB DB 인스턴스의 IP 주소를 찾으려면 host 명령을 사용합니다.

```
host <RDS_MariaDB_DB_host_name>
```

호스트 이름은 Amazon RDS MariaDB DB 인스턴스 엔드포인트의 DNS 이름입니다.

6. 선택한 클라이언트를 사용하여 외부 MariaDB 인스턴스에 연결하고 복제에 사용될 MariaDB 사용자를 만듭니다. 이 계정은 오직 복제용으로만 사용되며 보안 향상을 위해 사용자의 도메인으로 제한되어야 합니다. 다음은 그 한 예입니다.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY '<password>';
```

7. 외부 MariaDB 인스턴스의 경우 복제 사용자에게 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여합니다. 예를 들어 도메인의 'repl_user' 사용자를 위해 모든 데이터베이스에서 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여하려면 다음 명령을 실행합니다.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.*  
TO 'repl_user'@'mydomain.com'  
IDENTIFIED BY '<password>';
```

8. Amazon RDS MariaDB DB 인스턴스를 복제본으로 만듭니다. 마스터 사용자로서 Amazon RDS MariaDB DB 인스턴스에 연결하고 [mysql.rds_set_external_master_gtid \(p. 474\)](#) 명령을 사용하여 외부 MariaDB 데이터베이스를 복제 마스터로 식별합니다. 2단계에서 확인된 GTID를 사용하십시오. 다음은 그 한 예입니다.

```
CALL mysql.rds_set_external_master_gtid ('mymasterserver.mydomain.com', 3306,  
'repl_user', '<password>', '<GTID>', 0);
```

9. Amazon RDS MariaDB DB 인스턴스에서 [mysql.rds_start_replication \(p. 688\)](#) 명령을 실행하여 복제를 시작합니다.

```
CALL mysql.rds_start_replication;
```

MariaDB DB 인스턴스로 데이터 가져오기

다음에서는 MariaDB를 실행하는 Amazon RDS DB 인스턴스로 MariaDB 데이터를 가져오는 방법에 대한 정보를 찾을 수 있습니다.

최초로 MariaDB DB 인스턴스로 데이터를 가져오려면, 다음과 같이 [Amazon RDS MySQL DB 인스턴스로 백업 복원 \(p. 619\)](#)에서 설명하는 프로시저를 사용할 수 있습니다.

- Amazon RDS MySQL DB 인스턴스, 사용자의 Amazon RDS MariaDB DB 인스턴스와 동일한 VPC에 설치된 Amazon Elastic Compute Cloud(Amazon EC2) 내의 MariaDB 또는 MySQL 인스턴스, MariaDB 또는 MySQL의 스몰 온프레미스 인스턴스로부터 데이터를 가져오려면 [MySQL 또는 MariaDB DB에서 Amazon RDS MySQL 또는 MariaDB DB 인스턴스로 데이터 가져오기 \(p. 626\)](#) 단원에서 설명하는 프로시저를 사용할 수 있습니다.
- MariaDB 또는 MySQL의 라지 또는 프로덕션 온프레미스 인스턴스로부터 데이터를 가져오려면 [가동 중지 시간을 단축하여 Amazon RDS MySQL MariaDB DB 인스턴스로 데이터 가져오기 \(p. 628\)](#) 단원에서 설명하는 프로시저를 사용할 수 있습니다.
- Amazon RDS MariaDB DB 인스턴스와 다른 VPC에 설치된 EC2의 MariaDB 또는 MySQL 인스턴스로부터 데이터를 가져오거나, 구분 기호로 분리된 텍스트 파일을 출력할 수 있는 데이터 원본으로부터 데이터를 가져오려면 [임의의 소스에서 MySQL 또는 MariaDB DB 인스턴스로 데이터 가져오기 \(p. 640\)](#)에서 설명하는 프로시저를 사용할 수 있습니다.

AWS Database Migration Service(AWS DMS)를 사용하여 데이터를 Amazon RDS DB 인스턴스로 가져올 수 있습니다. AWS DMS는 가동 중지 없이 데이터베이스를 마이그레이션할 수 있으며 대부분의 데이터베이스 엔진에서는 대상 데이터베이스로 전환할 준비가 될 때까지 지속적으로 복제를 계속할 수 있습니다. 이를 사용하여 동일한 데이터베이스 엔진 또는 다른 데이터베이스 엔진에서 MariaDB로 마이그레이션할 수 있습니다. 다른 데이터베이스 엔진에서 마이그레이션하는 경우 AWS Schema Conversion Tool을 사용하여 AWS DMS를 통해 마이그레이션되지 않는 스키마 객체를 마이그레이션할 수 있습니다. AWS DMS에 대한 자세한 정보는 [AWS Database Migration Service란 무엇입니까?](#) 단원을 참조하십시오.

외부 인스턴스가 MariaDB 버전 10.0.24 이상인 경우 MariaDB GTID(글로벌 트랜잭션 식별자)를 사용하거나, MySQL 인스턴스 또는 10.0.2보다 이전 버전의 MariaDB 인스턴스인 경우 이진 로그 좌표를 사용하여 Amazon RDS MariaDB DB 인스턴스로 복제를 구성할 수 있습니다. MariaDB GTID는 MySQL GTID와 다르게 구현되며, MySQL GTID는 Amazon RDS에서 지원되지 않습니다.

MariaDB DB 인스턴스로의 복제를 구성하려면 다음 프로시저를 사용할 수 있습니다.

- 외부 MySQL 인스턴스, 또는 10.0.24보다 이전 버전을 실행하는 외부 MariaDB 인스턴스에서 MariaDB DB 인스턴스로의 복제를 구성하려면 [Amazon RDS 외부에서 실행 중인 MySQL 또는 MariaDB 인스턴스를 사용한 복제 \(p. 655\)](#)에서 설명하는 절차를 사용할 수 있습니다.
- 버전 10.0.24 이상을 실행하는 외부 MariaDB 인스턴스에서 MariaDB DB 인스턴스로의 복제를 구성하려면 [Amazon RDS MariaDB DB 인스턴스로의 GTID 기반 복제 구성 \(p. 464\)](#)에서 설명하는 프로시저를 사용할 수 있습니다.

Note

mysql 시스템 데이터베이스에는 DB 인스턴스에 로그인하고 데이터에 액세스하는 데 필요한 인증 및 권한 부여 정보가 포함되어 있습니다. DB 인스턴스에 있는 mysql 데이터베이스의 각종 테이블, 데이터 또는 기타 콘텐츠를 삭제하거나 변경하거나 이름을 바꾸거나 자르면 오류가 발생하여 DB 인스턴스와 데이터에 액세스할 수 없게 될 수 있습니다. 이 문제가 발생할 경우 AWS CLI `restore-db-instance-from-db-snapshot` 명령을 사용하여 DB 인스턴스를 스냅샷에서 복원하거나 `restore-db-instance-to-point-in-time` 명령을 사용하여 DB 인스턴스를 복구할 수 있습니다.

MariaDB 데이터베이스 엔진을 위한 옵션

이번 부록에서는 MariaDB DB 엔진 기반 Amazon RDS 인스턴스의 옵션과 추가 기능에 대해 설명합니다. 이러한 옵션을 활성화하려면 먼저 사용자 지정 옵션 그룹에 추가한 다음 옵션 그룹과 DB 인스턴스를 연동시켜야 합니다. 옵션 그룹 작업에 대한 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오.

Amazon RDS는 다음의 MariaDB 옵션을 지원합니다.

옵션 ID	엔진 버전
MARIADB_AUDIT_PLUGIN	MariaDB 10.0.24 이상

MariaDB 감사 플러그인 지원

Amazon RDS는 MariaDB 데이터베이스 인스턴스에서의 MariaDB 감사 플러그인 사용을 지원합니다. MariaDB 감사 플러그인은 사용자의 데이터베이스 로그온, 데이터베이스에 대해 실행되는 쿼리 등의 데이터베이스 활동을 기록합니다. 데이터베이스 활동 기록은 로그 파일에 저장됩니다.

감사 플러그인 옵션 설정

Amazon RDS는 MariaDB 감사 플러그인 옵션의 다음 설정을 지원합니다.

옵션 설정	유효한 값	기본값	설명
SERVER_AUDIT_FILE_NAME	/rdsdbdata/log/audit/	/rdsdbdata/log/audit/	로그 파일의 위치. 로그 파일에는 SERVER_AUDIT_EVENTS에서 지정된 활동 기록이 포함되어 있습니다. 자세한 내용은 데이터베이스 로그 파일 보기 및 나열 (p. 307) 및 MariaDB 데이터베이스 로그 파일 (p. 310) 단원을 참조하십시오.
SERVER_AUDIT_SIZE	1000000000	1000000000	도달 시 파일 로테이션을 초래하는 바이트 크기. 자세한 내용은 로그 파일 크기 (p. 314) 단원을 참조하십시오.

옵션 설정	유효한 값	기본값	설명
SERVER_AUDIT_ ON ¹⁰⁰ _ROTATION	9		저장할 로그 루테이션 수. 자세한 내용은 로그 파일 크기 (p. 314) 및 데이터베이스 로그 파일 다운로드 (p. 307) 단원을 참조하십시오.
SERVER_AUDIT_ CONNECT, QUERY, TABLE	CONNECT, QUERY		<p>로그에 기록할 활동 유형. MariaDB 감사 플러그인 설치 자체가 로깅됩니다.</p> <ul style="list-style-type: none"> CONNECT: 성공/실패한 데이터베이스 연결과 데이터베이스 연결 해제를 로깅합니다. QUERY: 데이터베이스에 대해 실행된 모든 쿼리의 텍스트를 로깅합니다. TABLE: 데이터베이스에 대해 쿼리가 실행될 때 쿼리의 영향을 받는 테이블을 로깅합니다. <p>MariaDB에서는 CONNECT, QUERY 및 TABLE을 지원합니다.</p> <p>MySQL에서는 CONNECT 및 QUERY를 지원합니다.</p>
SERVER_AUDIT_ 에서의 ^{으로} 분리된 값	없음		<p>지정된 사용자들의 활동만을 포함하십시오. 기본적으로 모든 사용자의 활동이 기록됩니다. 사용자가 SERVER_AUDIT_EXCL_USERS 및 SERVER_AUDIT_INCL_USERS에서 모두 지정된 경우, 해당 사용자에 대해 활동이 기록됩니다.</p>
SERVER_AUDIT_ 에서의 ^{으로} 분리된 값	없음		<p>지정된 사용자들의 활동을 제외하십시오. 기본적으로 모든 사용자의 활동이 기록됩니다. 사용자가 SERVER_AUDIT_EXCL_USERS 및 SERVER_AUDIT_INCL_USERS에서 모두 지정된 경우, 해당 사용자에 대해 활동이 기록됩니다.</p> <p>rdsadmin 사용자는 데이터베이스 상태를 확인하기 위해 1초마다 데이터베이스에 쿼리를 요청합니다. 다른 설정에 따라 이 활동은 로그 파일의 크기를 아주 빨리 대폭 증가시킬 수 있습니다. 이 활동을 기록할 필요가 없는 경우, rdsadmin 사용자를 SERVER_AUDIT_EXCL_USERS 목록에 추가하십시오.</p>
SERVER_AUDIT_LOGGING	ON		<p>로깅이 활성화되었습니다. 유효한 값은 ON뿐입니다. Amazon RDS는 로깅 비활성화를 지원하지 않습니다. 로깅을 비활성화하려면 MariaDB 감사 플러그인을 제거하십시오. 자세한 내용은 MariaDB 감사 플러그인 제거하기 (p. 469) 단원을 참조하십시오.</p>
SERVER_AUDIT_ 00247483647IMI1024			레코드에서 쿼리 문자열의 길이 제한.

MariaDB 감사 플러그인 추가하기

MariaDB 감사 플러그인을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동시킵니다.

MariaDB 감사 플러그인을 추가한 후 DB 인스턴스를 재시작할 필요가 없습니다. 옵션 그룹이 활성화되자마자 감사가 즉시 시작됩니다.

MariaDB 감사 플러그인을 추가하려면,

1. 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않다면 사용자 지정 DB 옵션 그룹을 생성합니다. [Engine]의 경우 [mariadb]를 선택하고, [Major engine version]의 경우 [10.0] 이상을 선택합니다. 자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.
2. MARIADB_AUDIT_PLUGIN 옵션을 옵션 그룹에 추가하고 옵션 설정을 구성하십시오. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#)을(를) 참조하십시오. 각 설정에 대한 자세한 내용은 [감사 플러그인 옵션 설정 \(p. 467\)](#) 단원을 참조하십시오.
3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다.
 - 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 내용은 [MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 437\)](#) 단원을 참조하십시오.
 - 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 자세한 내용은 [MariaDB 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 449\)](#) 단원을 참조하십시오.

MariaDB 감사 플러그인 로그 보기 및 다운로드

MariaDB 감사 플러그인을 활성화한 후 다른 텍스트 기반 로그 파일에 액세스하는 것과 동일한 방식으로 그 파일의 결과에 액세스할 수 있습니다. 감사 로그 파일은 /rdsdbdata/log/audit/에 있습니다. 콘솔에서 로그 파일 보기에 대한 자세한 내용은 [데이터베이스 로그 파일 보기 및 나열 \(p. 307\)](#)을(를) 참조하십시오. 로그 파일 다운로드에 대한 자세한 내용은 [데이터베이스 로그 파일 다운로드 \(p. 307\)](#)을(를) 참조하십시오.

MariaDB 감사 플러그인 설정 수정

MariaDB 감사 플러그인을 활성화한 후 플러그인 설정을 수정할 수 있습니다. 옵션 설정을 변경하는 방법에 대한 자세한 내용은 [옵션 설정 변경 \(p. 156\)](#)을(를) 참조하십시오. 각 설정에 대한 자세한 내용은 [감사 플러그인 옵션 설정 \(p. 467\)](#) 단원을 참조하십시오.

MariaDB 감사 플러그인 제거하기

Amazon RDS는 MariaDB 감사 플러그인에서의 로깅 끄기를 지원하지 않습니다. 다만 DB 인스턴스에서 플러그인을 제거할 수는 있습니다. MariaDB 감사 플러그인을 제거할 때 DB 인스턴스가 자동으로 재시작하여 감사가 중지됩니다.

MariaDB 감사 플러그인을 DB 인스턴스에서 제거하려면 다음 중 하나를 실행하십시오.

- MariaDB 감사 플러그인이 속한 옵션 그룹에서 MariaDB 감사 플러그인 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- DB 인스턴스를 수정하고, 플러그인이 포함되지 않은 다른 옵션 그룹을 지정합니다. 이 변경은 단일 DB 인스턴스에 영향을 미칩니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있!

습니다. 자세한 내용은 [MariaDB 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 449\)](#) 단원을 참조하십시오.

MariaDB에 대한 파라미터

기본적으로, MariaDB DB 인스턴스는 MariaDB 데이터베이스에만 해당되는 DB 파라미터 그룹을 사용합니다. 이 파라미터 그룹에는 MySQL 데이터베이스 엔진용 Amazon RDS DB 파라미터 그룹의 파라미터 중 일부만 포함되어 있습니다. 또한 몇 개의 새로운 MariaDB 고유 파라미터도 포함되어 있습니다. 다음의 MySQL 파라미터는 MariaDB 고유의 DB 파라미터 그룹에서 사용할 수 없습니다.

- 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

MySQL 5.6 파라미터에 대한 자세한 내용은 [MySQL 5.6 설명서](#)를 참조하십시오.

MariaDB 고유의 DB 파라미터 그룹에는 다음과 같이 MariaDB에만 적용되는 파라미터도 포함되어 있습니다. 수정 가능한 파라미터의 허용 범위는 따로 지정된 경우를 제외하고 MariaDB 설명서에 지정된 범위와 동일합니다. Amazon RDS MariaDB 파라미터는 사용자가 선택한 스토리지 엔진의 기본값으로 설정됩니다.

- 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

Amazon RDS MariaDB는 NORMAL, OFF 및 QUICK 값을 지원하고, FORCE 또는 BACKUP 값은 지원하지 않습니다.

- 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 버전 10.1 이상)
- deadlock_search_depth_long
- deadlock_search_depth_short
- deadlock_timeout_long

- deadlock_timeout_short
- explicit_defaults_for_timestamp(MariaDB 버전 10.1 이상)
- extra_max_connections
- extra_port
- 의견
- 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 버전 10.1 이상)
- innodb_background_scrub_data_compressed(MariaDB 버전 10.1 이상)
- innodb_background_scrub_data_interval(MariaDB 버전 10.1 이상)
- innodb_background_scrub_data_uncompressed(MariaDB 버전 10.1 이상)
- innodb_buf_dump_status_frequency(MariaDB 버전 10.1 이상)
- innodb_buffer_pool_populate
- innodb_cleaner_lsn_age_factor
- innodb_compression_algorithm(MariaDB 버전 10.1 이상)
- innodb_corrupt_table_action
- innodb_defragment(MariaDB 버전 10.1 이상)
- innodb_defragment_fill_factor(MariaDB 버전 10.1 이상)
- innodb_defragment_fill_factor_n_recs(MariaDB 버전 10.1 이상)
- innodb_defragment_frequency(MariaDB 버전 10.1 이상)
- innodb_defragment_n_pages(MariaDB 버전 10.1 이상)
- innodb_defragment_stats_accuracy(MariaDB 버전 10.1 이상)
- innodb_empty_free_List_algorithm
- innodb_fake_changes
- innodb_fatal_semaphore_wait_threshold(MariaDB 버전 10.1 이상)
- innodb_foreground_preflush
- innodb_idle_flush_pct(MariaDB 버전 10.1 이상)
- innodb_immediate_scrub_data_uncompressed(MariaDB 버전 10.1 이상)
- innodb_instrument_semaphores(MariaDB 버전 10.1 이상)
- 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 버전 10.1 이상)

- innodb_sched_priority_cleaner
- innodb_scrub_log(MariaDB 버전 10.1 이상)
- innodb_scrub_log_speed(MariaDB 버전 10.1 이상)
- 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 버전 10.1 이상)
- join_buffer_space_limit
- join_cache_level
- key_cache_file_hash_size
- key_cache_segments
- max_digest_length(MariaDB 버전 10.1 이상)
- max_statement_time(MariaDB 버전 10.1 이상)
- mysql56_temporal_format(MariaDB 버전 10.1 이상)
- progress_report_time
- query_cache_strip_comments
- replicate_annotate_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 버전 10.1 이상)
- slave_parallel_threads
- slave_run_triggers_for_rbr(MariaDB 버전 10.1 이상)
- 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

MariaDB 파라미터에 대한 자세한 내용은 [MariaDB 설명서](#)를 참조하십시오.

Amazon RDS SQL의 MariaDB 참조

이 부록에서는 MariaDB DB 엔진을 실행 중인 Amazon RDS 인스턴스에 사용할 수 있는 시스템 저장 프로시저를 설명합니다.

Amazon RDS MySQL DB 인스턴스에 사용할 수 있는 모든 시스템 저장 프로시저를 MariaDB DB 인스턴스에도 사용할 수 있습니다. 이러한 저장 프로시저는 [Amazon RDS SQL의 MySQL 참조 \(p. 676\)](#)에 소개되어 있습니다.

또한 다음은 MariaDB를 실행 중인 Amazon RDS DB 인스턴스에 대해서만 지원되는 시스템 저장 프로시자입니다.

- [mysql.rds_set_external_master_gtid \(p. 474\)](#)
- [mysql.rds_kill_query_id \(p. 476\)](#)

mysql.rds_set_external_master_gtid

Amazon RDS 외부에서 실행 중인 MariaDB 인스턴스에서 Amazon RDS MariaDB DB 인스턴스로의 GTID 기반 복제를 구성합니다. 이 프로시저는 외부 MariaDB 인스턴스가 버전 10.0.24 이상인 경우에만 지원됩니다. 두 인스턴스 중 하나 또는 모두가 MariaDB 전역 트랜잭션 ID(GTID)를 지원하지 않는 복제를 설정하는 경우 다음([mysql.rds_set_external_master \(p. 678\)](#))을 사용하십시오.

복제 시 GTID를 사용하면 이진 로그 복제에서는 제공하지 않는 충돌 안정성 기능을 제공하므로 복제 인스턴스가 지원하는 경우에는 GTID 기반 복제를 사용하는 것이 좋습니다.

구문

```
CALL mysql.rds_set_external_master_gtid(
    host_name
    , host_port
    , replication_user_name
    , replication_user_password
    , gtid
    , ssl_encryption
);
```

파라미터

host_name

문자열. 복제 마스터가 될 Amazon RDS 외부에서 실행 중인 MariaDB 인스턴스의 호스트 이름 또는 IP 주소입니다.

host_port

정수. 복제 마스터로 구성될 Amazon RDS 외부에서 실행 중인 MariaDB 인스턴스에서 사용하는 포트입니다. 네트워크 구성에 포트 번호를 변환하는 SSH 포트 복제가 포함되는 경우 SSH에 의해 공개되는 포트 이름을 지정하십시오.

replication_user_name

문자열. 읽기 전용 복제본으로 구성될 MariaDB DB 인스턴스에서 REPLICATION SLAVE 권한이 있는 사용자의 ID입니다.

replication_user_password

문자열. replication_user_name에 지정된 사용자 ID의 암호입니다.

gtid

문자열. 복제가 시작해야 하는 마스터에서의 전역 트랜잭션 ID.

복제를 구성하는 동안 복제 마스터가 잠긴다면 @@gtid_current_pos를 사용하여 현재 GTID를 가져올 수 있습니다. 그러므로 GTID를 가져오는 시점과 복제가 시작하는 시점 사이에서 이진 로그가 변경되지 않습니다.

또는, 복제를 시작하기 전에 mysqldump 버전 10.0.13 이상을 사용하여 슬레이브 인스턴스를 채우는 경우 --master-data 또는 --dump-slave 옵션을 사용하여 출력 내 GTID 위치를 가져올 수 있습니다. mysqldump 버전 10.0.13 이상을 사용하지 않는 경우에는 SHOW MASTER STATUS를 실행하거나 동일한 mysqldump 옵션을 사용하여 이진 로그 파일 이름 및 위치를 가져온 다음, 외부 MariaDB 인스턴스에서 BINLOG_GTID_POS옵션을 실행해 GTID로 변환할 수 있습니다.

```
SELECT BINLOG_GTID_POS('<binary log file name>', <binary log file position>);
```

GTID의 MariaDB 구현에 대한 자세한 내용은 MariaDB 설명서에서 [전역 트랜잭션 ID](#)를 참조하십시오.

ssl_encryption

정수. 이 옵션은 현재 구현되지 않습니다. 기본값은 0입니다.

사용 시 주의사항

mysql.rds_set_external_master_gtid 프로시저는 마스터 사용자가 실행해야 합니다. Amazon RDS 외부에서 실행 중인 MariaDB DB 인스턴스의 복제 슬레이브로 구성하려는 MariaDB DB 인스턴스에서 실행해야 합니다. mysql.rds_set_external_master_gtid를 실행하기 전에 Amazon RDS 외부에서 실행 중인 MariaDB의 인스턴스를 복제 마스터로 구성해야 합니다. 자세한 내용은 [MariaDB DB 인스턴스로 데이터 가져오기 \(p. 466\)](#) 단원을 참조하십시오.

Warning

mysql.rds_set_external_master_gtid를 사용하여 두 Amazon RDS DB 인스턴스 간 복제를 관리하지 마십시오. RDS 외부에서 실행 중인 MariaDB 인스턴스를 복제할 때만 사용합니다. Amazon RDS DB 인스턴스 간 복제 관리에 대한 자세한 내용은 [읽기 전용 복제본 작업 \(p. 135\)](#) 주제를 참조하십시오.

mysql.rds_set_external_master_gtid를 호출하여 Amazon RDS DB 인스턴스를 읽기 전용 복제본으로 구성한 후 복제본에서 mysql.rds_start_replication (p. 688)을 호출하여 복제 프로세스를 시작할 수 있습니다. mysql.rds_reset_external_master (p. 685)를 호출하여 읽기 전용 복제본 구성을 제거할 수 있습니다.

mysql.rds_set_external_master_gtid가 호출되면 Amazon RDS에서 mysql.rds_history 및 mysql.rds_replication_status 테이블에 시간, 사용자 및 "set master" 작업을 기록합니다.

예제

MariaDB DB 인스턴스에서 다음 예제를 실행하면 MariaDB DB 인스턴스가 Amazon RDS 외부에서 실행 중인 MariaDB 인스턴스의 복제 슬레이브로 구성됩니다.

```
call mysql.rds_set_external_master_gtid
('Sourcedb.some.com',3306,'ReplicationUser','SomePassWOrd','0-123-456',0);
```

관련 항목

- [mysql.rds_reset_external_master](#) (p. 685)
- [mysql.rds_start_replication](#) (p. 688)
- [mysql.rds_stop_replication](#) (p. 691)

mysql.rds_kill_query_id

MariaDB 서버에서 실행 중인 쿼리를 종료합니다.

구문

```
CALL mysql.rds_kill_query_id(queryID);
```

파라미터

queryID

정수. 종료할 쿼리의 ID입니다.

사용 시 주의사항

MariaDB 서버에서 실행 중인 쿼리를 종료하려면 `mysql.rds_kill_query_id` 프로시저를 사용하여 해당 쿼리의 ID를 전달합니다. 쿼리 ID를 가져오려면, 다음과 같이 MariaDB [Information Schema PROCESSLIST 테이블](#)을 쿼리합니다

```
SELECT USER, HOST, COMMAND, TIME, STATE, INFO, QUERY_ID FROM
INFORMATION_SCHEMA.PROCESSLIST WHERE USER = '<user name>';
```

MariaDB 서버와의 연결은 유지됩니다.

관련 주제

- [mysql.rds_kill](#) (p. 698)
- [mysql.rds_kill_query](#) (p. 698)

예

다음 예제는 쿼리 ID가 230040인 쿼리를 종료합니다.

```
call mysql.rds_kill_query_id(230040);
```

Amazon RDS의 Microsoft SQL Server

Amazon RDS는 Microsoft SQL Server의 여러 버전 및 에디션을 실행하는 DB 인스턴스를 지원합니다. 다음은 각 메이저 버전의 지원되는 최신 버전입니다. 지원되는 버전, 에디션 및 RDS 엔진 버전의 전체 목록은 [Amazon RDS에서의 버전 및 기능 지원 \(p. 483\)](#) 단원을 참조하십시오.

- 2019년 1월 1일 [KB4466404](#)에 따라 SQL Server 2017 RTM(CU13) 14.00.3049.10이 릴리스되었습니다.
- 2018년 9월 20일 [KB4466404](#)에 따라 SQL Server 2016 SP2 (CU3) 13.00.5216.0이 릴리스되었습니다.
- 2018년 1월 16일 [KB4052725](#)에 따라 SQL Server 2014 SP2 CU10 12.00.5571.0이 릴리스되었습니다.
- 2017년 1월 12일 [KB4057116](#)에 따라 SQL Server 2012 SP4 GDR 11.00.7462.6이 릴리스되었습니다.
- 2018년 1월 6일 [KB4057113](#)에 따라 SQL Server 2008 R2 SP3 GDR 10.50.6560.0이 릴리스되었습니다. 이 릴리스는 미국 동부(오하이오), 캐나다(중부), EU(런던) AWS 리전에서 사용할 수 없습니다. 사용되지 않습니다! 이 버전을 설치하지 마십시오. 자동 업그레이드를 피하기 위해 2019년 6월 1일 이전에 기존 인스턴스를 업그레이드하십시오.

SQL Server 라이선스에 대한 자세한 내용은 [Amazon RDS의 Microsoft SQL Server 라이선싱 \(p. 493\)](#) 단원을 참조하십시오. SQL Server 빌드에 대한 자세한 내용은 [최신 SQL Server 빌드](#)에 관한 Microsoft 지원 문서를 참조하십시오.

Amazon RDS를 통해 DB 인스턴스 및 DB 스냅샷, 특정 시점으로 복원 및 자동 또는 수동 백업을 만들 수 있습니다. SQL Server를 실행 중인 DB 인스턴스를 VPC 내에서 사용할 수 있습니다. 또한, SSL을 사용하여 SQL Server를 실행하는 DB 인스턴스에 연결할 수 있고, TDE를 사용하여 유 휴 데이터를 암호화 할 수 있습니다. Amazon RDS는 현재 SQL Server 데이터베이스 미러링(DBM) 또는 상시 가동 가용성 그룹(AG)을 고가용성, 장애 조치 솔루션으로 사용하여 SQL Server의 다중 AZ 배포를 지원합니다.

관리되는 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 셀 액세스를 제공하지 않으며, 고급 권한을 필요로 하는 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다. Amazon RDS는 Microsoft SQL Server Management Studio와 같은 표준 SQL 클라이언트 애플리케이션을 사용하여 DB 인스턴스의 데이터베이스에 대한 액세스를 지원합니다. Amazon RDS는 Telnet, Secure Shell (SSH), 또는 Windows 원격 데스크톱 연결을 통해 DB 인스턴스에 직접 호스트 액세스하는 것을 허용하지 않습니다. DB 인스턴스를 생성할 때 사용자는 해당 인스턴스의 모든 데이터베이스에 대한 db_owner 역할을 할당받게 되며, 백업을 위해 사용된 권한을 제외한 모든 데이터베이스 수준의 권한을 갖게 됩니다. Amazon RDS가 사용자를 위해 백업을 관리합니다.

첫 번째 DB 인스턴스를 생성하기 전에 이 설명서의 설정 섹션에 나오는 단계를 완료해야 합니다. 자세한 내용은 [Amazon RDS 설정 \(p. 5\)](#) 단원을 참조하십시오.

Amazon RDS에서 Microsoft SQL Server에 대한 일반 관리 작업

다음은 Amazon RDS SQL Server DB 인스턴스로 수행하는 일반적인 관리 작업과 각 작업에 해당하는 문서 링크입니다.

작업 영역	관련 문서
인스턴스 클래스, 스토리지 및 PIOPS 프로덕션 목적으로 DB 인스턴스를 만들 경우에는 Amazon RDS에서 인스턴스 클래스, 스토리지 유형 및 프로비저닝된 IOPS이 작동하는 방식을 이해해야 합니다.	Microsoft SQL Server를 위한 DB 인스턴스 클래스 지원 (p. 481) Amazon RDS 스토리지 유형 (p. 101)
다중 AZ 배포 프로덕션 DB 인스턴스에서는 다중 AZ 배포를 사용해야 합니다. 다중 AZ 배포는 DB 인스턴스를 위해 향상된 가용성, 데이터 내구성 및 내결 함성을 제공합니다. SQL Server용 다중 AZ 배포는 SQL Server의 기본 DBM 또는 AG 기술을 사용하여 구현됩니다.	Amazon RDS를 위한 고가용성(다중 AZ) (p. 109) Microsoft SQL Server 데이터베이스 미러링 또는 상시 가동 가용성 그룹을 사용하여 다중 AZ 배포 (p. 489)
Amazon Virtual Private Cloud(VPC) AWS 계정에 기본 VPC가 있는 경우에는 DB 인스턴스가 기본 VPC 내부에 자동으로 생성됩니다. 계정에 기본 VPC가 없는데 VPC 안에 DB 인스턴스를 만들려면 VPC와 서브넷 그룹을 만든 후 DB 인스턴스를 만들어야 합니다.	EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 (p. 407) VPC에서 Amazon RDS DB 인스턴스를 사용한 작업 (p. 415)
보안 그룹 기본적으로, DB 인스턴스와 함께 인스턴스에 대한 액세스를 막는 방화벽도 생성됩니다. 따라서 DB 인스턴스에 액세스하기 위한 알맞은 IP 주소와 네트워크 구성으로 보안 그룹을 만들어야 합니다. 만들어야 할 보안 그룹은 DB 인스턴스가 어떤 Amazon EC2 플랫폼에 있고, Amazon EC2 인스턴스에서 DB 인스턴스에 액세스하는지 여부에 따라 결정됩니다. 일반적으로 DB 인스턴스가 EC2-Classic 플랫폼에 있으면 DB 보안 그룹을 생성해야 하고, DB 인스턴스가 EC2-VPC 플랫폼에 있으면 VPC 보안 그룹을 생성해야 합니다.	EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 (p. 407) 보안 그룹을 통한 액세스 제어 (p. 390)
파라미터 그룹 DB 인스턴스에 특정 데이터베이스 파라미터가 필요할 경우, 파라미터 그룹을 만든 후 DB 인스턴스를 만들어야 합니다.	DB 파라미터 그룹 작업 (p. 161)
옵션 그룹 DB 인스턴스에 특정 데이터베이스 옵션이 필요할 경우, 옵션 그룹을 만든 후 DB 인스턴스를 만들어야 합니다.	Microsoft SQL Server 데이터베이스 엔진의 옵션 (p. 544)
DB 인스턴스에 연결 보안 그룹을 만들고 이를 DB 인스턴스에 연결한 후, Microsoft SQL Server Management Studio와 같은 표준 SQL 클라이언트 애플리케이션을 사용하여 DB 인스턴스에 연결할 수 있습니다.	Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 (p. 503)
백업 및 복원 DB 인스턴스를 생성할 때 자동 백업을 하도록 구성할 수 있습니다. 또한 전체 백업 파일(.bak 파일)을 사용하여 데이터베이스를 수동으로 백업 및 복원할 수도 있습니다.	백업 작업 (p. 196) SQL Server 데이터베이스 가져오기 및 내보내기 (p. 520)
모니터링	DB 인스턴스 측정치 보기 (p. 236)

작업 영역	관련 문서
CloudWatch Amazon RDS 측정치, 이벤트 및 향상된 모니터링 기능을 통해 SQL Server DB 인스턴스를 모니터링할 수 있습니다.	Amazon RDS 이벤트 보기 (p. 305)
로그 파일 SQL Server DB 인스턴스의 로그 파일에 액세스할 수 있습니다.	Amazon RDS 데이터베이스 로그 파일 (p. 306) Microsoft SQL Server 데이터베이스 로그 파일 (p. 318)

SQL Server DB 인스턴스 작업을 위한 고급 관리 작업도 있습니다. 자세한 내용은 다음 설명서를 참조하세요.

- [Microsoft SQL Server에 대한 일반 DBA 작업 \(p. 554\)](#)를 선택합니다.
- [SQL Server DB 인스턴스를 통한 Windows 인증 사용 \(p. 568\)](#)
- [tempdb 데이터베이스에 액세스 \(p. 555\)](#)

Microsoft SQL Server DB 인스턴스에 대한 제한

DB 인스턴스에서 Amazon RDS의 Microsoft SQL Server를 구현하려면 다음과 같은 몇 가지 제한 사항을 정확히 파악하고 있어야 합니다.

- DB 인스턴스에 지원되는 최대 데이터베이스 수는 인스턴스 클래스 유형과 가용성 모드—단일 AZ, 다중 AZ 데이터베이스 미러링(DBM) 또는 다중 AZ 가용성 그룹(AG)에 따라 다릅니다. Microsoft SQL Server 시스템 데이터베이스는 이 제한에 포함되지 않습니다.

다음 표는 각 인스턴스 클래스 유형 및 가용성 모드에 대해 지원되는 최대 데이터베이스 수를 보여줍니다. 이 표를 사용하면 하나의 인스턴스 클래스 유형에서 다른 유형으로 또는 한 가용성 모드에서 다른 모드로 이동할 수 있는지 알 수 있습니다. 원본 DB 인스턴스에 대상 인스턴스 클래스 유형이나 가용성 모드에서 지원할 수 있는 것보다 많은 데이터베이스가 있으면 DB 인스턴스 수정에 실패합니다. 이벤트 창에서 요청 상태를 확인할 수 있습니다.

인스턴스 클래스 유형	단일 AZ	DBM 기능이 있는 다중 AZ	상시 가동 AG 기능이 있는 다중 AZ
db.*.large	30	30	30
db.*.xlarge를 db.*.16xlarge로	100	50	75
db.*.24xlarge	100	50	100

* 서로 다른 인스턴스 클래스 유형을 나타냅니다.

예를 들어 DB 인스턴스가 단일 AZ를 사용한 db.*.16xlarge에서 실행되고 76개의 데이터베이스가 있다고 가정해봅시다. 다중 AZ 상시 가동 AG를 사용하여 업그레이드할 DB 인스턴스를 수정합니다. DB 인스턴스에 대상 구성에서 지원할 수 있는 것보다 많은 데이터베이스가 포함되어 있으므로 이 업그레이드는 실패합니다. 대신 인스턴스 클래스 유형을 db.*.24xlarge로 업그레이드하면 수정이 완료됩니다.

업그레이드에 실패하면 다음과 유사한 이벤트 및 메시지가 표시됩니다.

- 데이터베이스 인스턴스 클래스를 수정할 수 없습니다. 인스턴스에는 76개의 데이터베이스가 있지만 변환 후에는 75개만 지원합니다.

- DB 인스턴스를 다중 AZ로 변환할 수 없습니다. 인스턴스에 76개의 데이터베이스가 있지만 변환 후에는 75개만 지원합니다.

특정 시점으로 복원 또는 스냅샷 복원에 실패하면 다음과 유사한 이벤트 및 메시지가 표시됩니다.

- 데이터베이스 인스턴스가 호환되지 않는 복원으로 전환됩니다. 인스턴스에는 76개의 데이터베이스가 있지만 변환 후에는 75개만 지원합니다.
- 일부 포트는 Amazon RDS용으로 예약되어 있고 DB 인스턴스를 생성할 때 사용할 수 없습니다.
- SQL Server용 Amazon RDS는 msdb 데이터베이스로 데이터 가져오기를 지원하지 않습니다.
- SQL Server 다중 AZ 배포의 DB 인스턴스에 있는 데이터베이스의 이름은 바꿀 수 없습니다.
- SQL Server DB 인스턴스를 위한 최대 스토리지 크기는 다음과 같습니다.
 - 범용(SSD) 스토리지 – 모든 에디션에서 16TiB
 - 프로비저닝된 IOPS 스토리지 – 모든 에디션에서 16TiB
 - 마그네틱 스토리지 – 모든 에디션에서 1TiB

대량의 스토리지가 필요한 상황에서는 여러 DB 인스턴스에서 샤딩을 사용하여 이 제한을 우회할 수 있습니다. 이 접근 방식에서는 샤딩된 시스템에 연결하는 애플리케이션에 데이터 종속적인 라우팅 논리가 필요합니다. 기존 샤딩 프레임워크를 사용하거나 샤딩을 활성화하는 사용자 지정 코드를 작성할 수 있습니다. 기존 프레임워크를 사용하는 경우, 이 프레임워크는 DB 인스턴스와 같은 서버에 어떤 구성 요소도 설치할 수 없습니다.

- SQL Server DB 인스턴스를 위한 최소 스토리지 크기는 다음과 같습니다.
 - 범용 스토리지(SSD) – Enterprise 및 Standard Edition일 때 200GiB, Web 및 Express Edition일 때 20GiB
 - 프로비저닝된 IOPS 스토리지 – Enterprise 및 Standard Edition일 때 200GiB, Web 및 Express Edition일 때 100GiB
 - 마그네틱 스토리지 – Enterprise 및 Standard Edition일 때 200GiB, Web 및 Express Edition일 때 20GiB
- Amazon RDS는 Amazon RDS DB 인스턴스와 동일한 서버에서 이러한 서비스를 실행하는 것을 지원하지 않습니다.
 - SQL Server Analysis Services
 - SQL Server Integration Services
 - SQL Server Reporting Services
 - 데이터 품질 서비스
 - 마스터 데이터 서비스

이러한 기능을 사용하려면 Amazon EC2 인스턴스에 SQL Server를 설치하거나 온프레미스 SQL Server 인스턴스를 사용하는 것이 좋습니다. 이러한 경우 EC2 또는 SQL Server 인스턴스는 Amazon RDS에서 SQL Server DB 인스턴스의 보고, 분석, 통합 또는 마스터 데이터 서비스 서버 역할을 합니다. Microsoft 라이선싱 정책에 따라 SQL Server를 Amazon EBS 스토리지가 포함된 Amazon EC2 인스턴스에 설치할 수 있습니다.

- Microsoft SQL Server의 제한 사항 때문에, DROP DATABASE가 성공적으로 실행된 시점 이전의 시점으로 복원해도 그 시점에서 해당 데이터베이스의 상태가 반영되지 않을 수도 있습니다. 예를 들어, 끊긴 데이터베이스는 일반적으로 DROP DATABASE 명령이 실행되기 전 최대 5분까지 상태로 복원됩니다. 이 유형의 복원은 끊긴 데이터베이스에서 몇 분 동안 수행된 트랜잭션을 복원할 수 없음을 뜻합니다. 이 문제를 피하려면 복원 작업이 완료된 후 DROP DATABASE 명령을 다시 실행하면 됩니다. 데이터베이스를 삭제하면 그 데이터베이스에 대한 트랜잭션 로그가 삭제됩니다.

Microsoft SQL Server를 위한 DB 인스턴스 클래스 지원

DB 인스턴스의 계산 및 메모리 용량은 해당 DB 인스턴스 클래스에 의해 결정됩니다. 필요한 DB 인스턴스 클래스는 DB 인스턴스의 처리력 및 메모리 요구 사항에 따라 다릅니다. 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.

편의를 위해 Microsoft SQL Server에 지원되는 다음 DB 인스턴스 클래스 목록이 여기에서 제공됩니다. 최신 목록은 RDS 콘솔(<https://console.aws.amazon.com/rds/>)을 참조하십시오.

SQL Server Edition	2017 및 2016 지원 범위	2014, 2012, 2008 R2 지원 범위
Enterprise Edition	db.m4.xlarge-db.m4.16xlarge db.r3.xlarge-db.r3.8xlarge db.r4.xlarge-db.r4.16xlarge db.m5.xlarge-db.m5.24xlarge	db.m4.xlarge-db.m4.10xlarge db.r3.xlarge-db.r3.8xlarge db.r4.xlarge-db.r4.8xlarge db.m5.xlarge-db.m5.24xlarge
Standard Edition	db.m4.large-db.m4.16xlarge db.r4.large-db.r4.16xlarge db.m5.large-db.m5.24xlarge	db.m4.large-db.m4.10xlarge db.m3.medium-db.m3.2xlarge db.r3.large-db.r3.8xlarge db.r4.large-db.r4.8xlarge db.m5.large-db.m5.24xlarge
Web Edition	db.t2.small-db.t2.medium db.m4.large-db.m4.4xlarge db.r4.large-db.r4.2xlarge db.m5.large-db.m5.4xlarge	db.t2.small-db.t2.medium db.m4.large-db.m4.4xlarge db.m3.medium-db.m3.2xlarge db.r3.large-db.r3.2xlarge db.r4.large-db.r4.2xlarge db.m5.large-db.m5.4xlarge
Express Edition	db.t2.micro-db.t2.medium	db.t2.micro-db.t2.medium

Microsoft SQL Server 보안

Microsoft SQL Server 데이터베이스 엔진은 역할 기반 보안을 사용합니다. DB 인스턴스를 만들 때 사용하는 마스터 사용자 이름은 `processadmin`, `public` 및 `setupadmin` 고정 서버 역할의 구성원인 SQL Server 인증 로그인입니다.

데이터베이스를 만드는 사용자는 누구든 해당 데이터베이스에 대한 `db_owner` 역할에 할당되며, 백업에 사용되는 권한을 제외한 모든 데이터베이스 수준의 권한을 갖게 됩니다. Amazon RDS가 자동으로 백업을 관리합니다.

Amazon RDS에서는 현재 다음과 같은 서버 수준 역할을 사용할 수 없습니다.

- bulkadmin
- dbcreator
- diskadmin
- securityadmin
- serveradmin
- sysadmin

SQL Server DB 인스턴스에서는 다음과 같은 서버 수준 권한을 사용할 수 없습니다.

- ADMINISTER BULK OPERATIONS
- ALTER ANY CREDENTIAL
- ALTER ANY EVENT NOTIFICATION
- ALTER ANY EVENT SESSION
- ALTER ANY SERVER AUDIT
- ALTER RESOURCES
- ALTER SETTINGS(DB 파라미터 그룹 API 작업을 사용하여 파라미터를 수정할 수 있습니다. 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.)
- AUTHENTICATE SERVER
- CONTROL_SERVER
- CREATE DDL EVENT NOTIFICATION
- CREATE ENDPOINT
- CREATE TRACE EVENT NOTIFICATION
- EXTERNAL ACCESS ASSEMBLY
- SHUTDOWN(RDS 재부팅 옵션을 대신 사용할 수 있음)
- UNSAFE ASSEMBLY
- ALTER ANY AVAILABILITY GROUP(SQL Server 2012만 해당)
- CREATE ANY AVAILABILITY GROUP(SQL Server 2012만 해당)

Microsoft SQL Server DB 인스턴스에 대한 규정 준수 프로그램 지원

AWS 범위 내 서비스는 외부 감사 기관의 철저한 평가를 거쳐 인증, 규정 준수 증명 또는 운영 권한(ATO)을 받았습니다. 자세한 내용은 [규정 준수 프로그램 제공 AWS 범위 내 서비스](#) 단원을 참조하십시오.

Microsoft SQL Server DB 인스턴스에 대한 HIPAA 지원

Microsoft SQL Server 데이터베이스용 Amazon RDS를 사용하여 HIPAA 인증 애플리케이션을 개발할 수 있습니다. 예를 들어 AWS와 체결하여 이행 중인 비즈니스 제휴 계약(AWS)에 따라 보호 대상 건강 정보(PHI)를 비롯한 의료 관련 정보를 저장할 수 있습니다. 자세한 내용은 [HIPAA 규정 준수](#) 단원을 참조하십시오.

SQL Server용 Amazon RDS는 다음과 같은 버전 및 에디션에서 HIPAA를 지원합니다.

- SQL Server 2017, 2016, 2014 및 2012: Enterprise, Standard 및 Web Edition
- SQL Server 2008 R2: Enterprise Edition

DB 인스턴스에서 HIPAA 지원을 활성화하려면 다음과 같이 세 가지 구성 요소를 설정해야 합니다.

구성 요소	세부 정보
감사	감사를 설정하려면 파라미터 <code>rds.sqlserver_audit</code> 을 값 <code>fedramp_hipaa</code> 로 설정합니다. DB 인스턴스가 아직 사용자 지정 DB 파라미터 그룹을 사용하지 않는 경우에는 <code>rds.sqlserver_audit</code> 파라미터를 수정하기 전에 먼저 사용자 그룹 파라미터를 생성한 후 DB 인스턴스에 연결해야 합니다. 자세한 내용은 DB 파라미터 그룹 작업 (p. 161) 단원을 참조하십시오.
전송 데이터 암호화	전송 데이터 암호화를 설정하려면 모든 DB 인스턴스 연결에 강제로 SSL(Secure Sockets Layer)을 사용해야 합니다. 자세한 내용은 DB 인스턴스 연결이 SSL을 사용하도록 지정 (p. 541) 단원을 참조하십시오.
저장 데이터 암호화	저장 데이터 암호화의 설정은 다음과 같이 두 가지 옵션이 있습니다. <ol style="list-style-type: none">Enterprise Edition을 실행하는 경우에는 저장 데이터 암호화에 TDE(Transparent Data Encryption)를 사용하도록 선택할 수 있습니다. 자세한 내용은 Microsoft SQL Server Transparent Data Encryption(TDE) 지원 (p. 547) 단원을 참조하십시오.저장 데이터 암호화는 AWS Key Management Service(AWS KMS) 암호화 키를 사용하여 설정할 수 있습니다. 자세한 내용은 Amazon RDS 리소스 암호화 (p. 385) 단원을 참조하십시오.

Microsoft SQL Server 인스턴스를 위한 SSL 지원

SSL을 사용하여 애플리케이션과 Microsoft SQL Server를 실행하는 Amazon RDS DB 인스턴스 사이의 연결을 암호화할 수 있습니다. 또한 DB 인스턴스에 대한 모든 연결에서 SSL을 사용하도록 지정할 수도 있습니다. 연결이 SSL을 사용하도록 지정하면 클라이언트에 투명하게 발생하며, 클라이언트는 SSL 사용을 위해 작업을 수행할 필요가 없습니다.

SSL은 지원되는 모든 SQL Server 버전에 대해 모든 AWS 리전에서 사용할 수 있습니다. 자세한 내용은 [Microsoft SQL Server DB 인스턴스와 함께 SSL 사용 \(p. 541\)](#) 단원을 참조하십시오.

Amazon RDS에서의 버전 및 기능 지원

새 DB 인스턴스를 생성할 때는 현재 지원되는 모든 Microsoft SQL Server 버전을 지정할 수 있습니다. Microsoft SQL Server 메이저 버전(예: Microsoft SQL Server 14.00) 및 지정된 메이저 버전에 대해 지원되는 모든 마이너 버전을 지정할 수 있습니다. 버전이 지정되지 않은 경우 Amazon RDS는 지원되는 버전(보통 최신 버전)을 기본값으로 설정합니다. 메이저 버전이 지정되었지만 마이너 버전이 지정되지 않은 경우, Amazon RDS는 고객이 지정한 메이저 버전의 최근 릴리스를 기본값으로 설정합니다. 지원되는 버전 목록과 새로 만든 DB 인스턴스의 기본값을 보려면 `describe-db-engine-versions` AWS CLI 명령을 사용합니다.

Amazon RDS에서 지원되는 Microsoft SQL Server 2017 버전

Amazon RDS는 다음과 같은 SQL Server 2017 버전을 지원합니다.

- SQL Server 2017 RTM (CU13) 14.00.3049.1, 모든 에디션 및 모든 AWS 리전
RDS API EngineVersion 및 CLI engine-version: 14.00.3049.1.v1
- SQL Server 2017 RTM CU13 14.00.3015.40, 모든 에디션 및 모든 AWS 리전
RDS API EngineVersion 및 CLI engine-version: 14.00.3015.40.v1

- 버전 14.00.1000.169, RTM, 모든 에디션 및 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 14.00.1000.169.v1

SQL Server 2017은 다음과 같은 새 기능을 많이 제공합니다.

- 적응형 쿼리 처리
- 자동 계획 설정
- 그래프DB
- 다시 시작 가능한 인덱스 재작성

SQL Server 2012의 전체 기능 목록은 Microsoft 설명서의 [What's New in SQL Server 2017](#) 단원을 참조하십시오.

지원되지 않는 기능 목록은 [지원되지 않는 기능과 지원이 제한된 기능 \(p. 488\)](#) 단원을 참조하십시오.

Amazon RDS에서 지원되는 Microsoft SQL Server 2016 버전

Amazon RDS는 다음과 같은 SQL Server 2016 버전을 지원합니다.

- SQL Server 2016 SP2 (CU3) 13.00.5216.0, 모든 에디션 및 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 13.00.5216.0.v1

- SQL Server 2016 SP1 CU7 13.00.4466.4, 모든 에디션 및 모든 AWS 리전..

RDS API EngineVersion 및 CLI engine-version: 13.00.4466.4.v1

- 버전 13.00.4451.0, SP1 CU5, 모든 에디션 및 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 13.00.4451.0.v1

- 버전 13.00.4422.0, SP1 CU2, 모든 에디션 및 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 13.00.4422.0.v1

- 버전 13.00.2164.0, RTM CU2, 모든 에디션 및 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 13.00.2164.0.v1

Amazon RDS에서 지원되는 Microsoft SQL Server 2014 버전

Amazon RDS는 다음과 같은 SQL Server 2014 버전을 지원합니다.

- SQL Server 2014 SP2 CU10 12.00.5571.0, 모든 에디션 및 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 12.00.5571.0.v1

- 버전 12.00.5546.0, SP2 CU5, 모든 에디션 및 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 12.00.5546.0.v1

- 버전 12.00.5000.0, SP2, 모든 에디션 및 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 12.00.5000.0.v1

- 버전 12.00.4422.0, SP1 CU2, Enterprise Edition을 제외한 모든 에디션 및 캐나다(중부) 및 EU(런던)를 제외한 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 12.00.4422.0.v1

SQL Server 2012의 지원되는 기능 외에, Amazon RDS에서는 SQL Server 2014에 제공되는 새 쿼리 최적화 프로그램과 지연된 내구성 기능을 지원합니다.

지원되지 않는 기능 목록은 [지원되지 않는 기능과 지원이 제한된 기능 \(p. 488\)](#) 단원을 참조하십시오.

SQL Server 2014는 SQL Server 2012의 모든 파라미터를 지원하며 동일한 기본값을 사용합니다. SQL Server 2014에는 백업 체크섬 기본값이라는 새 파라미터가 하나 포함되어 있습니다. 자세한 내용은 Microsoft 설명서의 [How to enable the CHECKSUM option if backup utilities do not expose the option](#) 단원을 참조하십시오.

Amazon RDS에서 지원되는 Microsoft SQL Server 2012 버전

Amazon RDS는 다음과 같은 SQL Server 2012 버전을 지원합니다.

- SQL Server 2012 SP4 GDR 11.00.7462.6, 모든 에디션 및 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 11.00.7462.6.v1

- 버전 11.00.6594.0, SP3 CU8, 모든 에디션 및 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 11.00.6594.0.v1

- 버전 11.00.6020.0, SP3, 모든 에디션 및 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 11.00.6020.0.v1

- 버전 11.00.5058.0, SP2, 모든 에디션 및 미국 동부(오하이오), 캐나다(중부), EU(런던)를 제외한 모든 AWS 리전

RDS API EngineVersion 및 CLI engine-version: 11.00.5058.0.v1

SQL Server 2012에 대한 자세한 내용은 Microsoft 설명서의 [Features Supported by the Editions of SQL Server 2012](#) 단원을 참조하십시오.

지원되는 SQL Server 2008 R2의 기능 외에 Amazon RDS에서는 다음과 같은 SQL Server 2012 기능을 지원합니다.

- Columnstore 인덱스(Enterprise Edition)
- XML, varchar(max), nvarchar(max) 및 varbinary(max) 데이터 형식을 위한 온라인 인덱스 만들기, 리빌드 및 삭제(Enterprise Edition)
- 유연한 서버 역할
- 서비스 브로커는 지원되나, 서비스 브로커 엔드포인트는 지원 안 됩
- 부분적으로 포함된 데이터베이스
- 시퀀스
- 투명한 데이터 암호화(Enterprise Edition만 해당)
- THROW 문
- 신규 및 향상된 공간 형식
- UTF-16 지원
- ALTER ANY SERVER ROLE 서버 수준 권한

지원되지 않는 기능 목록은 [지원되지 않는 기능과 지원이 제한된 기능 \(p. 488\)](#) 단원을 참조하십시오.

SQL Server 2012에서는 일부 SQL Server 파라미터가 변경되었습니다.

- SQL Server 2012에서 `awe enabled`, `precompute rank` 및 `sql mail xps` 파라미터가 제거되었습니다. SQL Server DB 인스턴스에서는 이런 파라미터를 수정할 수 없었으며, 이런 파라미터를 제거해도 SQL Server 사용에는 아무런 영향도 미치면 안 됩니다.
- SQL Server 2012에서 새로운 `contained database authentication` 파라미터가 부분적으로 포함된 데이터베이스를 지원합니다. 이 파라미터를 활성화한 다음 부분적으로 포함된 데이터베이스를 만들면 마스터 데이터베이스 대신 부분적으로 포함된 데이터베이스 내에 권한이 있는 사용자의 사용자 이름과 암호가 저장됩니다. 부분적으로 포함된 데이터베이스에 대한 자세한 내용은 Microsoft 설명서의 [포함된 데이터베이스](#) 단원을 참조하십시오.

Amazon RDS에서 사용되지 않는 Microsoft SQL Server 2008 R2 버전

Warning

2019년 6월 1일부터 SQL Server 2008 R2를 계속 사용하는 기존 인스턴스를 모두 최신 마이너 버전의 SQL Server 2012로 업그레이드할 예정입니다. 자세한 내용은 [Amazon RDS의 Microsoft SQL Server 엔진 버전 관리 \(p. 487\)](#) 단원을 참조하십시오.

Amazon RDS는 모든 인스턴스의 업그레이드가 완료될 때까지 다음 버전의 SQL Server 2008 R2를 지원합니다.

- 2018년 1월 6일 [KB4057113](#)에 따라 SQL Server 2008 R2 SP3 GDR 10.50.6560.0이 릴리스되었습니다. 이 버전은 미국 동부(오하이오), 캐나다(중부), EU(런던) AWS 리전에서 사용할 수 없습니다.

RDS API `EngineVersion` 및 CLI `engine-version: 10.50.6560.0.v1`

- 버전 10.50.6529.0, SP3 QFE, 모든 에디션 및 미국 동부(오하이오), 캐나다(중부), EU(런던)를 제외한 모든 AWS 리전.

RDS API `EngineVersion` 및 CLI `engine-version: 10.50.6529.0.v1`

- 버전 10.50.6000.34, SP3, 모든 에디션 및 미국 동부(오하이오), 캐나다(중부), EU(런던)를 제외한 모든 AWS 리전

RDS API `EngineVersion` 및 CLI `engine-version: 10.50.6000.34.v1`

- 버전 10.50.2789.0, SP1, 모든 에디션 및 미국 동부(오하이오), 캐나다(중부), EU(런던)를 제외한 모든 AWS 리전

RDS API `EngineVersion` 및 CLI `engine-version: 10.50.2789.0.v1`

SQL Server 2008 R2에 대한 자세한 내용은 Microsoft 설명서의 [Features Supported by the Editions of SQL Server 2008 R2](#) 단원을 참조하십시오.

Amazon RDS에서는 다음과 같은 SQL Server 2008 R2 기능을 지원합니다.

- 주요 데이터베이스 엔진 기능
- SQL Server 개발 도구:
 - Visual Studio 통합
 - IntelliSense
- SQL Server 관리 도구:
 - SQL Server Management Studio(SMS)
 - sqlcmd

- SQL Server 프로파일러(클라이언트 쪽 추적 기능. 서버 쪽에 사용 가능한 차선책)
- SQL Server Migration Assistant(SSMA)
- 데이터베이스 엔진 투닝 관리자
- SQL Server 에이전트
- 안전한 CLR
- 전체 텍스트 검색(시맨틱 검색 제외)
- SSL
- 투명한 데이터 암호화(Enterprise Edition만 해당)
- 공간 및 위치 기능
- 서비스 브로커는 지원되나, 서비스 브로커 앤드포인트는 지원 안 됨
- 변경 내용 추적
- 데이터베이스 미러링(DBM) 또는 상시 가동 가용성 그룹(AG)
- Amazon RDS SQL DB 인스턴스를 별도의 서버에서 실행하는 보고, 분석 및 통합 서비스의 데이터 소스로 사용하는 기능

지원되지 않는 기능 목록은 [지원되지 않는 기능과 지원이 제한된 기능 \(p. 488\)](#) 단원을 참조하십시오.

Amazon RDS의 Microsoft SQL Server 엔진 버전 관리

Amazon RDS에는 DB 인스턴스가 패치되거나 업그레이드되는 시기와 방법을 제어할 수 있는 유연한 버전 관리가 포함됩니다. 이렇게 하면 DB 엔진에 대해 다음을 수행할 수 있습니다.

- 데이터베이스 엔진 패치 버전과의 호환성 유지
- 새로운 패치 버전을 테스트하여 프로덕션에 배포하기 전에 애플리케이션에서 잘 작동하는지 확인
- 서비스 수준 계약 및 타이밍 요구 사항을 충족하도록 버전 업그레이드 계획 및 수행

Amazon RDS의 Microsoft SQL Server 엔진 패치

Amazon RDS는 공식 Microsoft SQL Server 데이터베이스 패치를 정기적으로 Amazon RDS에 해당하는 DB 인스턴스 엔진 버전으로 집계합니다. 각 엔진 버전의 Microsoft SQL Server 패치에 대한 자세한 내용은 [Amazon RDS 기반 버전 및 기능 지원](#)을 참조하십시오.

현재 DB 인스턴스에서 모든 엔진 업그레이드를 수동으로 실행합니다. 자세한 내용은 [Microsoft SQL Server DB 엔진 업그레이드 \(p. 516\)](#) 단원을 참조하십시오.

Amazon RDS 기반 Microsoft SQL Server의 메이저 엔진 버전의 사용 중단 일정

다음 표에는 Microsoft SQL Server에 예정된 메이저 엔진 버전의 사용 중단 일정이 표시됩니다.

날짜	정보
2019년 6월 1일	Amazon RDS 팀은 Microsoft SQL Server 2008 R2에 대한 Amazon RDS 지원을 중단할 계획입니다. 2019년 6월 1일부터 Microsoft SQL Server 2008 R2의 나머지 인스턴스를 SQL Server 2012로 마이그레이션하기 시작합니다(최신 마이너 버전 사용 가능).

날짜	정보
	Microsoft SQL Server 2008 R2에서 자동으로 업그레이드되지 않도록, 편리하게 한 번에 업그레이드할 수 있습니다. 자세한 내용은 DB 인스턴스 엔진 버전 업그레이드 (p. 118) 단원을 참조하십시오.
2019년 4월 25일	2019년 4월 말 이전에는 더 이상 Microsoft SQL Server 2008 R2를 사용하여 SQL Server 데이터베이스 인스턴스에 대한 새로운 Amazon RDS를 만들 수 없습니다.

Microsoft SQL Server DB 인스턴스에 대한 변경 데이터 캡처 지원

Amazon RDS는 Microsoft SQL Server를 실행하는 DB 인스턴스에 대해 CDC(변경 데이터 캡처)를 지원합니다. CDC는 테이블의 데이터에 수행된 변경 사항을 캡처하고 각 변경 사항에 대해 나중에 액세스할 수 있는 메타데이터를 저장합니다. 자세한 내용은 Microsoft 설명서의 [변경 데이터 캡처](#) 단원을 참조하십시오.

Amazon RDS는 다음 SQL Server 에디션과 버전에 대해 CDC를 지원합니다.

- Microsoft SQL Server Enterprise Edition(모든 버전)
- Microsoft SQL Server Standard Edition:
 - 2017
 - 2016 버전 13.00.4422.0 SP1 CU2 이상

Amazon RDS DB 인스턴스에 CDC를 사용하려면 먼저 RDS의 저장 프로시저를 사용하여 데이터베이스 수준에서 CDC를 활성화하거나 비활성화합니다. 그런 후에는 해당 데이터베이스에 대해 db_owner 역할이 있는 모든 사용자가 기본 Microsoft 저장 프로시저를 사용하여 해당 데이터베이스에서 CDC를 제어할 수 있습니다. 자세한 내용은 [변경 데이터 캡처 사용 \(p. 561\)](#) 단원을 참조하십시오.

CDC와 AWS Database Migration Service를 사용하여 SQL Server DB 인스턴스에서 지속적 복제를 활성화 할 수 있습니다.

지원되지 않는 기능과 지원이 제한된 기능

다음 Microsoft SQL Server 기능은 Amazon RDS에서 지원되지 않습니다.

- Stretch 데이터베이스
- Microsoft Azure Blob Storage로 백업
- 버퍼 풀 확장
- BULK INSERT 및 OPENROWSET(BULK...) 기능
- 데이터 품질 서비스
- 데이터베이스 로그 전달
- 데이터베이스 메일
- 분산 트랜잭션 코디네이터(MSDTC)
- 파일 테이블
- FILESTREAM 지원
- 유지 관리 계획
- 성능 데이터 수집기
- 정책 기반 관리
- PolyBase

- 기계 학습 및 R 서비스(설치를 위해 OS 액세스 필요)
- 복제
- 리소스 거버너
- SQL Server 감사
- 서버 수준 트리거
- 서비스 브로커 엔드포인트
- T-SQL 엔드포인트(CREATE ENDPOINT를 사용하는 모든 작업을 사용할 수 없음)
- WCF 데이터 서비스

다음 Microsoft SQL Server 기능은 Amazon RDS에서 지원이 제한적입니다.

- 분산 쿼리/연결된 서버. 자세한 내용은 [Implementing Linked Servers with Amazon RDS for Microsoft SQL Server](#) 단원을 참조하십시오.

Microsoft SQL Server 데이터베이스 미러링 또는 상시 가동 가용성 그룹을 사용하여 다중 AZ 배포

Amazon RDS는 SQL Server 데이터베이스 미러링(DBM) 또는 상시 가동 가용성 그룹(AG)을 사용하여 Microsoft SQL Server 기반 DB 인스턴스의 다중 AZ 배포를 지원합니다. 다중 AZ 배포는 DB 인스턴스를 위해 향상된 가용성, 데이터 내구성 및 내결함성을 제공합니다. 계획된 데이터베이스 유지 관리 또는 예기치 않은 서비스 중단이 발생할 경우 Amazon RDS가 최신 보조 복제본으로 자동으로 장애 조치를 수행하므로 수동 개입 없이 데이터베이스 작업을 빠르게 재개할 수 있습니다. 기본 인스턴스 및 보조 인스턴스는 동일한 엔드포인트를 사용합니다. 이 엔드포인트의 물리적 네트워크 주소는 장애 조치 프로세스의 일환으로 수동 보조 복제본으로 전환됩니다. 장애 조치가 발생하는 경우 애플리케이션을 다시 구성할 필요가 없습니다.

Amazon RDS는 다중 AZ 배포를 능동적으로 모니터링하면서 기본 인스턴스에 문제가 발생할 때 장애 조치를 시작하여 장애 조치를 관리합니다. 대기 및 기본 인스턴스가 완벽히 동기화되어 있지 않으면 장애 조치가 이루어지지 않습니다. Amazon RDS는 비정상 DB 인스턴스를 자동으로 복구하고 동기 복제를 다시 설정하여 다중 AZ 배포를 능동적으로 유지합니다. 따라서 Amazon RDS가 기본 인스턴스, 감시, 대기 인스턴스를 자동으로 처리하기 때문에 사용자가 따로 관리할 것이 없습니다. SQL Server 다중 AZ를 설정하면 RDS가 인스턴스의 모든 데이터베이스에 대해 수동 보조 인스턴스를 구성합니다.

자세한 내용은 [Microsoft SQL Server의 다중 AZ 배포 \(p. 537\)](#) 단원을 참조하십시오.

Transparent Data Encryption을 사용하여 유 휴 데이터 암호화

Amazon RDS는 저장되어 있는 데이터를 투명하게 암호화할 수 있는 Microsoft SQL Server Transparent Data Encryption(TDE)을 지원합니다. Amazon RDS는 옵션 그룹을 사용하여 이러한 기능을 활성화 및 구성합니다. TDE 옵션에 대한 자세한 내용은 [Microsoft SQL Server Transparent Data Encryption\(TDE\) 지원 \(p. 547\)](#) 단원을 참조하십시오.

Microsoft SQL Server DB 인스턴스의 현지 시간대

Microsoft SQL Server를 실행 중인 Amazon RDS DB 인스턴스의 시간대가 기본적으로 설정되어 있습니다. 현재 기본값은 협정 세계시(UTC)입니다. DB 인스턴스의 시간대를 애플리케이션의 시간대와 일치하도록 현지 시간대로 설정할 수 있습니다.

DB 인스턴스를 처음 만들 때 시간대를 설정합니다. [AWS Management 콘솔](#), Amazon RDS API [CreateDBInstance](#) 작업 또는 AWS CLI `create-db-instance` 명령을 사용하여 DB 인스턴스를 만들 수 있습니다.

DB 인스턴스가 다중 AZ 배포의 일부인 경우(SQL Server DBM 또는 AG 사용) 장애 조치 중에 시간대가 설정된 현지 시간대로 유지됩니다. 자세한 내용은 [Microsoft SQL Server 데이터베이스 미러링 또는 상시 가동 가용성 그룹을 사용하여 다중 AZ 배포 \(p. 489\)](#) 단원을 참조하십시오.

시점 복원을 요청할 경우 복원 시간을 UTC로 지정합니다. 복원 프로세스 중에 시간은 DB 인스턴스의 시간대로 변환됩니다. 자세한 내용은 [DB 인스턴스를 지정된 시간으로 복원 \(p. 221\)](#) 단원을 참조하십시오.

다음은 DB 인스턴스에 대해 현지 시간대를 설정할 때 적용되는 제한 사항입니다.

- 기존 SQL Server DB 인스턴스의 시간대를 수정할 수 없습니다.
- DB 인스턴스의 스냅샷을 다른 시간대의 DB 인스턴스로 복원할 수 없습니다.
- 한 표준 시간대의 백업 파일을 다른 표준 시간대로 복원하지 않는 것이 좋습니다. 한 표준 시간대의 백업 파일을 다른 표준 시간대로 복원하는 경우 쿼리와 애플리케이션을 감사하여 표준 시간대 변경의 영향을 확인해야 합니다. 자세한 내용은 [SQL Server 데이터베이스 가져오기 및 내보내기 \(p. 520\)](#) 단원을 참조하십시오.

지원되는 시간대

현지 시간대를 다음 표에 나열된 값 중 하나로 설정할 수 있습니다.

시간대	표준 시간 오프셋	설명	참고
아프가니스탄 표준시	(UTC+04:30)	카불	
알래스카 표준시	(UTC–09:00)	알래스카	
아라비아 표준시	(UTC+04:00)	아부다비, 무스카트	
대서양 표준시	(UTC–04:00)	대서양 표준시(캐나다)	
AUS 중부 표준시	(UTC+09:30)	다윈	
AUS 동부 표준시	(UTC+10:00)	캔버라, 멜버른, 시드니	
벨라루스 표준시	(UTC+03:00)	민스크	이 시간대는 일광 절약 시간을 따르지 않습니다.
캐나다 중부 표준시	(UTC–06:00)	서스캐처원	
카포베르데 표준시	(UTC–01:00)	카포베르데 섬	
중부 오스트레일리아 표준시	(UTC+09:30)	애들레이드	
중앙 아메리카 표준시	(UTC–06:00)	중앙 아메리카	
중앙 아시아 표준시	(UTC+06:00)	아스타나	
브라질 중부 표준시	(UTC–04:00)	쿠이아바	
중앙 유럽 표준시	(UTC+01:00)	베오그라드, 브라티슬라바, 부다페스트, 류블랴나, 프라하	

시간대	표준 시간 오프셋	설명	참고
중앙 유럽 표준시	(UTC+01:00)	사라예보, 스코페, 바르샤바, 자그레브	
중앙 태평양 표준시	(UTC+11:00)	솔로몬 제도, 뉴칼레도니아	
중부 표준시	(UTC–06:00)	중부 표준시(미국과 캐나다)	
중부 표준시(멕시코)	(UTC–06:00)	과달라하라, 멕시코 시티, 몬테레이	
중국 표준시	(UTC+08:00)	베이징, 총칭, 홍콩 특별 행정구, 우루무치	
E. 아프리카 표준시	(UTC+03:00)	나이로비	이 시간대는 일광 절약 시간을 따르지 않습니다.
E. 오스트레일리아 표준시	(UTC+10:00)	브리즈번	
E. 유럽 표준시	(UTC+02:00)	키시나우	
E. 남아메리카 표준시	(UTC–03:00)	브라질리아	
동부 표준시	(UTC–05:00)	동부 표준시(미국과 캐나다)	
그루지야 표준시	(UTC+04:00)	트빌리시	
GMT 표준시	(UTC)	더블린, 에든버러, 리스본, 런던	이 시간대는 그리니치 표준시(GMT)와 다릅니다. 이 시간대는 일광 절약 시간을 따릅니다.
그린란드 표준시	(UTC–03:00)	그린란드	
그리니치 표준시	(UTC)	몬로비아, 레이캬비크	이 시간대는 일광 절약 시간을 따르지 않습니다.
GTB 표준시	(UTC+02:00)	아테네, 부쿠레슈티	
하와이 표준시	(UTC–10:00)	하와이	
인도 표준시	(UTC+05:30)	첸나이, 콜카타, 룸바이, 뉴델리	
요르단 표준시	(UTC+02:00)	암만	
대한민국 표준시	(UTC+09:00)	서울	
중동 표준시	(UTC+02:00)	베이루트	
산지 표준시	(UTC–07:00)	산지 표준시(미국과 캐나다)	

시간대	표준 시간 오프셋	설명	참고
산지 표준시(멕시코)	(UTC-07:00)	치와와, 라파스, 마사 틀란	
미국 산지 표준시	(UTC-07:00)	애리조나	이 시간대는 일광 절약 시간을 따르지 않습니다.
뉴질랜드 표준시	(UTC+12:00)	오클랜드, 웰링턴	
뉴펀들랜드 표준시	(UTC-03:30)	뉴펀들랜드	
태평양 SA 표준시	(UTC-03:00)	산티아고	
태평양 표준시	(UTC-08:00)	태평양 표준시(미국과 캐나다)	
태평양 표준시(멕시코)	(UTC-08:00)	바하 캘리포니아	
러시아 표준시	(UTC+03:00)	모스크바, 상트페테르 부르크, 볼고그라드	이 시간대는 일광 절약 시간을 따르지 않습니다.
태평양 SA 표준시	(UTC-05:00)	보고타, 리마, 키토, 리 오 브郎코	이 시간대는 일광 절약 시간을 따르지 않습니다.
동남아시아 표준시	(UTC+07:00)	방콕, 하노이, 자카르 타	
싱가포르 표준시	(UTC+08:00)	쿠알라룸푸르, 싱가포 르	
도쿄 표준시	(UTC+09:00)	오사카, 삿포로, 도쿄	
미국 동부 표준시	(UTC-05:00)	인디애나(동부)	
UTC	UTC	협정 세계시	이 시간대는 일광 절약 시간을 따르지 않습니다.
UTC-02	(UTC-02:00)	협정 세계시-02	
UTC-08	(UTC-08:00)	협정 세계시-08	
UTC-09	(UTC-09:00)	협정 세계시-09	
UTC-11	(UTC-11:00)	협정 세계시-11	
UTC+12	(UTC+12:00)	협정 세계시+12	
W. 오스트레일리아 표준시	(UTC+08:00)	퍼스	
W. 중앙 아프리카 표준시	(UTC+01:00)	서중앙 아프리카	
W. 유럽 표준시	(UTC+01:00)	암스테르담, 베를린, 베른, 로마, 스톡홀름, 비엔나	

Amazon RDS의 Microsoft SQL Server 라이선싱

Microsoft SQL Server용으로 Amazon RDS DB 인스턴스를 설정하는 경우 소프트웨어 라이선스가 포함됩니다.

따라서 SQL Server 라이선스를 별도로 구매할 필요가 없습니다. AWS는 SQL Server 데이터베이스 소프트웨어에 대한 라이선스를 보유합니다. Amazon RDS 요금에는 소프트웨어 라이선스, 기본 하드웨어 리소스 및 Amazon RDS 관리 기능이 포함됩니다.

Amazon RDS는 다음 Microsoft SQL Server 에디션을 지원합니다.

- 엔터프라이즈
- 표준
- 웹
- Express

Note

SQL Server Web Edition 라이선스를 사용해야만 퍼블릭 및 인터넷으로 액세스 가능한 웹 페이지, 웹 사이트, 웹 애플리케이션 및 웹 서비스를 지원할 수 있습니다. 이 지원 레벨은 Microsoft의 사용 권한을 준수하는 데 필요합니다. 자세한 정보는 [AWS 서비스 약관](#)을 참조하십시오.

Amazon RDS는 SQL Server 데이터베이스 미러링(DBM) 또는 상시 가동 가용성 그룹(AG)을 사용하여 Microsoft SQL Server 기반 DB 인스턴스의 다중 AZ 배포를 지원합니다. 다중 AZ 배포에 대한 추가 라이선스 요구 사항은 없습니다. 자세한 내용은 [Microsoft SQL Server의 다중 AZ 배포 \(p. 537\)](#) 단원을 참조하십시오.

라이선스가 종료된 DB 인스턴스 복원

Amazon RDS는 라이선스 종료된 DB 인스턴스의 스냅샷을 생성합니다. 라이선스 문제로 인스턴스가 종료되는 경우 스냅샷에서 새 DB 인스턴스로 복원할 수 있습니다. 새 DB 인스턴스에는 라이선스가 포함되어 있습니다.

자세한 정보는 [라이선스가 종료된 DB 인스턴스 복원 \(p. 564\)](#) 단원을 참조하십시오.

개발 및 테스트

라이선스 요구 사항으로 인해 Amazon RDS에서는 SQL Server Developer 에디션을 제공할 수 없습니다. 여러 개발, 테스트 및 기타 비 프로덕션 요구에 대해 Express 버전을 사용할 수 있습니다. 하지만 엔터프라이즈급 SQL Server 설치의 전체 기능이 필요한 경우 전용 호스트 환경을 사용해야 합니다. Amazon EC2에서 SQL Server Developer 에디션(및 다른 MSDN 제품)을 다운로드하고 설치할 수 있습니다. Developer 버전에는 전용 인프라가 필요하지 않습니다. 고유 호스트를 사용하여 Amazon RDS에서 액세스할 수 없는 다른 프로그래밍 기능에 대한 액세스 권한을 부여할 수도 있습니다. SQL Server 버전 간의 차이에 대한 자세한 정보는 Microsoft 문서의 [SQL Server 2017의 버전과 지원하는 기능](#)을 참조하십시오.

관련 주제

- [Amazon RDS의 Microsoft SQL Server \(p. 477\)](#)
- [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 생성 \(p. 494\)](#)

Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 생성

Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. Amazon RDS DB 인스턴스가 온프레미스 Microsoft SQL Server와 유사합니다. SQL Server DB 인스턴스를 만든 후에 1개 이상의 사용자 지정 데이터베이스를 추가할 수 있습니다.

Important

DB 인스턴스를 생성하거나 DB 인스턴스에 연결하려면 먼저 [Amazon RDS 설정 \(p. 5\)](#) 섹션의 작업을 완료해야 합니다.

사용자가 샘플 DB 인스턴스를 만들어 연결하는 절차를 실습하는 예제는 [Microsoft SQL Server DB 인스턴스 생성 및 DB 인스턴스에 연결 \(p. 17\)](#) 단원을 참조하십시오.

AWS Management 콘솔

SQL Server DB 인스턴스를 시작하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. Amazon RDS 콘솔의 오른쪽 위 모서리에서 DB 인스턴스를 생성하려는 AWS 리전을 선택합니다.
3. 탐색 창에서 데이터베이스를 선택합니다.
탐색 창이 닫혀 있는 경우 왼쪽 상단의 메뉴 아이콘을 선택하여 여십시오.
4. 데이터베이스 생성을 선택하여 엔진 선택 페이지를 여십시오.
5. Microsoft SQL Server 아이콘을 선택합니다.

Select engine

Engine options

Amazon Aurora
Amazon Aurora

MySQL


MariaDB


PostgreSQL


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. 사용할 SQL Server DB 엔진 에디션을 선택합니다. 사용 가능한 SQL Server 버전은 AWS 리전마다 다릅니다.
7. 일부 에디션의 경우, 사용 사례 단계에서 생성하려는 DB 인스턴스를 프로덕션에 사용할지 묻는 메시지가 나타납니다. 사용할 계획이라면 [Production]을 선택합니다. 프로덕션을 선택하면 다음 단계에서 다음 설정이 모두 미리 선택됩니다.
 - 다중 AZ 장애 조치 옵션

- 프로비저닝된 IOPS 스토리지 옵션
- 삭제 방지 활성화 옵션

어떤 프로덕션 환경이든 이 기능을 권장합니다.

8. 다음을 선택하여 계속 진행합니다. [Specify DB Details] 페이지가 나타납니다.

[Specify DB Details] 페이지에서 DB 인스턴스 정보를 지정합니다. 각 설정에 대한 자세한 내용은 [Microsoft SQL Server DB 인스턴스 설정 \(p. 500\)](#) 단원을 참조하십시오.

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](#)

9. 다음을 선택하여 계속 진행합니다. [Configure Advanced Settings] 페이지가 나타납니다.

Configure Advanced Settings 페이지에서 Amazon RDS가 DB 인스턴스를 시작하는 데 필요한 추가 정보를 입력합니다. 각 설정에 대한 자세한 내용은 [Microsoft SQL Server DB 인스턴스 설정 \(p. 500\)](#) 단원을 참조하십시오.

Configure advanced settings

Network & Security

Virtual Private Cloud (VPC) [Info](#)
VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-2aed394c) ▼ 

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.

default ▼

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](#)

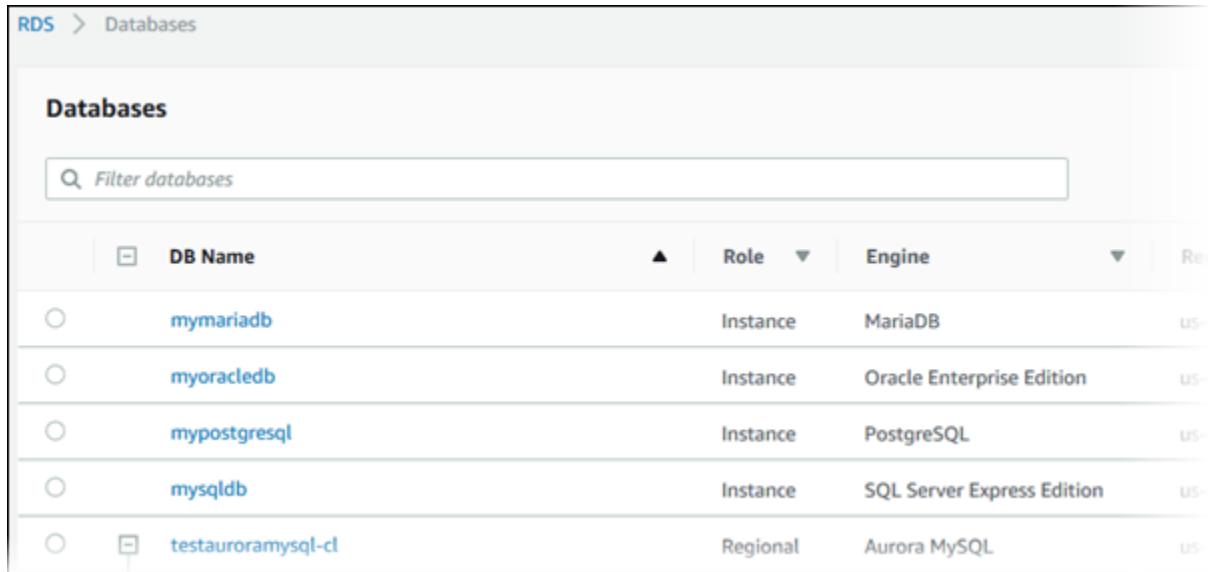
No preference ▼

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. [Launch DB Instance]를 선택합니다.
11. 마법사의 마지막 페이지에서 [Close]를 선택합니다.

RDS 콘솔의 DB 인터페이스 목록에 새 DB 인스턴스가 나타납니다. DB 인스턴스를 사용할 준비가 될 때까지 DB 인스턴스의 상태는 [creating]입니다. 상태가 available로 변경되면 DB 인스턴스에 연결할 수 있습니다. DB 인스턴스 클래스와 스토리지의 양에 따라 새 인스턴스를 사용할 수 있을 때까지 최대 20분이 걸릴 수 있습니다.



The screenshot shows the AWS RDS console with the 'Databases' page selected. At the top, there is a breadcrumb navigation: 'RDS > Databases'. Below the header, there is 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

CLI

AWS CLI를 사용해 Microsoft SQL Server DB 인스턴스를 생성하려면 다음 파라미터와 함께 [create-db-instance 명령을 호출합니다](#). 각 설정에 대한 자세한 내용은 [Microsoft SQL Server DB 인스턴스 설정 \(p. 500\)](#) 단원을 참조하십시오.

- `--db-instance-identifier`
- `--db-instance-class`
- `--db-security-groups`
- `--db-subnet-group`
- `--engine`
- `--master-user-name`
- `--master-user-password`
- `--allocated-storage`
- `--backup-retention-period`

Example

Linux, OS X, Unix의 경우:

```
aws rds create-db-instance
  --engine sqlserver-se \
  --db-instance-identifier mymssqlserver \
  --allocated-storage 250 \
  --db-instance-class db.m1.large \
  --db-security-groups mydbsecuritygroup \
  --db-subnet-group mydbsubnetgroup \
  --master-user-name masterawsuser \
  --master-user-password masteruserpassword \
  --backup-retention-period 3
```

Windows의 경우:

```
aws rds create-db-instance ^
--engine sqlserver-se ^
--db-instance-identifier mydbinstance ^
--allocated-storage 250 ^
--db-instance-class db.m1.large ^
--db-security-groups mydbsecuritygroup ^
--db-subnet-group mydbsubnetgroup ^
--master-user-name masterawsuser ^
--master-user-password masteruserpassword ^
--backup-retention-period 3
```

이 명령은 다음과 유사한 출력을 생성합니다.

```
DBINSTANCE mydbinstance db.m1.large sqlserver-se 250 sa creating 3 **** n
10.50.2789
SECGROUP default active
PARAMGRP default.sqlserver-se-10.5 in-sync
```

API

Amazon RDS API를 사용해 Microsoft SQL Server DB 인스턴스를 생성하려면 다음 파라미터와 함께 [CreateDBInstance](#) 작업을 호출합니다. 각 설정에 대한 자세한 내용은 [Microsoft SQL Server DB 인스턴스 설정 \(p. 500\)](#) 단원을 참조하십시오.

- AllocatedStorage
- BackupRetentionPeriod
- DBInstanceClass
- DBInstanceIdentifier
- DBSecurityGroups
- DBSubnetGroup
- Engine
- MasterUsername
- MasterUserPassword

Example

```
https://rds.amazonaws.com/
?Action=CreateDBInstance
&AllocatedStorage=250
&BackupRetentionPeriod=3
&DBInstanceClass=db.m1.large
&DBInstanceIdentifier=mydbinstance
&DBSecurityGroups.member.1=mysecuritygroup
&DBSubnetGroup=mydbsubnetgroup
&Engine=sqlserver-se
&MasterUserPassword=masteruserpassword
&MasterUsername=masterawsuser
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-10-31
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140305/us-west-1/rds/aws4_request
&X-Amz-Date=20140305T185838Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=b441901545441d3c7a48f63b5b1522c5b2b37c137500c93c45e209d4b3a064a3
```

Microsoft SQL Server DB 인스턴스 설정

다음 표에는 SQL Server DB 인스턴스를 생성할 때 선택하는 설정에 대한 세부 정보가 나와 있습니다.

설정	설정 설명
활당된 스토리지	DB 인스턴스용으로 할당할 스토리지(단위: GB)입니다. 경우에 따라 DB 인스턴스에 대해 데이터베이스의 크기보다 많은 양의 스토리지를 할당하면 I/O 성능을 개선할 수 있습니다. 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.
마이너 버전 자동 업그레이드	기본 마이너 DB 엔진 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다. Amazon RDS는 유지 관리 기간에 마이너 버전 자동 업그레이드를 수행합니다.
가용 영역	DB 인스턴스의 가용 영역입니다. 가용 영역을 지정하지 않으면 기본값으로 No Preference를 사용합니다. 자세한 내용은 리전 및 가용 영역 (p. 99) 단원을 참조하십시오.
백업 보존 기간	DB 인스턴스의 자동 백업을 보존할 기간(단위: 일)입니다. 중요한 DB 인스턴스의 경우 이 값을 1 이상으로 설정해야 합니다. 자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.
백업 기간	Amazon RDS가 자동으로 DB 인스턴스를 백업하는 기간입니다. 데이터베이스를 백업할 특정 시간을 지정하지 않으면 기본값으로 No Preference를 사용합니다. 자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.
스냅샷으로 태그 복사	스냅샷을 생성할 때 DB 인스턴스 태그를 DB 스냅샷에 복사하려면 이 옵션을 선택합니다. 자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.
데이터베이스 포트	DB 인스턴스에 액세스하는 데 사용할 포트입니다. SQL Server 설치의 기본값은 포트 1433으로 설정됩니다. DB 보안 그룹을 DB 인스턴스에 사용할 경우 DB 보안 그룹을 생성할 때 제공한 포트 값과 같아야 합니다.
DB 엔진 버전	사용할 Microsoft SQL Server의 버전입니다.
DB 인스턴스 클래스	DB 인스턴스에 대한 구성입니다. 예를 들어, [db.m1.small] 인스턴스 클래스는 1.7GiB 메모리, 1 ECU(1 ECU의 1 가상 코어), 64비트 플랫폼 및 보통 I/O 용량과 동일합니다. 가능하면 일반 쿼리 작업 세트가 메모리에 상주할 수 있을 정도로 큰 인스턴스 클래스를 선택합니다. 작업 세트가 메모리에 상주할 경우 시스템의 디스크 쓰기가 불필요하여 성능이 향상됩니다.

설정	설정 설명
	자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 및 Microsoft SQL Server를 위한 DB 인스턴스 클래스 지원 (p. 481) 단원을 참조하십시오.
DB 인스턴스 식별자	DB 인스턴스의 이름입니다. 온프레미스 서버와 동일한 방식으로 DB 인스턴스의 이름을 지정합니다. DB 인스턴스 식별자는 최대 63자의 영숫자 문자를 포함할 수 있으며 선택한 AWS 리전의 계정에 대해 고유해야 합니다. 예를 들어, sqlsv-instance1 처럼 선택한 AWS 리전과 DB 엔진을 포함하는 등 알기 쉬운 이름을 지정할 수 있습니다.
DB 파라미터 그룹	DB 인스턴스의 파라미터 그룹입니다. 기본 파라미터 그룹을 사용하거나 사용자 지정 파라미터 그룹을 생성할 수 있습니다. 자세한 내용은 DB 파라미터 그룹 작업 (p. 161) 단원을 참조하십시오.
삭제 방지 활성화	DB 인스턴스가 삭제되지 않도록 방지하려면, 삭제 방지를 활성화합니다. AWS Management 콘솔을 사용하여 프로덕션 DB 인스턴스를 생성할 경우 기본적으로 삭제 방지가 활성화됩니다. 자세한 내용은 DB 인스턴스 삭제 (p. 128) 단원을 참조하십시오.
[Enable Encryption]	이 DB 인스턴스에 대해 비활성화되어 있는 암호화를 활성화하려면 [Yes]를 선택합니다. 자세한 내용은 Amazon RDS 리소스 암호화 (p. 385) 단원을 참조하십시오.
확장 모니터링 활성화	DB 인스턴스가 실행되는 운영 체제에 대한 실시간 지표를 수집하려면 [Yes]를 선택합니다. 자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.
라이선스 모델	사용할 라이선스 모델입니다. license-included를 선택하여 Microsoft SQL Server에 대한 일반 라이선스 계약을 사용합니다.
유지 관리 기간	대기 중인 DB 인스턴스 설정 변경이 적용되기 위해 경과해야 하는 기간(30분)입니다. 이 시간이 중요하지 않은 경우 [No Preference]를 선택합니다. 자세한 내용은 Amazon RDS 유지 관리 기간 (p. 116) 단원을 참조하십시오.
마스터 사용자 이름	모든 데이터베이스 권한을 사용해 DB 인스턴스에 로그인하기 위해 마스터 사용자 이름으로 사용할 이름입니다. 마스터 사용자 이름은 processadmin, public 및 setupadmin 고정 서버 역할의 구성원인 SQL Server 인증 로그인입니다. 자세한 내용은 Microsoft SQL Server 보안 (p. 481) 단원을 참조하십시오.
Master User Password	마스터 사용자 계정의 암호입니다. 암호는 8~128자 인쇄 가능한 ASCII 문자를 포함해야 합니다(/, ", 공백 및 @ 제외).

설정	설정 설명
다중 AZ 배포	<p>장애 조치를 위해 다른 가용 영역에 DB 인스턴스의 수동 보조 복제본을 생성하려면 예를 선택합니다. 이때 고가용성을 유지 하려면 프로덕션 워크로드를 위한 다중 AZ를 권장합니다. 개발 및 테스팅을 위해 No를 선택할 수 있습니다.</p> <p>자세한 내용은 Microsoft SQL Server의 다중 AZ 배포 (p. 537) 단원을 참조하십시오.</p>
옵션 그룹	<p>DB 인스턴스의 옵션 그룹입니다. 기본 옵션 그룹을 사용하거나 사용자 지정 옵션 그룹을 생성할 수 있습니다.</p> <p>자세한 내용은 옵션 그룹 작업 (p. 148) 단원을 참조하십시오.</p>
Publicly Accessible	<p>DB 인스턴스에 퍼블릭 IP 주소를 부여하려면 [Yes]를 선택합니다. 그러면 VPC 외부에서 DB 인스턴스에 액세스할 수 있습니다(DB 인스턴스도 VPC의 퍼블릭 서브넷에 있어야 함). VPC 내부에서만 DB 인스턴스에 액세스할 수 있게 하려면 [No]를 선택합니다.</p> <p>자세한 내용은 VPC에 있는 DB 인스턴스를 인터넷에서 숨기기 (p. 416) 단원을 참조하십시오.</p>
[Storage Type]	<p>DB 인스턴스의 스토리지 유형입니다.</p> <p>자세한 내용은 Amazon RDS 스토리지 유형 (p. 101) 단원을 참조하십시오.</p>
[Subnet Group]	<p>이 설정은 현재 사용 중인 플랫폼에 따라 다릅니다. AWS를 처음 사용하는 고객은 [default]를 선택합니다. 이 DB 서브넷 그룹은 계정에 대해 생성된 기본 DB 서브넷 그룹입니다. 이전 E2-Classic 플랫폼에서 DB 인스턴스를 만들어서 특정 VPC에 배포 하려면 해당 VPC로 생성된 DB 서브넷 그룹을 선택합니다.</p>
시간대	<p>DB 인스턴스의 시간대입니다. 표준 시간대를 선택하지 않으면 DB 인스턴스는 기본 표준 시간대를 사용합니다.</p> <p>자세한 내용은 Microsoft SQL Server DB 인스턴스의 현지 시간대 (p. 489) 단원을 참조하십시오.</p>
VPC	<p>이 설정은 현재 사용 중인 플랫폼에 따라 다릅니다. AWS를 처음 사용하는 경우에는 기본 VPC를 선택합니다. 이전 E2-Classic 플랫폼에서 VPC를 사용하지 않는 DB 인스턴스를 만드는 경우에는 [Not in VPC]를 선택합니다.</p> <p>자세한 내용은 Amazon Virtual Private Cloud(VPC) 및 Amazon RDS (p. 406) 단원을 참조하십시오.</p>
[VPC Security Group]	<p>AWS를 처음 사용하는 경우에는 기본 VPC를 선택합니다. 그렇지 않으면 이전에 생성한 VPC 보안 그룹을 선택합니다.</p> <p>RDS 콘솔에서 새 VPC 보안 그룹 생성을 선택하는 경우 브라우저에서 검색된 IP 주소에서 DB 인스턴스에 액세스하도록 허용하는 발신 규칙을 사용하여 새 보안 그룹이 생성됩니다.</p> <p>자세한 내용은 DB 보안 그룹 작업(EC2-Classic 플랫폼) (p. 394) 단원을 참조하십시오.</p>

관련 주제

- 자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 (p. 421)
- Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 (p. 503)
- Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 (p. 509)
- DB 인스턴스 삭제 (p. 128)

Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스에 연결하기

Amazon RDS가 DB 인스턴스를 프로비저닝한 후에는 표준 SQL 클라이언트 애플리케이션을 사용해 DB 인스턴스에 연결할 수 있습니다. 이 주제에서는 Microsoft SQL Server Management Studio(SSMS) 또는 SQL Workbench/J를 사용하여 DB 인스턴스에 연결합니다.

사용자가 샘플 DB 인스턴스를 만들어 연결하는 절차를 실습하는 예제는 [Microsoft SQL Server DB 인스턴스 생성 및 DB 인스턴스에 연결 \(p. 17\)](#) 단원을 참조하십시오.

Microsoft SQL Server Management Studio로 DB 인스턴스에 연결

이 절차에서는 Microsoft SQL Server Management Studio(SSMS)를 사용하여 샘플 DB 인스턴스에 연결합니다. 이 유ти리티의 둑립 실행형 버전을 다운로드하려면 Microsoft 설명서의 [SQL Server Management Studio\(SSMS\)](#)를 참조하십시오.

SSMS를 사용하여 DB 인스턴스에 연결하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. Amazon RDS 콘솔의 오른쪽 상단 모서리에서 DB 인스턴스의 AWS 리전을 선택합니다.
3. DB 인스턴스에 대한 DNS 이름과 포트 번호를 찾습니다.
 - a. RDS 콘솔을 연 다음 데이터베이스를 선택하여 DB 인스턴스의 목록을 표시합니다.
 - b. 세부 정보를 표시할 SQL Server DB 인스턴스의 이름을 선택합니다.

mysqldb

Summary

DB Name	CPU	Info
mysqldb		Backing-up
Role	Current activity	Engine
Instance	0 Connections	SQL Server Express Edition

Connectivity Monitoring Logs & events Configuration Maintenance & backups Tags

Connectivity

Endpoint & port	Networking
Endpoint mysqldb	Availability zone us-west-2b
Port 1433	VPC vpc-2aed394c
	Subnet group

- c. 연결 탭에서 엔드포인트를 복사합니다. 또한, DB 인스턴스에 사용되는 포트를 적어둡니다.
4. SQL Server Management Studio를 시작합니다.

Connect to Server 대화 상자가 나타납니다.



5. DB 인스턴스에 대한 정보를 제공합니다.

- a. [Server type]에서 [Database Engine]을 선택합니다.
- b. 서버 이름에 쉼표로 구분한 DB 인스턴스의 DNS 이름과 포트 번호를 입력하거나 붙여 넣습니다.

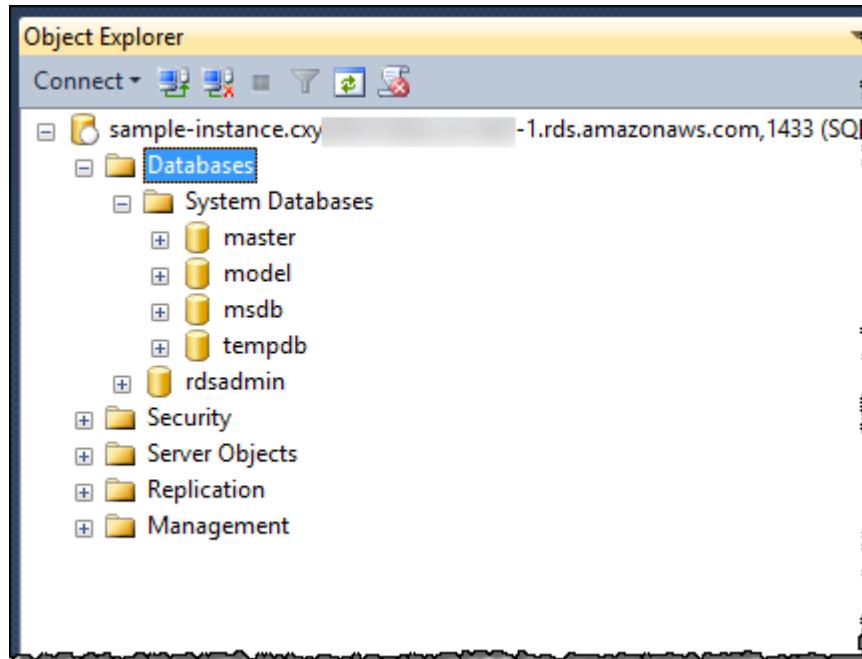
Important

DNS 이름과 포트 번호 사이의 콜론을 쉼표로 바꿉니다.

예를 들면 서버 이름은 다음과 같은 형식이어야 합니다.

```
sample-instance.cg034hpkmmjt.us-east-1.rds.amazonaws.com,1433
```

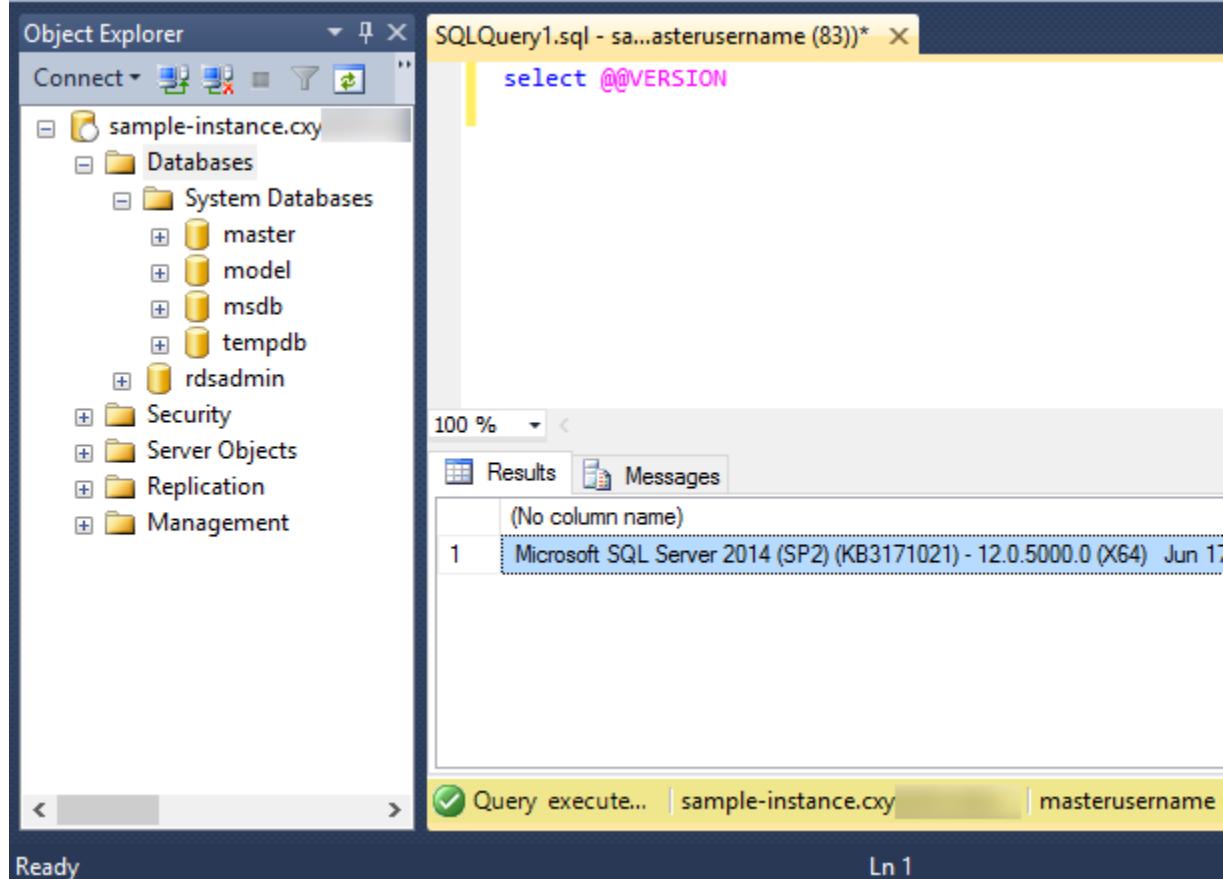
- c. [Authentication]의 경우 [SQL Server Authentication]을 선택합니다.
 - d. 로그인에는 DB 인스턴스의 마스터 사용자 이름을 입력합니다.
 - e. 암호에는 DB 인스턴스의 암호를 입력합니다.
 6. [Connect]를 선택합니다.
- 몇 분 정도 지나면 SSMS가 DB 인스턴스에 연결됩니다. DB 인스턴스에 연결할 수 없는 경우 [보안 그룹 고려 사항 \(p. 508\)](#) 및 [SQL Server DB 인스턴스 연결 문제 해결 \(p. 509\)](#) 단원을 참조하십시오.
7. SQL 서버 DB 인스턴스는 SQL 서버의 표준 기본 제공 시스템 데이터베이스(마스터, 모델, msdb 및 tempdb)와 함께 제공됩니다. 시스템 데이터베이스를 탐색하려면 다음을 수행하십시오.
- a. SSMS의 [View] 메뉴에서 [Object Explorer]를 선택합니다.
 - b. DB 인스턴스와 [Databases]를 확장하고, 다음과 같이 [System Databases]를 확장합니다.



8. SQL Server DB 인스턴스는 rdsadmin이라는 데이터베이스와 함께 제공됩니다. Amazon RDS는 이 데 이터베이스를 사용하여 데이터베이스를 관리하는 데 사용하는 객체를 저장합니다. rdsadmin 데이터베이스에도 고급 작업 수행을 위해 실행할 수 있는 저장 절차가 포함됩니다. 자세한 내용은 [Microsoft SQL Server에 대한 일반 DBA 작업 \(p. 554\)](#) 단원을 참조하십시오.
 9. 이제 자체 데이터베이스 생성을 시작하고 평소대로 DB 인스턴스와 데이터베이스에 대한 쿼리 실행을 시작할 수 있습니다. DB 인스턴스에 대한 테스트 쿼리를 실행하려면 다음 중 하나를 수행합니다.
- a. SSMS의 [File] 메뉴에서 [New]를 가리킨 후 [Query with Current Connection]을 선택합니다.
 - b. 다음 SQL 쿼리를 입력합니다.

```
select @@VERSION
```

- c. 쿼리를 실행합니다. SSMS가 Amazon RDS DB 인스턴스의 SQL Server 버전을 반환합니다.



SQL Workbench/J로 DB 인스턴스에 연결

이번 예에서는 SQL Workbench/J 데이터베이스 도구를 사용하여 Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스에 연결하는 방법을 나타냅니다. SQL Workbench/J를 다운로드하려면 [SQL Workbench/J](#)를 참조하십시오.

SQL Workbench/J가 JDBC를 이용해 DB 인스턴스에 연결합니다. 그 밖에 SQL Server용 JDBC 드라이버도 필요합니다. 이 드라이버를 다운로드하려면 [SQL Server용 Microsoft JDBC 드라이버 4.1\(프리뷰\)](#) 및 [4.0](#)을 참조하십시오.

SQL Workbench를 사용하여 DB 인스턴스에 연결하는 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. Amazon RDS 콘솔의 오른쪽 상단 모서리에서 DB 인스턴스의 AWS 리전을 선택합니다.
3. DB 인스턴스에 대한 DNS 이름과 포트 번호를 찾습니다.
 - a. RDS 콘솔을 연 다음 데이터베이스를 선택해 DB 인스턴스의 목록을 표시합니다.
 - b. 세부 정보를 표시할 SQL Server DB 인스턴스의 이름을 선택합니다.

mysqldb

Summary

DB Name	CPU	Info
mysqldb		Backing-up
Role	Current activity	Engine
Instance	0 Connections	SQL Server Express Edition

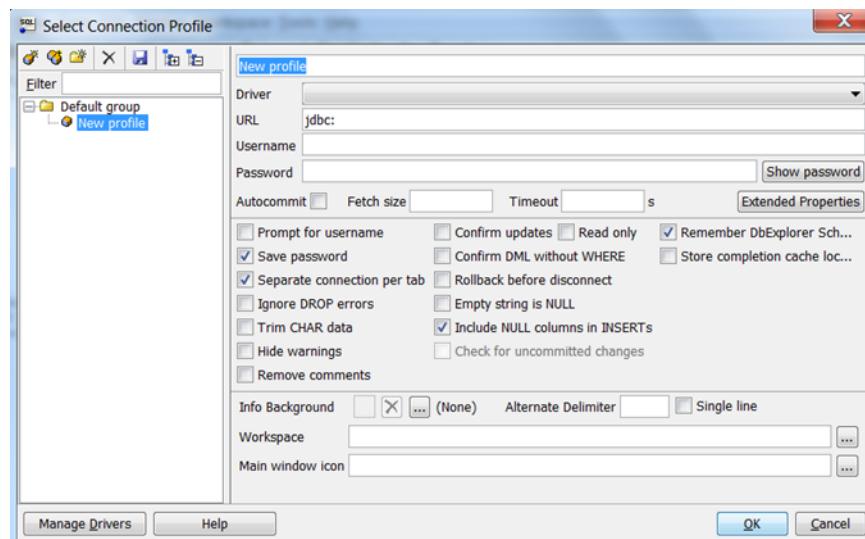
Connectivity Monitoring Logs & events Configuration Maintenance & backups Tags

Connectivity

Endpoint & port		Networking
Endpoint	rds.amazonaws.com	Availability zone
mysqldb		us-west-2b
Port		VPC
1433		vpc-2aed394c

c. 연결 탭에서 엔드포인트를 복사합니다. 또한, DB 인스턴스에 사용되는 포트를 적어둡니다.

4. SQL Workbench/J를 엽니다. 아래 그림과 같이 연결 프로파일 선택 대화 상자가 나타납니다.

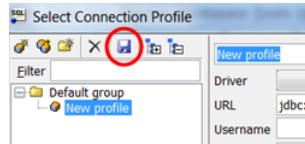


5. 대화 상자 상단의 첫 번째 상자에 프로파일 이름을 입력합니다.
6. 드라이버에서 **SQL JDBC 4.0**을 선택합니다.

7. URL에 `jdbc:sqlserver://`를 입력한 후, DB 인스턴스의 엔드포인트를 입력합니다. 예를 들면 URL 값은 다음과 같습니다.

```
jdbc:sqlserver://sqlsvr-pdz.abcd12340.us-west-2.rds.amazonaws.com:1433
```

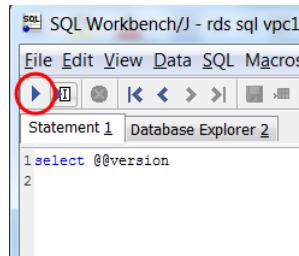
8. 사용자 이름에 DB 인스턴스의 마스터 사용자 이름을 입력하거나 붙여 넣습니다.
9. 암호에 마스터 사용자 암호를 입력합니다.
10. 아래 그림과 같이 대화 상자 도구 모음에서 저장 아이콘을 선택합니다.



11. 확인을 선택합니다. 몇 분 정도 지나면 SQL Workbench/J가 DB 인스턴스에 연결됩니다. DB 인스턴스에 연결할 수 없는 경우 [보안 그룹 고려 사항 \(p. 508\)](#) 및 [SQL Server DB 인스턴스 연결 문제 해결 \(p. 509\)](#) 단원을 참조하십시오.
12. 쿼리 창에 다음과 같이 SQL 쿼리를 입력합니다.

```
select @@VERSION
```

13. 아래 그림과 같이 도구 모음에서 실행 아이콘을 선택합니다.



쿼리가 다음과 같이 DB 인스턴스의 버전 정보를 반환합니다.

```
Microsoft SQL Server 2012 - 11.0.2100.60 (X64)
```

보안 그룹 고려 사항

DB 인스턴스에 연결하려면 DB 인스턴스가 DB 인스턴스에 액세스할 때 사용하는 IP 주소와 네트워크 구성이 할당되어 있는 보안 그룹과 연동되어야 합니다. DB 인스턴스를 만들 때 DB 인스턴스를 적당한 보안 그룹과 연동할 수 있습니다. 하지만 DB 인스턴스를 생성하면서 따로 설정할 필요가 없는 기본 보안 그룹을 할당한 경우에는 DB 인스턴스 방화벽이 연결을 차단합니다.

새로운 보안 그룹을 생성하여 액세스를 활성화해야 하는 경우에는 DB 인스턴스의 기반을 이루는 Amazon EC2 플랫폼에 따라 생성할 보안 그룹 유형이 결정됩니다. 플랫폼을 확인하려면 [EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 \(p. 407\)](#) 단원을 참조하십시오. 일반적으로 DB 인스턴스가 EC2-Classic 플랫폼에 있으면 DB 보안 그룹을 만들고, DB 인스턴스가 VPC 플랫폼에 있으면 VPC 보안 그룹을 만듭니다. 새 보안 그룹 생성에 대한 자세한 내용은 [보안 그룹을 통한 액세스 제어 \(p. 390\)](#) 단원을 참조하십시오.

새 보안 그룹을 생성하였으면 보안 그룹과 연동되도록 DB 인스턴스 설정을 변경합니다. 자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#) 단원을 참조하십시오.

SSL을 사용하여 DB 인스턴스 연결을 암호화함으로써 보안을 강화할 수 있습니다. 자세한 내용은 [Microsoft SQL Server DB 인스턴스와 함께 SSL 사용 \(p. 541\)](#) 단원을 참조하십시오.

SQL Server DB 인스턴스 연결 문제 해결

다음은 SQL Server DB 인스턴스 연결을 시도할 때 발생할 수 있는 문제입니다.

문제	문제 해결 제안
DB 인스턴스에 연결할 수 없습니다.	새로 만든 DB 인스턴스의 경우, DB 인스턴스를 사용할 준비가 될 때까지 DB 인스턴스의 상태는 [creating]입니다. 상태가 available로 변경되면 DB 인스턴스에 연결할 수 있습니다. DB 인스턴스 클래스와 스토리지의 양에 따라 새 인스턴스를 사용할 수 있을 때까지 최대 20분이 걸릴 수 있습니다.
DB 인스턴스에 연결할 수 없습니다.	DB 인스턴스를 만들 때 지정한 포트를 통해 통신을 보내거나 받을 수 없으면 DB 인스턴스에 연결할 수 없습니다. 네트워크 관리자에게 문의해 DB 인스턴스에 대해 지정한 포트가 인바운드 및 아웃바운드 통신을 허용하는지 확인하십시오.
DB 인스턴스에 연결할 수 없습니다.	로컬 방화벽에서 적용되는 액세스 규칙과 DB 인스턴스의 보안 그룹에 있는 DB 인스턴스에 액세스하기 위한 권한을 부여한 IP 주소가 일치하지 않을 수 있습니다. 방화벽의 아웃바운드 또는 인바운드 규칙에 문제가 있을 가능성이 높습니다. 보안 그룹에 대한 자세한 내용은 보안 그룹을 통한 액세스 제어 (p. 390) 단원을 참조하십시오. 보안 그룹의 규칙 설정 절차 안내에 관한 주제는 자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 (p. 421) 단원을 참조하십시오.
Could not open a connection to SQL Server – Microsoft SQL Server, Error: 53(SQL Server에 대한 연결을 열 수 없습니다 - Microsoft SQL Server, 오류: 53)	서버 이름을 정확하게 지정했는지 확인하십시오. 서버 이름에서 쉼표로 구분한 샘플 DB 인스턴스의 DNS 이름과 포트 번호를 입력합니다. Important DNS 이름과 포트 번호 사이의 콜론을 쉼표로 바꿉니다. 예를 들면 서버 이름은 다음과 같은 형식이어야 합니다. <code>sample-instance.cg034hpkmmt.us-east-1.rds.amazonaws.com,1433</code>
No connection could be made because the target machine actively refused it – Microsoft SQL Server, Error: 10061(대상 컴퓨터에서 연결을 거부했으므로 연결하지 못했습니다 - Microsoft SQL Server, 오류: 10061)	DB 인스턴스에 연결할 수 있지만 연결이 거부되었습니다. 이 문제는 주로 사용자 이름이나 암호를 잘못 지정하면 발생합니다. 사용자 이름과 암호를 확인하고 다시 시도하십시오.

Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정

DB 인스턴스의 설정을 변경하여 인스턴스 클래스 또는 인스턴스 이름을 변경하는 등 여러 작업을 실행할 수 있습니다. 이번 주제에서는 Microsoft SQL Server 기반 Amazon RDS DB 인스턴스의 수정 방법과 SQL Server DB 인스턴스의 설정에 대해서 설명하겠습니다.

프로덕션 인스턴스를 수정하기 전에 테스트 인스턴스에서 변경 사항을 테스트하면 각 변경 사항이 미칠 영향을 완전히 이해하는 데 도움이 됩니다. 이는 특히 데이터베이스 버전을 업그레이드할 때 중요합니다.

DB 인스턴스 설정을 수정한 후 변경 사항을 즉시 적용할 수도 있고, DB 인스턴스에 대한 다음 유지 관리 기간에 적용할 수도 있습니다. 일부 변경 사항의 경우 DB 인스턴스가 재시작될 수 있습니다.

AWS Management 콘솔

SQL Server DB 인스턴스의 설정을 변경하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 변경하려는 DB 인스턴스를 선택합니다.
3. [Modify]를 선택합니다. [Modify DB Instance] 페이지가 나타납니다.
4. 원하는 설정을 모두 변경합니다. 각 설정에 대한 자세한 내용은 [Microsoft SQL Server DB 인스턴스 설정 \(p. 511\)](#) 단원을 참조하십시오.
5. 원하는 대로 모두 변경되었으면 [Continue]를 선택합니다.
6. 변경 사항을 즉시 적용하려면 즉시 적용을 선택합니다. 일부의 경우 이 옵션을 선택하면 중단이 발생할 수 있습니다. 자세한 내용은 [즉시 적용 파라미터 사용 \(p. 112\)](#) 단원을 참조하십시오.
7. 확인 페이지에서 변경 내용을 검토합니다. 변경 내용이 정확할 경우 [Modify DB Instance]를 선택하여 변경 내용을 저장합니다.

그렇지 않으면 [Back]을 선택하여 변경 내용을 편집하거나 [Cancel]을 선택하여 변경 내용을 취소합니다.

CLI

AWS CLI를 사용해 Microsoft SQL Server DB 인스턴스를 수정하려면 `modify-db-instance` 명령을 호출합니다. DB 인스턴스 식별자와 수정하려는 설정의 파라미터를 지정합니다. 각 파라미터에 대한 자세한 내용은 [Microsoft SQL Server DB 인스턴스 설정 \(p. 511\)](#) 단원을 참조하십시오.

Example

다음 코드는 백업 보존 기간을 1주(7일)로 설정하여 `mydbinstance`를 수정합니다. 이 코드는 `--auto-minor-version-upgrade`를 사용하여 마이너 버전 자동 업그레이드를 활성화합니다. 마이너 버전 자동 업그레이드를 비활성화하려면 `--no-auto-minor-version-upgrade`를 사용합니다. 변경 사항은 `--no-apply-immediately`를 사용하여 다음 유지 관리 기간에 적용됩니다. 변경 사항을 바로 적용하려면 `--apply-immediately`를 사용합니다. 자세한 내용은 [즉시 적용 파라미터 사용 \(p. 112\)](#) 단원을 참조하십시오.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--backup-retention-period 7 \
--auto-minor-version-upgrade \
--no-apply-immediately
```

Windows의 경우:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--backup-retention-period 7 ^
--auto-minor-version-upgrade ^
--no-apply-immediately
```

API

Amazon RDS API를 사용하여 Microsoft SQL Server DB 인스턴스를 수정하려면 [ModifyDBInstance](#) 작업을 호출합니다. DB 인스턴스 식별자와 수정하려는 설정의 파라미터를 지정합니다. 각 파라미터에 대한 자세한 내용은 [Microsoft SQL Server DB 인스턴스 설정 \(p. 511\)](#) 단원을 참조하십시오.

Example

다음은 백업 보존 기간을 1주일(7일)로 설정하고 마이너 버전 자동 업그레이드를 활성화하여 mydbinstance를 수정하는 코드입니다. 이 변경 사항은 다음 유지 관리 기간에 적용됩니다.

```
https://rds.amazonaws.com/  
?Action=ModifyDBInstance  
&ApplyImmediately=false  
&AutoMinorVersionUpgrade=true  
&BackupRetentionPeriod=7  
&DBInstanceIdentifier=mydbinstance  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4  
&Version=2014-10-31  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Credential=AKIADQKE4SARGYLE/20131016/us-west-1/rds/aws4_request  
&X-Amz-Date=20131016T233051Z  
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date  
&X-Amz-Signature=087a8eb41cb1ab0fc9ec1575f23e73757ffc6ale42d7d2b30b9cc0be988cff97
```

Microsoft SQL Server DB 인스턴스 설정

다음 표에는 수정이 가능한 설정, 수정이 불가능한 설정, 변경 사항을 적용할 수 있는 시점, 변경 사항으로 인한 DB 인스턴스 가동 중지 여부에 대한 자세한 내용이 나와 있습니다.

설정	설정 설명	변경이 발생할 때	가동 중지 참고 사항
활당된 스토리지	<p>DB 인스턴스에 할당할 스토리지(단위: GiB). 할당된 스토리지를 늘릴 수만 있고 줄일 수는 없습니다. 허용되는 최대 스토리지는 16TiB입니다.</p> <p>Warning</p> <p>Amazon RDS가 DB 인스턴스를 수정하여 스토리지 크기나 형식을 늘리기 시작하면 이후 6시간 동안은 스토리지 크기 또는 형식을 늘리는 또 다른 요청을 제출할 수 없습니다.</p> <p>일부 이전 DB 인스턴스의 스토리지 및 이전 DB 스냅샷에서 복원된 DB 인스턴스는 수정할 수 없습니다. DB 인스턴스를 사용할 수 없는 경우 콘솔에서 할당된 스토리지 옵션이 비활성화됩니다. DB 인스턴스의 유 효한 스토리지 옵션을 반환하는 AWS CLI 명령 describe-valid-db-instance-modifications를 사용하여 자격을 확인할 수도 있습니다.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	몇 분 동안 짧게 중단될 수 있습니다. 그 후 DB 인스턴스는 온라인 상태이나 스토리지 최적화 상태에 있습니다. 스토리지 최적화 중에 성능이 저하될 수 있습니다. 스토리지 최적화 프로세스는 보통 짧지만, 가끔 최대 24 시간까지 걸릴 수도 있습니다.

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
	자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.		
마이너 버전 자동 업그레이드	기본 마이너 DB 엔진 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다. Amazon RDS는 유지 관리 기간에 마이너 버전 자동 업그레이드를 수행합니다.	-	-
백업 보존 기간	자동 백업을 보존할 일수. 자동 백업을 비활성화하기 위해 백업 보존 기간을 0으로 설정합니다. 자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.	[Apply Immediately]가 true로 설정되는 즉시 변경됩니다. [Apply Immediately]가 false로 설정되고 이 설정을 0이 아닌 값에서 다른 0이 아닌 값으로 변경하면 비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이때 적용되지 않을 경우, 다음 유지 관리 기간에 변경 사항이 적용됩니다.	백업 보존 기간을 0에서 0이 아닌 값으로 또는 0이 아닌 값에서 0으로 변경할 경우 인스턴스가 중단됩니다.
백업 기간	데이터베이스의 자동 백업이 실행되는 기간. 백업 기간은 국제 표준시(UTC)의 시작 시간과 시간 단위의 지속 기간으로 구성됩니다. 자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다.	-
인증 기관	사용할 인증서.	-	-
스냅샷으로 태그 복사	DB 인스턴스 태그가 있는 경우 이 옵션은 DB 스냅샷을 만들 때 태그를 복사합니다. 자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-
데이터베이스 포트	데이터베이스에 액세스하는 데 사용할 포트. 포트 값은 DB 인스턴스의 옵션 그룹에서 옵션에 대해 지정한 포트 값과 일치하지 않아야 합니다.	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply Immediately] 설정을 무시합니다.	DB 인스턴스가 즉시 재부팅됩니다.

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
DB 엔진 버전	<p>사용할 SQL Server 데이터베이스 엔진의 버전. 프로덕션 DB 인스턴스를 업그레이드 하려면 먼저 테스트 인스턴스에서 업그레이드 프로세스를 테스트하여 업그레이드 시간을 확인하고 애플리케이션을 검증하는 것이 좋습니다.</p> <p>자세한 내용은 Microsoft SQL Server DB 엔진 업그레이드 (p. 516) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다.
DB 인스턴스 클래스	<p>사용할 DB 인스턴스 클래스.</p> <p>자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 및 Microsoft SQL Server를 위한 DB 인스턴스 클래스 지원 (p. 481) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다.
DB 인스턴스 식별자	<p>DB 인스턴스 식별자</p> <p>DB 인스턴스 이름 바꾸기의 영향에 대한 자세한 내용은 DB 인스턴스 이름 바꾸기 (p. 121) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다. DB 인스턴스가 재부팅됩니다.
DB 파라미터 그룹	<p>DB 인스턴스와 연결할 파라미터 그룹.</p> <p>자세한 내용은 DB 파라미터 그룹 작업 (p. 161) 단원을 참조하십시오.</p>	파라미터 그룹 변경 사항은 즉시 적용됩니다.	<p>이 변경 도중 인스턴스가 중단되지 않습니다. 파라미터 그룹을 변경하는 경우 일부 파라미터에 대한 변경 내용은 재부팅 없이 DB 인스턴스에 즉시 적용됩니다. 다른 파라미터에 대한 변경 내용은 DB 인스턴스를 재부팅한 후에만 적용됩니다.</p> <p>자세한 내용은 DB 클러스터에서 DB 인스턴스 (p. 123) 단원을 참조하십시오.</p>
도메인	<p>인스턴스를 이동할 Active Directory 도메인입니다. 현재 도메인에서 인스턴스를 제거하려면 [none]을 지정합니다. 도메인은 이 작업 이전에 존재해야 합니다.</p> <p>자세한 내용은 Microsoft SQL Server DB 인스턴스를 통한 Windows 인증 사용 (p. 568) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	<p>이 변경 도중 잠시 작동이 중단됩니다. 단일 AZ DB 인스턴스의 경우 약 5~10분 정도 중단됩니다. 다중 AZ DB 인스턴스의 경우 약 1분 정도 중단됩니다.</p>

설정	설정 설명	변경이 발생할 때	자동 종지 참고 사항
도메인 IAM 역할 이름	<p>Active Directory 서비스를 액세스할 때 사용하는 IAM 역할의 이름입니다.</p> <p>자세한 내용은 Microsoft SQL Server DB 인스턴스를 통한 Windows 인증 사용 (p. 568) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 잠시 작동이 중단됩니다.
삭제 방지 활성화	DB 인스턴스가 삭제되지 않도록 방지하려면, 삭제 방지를 활성화합니다. 자세한 내용은 DB 인스턴스 삭제 (p. 128) 단원을 참조하십시오.	-	-
확장 모니터링 활성화	<p>DB 인스턴스가 실행되는 운영 체제에 대한 실시간 측정치 수집을 활성화하려면 [Yes].</p> <p>자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.</p>	-	-
라이선스 모델	[license-included]를 선택하여 Microsoft SQL Server에 대한 일반 라이선스 계약을 사용합니다.	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다.
유지 관리 기간	<p>시스템 유지 관리를 실행하는 기간. 시스템 유지 관리는 업그레이드를 포함합니다(해당 할 경우). 유지 관리 기간은 국제 표준시(UTC)의 시작 시간과 시간 단위의 지속 기간으로 구성됩니다.</p> <p>이 기간을 현재 시간으로 설정하려면 대기 중인 변경 사항이 모두 적용될 수 있도록 현재 시간과 기간 종료 시간 사이에 최소 30분 이상 필요합니다.</p> <p>자세한 내용은 Amazon RDS 유지 관리 기간 (p. 116) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply Immediately] 설정을 무시합니다.	인스턴스가 중단될 수 있는 작업이 하나 이상 대기 중이고, 유지 관리 기간이 현재 시간을 포함하여 변경된 경우 대기 중인 작업들이 즉시 적용되고 인스턴스가 중단됩니다.

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
다중 AZ 배포	<p>예를 선택하면 장애 조치를 위해 다른 가용 영역에 DB 인스턴스의 수동 보조 복제본이 생성됩니다. 이때 고가용성을 유지하려면 프로덕션 워크로드를 위한 다중 AZ를 권장합니다. 개발 및 테스트용이면 [No].</p> <p>DB 인스턴스가 SQL Server 2014, 2016 또는 2017 Enterprise Edition에서 데이터베이스 미러링(DBM)-상시 작동 가능성 그룹(AG) 아님-을 실행 중이고 인 메모리 최적화가 활성화되어 있는 경우, 다중 AZ를 추가하기 전에 인 메모리 최적화를 비활성화합니다. AG를 실행 중인 경우 이 단계가 필요하지 않습니다.</p> <p>자세한 내용은 Microsoft SQL Server의 다중 AZ 배포 (p. 537) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	-
새 마스터 암호	<p>마스터 사용자의 암호. 암호는 8~128자 인쇄 가능한 ASCII 문자를 포함해야 합니다(/, ", 공백 및 @ 제외). 마스터 암호를 재설정하여 DB 인스턴스에 대한 권한을 재설정할 수도 있습니다.</p> <p>자세한 내용은 DB 인스턴스 소유자 역할 암호 재설정 (p. 1070) 단원을 참조하십시오.</p>	<p>비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이 설정은 [Apply Immediately] 설정을 무시합니다.</p>	-
옵션 그룹	<p>DB 인스턴스와 연결할 옵션 그룹.</p> <p>자세한 내용은 옵션 그룹 작업 (p. 148) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	-
공개적으로 액세스할 수 있음	<p>DB 인스턴스에 퍼블릭 IP 주소를 부여 하려면(즉 VPC 외부에서 액세스할 수 있음) [Yes]를 선택합니다. 공개적으로 액세스가 가능하려면 DB 인스턴스도 VPC의 퍼블릭 서브넷에 있어야 합니다. VPC 내부에서만 DB 인스턴스에 액세스할 수 있게 하려면 [No].</p> <p>자세한 내용은 VPC에 있는 DB 인스턴스를 인터넷에서 습기기 (p. 416) 단원을 참조하십시오.</p>	<p>변경 사항이 즉시 적용됩니다. 이 설정은 [Apply Immediately] 설정을 무시합니다.</p>	-

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
보안 그룹	DB 인스턴스와 연결할 보안 그룹. 자세한 내용은 DB 보안 그룹 작업 (EC2-Classic 플랫폼) (p. 394) 단원을 참조하십시오.	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이 설정은 [Apply Immediately] 설정을 무시합니다.	-
스토리지 유형	사용할 스토리지 유형. 마그네틱 스토리지에서 변경하거나 마그네틱 스토리지로 변경할 수 없습니다. 자세한 내용은 Amazon RDS 스토리지 유형 (p. 101) 단원을 참조하십시오. Warning Amazon RDS가 DB 인스턴스를 수정하여 스토리지 크기나 형식을 변경하면 이후 6시간 동안은 스토리지 크기 또는 형식을 변경하는 또 다른 요청을 제출할 수 없습니다.	[Apply Immediately]가 true로 설정되는 즉시 변경됩니다. [Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.	다음과 같이 변경하면 프로세스가 시작되는 동안 잠시 작동이 중단됩니다. 그 이후에 변경 사항이 적용되는 동안 데이터베이스를 정상적으로 사용할 수 있습니다. <ul style="list-style-type: none">범용(SSD)에서 프로비저닝된 IOPS(SSD)로프로비저닝된 IOPS(SSD)에서 범용 (SSD)로
[Subnet Group]	DB 인스턴스에 대한 서브넷 그룹. 이 설정을 사용하여 DB 인스턴스를 다른 VPC로 이동할 수 있습니다. DB 인스턴스가 VPC 내에 있지 않을 경우 이 설정을 사용하여 DB 인스턴스를 VPC 내로 이동할 수 있습니다. 자세한 내용은 VPC에 있지 않은 DB 인스턴스를 VPC로 이동 (p. 420) 단원을 참조하십시오.	[Apply Immediately]가 true로 설정되는 즉시 변경됩니다. [Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.	이 변경 도중 인스턴스가 중단됩니다. DB 인스턴스가 재부팅됩니다.

관련 주제

- [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#)
- [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 503\)](#)
- [Microsoft SQL Server DB 엔진 업그레이드 \(p. 516\)](#)
- [DB 인스턴스 삭제 \(p. 128\)](#)

Microsoft SQL Server DB 엔진 업그레이드

Amazon RDS에서 새 버전의 Microsoft SQL Server를 지원하는 경우 DB 인스턴스를 새 버전으로 업그레이드할 수 있습니다. Amazon RDS는 Microsoft SQL Server DB 인스턴스에 대해 다음 업그레이드를 지원합니다.

- [메이저 버전 업그레이드](#)
- [マイ너 버전 업그레이드](#)

일반적으로 메이저 업그레이드는 기존 애플리케이션과 호환되지 않는 변경 사항을 가져올 수 있습니다. 반대로 마이너 버전 업그레이드에는 기존 애플리케이션과 호환되는 변경 사항만 포함됩니다.

DB 인스턴스를 수동으로 수정하여 메이저 버전 업그레이드를 수행해야 합니다. DB 인스턴스에서 마이너 버전 자동 업그레이드를 사용하도록 설정하면 마이너 버전 업그레이드가 자동으로 수행됩니다. 그 밖의 경우에는 DB 인스턴스를 수동으로 수정하여 마이너 버전 업그레이드를 수행해야 합니다.

Amazon RDS에서 사용할 수 있는 SQL Server 버전에 대한 자세한 내용은 [Amazon RDS의 Microsoft SQL Server \(p. 477\)](#) 단원을 참조하십시오.

주제

- [업그레이드 개요 \(p. 517\)](#)
- [메이저 버전 업그레이드 \(p. 517\)](#)
- [다중 AZ와 인 메모리 최적화 활성화의 고려 사항 \(p. 518\)](#)
- [옵션 및 파라미터 그룹 고려 사항 \(p. 518\)](#)
- [업그레이드 테스트 \(p. 519\)](#)
- [SQL Server DB 인스턴스 업그레이드 \(p. 519\)](#)
- [지원 종료 전에 사용되지 않는 DB 인스턴스 업그레이드 \(p. 520\)](#)

업그레이드 개요

Amazon RDS는 업그레이드 프로세스 중에 DB 스냅샷을 2개 캡처합니다. 첫 번째 DB 스냅샷은 업그레이드 변경 이전 DB 인스턴스의 스냅샷입니다. 업그레이드가 데이터베이스에 맞지 않는 경우에는 이 스냅샷을 복구하여 이전 버전의 DB 인스턴스를 생성할 수 있습니다. 두 번째 DB 스냅샷은 업그레이드 완료 이후에 캡처됩니다.

Note

DB 인스턴스에 대한 백업 보존 기간을 0보다 큰 수로 설정하면 Amazon RDS는 DB 스냅샷만 캡처합니다. 백업 보존 기간을 변경하려면 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#) 단원을 참조하십시오.

업그레이드가 완료되면 이전 버전의 데이터베이스 엔진으로 되돌릴 수 없습니다. 이때 이전 버전으로 되돌리려면 업그레이드 전에 캡처한 DB 스냅샷을 복구하여 새로운 DB 인스턴스를 생성해야 합니다.

SQL Server의 마이너 버전 또는 메이저 버전 업그레이드 중에는 [Free Storage Space] 및 [Disk Queue Depth] 측정치가 -1로 표시됩니다. 이후 업그레이드가 끝나면 두 측정치 모두 정상적으로 돌아옵니다.

메이저 버전 업그레이드

Amazon RDS는 현재 Microsoft SQL Server DB 인스턴스에 대해 다음 메이저 버전의 업그레이드를 지원합니다.

SQL Server 2008을 제외한 어떤 버전에서든 SQL Server 2017로 기존 DB 인스턴스를 업그레이드할 수 있습니다. SQL Server 2008에서 업그레이드하려면 다음 버전 중 하나로 업그레이드하십시오.

현재 버전	지원하는 업그레이드 버전
SQL Server 2016	SQL Server 2017
SQL Server 2014	SQL Server 2017 SQL Server 2016

현재 버전	지원하는 업그레이드 버전
SQL Server 2012	SQL Server 2017
	SQL Server 2016
	SQL Server 2014
SQL Server 2008 R2(사용되지 않음)	SQL Server 2016
	SQL Server 2014
	SQL Server 2012

데이터베이스 호환성 수준

Microsoft SQL Server 데이터베이스 호환성 수준을 이용해 일부 데이터베이스 동작이 이전 버전의 SQL Server를 모방하도록 조정할 수 있습니다. 자세한 내용은 Microsoft 설명서의 [Compatibility Level](#)을 참조하십시오.

DB 인스턴스를 업그레이드할 때 기존의 모든 데이터베이스는 원래 호환성 수준으로 유지됩니다. 예를 들어, SQL Server 2012에서 SQL Server 2014로 업그레이드할 경우 모든 기존 데이터베이스의 호환성 수준은 110입니다. 업그레이드 이후에 생성된 새 데이터베이스의 호환성 수준은 120입니다.

ALTER DATABASE 명령을 사용하여 데이터베이스의 호환성 수준을 변경할 수 있습니다. 예를 들어, customeracct라는 이름의 데이터베이스를 SQL Server 2014와 호환되도록 변경하려면 다음 명령을 실행합니다.

```
ALTER DATABASE customeracct SET COMPATIBILITY_LEVEL = 120
```

다중 AZ와 인 메모리 최적화 활성화의 고려 사항

Amazon RDS는 SQL Server 데이터베이스 미러링(DBM) 또는 상시 가동 가용성 그룹(AG)을 사용하여 Microsoft SQL Server 기반 DB 인스턴스의 다중 AZ 배포를 지원합니다. 자세한 정보는 [Microsoft SQL Server의 다중 AZ 배포 \(p. 537\)](#) 단원을 참조하십시오.

DB 인스턴스를 다중 AZ 배포로 생성한 경우 기본 DB 인스턴스와 예비 DB 인스턴스가 모두 업그레이드됩니다. Amazon RDS는 룰링 업그레이드를 수행합니다. 장애 조치 기간에 대해서만 중단됩니다.

SQL Server 2014/2016/2017 Enterprise Edition은 인 메모리 최적화를 지원합니다.

옵션 및 파라미터 그룹 고려 사항

옵션 그룹 고려 사항

DB 인스턴스에서 사용자 지정 옵션 그룹을 사용할 경우 Amazon RDS에서 DB 인스턴스에 새 옵션 그룹을 할당할 수 없는 경우도 있습니다. 예를 들어 새로운 메이저 버전으로 업그레이드할 경우, 새 옵션 그룹을 지정해야 합니다. 새 옵션 그룹을 생성하고 동일한 옵션을 기존 사용자 지정 옵션 그룹에 추가하는 것이 좋습니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 또는 [옵션 그룹 복사 \(p. 151\)](#) 단원을 참조하십시오.

파라미터 그룹 고려 사항

DB 인스턴스에서 사용자 지정 파라미터 그룹을 사용할 경우 Amazon RDS에서 DB 인스턴스에 새 파라미터 그룹을 할당할 수 없는 경우도 있습니다. 예를 들어 새로운 메이저 버전으로 업그레이드할 경우, 새 파라미터

그룹을 지정해야 합니다. 새 파라미터 그룹을 생성하고 기존 사용자 지정 파라미터 그룹에서와 같은 방법으로 파라미터를 구성하는 것이 좋습니다.

자세한 내용은 [DB 파라미터 그룹 생성 \(p. 163\)](#) 또는 [DB 파라미터 그룹 복사 \(p. 166\)](#) 단원을 참조하십시오.

업그레이드 테스트

DB 인스턴스에 대한 메이저 버전 업그레이드를 수행하기 전에 데이터베이스 및 해당 데이터베이스에 액세스하는 모든 애플리케이션이 새 버전과 호환되는지 여부를 철저하게 테스트해야 합니다. 다음 절차를 참조하는 것이 좋습니다.

메이저 버전 업그레이드를 테스트하려면

1. 다음과 같이 새 버전의 데이터베이스 엔진에 대한 업그레이드 문서를 검토하여 데이터베이스나 애플리케이션에 영향을 끼칠 수도 있는 호환성 문제가 있는지 살펴봅니다.
 - [SQL Server 2016 또는 2017 업그레이드](#)
 - [SQL Server 2014 업그레이드](#)
 - [SQL Server 2012 업그레이드](#)
2. DB 인스턴스에서 사용자 지정 옵션 그룹을 사용할 경우 업그레이드하려는 새 버전과 호환되는 새 옵션 그룹을 생성합니다. 자세한 내용은 [옵션 그룹 고려 사항 \(p. 518\)](#) 단원을 참조하십시오.
3. DB 인스턴스에서 사용자 지정 파라미터 그룹을 사용할 경우 업그레이드하려는 새 버전과 호환되는 새 파라미터 그룹을 생성합니다. 자세한 내용은 [파라미터 그룹 고려 사항 \(p. 518\)](#) 단원을 참조하십시오.
4. 업그레이드할 DB 인스턴스의 DB 스냅샷을 생성합니다. 자세한 내용은 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.
5. DB 스냅샷을 복구하여 새로운 테스트 DB 인스턴스를 생성합니다. 자세한 내용은 [DB 스냅샷에서 복원 \(p. 205\)](#) 단원을 참조하십시오.
6. 다음 방법 중 한 가지를 사용하여 이 새로운 테스트 DB 인스턴스를 변경하고 새로운 버전으로 업그레이드합니다.
 - [콘솔을 사용하여 DB 인스턴스의 엔진 버전 업그레이드 \(p. 119\)](#)
 - [AWS CLI를 사용하여 DB 인스턴스의 엔진 버전 업그레이드 \(p. 119\)](#)
 - [RDS API를 사용하여 DB 인스턴스의 엔진 버전 업그레이드 \(p. 120\)](#)
7. 업그레이드한 인스턴스에서 사용할 스토리지를 평가하여 업그레이드 시 추가 스토리지의 필요 여부를 결정합니다.
8. 업그레이드한 DB 인스턴스와 관련하여 데이터베이스 및 애플리케이션과 새로운 버전의 호환성을 보장하는 데 필요하다면 최대한 많은 수의 품질 보증 테스트를 실행합니다. 또한 1단계에서 발견된 호환성 문제의 영향을 평가하는 데 필요한 새로운 테스트도 모두 실행합니다. 저장된 프로시저와 함수를 모두 테스트합니다. 업그레이드한 DB 인스턴스에 대해 애플리케이션의 테스트 버전을 실행합니다.
9. 모든 테스트가 통과되면 프로덕션 환경의 DB 인스턴스에도 업그레이드를 실행합니다. 단, 모든 기능이 정상 작동하는 것을 확인할 때까지 쓰기 연산은 DB 인스턴스에 실행하지 않는 것이 좋습니다.

SQL Server DB 인스턴스 업그레이드

SQL Server DB 인스턴스의 수동 또는 자동 업그레이드에 대한 자세한 내용은 다음을 참조하십시오.

- [DB 인스턴스 엔진 버전 업그레이드 \(p. 118\)](#)
- [Amazon RDS for SQL Server의 SQL Server 2008 R2를 SQL Server 2016으로 업그레이드하는 모범 사례](#)

Important

KMS를 사용하여 암호화한 스냅샷이 있는 경우, 지원이 끝나기 전에 업그레이드를 시작하는 것이 좋습니다.

지원 종료 전에 사용되지 않는 DB 인스턴스 업그레이드

메이저 버전 사용이 중단되면 새 DB 인스턴스에 설치할 수 없습니다. RDS는 기존의 모든 DB 인스턴스를 자동으로 업그레이드하려고 시도합니다.

사용되지 않는 DB 인스턴스를 복원해야 하는 경우 특정 시점으로 복원(PITR)하거나 스냅샷을 복원할 수 있습니다. 이렇게 하면 사용되지 않는 버전을 사용하는 DB 인스턴스에 임시 액세스할 수 있습니다. 그러나 메이저 버전이 완전히 사용되지 않게 되면 이 DB 인스턴스도 지원되는 버전으로 자동 업그레이드됩니다.

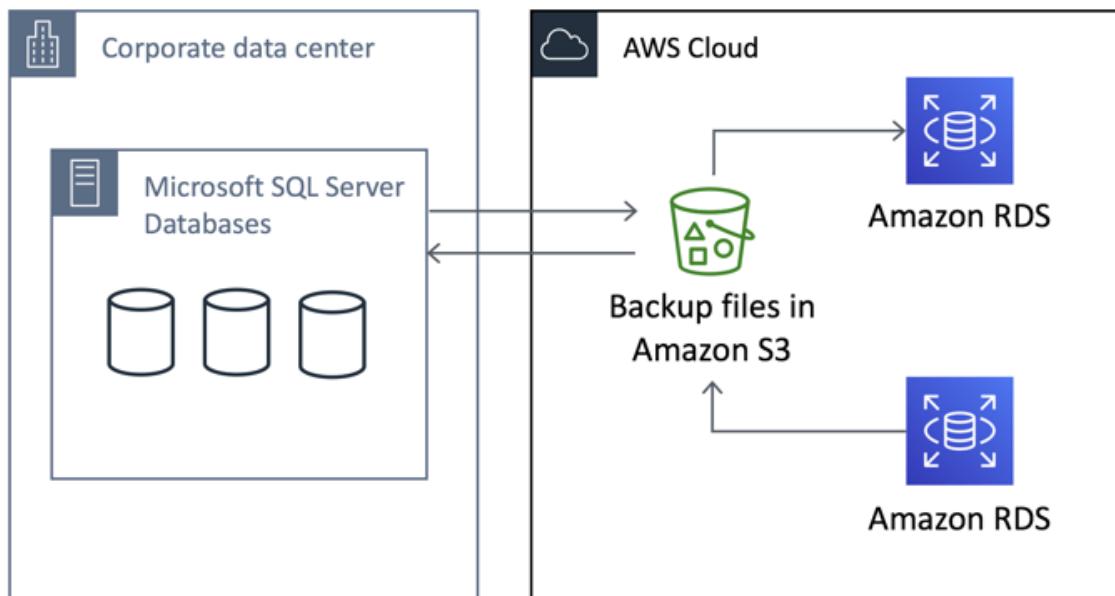
SQL Server 데이터베이스 가져오기 및 내보내기

Amazon RDS는 전체 백업 파일(.bak 파일)을 사용하여 Microsoft SQL Server 데이터베이스에 기본 백업 및 복원을 할 수 있도록 지원합니다. RDS를 사용할 때 데이터베이스 서버의 로컬 파일 시스템을 사용하는 대신 Amazon S3에 저장된 파일에 액세스합니다.

예를 들어 로컬 서버에서 전체 백업을 생성하고, 이를 S3에 저장한 후, 기존 Amazon RDS DB 인스턴스에서 복원할 수 있습니다. 또한 RDS에서 백업을 만들고, 이를 S3에 저장한 후, 어디든 원하는 곳에서 복원할 수 있습니다.

모든 AWS 리전에서 단일 AZ 및 다중 AZ DB 인스턴스에 대해 기본 백업 및 복원을 사용할 수 있습니다. Amazon RDS에서 지원되는 Microsoft SQL Server의 모든 버전에 기본 백업 및 복원이 제공됩니다.

다음 다이어그램은 지원되는 시나리오를 보여 줍니다.



기본 .bak 파일을 사용하여 데이터베이스를 백업 및 복원하는 과정은 대개의 경우 데이터베이스를 가장 빨리 백업하고 복원할 수 있는 방법입니다. 또한 기본 백업 및 복원을 이용하는 것보다 장점이 더 많습니다. 예를 들면,

- Amazon RDS로/에서 데이터베이스 마이그레이션
- Amazon RDS SQL Server DB 인스턴스 간에 데이터베이스를 이동합니다.
- .bak 파일 내부의 데이터, 스키마, 저장 프로시저, 트리거 및 기타 데이터베이스 코드를 마이그레이션합니다.

- DB 인스턴스 전체가 아닌 데이터베이스 하나를 백업 및 복원합니다.
- 개발, 테스트, 교육, 데모를 위해 데이터베이스 사본을 만듭니다.
- 재해 복구를 위한 추가 보호 계층을 위해 Amazon S3를 통해 백업 파일을 저장 및 전송합니다.

제한 및 권장 사항

다음은 기본 백업 및 복원을 사용할 때 적용되는 몇 가지 제한 사항입니다.

- Amazon RDS DB 인스턴스와 다른 AWS 리전에서는 Amazon S3 버킷으로 백업하거나 복원할 수 없습니다.
- 한 표준 시간대의 백업 파일을 다른 표준 시간대로 복원하지 않는 것이 좋습니다. 한 표준 시간대의 백업 파일을 다른 표준 시간대로 복원하는 경우 쿼리와 애플리케이션을 감사하여 표준 시간대 변경의 영향을 확인해야 합니다.
- 1TB보다 큰 데이터베이스의 기본 백업은 지원되지 않습니다.
- RDS는 데이터베이스의 최대 5TB의 기본 복원을 지원합니다. SQL-Server Express에서 데이터베이스 기본 복원은 MSSQL 에디션에 의해 10GB 미만으로 제한됩니다.
- 유지 관리 기간 또는 Amazon RDS에서 데이터베이스 스냅샷을 만드는 동안에는 기본 백업을 수행할 수 없습니다.
- 다중 AZ DB 인스턴스에서는 전체 복구 모델로 백업된 데이터베이스만 기본적으로 복원할 수 있습니다.
- 트랜잭션 내에서 기본 백업 및 복원을 위한 RDS 프로시저 호출은 지원되지 않습니다.
- 기본 백업 파일은 "암호화 전용" 암호 모드를 사용하여 지정된 AWS Key Management Service 키로 암호화됩니다. 암호화된 백업 파일을 복원할 때 파일이 "암호화 전용" 암호 모드로 암호화되었음을 알고 있어야 합니다.

백업 파일을 만들고, 복사하고, 복원하는 동안 데이터베이스를 오프라인 상태로 둘 수 있다면 RDS로 마이그레이션할 때 기본 백업 및 복원을 사용하는 것이 좋습니다. 온프레미스 데이터베이스를 오프라인으로 전환할 수 없다면 AWS Database Migration Service를 사용하여 데이터베이스를 Amazon RDS로 마이그레이션하는 것이 좋습니다. 자세한 내용은 [AWS Database Migration Service란 무엇입니까?](#)를 참조하십시오.

기본 백업 및 복원은 리전 간 스냅샷 복사 기능의 데이터 복구 대신 사용하기 위한 기능이 아닙니다. Amazon RDS에서 리전 간 재해 복구를 위해 데이터베이스 스냅샷을 다른 리전으로 복사할 때는 스냅샷 복사를 이용하는 것이 좋습니다. 자세한 내용은 [스냅샷 복사 \(p. 207\)](#) 단원을 참조하십시오.

기본 백업 및 복원 설정

기본 백업 및 복원을 설정하려면 다음 세 가지 구성 요소가 필요합니다.

1. 백업 파일을 저장할 Amazon S3 버킷.

백업 파일에 사용한 후 RDS로 마이그레이션하려는 백업을 업로드할 S3 버킷이 있어야 합니다. 이미 Amazon S3 버킷이 있으면 그 버킷을 사용하면 됩니다. 없는 경우 [버킷을 생성](#)할 수 있습니다. 또는 AWS Management 콘솔을 사용하여 `SQlSERVER_BACKUP_RESTORE` 옵션을 추가할 때 새 버킷이 생성되도록 선택할 수도 있습니다.

S3 사용에 대한 정보는 Amazon Simple Storage Service 시작 안내서의 간단한 소개를 참조하십시오. 자세한 내용은 Amazon Simple Storage Service 콘솔 사용 설명서를 참조하십시오.

2. 버킷에 액세스하기 위한 AWS Identity and Access Management(IAM) 역할.

이미 IAM 역할이 있으면 그 역할을 사용하면 됩니다. IAM 역할이 없는 경우에는 수동으로 새 역할을 만들 수 있습니다. 또는 AWS Management 콘솔을 사용하여 `SQlSERVER_BACKUP_RESTORE` 옵션을 추가할 때 새 IAM 역할이 생성되도록 선택할 수도 있습니다.

수동으로 새 IAM 역할을 만들려면 다음 단원에서 설명하는 방법을 사용하십시오. 수동으로 기존의 IAM 역할에 신뢰 및 권한 정책을 연결하려면 다음 단원에서 설명하는 접근 방식을 사용하십시오.

3. DB 인스턴스의 옵션 그룹에 SQLSERVER_BACKUP_RESTORE 옵션 추가.

DB 인스턴스에서 기본 백업 및 복원을 활성화하려면 DB 인스턴스의 옵션 그룹에 SQLSERVER_BACKUP_RESTORE 옵션을 추가합니다. 자세한 정보와 지침은 [Microsoft SQL Server에서 기본 백업 및 복원 지원 \(p. 545\)](#) 단원을 참조하십시오.

기본 백업 및 복원을 위한 IAM 역할 수동으로 만들기

기본 백업 및 복원에 사용할 새 IAM 역할을 수동으로 만들려면 먼저 Amazon RDS 서비스에서 Amazon S3 버킷으로 권한을 위임하기 위한 역할을 만들어야 합니다. IAM 역할을 만들 때 신뢰 및 권한 정책을 연결합니다. 신뢰 정책은 RDS가 이 역할을 맡도록 허용합니다. 권한 정책은 이 역할이 수행할 수 있는 작업을 정의합니다. 역할을 만드는 방법에 대한 자세한 내용은 [AWS 서비스에 대한 권한을 위임할 역할 생성](#)을 참조하십시오.

기본 백업 및 복원 기능의 경우, 이 단원의 예제와 비슷한 신뢰 및 권한 정책을 사용합니다. 다음 예에서 서비스 원칙 이름인 `rds.amazonaws.com`를 모든 서비스 계정의 별칭으로 사용했습니다. 다른 예에서는 신뢰 정책에서 액세스 권한을 부여할 다른 계정, 사용자, 역할을 식별하기 위해 Amazon 리소스 이름(ARN)을 지정합니다.

Example 기본 백업 및 복원을 위한 신뢰 정책

```
{  
    "Version": "2012-10-17",  
    "Statement":  
    [ {  
        "Effect": "Allow",  
        "Principal": {"Service": "rds.amazonaws.com"},  
        "Action": "sts:AssumeRole"  
    }]  
}
```

다음 예에서는 ARN을 사용하여 리소스를 지정합니다. ARN 사용에 대한 자세한 내용은 [Amazon 리소스 이름\(ARN\)](#)을 참조하십시오.

Example 암호화 지원 없는 기본 백업 및 복원을 위한 권한 정책

```
{  
    "Version": "2012-10-17",  
    "Statement":  
    [ {  
        "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/*"
    }
}
```

Example 암호화 지원 있는 기본 백업 및 복원을 위한 권한 정책

백업 파일을 암호화하려면 권한 정책에 암호화 키를 포함시켜야 합니다. 암호화 키에 대한 자세한 내용은 AWS Key Management Service(AWS KMS) 설명서의 [시작하기](#)를 참조하십시오.

```
{
    "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/*"
        }
    ]
}
```

기본 백업 및 복원 사용

기본 백업 및 복원을 활성화하고 구성한 다음에는 사용하기 시작할 수 있습니다. 먼저 Microsoft SQL Server 데이터베이스에 연결한 다음 해당 작업을 위한 Amazon RDS 저장 프로시저를 호출합니다. 데이터베이스 연결 방법에 대한 자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 503\)](#) 단원을 참조하십시오.

저장 프로시저 중에는 Amazon S3 버킷과 파일에 ARN(Amazon 리소스 이름)을 제공해야 하는 것도 있습니다. ARN의 형식은 `arn:aws:s3:::bucket_name/file_name`입니다. Amazon S3에서는 ARN에 계정 번호나 리전을 요구하지 않습니다. 선택적 AWS KMS 암호화 키도 제공하는 경우 ARN 형식은 `arn:aws:kms:region:account-id:key/key-id`입니다. 자세한 내용은 [ARN\(Amazon 리소스 이름\)](#) 및 [AWS 서비스 네임스페이스](#) 단원을 참조하십시오.

데이터베이스를 백업하고, 복원하고, 진행 중인 작업을 취소하고, 백업 및 복원 작업의 상태를 추적하기 위한 저장 프로시저가 있습니다. 각각의 저장 프로시저를 호출하는 방법에 대한 자침은 다음 하위 단원을 참조하십시오.

- 데이터베이스 백업 (p. 524)
- 데이터베이스 복원 (p. 525)
- 작업 취소 (p. 526)
- 작업 상태 추적 (p. 526)

데이터베이스 백업

데이터베이스를 백업하려면 `rds_backup_database` 저장 프로시저를 호출해야 합니다.

Note

유지 관리 기간 중 또는 Amazon RDS에서 스냅샷을 만드는 동안에는 데이터베이스를 백업할 수 없습니다.

다음 파라미터는 필수 파라미터입니다.

- `@source_db_name` – 백업할 데이터베이스의 이름입니다.
- `@s3_arn_to_backup_to` – 백업에 사용할 버킷과 파일 이름입니다(Amazon S3 버킷 + 키 ARN).

파일에는 어떤 확장명이든 있을 수 있지만 `.bak`이 기본적입니다.

다음 파라미터는 선택적입니다.

- `@kms_master_key_arn` – 백업을 암호화 할 키입니다(KMS 고객 마스터 키 ARN). AWS KMS 키 식별자 를 지정하지 않으면 Amazon RDS가 새 DB 인스턴스에 대해 기본 암호화 키를 사용합니다. AWS KMS는 AWS 계정의 Amazon RDS에 대한 기본 암호화 키를 생성합니다. AWS 계정에는 AWS 리전마다 다른 기본 암호화 키가 있습니다.

자세한 내용은 [Amazon RDS 리소스 암호화](#) 단원을 참조하십시오.

- `@overwrite_S3_backup_file` – 기본값은 0입니다.
 - 0 – 기존 파일을 덮어쓰지 않습니다. 파일이 이미 있는 경우 그 대신 오류를 반환합니다.
 - 1 – 백업 파일이 아니더라도 지정된 이름이 있는 기존 파일을 덮어씁니다.
- `@type` – 기본적으로 `FULL`로 설정되며 대소문자를 구분하지 않습니다.
 - `differential` – 차등 백업을 수행합니다.
 - `full` – 전체 백업을 수행합니다.

Example 암호화 없는 차등 백업

```
exec msdb.dbo.rds_backup_database
    @source_db_name='database_name',
    @s3_arn_to_backup_to='arn:aws:s3:::bucket_name/file_name_and_extension',
    @overwrite_S3_backup_file=1,
    @type='differential';
```

Example 암호화를 사용하는 전체 백업

```
exec msdb.dbo.rds_backup_database
    @source_db_name='database_name',
    @s3_arn_to_backup_to='arn:aws:s3:::bucket_name/file_name_and_extension',
    @kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id',
```

```
@overwrite_S3_backup_file=1,  
@type='FULL';
```

차등 백업은 마지막 전체 백업을 기반을 합니다. 차등 백업이 작동할 수 있도록 마지막 전체 백업과 차등 백업 간에는 스냅샷을 만들 수 없습니다. 차등 백업을 만들려고 할 때 스냅샷이 이미 있으면 차등 백업을 진행하기 전에 다른 전체 백업을 만드십시오.

다음 샘플 SQL을 사용하여 마지막 전체 백업 또는 스냅샷을 찾을 수 있습니다.

```
select top 1  
database_name  
, backup_start_date  
, backup_finish_date  
from msdb.dbo.backupset  
where database_name='name-of-database'  
and type = 'D'  
order by backup_start_date desc;
```

데이터베이스 복원

데이터베이스를 복원하려면 `rds_restore_database` 저장 프로시저를 호출해야 합니다.

다음 파라미터는 필수 파라미터입니다.

- `@restore_db_name` – 복원할 데이터베이스의 이름입니다.
- `@s3_arn_to_restore_from` – 데이터베이스를 복원하는 데 사용된 백업 파일의 Amazon S3 ARN 접두사입니다. 단일 파일 백업의 경우 전체 이름을 입력하십시오. 다중 파일 백업에서 복원하려면 파일에 공통으로 사용되는 접두사를 입력한 다음 접미사로 별표(*)를 붙입니다. 다음은 예제입니다.

다음 예제에서는 단일 파일 복원을 보여줍니다.

Example

```
exec msdb.dbo.rds_restore_database  
@restore_db_name='database_name',  
@s3_arn_to_restore_from='arn:aws:s3:::bucket_name/bakup_file'
```

여러 파일을 복원하는 동안 오류를 방지하려면 모든 백업 파일의 접두사가 같고 다른 파일은 이 접두사를 사용하지 않아야 합니다.

다음 예제에서는 다중 파일 복원을 보여줍니다.

Example

```
exec msdb.dbo.rds_restore_database  
@restore_db_name='database_name',  
@s3_arn_to_restore_from='arn:aws:s3:::bucket_name/bakup_file_*'
```

`@s3_arn_to_restore_from`이 비어 있으면 다음 오류가 반환됩니다. "S3 ARN prefix cannot be empty(S3 ARN 접두사는 비워 둘 수 없습니다)".

다음 파라미터는 선택적입니다.

- `@kms_master_key_arn` – 백업 파일을 암호화한 경우 파일 해독에 사용할 키입니다.

다음 예제는 암호화를 사용하지 않는 복원을 보여줍니다.

Example

```
exec msdb.dbo.rds_restore_database
    @restore_db_name='database_name',
    @s3_arn_to_restore_from='arn:aws:s3:::bucket_name/file_name_and_extension';
```

다음 예제에서는 암호화를 사용한 복원을 보여줍니다.

Example

```
exec msdb.dbo.rds_restore_database
    @restore_db_name='database_name',
    @s3_arn_to_restore_from='arn:aws:s3:::bucket_name/file_name_and_extension',
    @kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id';
```

작업 취소

백업 또는 복원 작업을 취소하려면 `rds_cancel_task` 저장 프로시저를 호출합니다.

다음 파라미터는 선택적입니다.

- `@db_name` – 작업을 취소할 데이터베이스의 이름입니다.
- `@task_id` – 취소할 작업의 ID입니다. 작업 ID는 `rds_task_status`를 호출하여 확인할 수 있습니다.

Example

```
exec msdb.dbo.rds_cancel_task @task_id=1234;
```

작업 상태 추적

백업 및 복원 작업의 상태를 추적하려면 `rds_task_status` 저장 프로시저를 호출합니다. 파라미터를 제공하지 않으면 이 저장 프로시저는 모든 작업의 상태를 반환합니다. 작업 상태는 약 2분마다 업데이트됩니다.

다음 파라미터는 선택적입니다.

- `@db_name` – 작업 상태를 표시할 데이터베이스의 이름입니다.
- `@task_id` – 작업 상태를 표시할 작업의 ID입니다.

Example

```
exec msdb.dbo.rds_task_status @db_name='database_name';
```

`rds_task_status` 저장 프로시저는 다음과 같은 열을 반환합니다.

열	설명
<code>task_id</code>	작업의 ID입니다.
<code>task_type</code>	백업 작업에 대한 <code>BACKUP_DB</code> 또는 복원 작업에 대한 <code>RESTORE_DB</code> 중 하나입니다.
<code>database_name</code>	작업이 연결되어 있는 데이터베이스의 이름입니다.

열	설명
% complete	백분율로 나타낸 작업의 진행률입니다.
duration (mins)	작업에 소요된 시간입니다(분).
lifecycle	<p>작업의 상태입니다. 가능한 작업 상태는 다음과 같습니다.</p> <ul style="list-style-type: none"> • CREATED – rds_backup_database 또는 rds_restore_database를 호출하는 즉시 작업이 생성되고 상태는 CREATED로 설정됩니다. • IN_PROGRESS – 백업 또는 복원 작업이 시작된 후 상태가 IN_PROGRESS로 설정됩니다. CREATED에서 IN_PROGRESS로 상태가 변경되려면 최대 5분이 걸릴 수 있습니다. • SUCCESS – 백업 또는 복원 작업이 완료된 후 상태가 SUCCESS로 설정됩니다. • ERROR – 백업 또는 복원 작업이 실패한 후 상태가 ERROR로 설정됩니다. 오류에 대한 자세한 내용은 task_info 열을 참조하십시오. • CANCEL_REQUESTED – rds_cancel_task를 호출하는 즉시 작업의 상태가 CANCEL_REQUESTED로 설정됩니다. • CANCELLED – 작업이 성공적으로 취소된 뒤에는 작업 상태가 CANCELLED로 설정됩니다.
task_info	<p>작업에 대한 추가 정보입니다.</p> <p>데이터베이스를 백업하거나 복원하는 동안 오류가 발생하면 이 열에 오류에 대한 정보가 포함됩니다. 가능한 오류 목록 및 완화 전략은 문제 해결 (p. 528) 단원을 참조하십시오.</p>
last_updated	작업 상태를 마지막으로 업데이트한 날짜와 시간입니다. 상태는 5% 진행 후마다 업데이트됩니다.
created_at	작업을 생성한 날짜와 시간입니다.
overwrite_S3_backup_file	백업 작업을 호출할 때 지정된 @overwrite_S3_backup_file 파라미터의 값입니다. 자세한 내용은 데이터베이스 백업 (p. 524) 단원을 참조하십시오.

백업 파일 압축

Amazon S3 버킷 용량 절약을 위해 백업 파일을 압축할 수 있습니다. 백업 파일 압축에 대한 자세한 내용은 Microsoft 설명서의 [Backup Compression](#)을 참조하십시오.

백업 파일 압축은 다음 데이터베이스 에디션에 지원됩니다.

- Microsoft SQL Server Enterprise Edition
- Microsoft SQL Server Standard Edition

백업 파일 압축을 설정하려면 다음 코드를 실행하십시오.

```
exec rdsadmin..rds_set_configuration 'S3 backup compression', 'true';
```

백업 파일 압축을 해제하려면 다음 코드를 실행하십시오.

```
exec rdsadmin..rds_set_configuration 'S3 backup compression', 'false';
```

문제 해결

다음은 기본 백업 및 복원을 사용할 때 생길 수 있는 문제들입니다.

문제	문제 해결 제안
Access Denied	<p>백업 또는 복원 프로세스는 백업 파일에 액세스할 수 없습니다. 이것의 원인이 되는 문제는 일반적으로 다음과 같습니다.</p> <ul style="list-style-type: none">올바르지 않은 버킷 참조. 올바르지 않은 형식을 사용하여 버킷 참조 ARN을 사용하지 않고 파일 이름 참조.버킷 파일에 올바르지 않은 권한. 예컨대 지금 액세스를 시도하는 계정과 다른 계정에서 생성한 경우 올바른 권한을 추가하십시오.올바르지 않거나 완전하지 않은 IAM 정책. IAM 역할에는 올바른 버전을 비롯하여 필요한 요소가 모두 포함되어야 합니다. 이에 대해서는 SQL Server 데이터베이스 가져오기 및 내보내기 (p. 520)에 강조 표시되어 있습니다.
BACKUP DATABASE WITH COMPRESSION is not supported on <edition_name> Edition	<p>백업 파일 압축은 Microsoft SQL Server Enterprise Edition과 Standard Edition에서만 지원됩니다.</p> <p>자세한 내용은 백업 파일 압축 (p. 527) 단원을 참조하십시오.</p>
Key <ARN> does not exist	<p>암호화된 백업을 복원하려고 시도했으나 유효한 암호화 키를 제공하지 않았습니다. 암호화 키를 확인한 후 다시 시도하십시오.</p> <p>자세한 내용은 데이터베이스 복원 (p. 525) 단원을 참조하십시오.</p>
Please reissue task with correct type and overwrite property	<p>데이터베이스 백업을 시도할 때 이미 존재하는 파일 이름을 입력하고 덮어쓰기 속성을 false로 설정하면 저장 작업이 실패합니다. 이 오류를 해결하려면 존재하지 않는 파일 이름을 입력하거나 덮어쓰기 속성을 true로 설정하십시오.</p> <p>자세한 내용은 데이터베이스 백업 (p. 524) 단원을 참조하십시오.</p> <p>데이터베이스를 복원할 때 잘못하여 rds_backup_database 저장 프로시저를 호출했을 수도 있습니다. 이 경우 대신 rds_restore_database 저장 프로시저를 호출하십시오.</p> <p>자세한 내용은 데이터베이스 복원 (p. 525) 단원을 참조하십시오.</p> <p>데이터베이스를 복원할 때 rds_restore_database 저장 프로시저를 호출한 경우, 유효한 백업 파일 이름을 입력했는지 확인하십시오.</p> <p>자세한 내용은 기본 백업 및 복원 사용 (p. 523) 단원을 참조하십시오.</p>
Please specify a bucket that is in the same region as RDS instance	<p>Amazon RDS DB 인스턴스와 다른 AWS 리전에서는 Amazon S3 버킷으로 백업하거나 복원할 수 없습니다. Amazon S3 복제를 사용하여 백업 파일을 올바른 리전에 복사할 수 있습니다.</p> <p>자세한 내용은 Amazon S3 설명서의 리전 간 복제 단원을 참조하십시오.</p>
The specified bucket does not exist	<p>버킷 및 파일에 대해 올바른 ARN을 올바른 형식으로 입력했는지 확인하십시오.</p> <p>자세한 내용은 기본 백업 및 복원 사용 (p. 523) 단원을 참조하십시오.</p>

문제	문제 해결 제안
User <ARN> is not authorized to perform <kms action> on resource <ARN>	암호화된 작업을 요청했으나 올바른 AWS KMS 권한을 제공하지 않았습니다. 올바른 권한이 있는지 확인하거나 올바른 권한을 추가하십시오. 자세한 내용은 기본 백업 및 복원 설정 (p. 521) 단원을 참조하십시오.
The Restore task is unable to restore from more than n backup file(s). Please reduce the number of files matched and try again.	복원하려는 파일 수를 줄입니다. 필요한 경우 개별 파일을 더 크게 만들 수 있습니다.

관련 주제

- 다른 방법으로 SQL Server 데이터 가져오기 및 내보내기 ([p. 529](#))
- Amazon RDS DB 인스턴스 백업 및 복원 ([p. 196](#))

다른 방법으로 SQL Server 데이터 가져오기 및 내보내기

다음에는 스냅샷을 사용하여 Microsoft SQL Server 데이터를 Amazon RDS로 가져오는 방법과 SQL Server를 실행하는 Amazon RDS DB 인스턴스에서 데이터를 내보내는 방법에 대한 정보를 찾을 수 있습니다.

이 방법이 가능한 시나리오라면 기본 백업 및 복원 기능을 사용하여 Amazon RDS 안팎으로 데이터를 옮기는 것이 더 쉽습니다. 자세한 내용은 [SQL Server 데이터베이스 가져오기 및 내보내기 \(p. 520\)](#) 단원을 참조하십시오.

Note

Microsoft SQL Server용 Amazon RDS는 msdb 데이터베이스로 데이터 가져오기를 지원하지 않습니다.

스냅샷을 사용하여 Amazon RDS의 SQL Server로 데이터 가져오기

스냅샷을 사용하여 SQL Server DB 인스턴스로 데이터를 가져오려면

1. DB 인스턴스를 만듭니다. 자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 생성 \(p. 494\)](#) 단원을 참조하십시오.
2. 애플리케이션이 대상 DB 인스턴스에 액세스하지 못하게 차단합니다.

데이터를 가져오는 동안 DB 인스턴스에 대한 액세스를 금지하면 데이터 전송이 더 빨라집니다. 또한, 다른 애플리케이션이 DB 인스턴스에 동시에 쓸 수 없는 경우 데이터가 로드되는 동안 충돌에 대해 걱정할 필요가 없습니다. 원가 잘못되어 이전의 데이터베이스 스냅샷으로 롤백해야 할 때 잃게 되는 유일한 변경 내용은 가져온 데이터이며, 문제를 해결한 후에 다시 가져올 수 있습니다.

DB 인스턴스 액세스 제어에 대한 자세한 정보는 [DB 보안 그룹 작업\(EC2-Classic 플랫폼\) \(p. 394\)](#)을(를) 참조하십시오.

3. 대상 데이터베이스의 스냅샷을 만듭니다.

대상 데이터베이스가 이미 데이터로 채워져 있으면 데이터베이스의 스냅샷을 만든 후에 데이터를 가져오는 것이 좋습니다. 데이터 가져오기에 문제가 있거나 변경 내용을 취소하려는 경우 스냅샷을 사용하여 데이터베이스를 이전 상태로 복원할 수 있습니다. 데이터베이스 스냅샷에 대한 자세한 정보는 [DB 스냅샷 생성 \(p. 203\)](#)을(를) 참조하십시오.

Note

데이터베이스 스냅샷을 만들 때, 백업이 진행되는 동안 데이터베이스에 대한 I/O 작업이 약 10초간 일시 중단됩니다.

4. 대상 데이터베이스에서 자동 백업을 비활성화합니다.

자동 백업이 비활성화되어 있을 때는 Amazon RDS가 트랜잭션을 로그에 기록하지 않기 때문에 대상 DB 인스턴스에서 자동 백업을 비활성화하면 데이터를 가져오는 동안 성능이 향상됩니다. 하지만 몇 가지 고려해야 할 사항이 있습니다. 자동 백업은 특정 시점으로 복구하는 데 꼭 필요하기 때문에, 데이터를 가져오는 동안 특정 시점으로 데이터베이스를 복원할 수는 없습니다. 뿐만 아니라, DB 인스턴스에서 만든 자동 백업이 지워집니다. 그래도 이전 스냅샷을 사용하여 데이터베이스를 복구할 수 있고, 지금까지 만든 스냅샷은 전부 그대로 사용할 수 있습니다. 자동 백업에 대한 자세한 내용은 [백업 작업 \(p. 196\)](#) 단원을 참조하십시오.

5. 해당될 경우 외래 키 제약 조건을 비활성화합니다.

외래 키 제약 조건을 비활성화할 필요가 있는 경우 다음 스크립트로 비활성화할 수 있습니다.

```
--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. 해당될 경우 인덱스를 삭제합니다.
7. 트리거가 있는 경우 비활성화합니다.

트리거를 비활성화해야 하는 경우 다음 스크립트로 비활성화할 수 있습니다.

```
--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. 원본 SQL Server 인스턴스에서 대상 DB 인스턴스로 가져오려는 로그인을 쿼리합니다.

SQL Server는 master 데이터베이스에 로그인과 암호를 저장합니다. Amazon RDS는 master 데이터베이스에 액세스 권한을 부여하지 않기 때문에, 로그인 정보와 암호를 대상 DB 인스턴스로 직접 가져올 수는 없습니다. 그 대신, 원본 SQL Server 인스턴스에서 master 데이터베이스를 쿼리하여 대상 DB 인스턴스에 추가하려는 모든 로그인 및 암호와 전송하려는 역할 멤버십과 권한도 포함하는 데이터 정의 언어(DDL) 파일을 생성해야 합니다.

master 데이터베이스 쿼리에 대한 자세한 정보는 Microsoft 기술 자료의 [SQL Server 2005와 SQL Server 2008 인스턴스 간에 로그인 및 암호를 전송하는 방법](#)을 참조하십시오.

스크립트의 출력은 대상 DB 인스턴스에서 실행할 수 있는 또 다른 스크립트입니다. 기술 자료 문서에 있는 스크립트에 다음 코드가 있을 경우

```
p.type IN
```

p.type이 나타나는 곳에서는 모두 다음 코드를 대신 사용합니다.

```
p.type = 'S'
```

9. 데이터 가져오기 ([p. 532](#))의 방법을 사용하여 데이터를 가져옵니다.
10. 애플리케이션에 대상 DB 인스턴스에 대한 액세스 권한을 부여합니다.

데이터 가져오기가 완료되면 가져오기 중에 차단한 애플리케이션에 DB 인스턴스 액세스 권한을 부여 할 수 있습니다. DB 인스턴스 액세스 제어에 대한 자세한 정보는 [DB 보안 그룹 작업\(EC2-Classic 플랫폼\) \(\[p. 394\]\(#\)\)](#)을(를) 참조하십시오.

11. 대상 DB 인스턴스에서 자동 백업을 활성화합니다.

자동 백업에 대한 자세한 내용은 [백업 작업 \(\[p. 196\]\(#\)\)](#) 단원을 참조하십시오.

12. 외래 키 제약 조건을 활성화합니다.

앞서 외래 키 제약 조건을 비활성화한 경우 다음 스크립트로 활성화할 수 있습니다.

```
--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. 인덱스가 있는 경우 인덱스를 활성화합니다.
14. 트리거가 있는 경우 트리거를 활성화합니다.

앞서 트리거를 비활성화했다면 다음 스크립트로 활성화할 수 있습니다.

```
--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;
```

데이터 가져오기

Microsoft SQL Server Management Studio는 Express Edition을 제외한 모든 Microsoft SQL Server 버전에 포함되는 그래픽 SQL Server 클라이언트입니다. SQL Server Management Studio Express는 Microsoft에서 무료 다운로드로 사용할 수 있습니다. 다운로드에 관해서는 [Microsoft 웹 사이트](#)를(를) 참조하십시오.

Note

SQL Server Management Studio는 Windows 기반 애플리케이션으로만 사용할 수 있습니다.

SQL Server Management Studio에는 SQL Server DB 인스턴스로 데이터를 가져오는 데 유용한 다음 도구가 포함됩니다.

- 스크립트 생성 및 게시 마법사
- 가져오기 및 내보내기 마법사
- 대량 복사

스크립트 생성 및 게시 마법사

스크립트 생성 및 게시 마법사는 데이터베이스의 스키마, 데이터 자체 또는 두 가지를 모두 포함한 스크립트를 만듭니다. 로컬 SQL Server 배포 위치에서 데이터베이스용 스크립트를 생성하면 그 스크립트를 실행하여 데이터베이스에 포함된 정보를 Amazon RDS DB 인스턴스로 전송할 수 있습니다.

Note

1GiB 이상의 데이터베이스에 대해서는 데이터베이스 스키마만 스크립트로 작성한 다음 가져오기 및 내보내기 마법사 또는 SQL Server의 대량 복사 기능을 사용하여 데이터를 전송하는 것이 더 효율적입니다.

스크립트 생성 및 게시 마법사에 대한 자세한 정보는 [Microsoft SQL Server 문서](#)를(를) 참조하십시오.

마법사에서는 특히 [스크립팅 옵션 설정] 페이지의 고급 옵션에 주의하여 스크립트에 포함하려는 모든 것이 선택되어 있는지 확인하십시오. 예를 들어 데이터베이스 트리거는 기본적으로 스크립트에 포함되지 않습니다.

스크립트가 생성되고 저장되면 SQL Server Management Studio를 사용하여 DB 인스턴스에 연결한 후 스크립트를 실행할 수 있습니다.

가져오기 및 내보내기 마법사

가져오기 및 내보내기 마법사는 특별한 통합 서비스 패키지를 만드는데, 이 패키지를 사용하여 로컬 SQL Server 데이터베이스에서 대상 DB 인스턴스로 데이터를 복사할 수 있습니다. 이 마법사에서는 대상 DB 인스턴스로 어떤 테이블, 심지어 테이블 내에 있는 어떤 풀을 복사할지 필터링할 수 있습니다.

Note

가져오기 및 내보내기 마법사는 큰 데이터 집합에 매우 효과적이지만, 로컬 배포 위치에서 원격으로 데이터를 내보내는 가장 빠른 방법은 아닐 수도 있습니다. 훨씬 더 빠른 방법을 원한다면, SQL Server 대량 복사 기능을 고려하십시오.

가져오기 및 내보내기 마법사에 대한 자세한 내용은 [Microsoft SQL Server 문서](#)를 참조하십시오.

이 마법사의 [대상 선택] 페이지에서 다음을 수행합니다.

- [서버 이름]에 DB 인스턴스의 앤드포인트 이름을 입력합니다.
- 서버 인증 모드를 위해 [SQL Server 인증 사용]을 선택합니다.
- [사용자 이름] 및 [암호]에 DB 인스턴스용으로 만든 마스터 사용자의 자격 증명을 입력합니다.

대량 복사

SQL Server 대량 복사 기능은 원본 데이터베이스에서 DB 인스턴스로 데이터를 복사하는 효율적인 수단입니다. 대량 복사는 ASCII 파일과 같은 데이터 파일에 사용자가 지정하는 데이터를 쓰는 기능입니다. 대량 복사를 다시 실행하여 파일의 내용을 대상 DB 인스턴스에 쓸 수 있습니다.

이 섹션에서는 모든 SQL Server 버전에 포함되어 있는 bcp 유ти리티를 사용합니다. 대량 가져오기 및 내보내기 작업에 대한 자세한 정보는 [Microsoft SQL Server 문서](#)를 참조하십시오.

Note

대량 복사 기능을 사용하기 전에, 우선 데이터베이스 스키마를 대상 DB 인스턴스로 가져와야 합니다. 이 주제의 앞 부분에서 설명한 스크립트 생성 및 게시 마법사는 이 목적으로 사용하기에 훌륭한 도구입니다.

다음 명령으로 로컬 SQL Server 인스턴스에 연결하여 기존 SQL Server 배포의 C:\ 루트 디렉터리에 지정된 테이블의 탭으로 구분된 파일을 생성할 수 있습니다. 테이블은 정규화된 이름으로 지정되고, 텍스트 파일은 복사되는 테이블과 이름이 같습니다.

```
bcp dbname.schema_name.table_name out C:\table_name.txt -n -S localhost -U username -P password -b 10000
```

앞에 나온 코드에는 다음 옵션이 포함됩니다.

- **-n**은 대량 복사에서 복사되는 데이터의 기본 데이터 형식을 사용할 것임을 지정합니다.
- **-S**는 bcp 유ти리티가 연결할 SQL Server 인스턴스를 지정합니다.
- **-U**는 SQL Server 인스턴스에 로그인할 계정의 사용자 이름을 지정합니다.
- **-P**는 **-U**로 지정된 사용자의 암호를 지정합니다.
- **-b**는 가져온 데이터의 배치당 행 수를 지정합니다.

Note

가져오기 작업에 중요한 다른 파라미터가 있을 수 있습니다. 예를 들어, 자격 증명 값에 속한 **-E** 파라미터가 필요할 수 있습니다. 자세한 내용은 [Microsoft SQL Server 문서](#)에서 bcp 유ти리티 관련 명령줄 구문에 대한 상세 설명을 참조하십시오.

예를 들어 기본 스키마 **dbo**를 사용하는 **store**라는 이름의 데이터베이스에 **customers**라는 이름의 테이블이 있다고 가정합시다. 암호가 **insecure**인 사용자 계정 **admin**이 **customers** 테이블에서 10,000개의 행을 **customers.txt**라는 파일로 복사합니다.

```
bcp store.dbo.customers out C:\customers.txt -n -S localhost -U admin -P insecure -b 10000
```

데이터 파일을 생성한 후, 대상 DB 인스턴스에 데이터베이스와 스키마를 만든 경우 비슷한 명령을 사용하여 DB 인스턴스로 데이터를 업로드할 수 있습니다. 이 경우에는 출력 파일을 지정하는 **out** 대신 입력 파일을 지정하는 **in** 인수를 사용합니다. 로컬 SQL Server 인스턴스를 지정하기 위해 **localhost**를 사용하는 대신, DB 인스턴스의 엔드포인트를 지정합니다. 1433 이외의 포트를 사용하는 경우 그 포트도 지정할 것입니다. 사용자 이름과 암호는 DB 인스턴스에 대한 마스터 사용자와 암호가 됩니다. 구문은 다음과 같습니다.

```
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
```

이전 예제를 진행하기 위해, 마스터 사용자 이름이 **admin**이고 암호가 **insecure**라고 가정합니다. DB 인스턴스의 엔드포인트는 **rds.ckz2kqd4qsn1.us-east-1.rds.amazonaws.com**이고, 포트 4080을 사용합니다. 명령은 다음과 같습니다.

```
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
```

Amazon RDS 상의 SQL Server에서 데이터 내보내기

다음 옵션 중 하나를 선택하여 Amazon RDS SQL DB 인스턴스에서 데이터를 내보낼 수 있습니다.

- 전체 백업 파일(.bak)을 사용한 기본 데이터베이스 백업 – .bak 파일을 사용하여 데이터베이스를 백업하는 과정은 철저히 최적화되어 있으며, 대개의 경우 데이터를 가장 빨리 내보내는 방법입니다. 자세한 내용은 [SQL Server 데이터베이스 가져오기 및 내보내기 \(p. 520\)](#) 단원을 참조하십시오.
- SQL Server 가져오기 및 내보내기 마법사 – 자세한 내용은 [SQL Server 가져오기 및 내보내기 마법사 \(p. 534\)](#) 단원을 참조하십시오.
- SQL Server 스크립트 생성 및 게시 마법사와 bcp 유ти리티 – 자세한 내용은 [SQL Server 스크립트 생성 및 게시 마법사와 bcp 유ти리티 \(p. 536\)](#) 단원을 참조하십시오.

SQL Server 가져오기 및 내보내기 마법사

SQL Server 가져오기 및 내보내기 마법사를 사용하여 Amazon RDS SQL DB 인스턴스에서 다른 데이터 스토어로 하나 이상의 테이블, 보기 또는 쿼리를 복사할 수 있습니다. 대상 데이터 스토어가 SQL Server가 아닌 경우 이 방법이 가장 좋습니다. 자세한 내용은 SQL Server 문서의 [SQL Server 가져오기 및 내보내기 마법사를](#) 참조하십시오.

SQL Server 가져오기 및 내보내기 마법사는 Express Edition을 제외한 모든 Microsoft SQL Server 버전에 포함되어 있는 그래픽 SQL Server 클라이언트인 Microsoft SQL Server Management Studio에 포함되어 있습니다. SQL Server Management Studio는 Windows 기반 애플리케이션으로만 사용할 수 있습니다. SQL Server Management Studio Express는 Microsoft에서 무료 다운로드로 사용할 수 있습니다. 다운로드에 관해서는 [Microsoft 웹 사이트](#)를 참조하십시오.

SQL Server 가져오기 및 내보내기 마법사를 사용하여 데이터를 내보내려면

1. SQL Server Management Studio에서 Amazon RDS SQL DB 인스턴스에 연결합니다. 이 작업을 수행하는 자세한 방법은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 503\)](#) 단원을 참조하십시오.
2. [개체 탐색기]에서 [데이터베이스]를 확장하고, 원본 데이터베이스에 대한 컨텍스트 메뉴를 마우스 오른쪽 버튼을 클릭하여 열고, [작업]을 선택한 다음, [데이터 내보내기]를 선택합니다. 그러면 마법사가 표시됩니다.
3. [데이터 소스 선택] 페이지에서 다음을 수행합니다.
 - a. Data source(데이터 원본)에 **SQL Server Native Client 11.0**을 선택합니다.
 - b. Server name(서버 이름) 상자에 Amazon RDS SQL DB 인스턴스의 앤드포인트가 표시되는지 확인합니다.
 - c. [SQL 서버 인증 사용]을 선택합니다. User name(사용자 이름) 및 Password(암호)에서 Amazon RDS SQL DB의 마스터 사용자 이름과 암호를 입력합니다.
 - d. [데이터베이스] 상자에 데이터를 내보내려는 원본 데이터베이스가 표시되는지 확인합니다.
 - e. [Next]를 선택합니다.
4. [대상 선택] 페이지에서 다음을 수행합니다.
 - a. Destination(대상)에 **SQL Server Native Client 11.0**을 선택합니다.

Note

.NET Framework 데이터 공급자, OLE DB 공급자, SQL Server Native Client 공급자, ADO.NET 공급자, Microsoft Office Excel, Microsoft Office Access 및 Flat File 소스를 포함한 다른 대상 데이터 소스를 사용할 수 있습니다. 이러한 데이터 소스 중 하나를 대상으로 선택할 경우, 4단계의 나머지 부분은 건너뜁니다. 제공할 연결 정보에 대한 세부 정보는 SQL Server 문서에서 [Choose a Destination](#)을 참조하십시오.

- b. [서버 이름]에 대상 SQL Server DB 인스턴스의 서버 이름을 입력합니다.
 - c. 알맞은 인증 유형을 선택합니다. 필요한 경우 사용자 이름과 암호를 입력합니다.
 - d. [데이터베이스]에서 대상 데이터베이스의 이름을 선택하거나, [새로 만들기]를 선택하여 내보낸 데이터를 포함하는 새 데이터베이스를 만듭니다.
- [새로 만들기]를 선택하는 경우, 제공할 데이터베이스 정보에 대한 자세한 내용은 SQL Server 문서의 [Create Database](#)를 참조하십시오.
- e. [Next]를 선택합니다.
 5. [테이블 복사 또는 쿼리] 페이지에서 [하나 이상의 테이블 또는 뷰에서 데이터 복사] 또는 [전송 데이터를 지정할 쿼리 작성]을 선택합니다. [Next]를 선택합니다.
 6. [전송 데이터를 지정할 쿼리 작성]을 선택한 경우 [원본 쿼리 지정] 페이지가 표시됩니다. SQL 쿼리를 입력하거나 붙여 넣은 다음, [구문 분석]을 선택하여 확인합니다. 쿼리 유효성 검사가 끝나면 [다음]을 선택합니다.
 7. [원본 테이블 및 뷰 선택] 페이지에서 다음을 수행합니다.
 - a. 내보내려는 테이블과 뷰를 선택하거나 사용자가 입력한 쿼리가 선택되어 있는지 확인합니다.
 - b. [매핑 편집]을 선택하고 데이터베이스 및 열 매핑 정보를 지정합니다. 자세한 내용은 SQL Server 문서의 [Column Mappings](#)를 참조하십시오.
 - c. (선택 사항) 내보낼 데이터의 미리 보기와 함께 테이블, 뷰 또는 쿼리를 선택한 후 [미리 보기]를 선택합니다.

- d. [Next]를 선택합니다.
8. [파키지 실행] 페이지에서 [즉시 실행]이 선택되어 있는지 확인합니다. [Next]를 선택합니다.
9. [마법사 완료] 페이지에서 데이터 내보내기 세부 정보가 예상한 대로인지 확인합니다. [마침]을 클릭합니다.
10. [실행이 성공했습니다.] 페이지에서 [닫기]를 선택합니다.

SQL Server 스크립트 생성 및 게시 마법사와 bcp 유ти리티

SQL Server 스크립트 생성 및 게시 마법사를 사용하여 데이터베이스 전체 또는 선택한 개체만을 위한 스크립트를 작성할 수 있습니다. 대상 SQL Server DB 인스턴스에서 이런 스크립트를 실행하여 스크립팅된 객체를 다시 만들 수 있습니다. 그런 다음, bcp 유ти리티를 사용하여 선택한 객체에 대한 데이터를 대상 DB 인스턴스로 대량으로 내보낼 수 있습니다. (테이블 이외의 객체를 포함한) 전체 데이터베이스를 이동하거나 두 SQL Server DB 인스턴스 사이에서 대량의 데이터를 이동하려는 경우에 이 마법사를 선택하는 것이 최선입니다. bcp 명령줄 구문에 대한 전체 설명은 Microsoft SQL Server 문서의 [bcp 유ти리티](#)를 참조하십시오.

SQL Server 스크립트 생성 및 게시 마법사는 Express Edition을 제외한 모든 Microsoft SQL Server 버전에 포함되어 있는 그래픽 SQL Server 클라이언트인 Microsoft SQL Server Management Studio에 포함되어 있습니다. SQL Server Management Studio는 Windows 기반 애플리케이션으로만 사용할 수 있습니다. SQL Server Management Studio Express는 Microsoft에서 [무료 다운로드](#)로 사용할 수 있습니다.

SQL Server 스크립트 생성 및 게시 마법사와 bcp 유ти리티를 사용하여 데이터를 내보내려면

1. SQL Server Management Studio에서 Amazon RDS SQL DB 인스턴스에 연결합니다. 이 작업을 수행하는 자세한 방법은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 503\)](#) 섹션을 참조하십시오.
2. [개체 탐색기]에서 [데이터베이스] 노드를 확장하고 스크립팅하려는 데이터베이스를 선택합니다.
3. 스크립트 파일을 만들려면 SQL Server 문서의 [스크립트 생성 및 게시 마법사](#)에 설명되어 있는 지침을 따르십시오.
4. SQL Server Management Studio에서 대상 SQL Server DB 인스턴스에 연결합니다.
5. [개체 탐색기]에서 대상 SQL Server DB 인스턴스를 선택한 상태에서 [파일] 메뉴에서 [열기]를 선택하고, [파일]을 선택한 후 스크립트 파일을 열니다.
6. 전체 데이터베이스를 스크립팅했으면 스크립트에서 CREATE DATABASE 문을 검토하여 자신이 원하는 위치에서 원하는 파라미터로 데이터베이스가 생성되고 있는지 확인합니다. 자세한 정보는 SQL Server 문서의 [CREATE DATABASE](#)를 참조하십시오.
7. 스크립트에서 데이터베이스 사용자를 만들 경우 대상 DB 인스턴스에 해당 사용자에 대한 서버 로그인이 존재하는지 확인합니다. 존재하지 않을 경우 해당 사용자에 대한 로그인을 만듭니다. 그렇지 않으면, 데이터베이스 사용자를 만들기 위해 스크립팅된 명령이 실패합니다. 자세한 정보는 SQL Server 문서의 [Create a Login](#)을 참조하십시오.
8. SQL 편집기 메뉴에서 [실행]을 선택하여 스크립트 파일을 실행하고 데이터베이스 개체를 만듭니다. 스크립트가 완료되면 모든 데이터베이스 개체가 예상한 대로 존재하는지 확인합니다.
9. bcp 유ти리티를 사용하여 Amazon RDS SQL DB 인스턴스에서 파일로 데이터를 내보냅니다. 명령 프롬프트를 열고 다음 명령을 입력합니다.

```
bcp database_name.schema_name.table_name out data_file -n -S aws_rds_sql_endpoint -U username -P password
```

앞에 나온 코드에는 다음 옵션이 포함됩니다.

- table_name은 대상 데이터베이스에서 다시 만들어 이제 데이터를 채우려는 테이블 중 하나의 이름입니다.
- data_file은 만들 데이터 파일의 전체 경로와 이름입니다.
- -n은 대량 복사에서 복사되는 데이터의 기본 데이터 형식을 사용할 것임을 지정합니다.
- -S는 데이터를 내보낼 원본 SQL Server DB 인스턴스를 지정합니다.

- -U는 SQL Server DB 인스턴스에 연결할 때 사용할 사용자 이름을 지정합니다.
- -P는 -U로 지정된 사용자의 암호를 지정합니다.

다음은 명령의 예시입니다.

```
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
```

내보낼 모든 테이블에 대한 데이터 파일을 확보할 때까지 이 단계를 반복하십시오.

10. SQL Server 문서의 [대량 데이터 가져오기 준비](#)에 설명되어 있는 지침에 따라 대량 데이터 가져오기를 위해 대상 DB 인스턴스를 준비하십시오.
11. SQL Server 문서의 [대량 가져오기 및 내보내기 작업 정보](#)에 설명되어 있는 성능 및 기타 문제를 고려한 후 사용할 대량 가져오기 방법을 결정하십시오.
12. 11단계에서 결정한 바에 따라, SQL Server 문서의 [bcp 유ти리티를 사용하여 대량 데이터 가져오기 및 내보내기 또는 BULK INSERT 또는 OPENROWSET\(BULK..\)을 사용하여 대량 데이터 가져오기](#)에 설명되어 있는 지침에 따라 bcp 유ти리티를 사용하여 사용자가 만든 데이터 파일에서 데이터를 대량으로 가져오십시오.

관련 주제

- [SQL Server 데이터베이스 가져오기 및 내보내기 \(p. 520\)](#)

Microsoft SQL Server의 다중 AZ 배포

다중 AZ 배포는 DB 인스턴스를 위해 향상된 가용성, 데이터 내구성 및 내결함성을 제공합니다. 계획된 데이터베이스 유지 관리 또는 예기치 않은 서비스 중단이 발생할 경우, Amazon RDS가 최신 보조 DB 인스턴스로 자동으로 장애 조치를 수행합니다. 이 기능을 통해 수동 개입 없이 데이터베이스 작업을 빠르게 재개할 수 있습니다. 기본 인스턴스 및 예비 인스턴스는 동일한 엔드포인트를 사용합니다. 이 엔드포인트의 물리적 네트워크 주소는 장애 조치 프로세스의 일환으로 보조 복제본으로 전환됩니다. 장애 조치가 발생하는 경우 애플리케이션을 다시 구성할 필요가 없습니다.

Amazon RDS는 SQL Server 데이터베이스 미러링(DBM) 또는 상시 작동 가용성 그룹(AG)을 사용하여 Microsoft SQL Server의 다중 AZ 배포를 지원합니다. Amazon RDS는 다중 AZ 배포의 상태를 모니터링하고 유지 관리합니다. 문제가 발생하면 RDS는 이상 있는 DB 인스턴스를 복구하고, 동기화를 재설정하며, 장애 조치를 시작합니다. 대기 및 기본 인스턴스가 완벽히 동기화되어 있는 경우에만 장애 조치가 이루어집니다. 사용자가 따로 관리할 것이 없습니다.

SQL Server 다중 AZ를 설정하면 RDS가 DBM 또는 AG를 사용하도록 인스턴스의 모든 데이터베이스를 자동으로 구성합니다. Amazon RDS는 사용자를 위해 기본, 감시 및 보조 DB 인스턴스를 처리합니다.

구성이 자동이므로 RDS는 사용자가 배포한 SQL Server 버전에 따라 DBM 또는 상시 작동 AG를 선택합니다. Amazon RDS는 다음 SQL Server 버전과 에디션에 DBM 또는 AG를 사용하는 다중 AZ를 지원합니다(예외 표시).

- SQL Server 2017: Standard 및 Enterprise Edition(상시 작동 AG는 Enterprise Edition 14.00.3049.1 이상에서 지원됩니다.)
- SQL Server 2016: Standard 및 Enterprise Edition(상시 작동 AG는 Enterprise Edition 13.00.5216.0 이상에서 지원됩니다.)
- SQL Server 2014: Standard 및 Enterprise Edition
- SQL Server 2012: Standard 및 Enterprise Edition
- SQL Server 2008 R2: Standard 및 Enterprise Edition

Amazon RDS는 모든 AWS 리전에서 SQL Server에 다중 AZ를 지원하지만 다음은 예외입니다.

- 미국 서부(캘리포니아 북부 지역): 여기에서는 DBM과 상시 작동 AG가 모두 지원되지 않음
- 아시아 태평양(시드니): [VPC의 DB 인스턴스](#)가 지원됨
- 아시아 태평양(도쿄): [VPC의 DB 인스턴스](#)가 지원됨
- 남아메리카(상파울루): m1 또는 m2를 제외한 모든 [DB 인스턴스 클래스](#)에서 지원됨
- EU(스톡홀름): 여기에서는 DBM과 상시 작동 AG가 모두 지원되지 않음

다중 AZ를 Microsoft SQL Server DB 인스턴스에 추가

AWS Management 콘솔을 사용하여 새로운 SQL Server DB 인스턴스를 만들 때, 데이터베이스 미러링 (DBM) 또는 상시 작동 AG를 사용하는 다중 AZ를 추가할 수 있습니다. 이렇게 하려면 DB 세부 정보 지정 페이지의 다중 AZ 배포 목록에서 예(미러링/상시 작동)를 선택합니다. 자세한 정보는 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 생성 \(p. 494\)](#) 단원을 참조하십시오.

AWS Management 콘솔을 사용하여 기존 SQL Server DB 인스턴스를 수정할 때 DB 인스턴스 수정 페이지의 다중 AZ 배포 목록에서 예(미러링/상시 작동)를 선택하여 DBM 또는 AG를 사용하는 다중 AZ를 추가할 수 있습니다. 자세한 정보는 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#) 단원을 참조하십시오.

Microsoft SQL Server 다중 AZ 배포의 참고 및 권장 사항

다음은 Microsoft SQL Server DB 인스턴스에서 다중 AZ 배포 작업 시 알아두어야 할 몇 가지 제약 조건입니다.

- 교차 리전 다중 AZ는 현재 지원되지 않습니다.
- 보조 인스턴스가 데이터베이스 읽기 작업을 허용하도록 구성할 수 없습니다.
- 상시 작동 가용성 그룹(AG)을 사용하는 다중 AZ는 인 메모리 최적화를 지원합니다.
- SQL Server 다중 AZ 배포에 있는 SQL Server DB 인스턴스의 데이터베이스는 이름을 변경할 수 없습니다. 그러한 인스턴스의 데이터베이스 이름을 바꿔야 하는 경우, 먼저 DB 인스턴스의 다중 AZ를 끈 후 데이터베이스 이름을 바꿉니다. 마지막으로 DB 인스턴스의 다중 AZ를 다시 캡니다.
- 전체 복구 모델을 사용하여 백업한 다중 AZ DB 인스턴스만 복원할 수 있습니다.

다음은 Microsoft SQL Server DB 인스턴스에서 다중 AZ 배포 작업 시 알아두어야 할 몇 가지 참고 사항입니다.

- Amazon RDS는 상시 작동 AG가용성 그룹 [리스너 앤드포인트](#)를 표시합니다. 이 앤드포인트는 콘솔에 표시되며, `DescribeDBInstances` API에 의해 앤드포인트 필드의 항목으로 반환됩니다.
- Amazon RDS는 [가용성 그룹 다중 서브넷 장애 조치](#)를 지원합니다.
- VPC에서 SQL Server DB 인스턴스와 함께 SQL Server 다중 AZ를 사용하려면 먼저 별개의 가용 영역 2개 이상에 서브넷이 있는 DB 서브넷 그룹을 만들어야 합니다. 그런 다음 SQL Server DB 인스턴스의 기본 복제본에 DB 서브넷 그룹을 할당합니다.
- DB 인스턴스를 다중 AZ 배포로 수정할 때 수정하는 동안 인스턴스의 상태는 `modifying`(수정 중)입니다. Amazon RDS는 대기를 생성하며 기본 DB 인스턴스의 백업을 생성합니다. 프로세스가 완료되면 기본 DB 인스턴스의 상태가 사용 가능이 됩니다.
- 다중 AZ 배포가 같은 노드 상의 모든 데이터베이스를 유지 관리합니다. 기본 호스트의 데이터베이스가 장애 조치하는 경우, 모든 SQL Server 데이터베이스가 하나의 원자 단위로 대기 호스트로 장애 조치합니다. Amazon RDS는 새로운 정상 호스트를 프로비저닝하여 비정상 호스트를 대체합니다.
- DBS 또는 AG를 사용하는 다중 AZ는 예비 복제본 하나를 지원합니다.

- 사용자, 로그인 및 권한은 보조에 자동으로 복제됩니다. 이런 것을 다시 만들 걱정을 할 필요가 없습니다. 사용자 정의 서버 역할(SQL Server 2012 기능)은 AG 인스턴스의 다른 AZ 인스턴스에서만 복제됩니다.
- SQL Server 에이전트 작업이 있는 경우, 보조에서 다시 생성합니다. 이러한 작업은 msdb 데이터베이스에 저장되므로 데이터베이스 미러링(DBM) 또는 상시 작동 가용성 그룹(AG)을 사용하여 이 데이터베이스를 복제할 수 없기 때문입니다. 원래의 기본 인스턴스에서 작업을 먼저 만든 다음, 장애 조치하고 새 기본 인스턴스에서 같은 작업을 만들어야 합니다.
- 동기식 데이터 복제로 인해 단일 가용 영역에서 표준 DB 인스턴스 배포와 비교했을 때 지연 시간이 증가할 수 있습니다.
- 장애 조치 시간은 복구 프로세스 완료에 걸리는 시간의 영향을 받습니다. 트랜잭션이 크면 장애 조치 시간이 늘어납니다.

다음은 Microsoft SQL Server DB 인스턴스에서 다중 AZ 배포 작업 시 알아두어야 할 몇 가지 권장 사항입니다.

- 프로덕션 또는 프리 프로덕션 환경에서 사용되는 데이터베이스의 경우, 고가용성, 빠르고 일관된 성능을 위한 프로비저닝된 IOPS 및 프로비저닝된 IOPS용으로 최적화된 인스턴스 클래스(m3.large 이상, m4.large 이상)에 다중 AZ 배포를 사용하는 것이 좋습니다.
- 보조 인스턴스에 대해 가용 영역(AZ)을 선택할 수 없으므로 애플리케이션 호스트를 배포할 경우 이 점을 고려하십시오. 데이터베이스를 다른 AZ로 장애 조치할 수 있으며, 애플리케이션 호스트가 데이터베이스와 다른 AZ에 있을 수도 있습니다. 이러한 이유로 지정된 AWS 리전의 모든 AZ에서 애플리케이션 호스트의 균형을 조정하는 것이 좋습니다.
- 최상의 성능을 위해, 큰 데이터 로드 작업 중에는 데이터베이스 미러링 또는 상시 작동 AG를 활성화하지 마십시오. 데이터 로드를 최대한 빠르게 수행하려면 데이터 로드를 완료한 후에 DB 인스턴스를 다중 AZ 배포로 변환합니다.
- SQL Server 데이터베이스에 액세스하는 애플리케이션에 연결 오류를 포착하는 예외 처리 기능이 있어야 합니다. 다음 코드 샘플에 통신 오류를 포착하는 try/catch 블록이 표시되어 있습니다.

```
for (int iRetryCount = 0; (iRetryCount < RetryMaxAttempts && keepInserting); iRetryCount++)
{
    using (SqlConnection connection = new SqlConnection(DatabaseConnectionString))
    {
        using (SqlCommand command = connection.CreateCommand())
        {
            command.CommandText = "INSERT INTO SOME_TABLE VALUES ('SomeValue');";

            try
            {
                connection.Open();

                while (keepInserting)
                {
                    command.ExecuteNonQuery();
                    intervalCount++;
                }
                connection.Close();
            }

            catch (Exception ex)
            {
                Logger(ex.Message);
            }
        }
    }

    if (iRetryCount < RetryMaxAttempts && keepInserting)
    {
        Thread.Sleep(RetryIntervalPeriodInSeconds * 1000);
    }
}
```

}

- 다중 AZ 인스턴스로 작업할 때는 `SET PARTNER OFF` 명령을 사용하지 마십시오. 예를 들어 다음을 수행하지 마십시오.

```
--Don't do this
ALTER DATABASE db1 SET PARTNER off
```

- 복구 모드를 `simple`로 설정하지 마십시오. 예를 들어 다음을 수행하지 마십시오.

```
--Don't do this
ALTER DATABASE db1 SET RECOVERY simple
```

- 이러한 설정은 대기 미러에 적용할 수 없으므로 다중 AZ DB 인스턴스에 새 로그인을 만들 때 `DEFAULT_DATABASE` 파라미터를 사용하지 마십시오. 예를 들어 다음을 수행하지 마십시오.

```
--Don't do this
CREATE LOGIN [test_dba] WITH PASSWORD=foo, DEFAULT_DATABASE=[db2]
```

또한 다음 작업을 수행하지 마십시오.

```
--Don't do this
ALTER LOGIN [test_dba] SET DEFAULT_DATABASE=[db3]
```

보조의 위치 확인

AWS Management 콘솔을 사용하여 보조 복제본의 위치를 확인할 수 있습니다. VPC에서 기본 DB 인스턴스를 설정할 경우 보조의 위치를 알아야 합니다.

Endpoint: sg-sqlsvr08r2.c6c8mntzhgv0.us-west-2.rds.amazonaws.com:8200 ([author](#))

Configuration Details		Security and Network	
DB Name:		Availability Zone:	us-west-2
Engine:	sqlserver-se(10.50.2789.0.v1)	VPC ID:	
Username:	sgawsuser	Subnet Group:	
Option Group(s):	default:sqlserver-se-10-50 (in-sync)	Subnets:	None
Parameter Group:	default.sqlserver-se-10.5 (in-sync)	Security Groups:	sg-db-se
Availability and Durability		Maintenance Details	
DB Instance Status:	available	Auto Minor Version Upgrades:	
Replication State:	-	Maintenance Window:	
Replication Error:	-	Backup Window:	
Multi AZ:	Yes		
Secondary Zone:	us-west-2c		
Automated Backups:	Enabled (1 Day)		
Latest Restore Time:	May 19, 2014 7:15:01 AM UTC-7		

AWS CLI 명령 `describe-db-instances` 또는 RDS API 작업 `DescribeDBInstances`를 사용하여 보조의 가용 영역을 확인할 수도 있습니다. 출력에는 대기 미러가 있는 보조 AZ가 표시됩니다.

데이터베이스 미러링에서 상시 작동 가능성 그룹으로 マイグレーション

Microsoft SQL Server Enterprise edition의 14.00.3049.1 버전에서는 상시 작동 가능성 그룹(AG)이 기본적으로 활성화됩니다.

데이터베이스 미러링(DBM)에서 AG로 마이그레이션하려면 먼저 버전을 확인하십시오. 버전이 13.00.5216.0 이전인 DB 인스턴스를 사용하는 경우, 13.00.5216.0으로 패치하도록 인스턴스를 수정합니다. Enterprise Edition 14.00.3049.1 이전 버전의 DB 인스턴스를 사용하는 경우, 14.00.3049.1 이상으로 패치하도록 인스턴스를 수정합니다.

AG를 사용하도록 미러링된 DB 인스턴스를 업그레이드하려면 먼저 업그레이드를 실행하고 다중 AZ를 제거하도록 인스턴스를 수정한 다음 다시 수정하여 다중 AZ를 추가하십시오. 그러면 인스턴스가 상시 작동 AG를 사용하도록 변환됩니다.

Microsoft SQL Server DB 인스턴스와 함께 SSL 사용

SSL(Secure Sockets Layer)을 사용하여 클라이언트 애플리케이션과 Microsoft SQL Server를 실행하는 Amazon RDS DB 인스턴스 사이의 연결을 암호화할 수 있습니다. 지원되는 모든 SQL Server 버전의 모든 AWS 리전에서 SSL 지원 기능을 사용할 수 있습니다.

SQL Server DB 인스턴스를 생성할 때 Amazon RDS는 인스턴스의 SSL 인증서를 만듭니다. SSL 인증서에는 스푸핑 공격으로부터 보호해주는 SSL 인증서를 위한 일반 이름(CN)으로 DB 인스턴스 엔드포인트가 포함되어 있습니다.

SSL을 사용하여 SQL Server DB 인스턴스에 연결하는 방법은 두 가지입니다.

- 모든 연결에 대해 SSL 지정 – 이것은 클라이언트에 투명하게 발생하며, 클라이언트는 SSL 사용을 위해 작업을 수행할 필요가 없습니다.
- 특정 연결 암호화 – 이것은 특정 클라이언트 컴퓨터로부터 SSL 연결을 설정하며, 연결 암호화를 위해 클라이언트에서 작업을 수행해야 합니다.

SQL Server의 TLS(전송 계층 보안)에 대한 자세한 내용은 [Microsoft SQL Server에 대한 TLS 1.2 지원](#)을 참조하십시오.

DB 인스턴스 연결이 SSL을 사용하도록 지정

DB 인스턴스에 대한 모든 연결에서 SSL을 사용하도록 지정할 수 있습니다. 연결이 SSL을 사용하도록 지정하면 클라이언트에 투명하게 발생하며, 클라이언트는 SSL 사용을 위해 작업을 수행할 필요가 없습니다.

SSL을 지정하려면 `rds.force_ssl` 파라미터를 사용하십시오. 기본적으로 `rds.force_ssl` 파라미터는 `false`로 설정됩니다. 연결이 SSL을 사용하도록 지정하려면 `rds.force_ssl` 파라미터를 `true`로 설정하십시오. `rds.force_ssl` 파라미터는 정적이기 때문에 값을 변경하면 DB 인스턴스를 재부팅해야만 변경사항이 적용됩니다.

모든 DB 인스턴스 연결이 SSL을 사용하도록 지정

- DB 인스턴스에 첨부할 파라미터 그룹을 결정합니다.
 - AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
 - Amazon RDS 콘솔의 오른쪽 상단 모서리에서 DB 인스턴스의 AWS 리전을 선택합니다.

- c. 탐색 창에서 데이터베이스를 선택한 후, DB 인스턴스의 이름을 선택하여 세부 정보를 표시합니다.
 - d. 구성 탭을 선택합니다. 섹션에서 파라미터 그룹을 찾습니다.
2. 필요하다면 새 파라미터 그룹을 만듭니다. DB 인스턴스가 기본 파라미터 그룹을 사용한다면, 새 파라미터 그룹을 생성해야 합니다. DB 인스턴스가 기본이 아닌 파라미터 그룹을 사용한다면, 기존 파라미터 그룹을 편집하거나 새 파라미터 그룹을 생성할 수 있습니다. 기존 파라미터 그룹을 편집하면, 해당 파라미터 그룹을 사용하면 모든 DB 인스턴스가 영향받게 됩니다.
- 새 파라미터 그룹을 만들려면, [DB 파라미터 그룹 생성 \(p. 163\)](#)의 지침을 따르십시오.
3. 신규 또는 기존 파라미터 그룹을 편집해 `rds.force_ssl` 파라미터를 `true`로 설정하십시오. 파라미터 그룹을 편집하려면, [DB 파라미터 그룹의 파라미터 수정 \(p. 164\)](#)의 지침을 따르십시오.
 4. 새 파라미터 그룹을 생성했다면, DB 인스턴스를 수정해 새 파라미터 그룹을 첨부하십시오. DB 인스턴스의 DB Parameter Group 설정을 수정합니다. 자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#) 단원을 참조하십시오.
 5. DB 인스턴스를 재부팅합니다. 자세한 내용은 [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#) 단원을 참조하십시오.

특정 연결 암호화

모든 DB 인스턴스 연결이 SSL을 사용하도록 지정하거나, 특정 클라이언트 컴퓨터 연결만 암호화할 수 있습니다. 특정 클라이언트에서 SSL을 사용하려면, 클라이언트 컴퓨터의 인증서를 획득하고 클라이언트 컴퓨터에서 인증서를 가져온 다음 클라이언트 연결을 암호화해야 합니다.

Note

2014년 8월 5일 이후에 만든 모든 SQL Server 인스턴스는 SSL 인증서의 일반 이름(CN) 필드에 DB 인스턴스 엔드포인트를 사용합니다. 2014년 8월 5일 이전에는 VPC 기반 SQL Server 인스턴스에 대해 SSL 인증서 확인 기능을 사용할 수 없었습니다. 2014년 8월 5일 이전에 생성된 VPC 기반 SQL Server DB 인스턴스가 있는 상태에서 SSL 인증서 확인을 사용하고 DB 인스턴스용 SSL 인증서에 대한 CN으로 인스턴스 엔드포인트를 반드시 포함시키려면, DB 인스턴스의 이름을 변경합니다. DB 인스턴스의 이름을 변경하면 새 인증서가 배포되고 인스턴스가 재부팅되어 새 인증서를 활성화합니다.

클라이언트 컴퓨터의 인증서 획득

클라이언트 컴퓨터와 Microsoft SQL Server를 실행하는 Amazon RDS DB 인스턴스 사이의 연결을 암호화하려면, 클라이언트 컴퓨터에 인증서가 있어야 합니다.

인증서를 얻으려면, 클라이언트 컴퓨터에 인증서를 다운로드하십시오. <https://s3.amazonaws.com/rds-downloads/rds-ca-2015-root.pem>에서 모든 리전에 적용되는 루트 인증서를 다운로드할 수 있습니다. <https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem>에서는 구형 및 신형 루트 인증서를 모두 포함하는 인증서 번들을 다운로드할 수 있습니다. 리전별 중간 인증서 및 자세한 내용은 [SSL을 사용하여 DB 인스턴스에 대한 연결 암호화 \(p. 388\)](#) 단원을 참조하십시오.

적절한 인증서를 다운로드했다면, 아래 항목에 있는 절차를 따라 Microsoft Windows 운영 체제로 인증서를 가져오십시오.

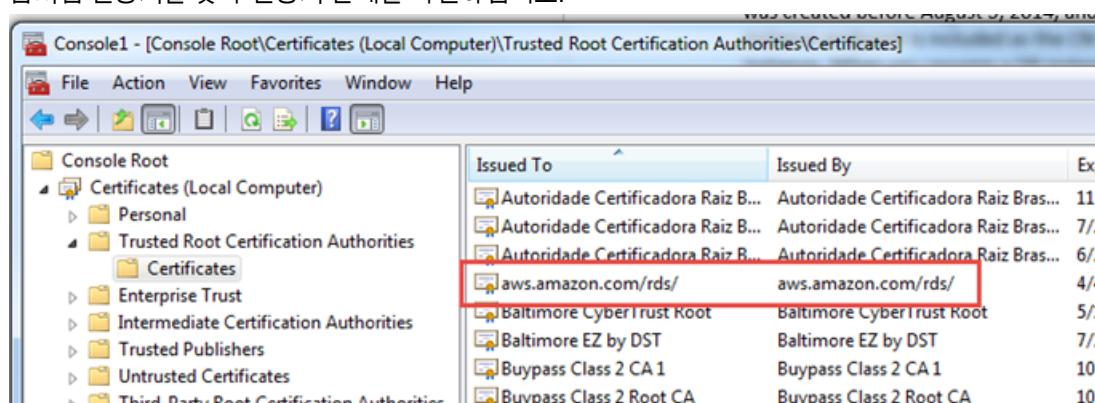
클라이언트 컴퓨터로 인증서 가져오기

다음 절차를 이용해 클라이언트 컴퓨터의 Microsoft Windows 운영 체제로 인증서를 가져올 수 있습니다.

Windows 운영 체제로 인증서를 가져오는 방법:

1. 시작 메뉴에서 검색 상자에 **Run**을 입력하고 Enter 키를 누릅니다.
2. 열기 상자에 **MMC**를 입력하고 확인을 선택합니다.
3. MMC 콘솔의 파일 메뉴에서 스냅인 추가/제거를 선택합니다.
4. 스냅인 추가/제거 대화 상자의 사용 가능한 스냅인에서 **Certificates**를 선택하고 추가를 선택합니다.

5. MMC 콘솔의 파일 메뉴에서 스냅인 추가/제거를 선택합니다.
6. 인증서 스냅인 대화 상자에서 컴퓨터 계정을 선택한 후 다음을 선택합니다.
7. 컴퓨터 선택 대화 상자에서 마침을 선택합니다.
8. 스냅인 추가/제거 대화 상자에서 확인을 선택합니다.
9. MMC 콘솔에서 인증서를 확장하고, 문맥 메뉴(우클릭)를 열어 신뢰할 수 있는 루트 인증 기관을 선택하고, 모든 작업을 선택한 후 가져오기를 선택합니다.
10. 인증서 가져오기 마법사의 첫 페이지에서 다음을 선택합니다.
11. 인증서 가져오기 마법사의 두 번째 페이지에서 찾아보기를 선택합니다. .pem은 표준 인증서 확장명이 아니기 때문에 브라우저 창에서 파일 유형을 모든 파일 (*.*)로 변경합니다. 이전에 다운로드한 .pem 파일을 찾습니다.
12. 열기를 선택해 인증서 파일을 선택하고 다음을 선택합니다.
13. 인증서 가져오기 마법사의 세 번째 페이지에서 다음을 선택합니다.
14. 인증서 가져오기 마법사의 네 번째 페이지에서 종료를 선택합니다. 가져오기에 성공했음을 표시하는 대화 상자가 나타날 것입니다.
15. MMC 콘솔에서 인증서와 신뢰할 수 있는 루트 인증 기관을 차례로 확장한 다음 인증서를 선택합니다. 다음처럼 인증서를 찾아 인증서 존재를 확인하십시오.



16. 컴퓨터를 다시 시작합니다.

Microsoft SQL Server 기반 Amazon RDS DB 인스턴스 연결 암호화

클라이언트 컴퓨터로 인증서를 가져오면, 클라이언트 컴퓨터와 Microsoft SQL Server를 실행하는 Amazon RDS DB 인스턴스 사이의 연결을 암호화할 수 있습니다.

SQL Server Management Studio의 경우, 다음 절차를 사용합니다. SQL Server Management Studio에 대한 자세한 내용은 [SQL Server Management Studio 사용](#)을 참조하십시오.

SQL Server Management Studio 연결 암호화

1. SQL Server Management Studio를 시작합니다.
2. [서버에 연결]에 서버 정보, 로그인 사용자 이름 및 암호를 입력합니다.
3. Options를 선택합니다.
4. [연결 암호화]를 선택합니다.
5. [Connect]를 선택합니다.
6. 다음 쿼리를 실행하여 연결이 암호화되는지 확인합니다. 쿼리가 encrypt_option에 대해 true를 반환하는지 확인합니다.

```
select ENCRYPT_OPTION from SYS.DM_EXEC_CONNECTIONS where SESSION_ID = @@SPID
```

다른 SQL 클라이언트에는 다음 절차를 적용하십시오.

다른 SQL 클라이언트 연결 암호화

- 연결 문자열에 `encrypt=true`를 추가합니다. 이 문자열은 GUI 도구의 연결 페이지에서 옵션 또는 속성으로 사용할 수 있습니다.

Note

JDBC를 사용하여 연결하는 클라이언트에 대해 SSL 암호화를 활성화하려면 Java CA 인증서 (cacert) 저장소에 Amazon RDS SQL 인증서를 추가해야 할 수도 있습니다. [keytool](#) 유틸리티를 사용하여 이 작업을 수행할 수 있습니다.

- 다음 쿼리를 실행하여 연결이 암호화되는지 확인합니다. 쿼리가 `encrypt_option`에 대해 `true`를 반환하는지 확인합니다.

```
select ENCRYPT_OPTION from SYS.DM_EXEC_CONNECTIONS where SESSION_ID = @@SPID
```

Microsoft SQL Server 데이터베이스 엔진의 옵션

이 단원에서는 Microsoft SQL Server DB 엔진을 실행하는 Amazon RDS 인스턴스에 사용할 수 있는 옵션 또는 추가 기능에 대한 설명을 볼 수 있습니다. 이러한 옵션을 활성화하려면 먼저 옵션 그룹에 추가한 다음 옵션 그룹을 DB 인스턴스에 연결해야 합니다. 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오.

Amazon RDS는 Microsoft SQL Server DB 인스턴스에 대해 다음 옵션을 지원합니다.

옵션	옵션 ID	엔진 버전
기본 백업 및 복원 (p. 545)	<code>SQLSERVER_BACKUP_RESTORE</code>	SQL Server Enterprise Edition SQL Server Standard Edition SQL Server Web Edition SQL Server Express Edition
투명한 데이터 암호화 (p. 547)	<code>TRANSPARENT_DATA_ENCRYPTION</code>	SQL Server Enterprise Edition
SQL Server Audit (p. 549)	<code>SQLSERVER_AUDIT</code>	SQL Server 2012부터는 RDS에서 모든 버전의 SQL Server가 서버 수준 감사를 지원하고 엔터프라이즈 에디션은 데이터베이스 수준 감사도 지원합니다. SQL Server SQL Server 2016(13.x) SP1부터는 모든 버전이 서버 및 데이터베이스 수준 감사를 모두 지원합니다. 자세한 내용은 SQL Server Audit(데이터베이스 수준 감사) 를 참조하십시오.

옵션	옵션 ID	엔진 버전
		스 엔진 를 참조하십시오.

Microsoft SQL Server에서 기본 백업 및 복원 지원

Amazon RDS는 전체 백업 파일(.bak 파일)을 사용하여 Microsoft SQL Server 데이터베이스에 기본 백업 및 복원을 할 수 있도록 지원합니다. 이 기능을 사용하면 온프레미스 데이터베이스의 전체 백업을 생성하고 Amazon Simple Storage Service(Amazon S3)에 백업 파일을 저장할 수 있습니다. 그런 다음 SQL Server를 실행하는 기존 Amazon RDS DB 인스턴스로 복원할 수 있습니다. RDS SQL Server 데이터베이스를 백업하고 Amazon S3에 저장하고 다른 위치에 복원할 수도 있습니다. 온프레미스 서버 또는 SQL Server를 실행 중인 다른 Amazon RDS DB 인스턴스로 복원할 수 있습니다. 자세한 내용은 [SQL Server 데이터베이스 가져오기 및 내보내기 \(p. 520\)](#) 단원을 참조하십시오.

기본 백업 및 복원 옵션 설정

Amazon RDS는 기본 백업 및 복원 옵션에 대해 다음과 같은 설정을 지원합니다.

옵션 설정	유효한 값	설명
IAM_ROLE_ARN	arn:aws:iam::account-id:role/ <i>role-name</i> 형식의 유효한 ARN(Amazon 리소스 이름).	백업 파일이 들어 있는 Amazon S3 버킷에 액세스하기 위한 AWS Identity and Access Management(IAM) 역할의 ARN. 자세한 내용은 AWS Identity and Access Management(IAM) 단원을 참조하십시오.

기본 백업 및 복원 옵션 추가

기본 백업 및 복원 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

- 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
- 옵션을 옵션 그룹에 추가합니다.
- 옵션 그룹을 DB 인스턴스에 연동시킵니다.

기본 백업 및 복원 옵션을 추가한 후 DB 인스턴스를 재시작할 필요가 없습니다. 옵션 그룹이 활성화되는 즉시 백업 및 복원을 시작할 수 있습니다.

기본 백업 및 복원 옵션을 추가하려면

- 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않다면 사용자 지정 DB 옵션 그룹을 생성합니다. 자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.
- 옵션 그룹에 [SQLSERVER_BACKUP_RESTORE] 옵션을 추가하고 옵션 설정을 구성합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#)을(를) 참조하십시오.

다음 중 하나를 선택합니다.

- 기존 역할과 Amazon S3 설정을 사용하려면:

IAM 역할에서 기존 IAM 역할을 선택합니다. 기존 IAM 역할을 다시 사용하면 RDS는 이전에 이 역할에 구성한 Amazon S3 설정을 사용합니다.

2. 새 역할 및 Amazon S3 설정을 구성하려면:

- IAM 역할에서 새 역할 생성을 선택합니다.
- Select S3 Bucket(S3 버킷 선택)에서 새로 만들거나, 이미 가지고 있는 것을 사용할 수 있습니다. 새 버킷을 만들려면 S3 버킷 새로 만들기를 선택합니다. 기존 버킷을 사용하려면 목록에서 선택하십시오.
- S3 folder path prefix (optional)(S3 폴더 경로 접두사(선택 사항)에서 Amazon S3 버킷에 저장된 파일에 사용할 접두사를 지정하십시오. 파일 경로를 포함할 수 있지만 필수는 아닙니다. 항목을 입력할 경우, RDS는 모든 백업 파일에 접두사를 붙이고, 복원 중에 접두사를 사용하여 관련 파일을 식별하고, 관련 없는 파일을 무시합니다. 버킷을 다른 용도로 사용하는 경우, 접두사를 사용하여 RDS가 특정 폴더와 그 하위 폴더에만 액세스할 수 있도록 기본 백업과 복원을 제한할 수 있습니다. 접두사를 사용하면 파일 이름 앞에 접두사를 붙여 폴더에 있는 파일의 하위 집합에 집중할 수 있습니다.)

접두사를 공백으로 두면 RDS가 백업 파일에 접두사를 붙이지 않고 복원 중에 접두사를 사용하여 파일을 식별하지 않습니다. 그러나 다중 파일 복원 중에 RDS는 이 버킷의 모든 폴더에 있는 모든 파일을 복원하려고 시도합니다. 예를 들어 여러 파일에서 데이터베이스를 복원하는 데 사용하되, 백업과 관련 없는 다른 파일을 버킷에 포함하려면 경로 접두사를 지정해야 합니다. 그러면 RDS가 모든 관련 파일을 복원하고 관련 없는 파일을 무시합니다.

- [Enable Encryption]에 대해 [Yes]를 선택하여 백업 파일을 암호화합니다. [Yes]를 선택하면 [Master Key]에도 암호화 키를 선택해야 합니다. 암호화 키에 대한 자세한 내용은 AWS Key Management Service(AWS KMS) 설명서의 [시작하기](#)를 참조하십시오.

3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:

- 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 생성 \(p. 494\)](#) 단원을 참조하십시오.
- 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#) 단원을 참조하십시오.

기본 백업 및 복원 옵션 설정 수정

기본 백업 및 복원 옵션을 활성화한 후 옵션의 설정을 수정할 수 있습니다. 옵션 설정을 변경하는 방법에 대한 자세한 내용은 [옵션 설정 변경 \(p. 156\)](#) 단원을 참조하십시오. 각 설정에 대한 자세한 내용은 [기본 백업 및 복원 옵션 설정 \(p. 545\)](#) 단원을 참조하십시오.

기본 백업 및 복원 옵션 제거

DB 인스턴스에서 옵션을 제거하여 기본 백업 및 복원 기능을 끌 수 있습니다. 기본 백업 및 복원 옵션을 제거한 후 DB 인스턴스를 재시작할 필요가 없습니다.

DB 인스턴스에서 기본 백업 및 복원 옵션을 제거하려면 다음 중 하나를 실시합니다.

- 기본 백업 및 복원 옵션이 속하는 옵션 그룹에서 이 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- DB 인스턴스를 수정하고, 기본 백업 및 복원 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 이 변경은 단일 DB 인스턴스에 영향을 미칩니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. 자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#) 단원을 참조하십시오.

Microsoft SQL Server Transparent Data Encryption(TDE) 지원

Amazon RDS는 Microsoft SQL Server를 실행하는 DB 인스턴스에 저장된 데이터를 암호화하기 위하여 Transparent Data Encryption(TDE) 이용을 지원합니다. TDE는 스토리지에 데이터를 쓰기 전에 자동으로 데이터를 암호화한 뒤에 데이터를 스토리지에서 읽을 때 다시 자동으로 복호화합니다.

Amazon RDS는 다음 SQL Server 버전에 TDE를 지원합니다.

- SQL Server 2017 Enterprise Edition
- SQL Server 2016 Enterprise Edition
- SQL Server 2014 Enterprise Edition
- SQL Server 2012 Enterprise Edition
- SQL Server 2008 R2 Enterprise Edition

RDS SQL Server DB 인스턴스에서 TDE를 활성화하려면 해당 DB 인스턴스와 연동되어 있는 RDS 옵션 그룹에서 TDE 옵션을 지정해야 합니다.

SQL Server의 TDE는 2계층 키 아키텍처를 사용하여 암호화 키 관리 기능을 지원합니다. 데이터베이스 마스터 키에서 생성된 인증서는 데이터 암호화 키를 보호하는 데 사용됩니다. 사용자 데이터베이스의 데이터를 실제로 암호화 및 복호화하는 것이 바로 이 데이터베이스 암호화 키입니다. Amazon RDS는 데이터베이스 마스터 키와 TDE 인증서를 백업 및 관리합니다. 또한 몇 가지 보안 기준을 준수하기 위해 Amazon RDS는 주기적으로 마스터 키를 자동으로 교체하고 있습니다.

투명한 데이터 암호화는 중요한 데이터를 암호화해야 하는 경우에 사용됩니다. 예를 들어 타사에 데이터 파일과 백업을 제공하거나 보안 관련 규정 준수 문제를 해결하고 싶을 수 있습니다. Model이나 Master 데이터베이스 같은 SQL Server의 시스템 데이터베이스를 암호화할 수 없습니다.

TDE에 대한 자세한 설명은 본 문서의 범위에서 벗어나지만 보안과 관련하여 각 암호화 알고리즘과 키의 장단점은 잘 알고 있어야 합니다. SQL Server의 TDE에 대한 자세한 내용은 Microsoft 웹사이트에서 [Transparent Data Encryption\(TDE\)](#)을 참조하십시오.

혹시 DB 인스턴스가 TDE 옵션이 추가된 옵션 그룹과 이미 연동되어 있지 않은지 먼저 확인해야 합니다. DB 인스턴스와 연동되어 있는 옵션 그룹은 RDS 콘솔, [describe-db-instance](#) AWS CLI 명령, 또는 API 작업 [DescribeDBInstances](#)를 사용하여 확인할 수 있습니다.

SQL Server DB 인스턴스에서 TDE를 활성화하는 방법은 다음과 같습니다.

1. DB 인스턴스가 TDE가 활성화된 옵션 그룹과 연결되어 있지 않으면 두 가지 옵션이 있습니다. 옵션 그룹을 생성하고 TDE 옵션을 추가하거나, 연결된 옵션 그룹을 수정하여 추가할 수 있습니다. 옵션 그룹의 생성 및 변경에 대한 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오. 옵션 그룹에 옵션을 추가하는 방법에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#) 단원을 참조하십시오.
2. DB 인스턴스를 TDE 옵션이 있는 옵션 그룹과 연결합니다. DB 인스턴스와 옵션 그룹의 연동에 대한 자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#) 단원을 참조하십시오.

TDE 옵션을 옵션 그룹에 추가하면 Amazon RDS가 암호화 프로세스에 사용할 인증서를 생성합니다. 그러면 이 인증서를 사용하여 DB 인스턴스의 데이터베이스에 저장된 데이터를 암호화하는 SQL 문을 실행할 수 있습니다. 다음은 RDSTDECertificateName이라고 하는 RDS 생성 인증서를 사용하여 customerDatabase라는 데이터베이스를 암호화하는 예제입니다.

```
----- 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
```

TDE를 사용하여 SQL Server 데이터베이스를 암호화하는 데 걸리는 시간은 몇 가지 요인에 따라 다릅니다. 여기에는 DB 인스턴스의 크기, 인스턴스에 대한 PIOPS 활성화 여부, 데이터 양, 기타 요소가 포함됩니다.

TDE는 영구 옵션이기 때문에 DB 인스턴스 및 백업 파일과 옵션 그룹의 연동을 끊지 않는 한 제거하지 못합니다. TDE 옵션을 옵션 그룹에 추가하면 이 옵션 그룹은 TDE를 사용하는 DB 인스턴스에 한해 연동이 가능합니다. 옵션 그룹의 영구 옵션에 대한 자세한 내용은 [옵션 그룹 개요 \(p. 148\)](#) 단원을 참조하십시오.

TDE 옵션은 영구 옵션이기 때문에 옵션 그룹과 연결된 DB 인스턴스 사이에 충돌이 일어날 수 있습니다. 이러한 충돌이 일어나는 상황은 다음과 같습니다.

- 현재 TDE 옵션이 있는 옵션 그룹을 TDE 옵션이 없는 옵션 그룹으로 변경하는 경우
- DB 스냅샷에서 TDE 옵션을 포함하는 옵션 그룹이 연결되지 않은 새 DB 인스턴스로 복원하는 경우 이 시나리오에 대한 자세한 내용은 [옵션 그룹 고려 사항 \(p. 209\)](#)를 참조하십시오.

DB 인스턴스에서 TDE를 비활성화하려면 먼저 객체를 복호화하거나 삭제하여 DB 인스턴스에 암호화된 객체가 하나도 없어야 합니다. DB 인스턴스에 암호화된 객체가 남아 있을 경우에는 DB 인스턴스에서 TDE를 비활성화할 수 없습니다. 콘솔을 사용하여 옵션 그룹에서 TDE 옵션을 제거하면 콘솔에 처리 중으로 표시됩니다. 또한 옵션 그룹이 암호화된 DB 인스턴스 또는 DB 스냅샷과 연결되어 있으면 오류 이벤트가 생성됩니다.

다음은 customerDatabase라고 하는 데이터베이스에서 TDE 암호화를 제거하는 예제입니다.

```
----- 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
```

모든 객체가 복호화되면 DB 인스턴스의 설정을 변경하여 TDE 옵션이 없는 옵션 그룹과 연동하거나, TDE 옵션을 옵션 그룹에서 제거할 수 있습니다.

Microsoft SQL Server 성능 고려 사항

SQL Server DB 인스턴스는 TDE 사용 시 성능에 영향을 받을 수 있습니다.

DB 인스턴스의 데이터베이스 중 암호화된 데이터베이스가 하나 이상만 있어도 마찬가지로 암호화되지 않은 데이터베이스의 성능이 떨어질 수 있습니다. 따라서 암호화되지 않은 데이터베이스와 암호화된 데이터베이스는 별도의 DB 인스턴스에서 관리하는 것이 좋습니다.

암호화의 특성상 데이터베이스 크기와 트랜잭션 로그의 크기가 암호화되지 않은 데이터베이스보다 클 수밖에 없습니다. 그 결과 백업을 위해 할당한 여유 공간까지 침범하기도 합니다. TDE로 인한 성능 저하는 어쩔 수 없습니다. 높은 성능과 TDE가 동시에 필요하다면 성능에 미치는 영향을 평가하여 요건에 맞춰야 합니다. 프로비저닝된 IOPS와 M3.Large DB 인스턴스 클래스만 사용해도 성능에 미치는 영향을 줄일 수 있습니다.

SQL Server Audit

Amazon RDS에서는 기본 제공 SQL Server Audit 메커니즘을 사용하여 Microsoft SQL Server 데이터베이스를 감사할 수 있습니다. 온프레미스 데이터베이스 서버용으로 생성하는 경우와 같은 방식으로 감사 및 감사 사양을 생성할 수 있습니다.

RDS는 사용자가 제공한 IAM 역할을 사용하여 완료된 감사 로그를 S3 버킷에 업로드합니다. 보존을 활성화하면 RDS는 구성된 기간 동안 DB 인스턴스에 대한 감사 로그를 유지합니다.

자세한 내용은 Microsoft SQL Server 설명서의 [SQL Server Audit\(데이터베이스 엔진\)](#)를 참조하십시오.

주제

- [SQL Server Audit 지원 \(p. 549\)](#)
- [DB 인스턴스 옵션에 SQL Server Audit 추가 \(p. 550\)](#)
- [SQL Server Audit의 사용 \(p. 551\)](#)
- [감사 로그 보기 \(p. 552\)](#)
- [다중 AZ 인스턴스에서 SQL Server Audit 사용 \(p. 552\)](#)
- [S3 버킷 구성 \(p. 553\)](#)
- [수동으로 SQL Server Audit에 대한 IAM 역할 생성 \(p. 553\)](#)

SQL Server Audit 지원

SQL Server 2012부터는 Amazon RDS에서 모든 버전의 SQL Server가 서버 수준 감사를 지원하고 엔터프라이즈 에디션은 데이터베이스 수준 감사도 지원합니다. SQL Server SQL Server 2016(13.x) SP1부터는 모든 버전이 서버 및 데이터베이스 수준 감사를 모두 지원합니다. 자세한 내용은 Microsoft SQL Server 설명서의 [SQL Server Audit\(데이터베이스 엔진\)](#)를 참조하십시오.

RDS는 SQL Server Audit에 대한 다음 옵션 설정 구성을 지원합니다.

옵션 설정	유효한 값	설명
IAM_ROLE_ARN	arn:aws:iam::account-id:role/role-name 형식의 유효한 ARN(Amazon 리소스 이름).	감사 로그를 저장할 S3 버킷에 액세스 권한을 부여하는 IAM 역할의 ARN입니다. 자세한 내용은 AWS Identity and Access Management(IAM) 을 참조하십시오.
S3_BUCKET_ARN	arn:aws:s3:::bucket-name 또는 arn:aws:s3:::bucket-name/key-prefix 형식의 유효한 ARN	감사 로그를 저장할 S3 버킷의 ARN입니다.
ENABLE_COMPRESSION	true 또는 false	감사 로그 압축을 제어합니다. 기본적으로 압축은 활성화됩니다 (true로 설정).
RETENTION_TIME	0 840	SQL Server Audit 레코드가 RDS 인스턴스에 유지되는 보존 기간 (시간 단위)입니다. 기본적으로 보존은 비활성화됩니다.

DB 인스턴스 옵션에 SQL Server Audit 추가

SQL Server Audit를 활성화하려면 DB 인스턴스에 옵션을 활성화하고 SQL Server 내에서 이 기능을 활성화하는 두 단계가 필요합니다. SQL Server Audit 옵션을 DB 인스턴스에 추가하는 프로세스는 다음과 같습니다.

- 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
- 필요한 모든 옵션을 추가하고 구성하십시오.
- 옵션 그룹을 DB 인스턴스에 연동시킵니다.

SQL Server Audit 옵션을 추가한 후 DB 인스턴스를 다시 시작할 필요가 없습니다. 옵션 그룹이 활성화되면 S3 버킷에 감사를 생성하고 감사 로그를 저장할 수 있습니다.

DB 인스턴스의 옵션 그룹에 SQL Server Audit을 추가하고 구성하려면

- 다음 중 하나를 선택합니다.
 - 기존 옵션 그룹을 사용합니다.
 - 사용자 지정 DB 옵션 그룹을 생성하고 해당 옵션 그룹을 사용합니다. 자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.
- 옵션 그룹에 SQLSERVER_AUDIT 옵션을 추가하고 옵션 설정을 구성합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#) 단원을 참조하십시오.
 - IAM 역할에 필요한 정책의 IAM 역할이 이미 있는 경우 해당 역할을 선택할 수 있습니다. 새 IAM 역할을 만들려면 새 역할 생성을 선택하십시오. 필요한 정책에 대한 자세한 내용은 [수동으로 SQL Server Audit에 대한 IAM 역할 생성 \(p. 553\)](#) 단원을 참조하십시오.
 - Select S3 destination(S3 대상 선택)의 경우, 사용할 S3 버킷이 이미 있는 경우 선택합니다. S3 버킷을 생성하려면 S3 버킷 새로 만들기를 선택하십시오.
 - Enable Compression(압축 활성화)의 경우, 감사 파일을 압축하려면 이 옵션을 선택한 상태로 듭니다. 압축은 기본적으로 활성화되어 있습니다. 압축을 비활성화하려면 Enable Compression(압축 활성화) 선택을 취소하십시오.

- Audit log retention(감사 로그 보존)의 경우, DB 인스턴스에 감사 레코드를 보존하려면 이 옵션을 선택합니다. 보존 시간을 시간 단위로 지정하십시오. 최대 보존 기간은 35일입니다.
3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다. 다음 중 하나를 선택합니다.
- 새 DB 인스턴스를 생성하는 경우, 인스턴스를 시작할 때 옵션 그룹을 적용하십시오.
 - 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정한 후 새 옵션 그룹을 연결하여 옵션 그룹을 적용하십시오. 자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#) 단원을 참조하십시오.

SQL Server Audit 옵션 설정

SQL Server Audit 옵션을 활성화하면 설정을 수정할 수 있습니다. 옵션 설정을 변경하는 방법에 대한 자세한 내용은 [옵션 설정 변경 \(p. 156\)](#) 단원을 참조하십시오.

DB 인스턴스 옵션에서 SQL Server Audit 제거

감사를 비활성화한 다음 옵션을 삭제하여 SQL Server Audit 기능을 해제할 수 있습니다.

감사를 제거하려면

1. SQL Server 내부의 모든 감사 설정을 비활성화합니다. 감사가 실행되는 위치를 확인하려면 SQL Server 보안 카탈로그 보기를 퀼리하십시오. 자세한 내용은 [Microsoft SQL Server 설명서의 보안 카탈로그 보기](#)를 참조하십시오.
2. DB 인스턴스에서 SQL Server Audit 옵션을 삭제하십시오. 다음 중 하나를 선택합니다.
 - DB 인스턴스가 사용하는 옵션 그룹에서 SQL Server Audit 옵션을 삭제하십시오. 이 변경은 동일한 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
 - DB 인스턴스를 수정한 다음 SQL Server Audit 옵션이 없는 옵션 그룹을 선택합니다. 이 변경은 수정하는 DB 인스턴스에만 영향을 줍니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. 자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#) 단원을 참조하십시오.
3. DB 인스턴스에서 SQL Server Audit 옵션을 삭제한 후 인스턴스를 다시 시작할 필요가 없습니다. S3 빠킷에서 불필요한 감사 파일을 제거하십시오.

SQL Server Audit의 사용

온프레미스 데이터베이스 서버에서 제어하는 것과 동일한 방식으로 서버 감사, 서버 감사 사양 및 데이터베이스 감사 사양을 제어할 수 있습니다.

감사 생성

온프레미스 데이터베이스 서버에 대해 생성하는 것과 같은 방법으로 서버 감사를 생성합니다. 서버 감사를 생성하는 방법에 대한 자세한 내용은 Microsoft SQL Server 설명서에서 [서버 감사 생성](#)을 참조하십시오.

오류를 방지하려면 다음 제한 사항을 준수하십시오.

- 인스턴스당 지원되는 최대 50개의 서버 감사를 초과하지 마십시오.
- 이진 파일에 데이터를 쓰도록 SQL Server에 지시하십시오.
- 서버 감사 이름의 접두사로 RDS_를 사용하지 마십시오.
- FILEPATH에 D:\rdsdbdata\SQLAudit을 지정합니다.
- MAXSIZE에 2MB ~ 50MB 사이의 크기를 지정하십시오.

- MAX_ROLLOVER_FILES 또는 MAX_FILES를 구성하지 마십시오.
- SQL Server가 감사 레코드를 쓰지 못할 경우 DB 인스턴스를 종료하도록 구성하지 마십시오.

감사 사양 생성

온프레미스 데이터베이스 서버에서 생성하는 것과 동일한 방식으로 서버 감사 사양 및 데이터베이스 감사 사양을 생성합니다. 감사 사양 생성에 대한 자세한 내용은 Microsoft SQL Server 설명서에서 [서버 감사 사양 생성 및 데이터베이스 감사 사양 생성](#)을 참조하십시오.

오류를 피하려면 데이터베이스 감사 사양 또는 서버 감사 사양의 접두사로 RDS_를 사용하지 마십시오.

감사 로그 보기

감사 로그는 D:\rdsdbdata\SQLAudit에 저장됩니다.

SQL Server가 감사 로그 파일에 파일 쓰기를 완료하면 파일이 크기 제한에 도달하면 Amazon RDS가 S3 버킷에 파일을 업로드합니다. 보존을 활성화하면 Amazon RDS가 파일을 보존 폴더 D:\rdsdbdata\SQLAudit\transmitted로 옮깁니다.

보존 구성에 대한 자세한 내용은 [DB 인스턴스 옵션에 SQL Server Audit 추가 \(p. 550\)](#) 단원을 참조하십시오.

감사 레코드는 감사 로그 파일이 업로드될 때까지 DB 인스턴스에 보관됩니다. 다음 명령을 실행하여 감사 레코드를 볼 수 있습니다.

```
SELECT *
FROM msdb.dbo.rds_fn_get_audit_file
('D:\rdsdbdata\SQLAudit\*.sqlaudit'
, default
, default )
```

같은 명령을 사용하여 필터를 D:\rdsdbdata\SQLAudit\transmitted*.sqlaudit로 변경하여 보존 폴더의 감사 레코드를 볼 수 있습니다.

```
SELECT *
FROM msdb.dbo.rds_fn_get_audit_file
('D:\rdsdbdata\SQLAudit\transmitted\*.sqlaudit'
, default
, default )
```

다중 AZ 인스턴스에서 SQL Server Audit 사용

다중 AZ 인스턴스의 경우 감사 로그 파일을 Amazon S3로 보내는 프로세스는 단일 AZ 인스턴스의 프로세스와 유사합니다. 그러나 몇 가지 중요한 차이점이 있습니다.

- 데이터베이스 감사 사양 객체는 모든 노드에 복제됩니다.
- 서버 감사 및 서버 감사 사양은 보조 노드에 복제되지 않습니다. 대신, 수동으로 생성하거나 수정해야 합니다.

두 노드에서 서버 감사 사양 또는 서버 감사를 캡처하려면

1. 기본 노드에서 서버 감사 또는 서버 감사 사양을 생성하십시오.
2. 보조 노드로 장애 조치하고 보조 노드에서 동일한 이름과 GUID로 서버 감사 또는 서버 감사 사양을 생성합니다. AUDIT_GUID 파라미터를 사용하여 GUID를 지정합니다.

S3 버킷 구성

감사 로그 파일은 DB 인스턴스에서 S3 버킷으로 자동 업로드됩니다. 감사 파일의 대상으로 사용하는 S3 버킷에는 다음 제한이 적용됩니다.

- DB 인스턴스와 동일한 AWS 리전에 있어야 합니다.
- 대중에게 공개되어서는 안 됩니다.
- 버킷 소유자는 IAM 역할 소유자여야 합니다.

데이터를 저장하는 데 사용되는 대상 키는 `bucket-name/key-prefix/instance-name/audit-name/node_file-name.ext` 명명 스키마를 따릅니다.

Note

버킷 이름과 키 접두사 값을 모두 `s3_BUCKET_ARN` 옵션 설정으로 설정합니다.

스키마는 다음 요소로 구성됩니다.

- **bucket-name** – S3 버킷의 이름.
- **key-prefix** – 감사 로그에 사용할 사용자 지정 키 접두사.
- **instance-name** – Amazon RDS 인스턴스의 이름.
- **audit-name** – 감사의 이름.
- **node** – 감사 로그의 소스인 노드의 식별자(또는). 단일 AZ 인스턴스에는 하나의 노드가 있고 다중 AZ 인스턴스에는 두 개의 복제 노드가 있습니다. 기본 노드와 보조 노드의 역할은 시간이 지남에 따라 변하기 때문에 이들은 기본 노드와 보조 노드가 아닙니다. 대신, 노드 식별자는 단순한 레이블입니다.
 - **node1** – 첫 번째 복제 노드(단일 AZ에는 하나의 노드만 있음).
 - **node2** – 두 번째 복제 노드(다중 AZ에는 두 개의 노드가 있음).
- **file-name** – 대상 파일 이름. 파일 이름은 SQL Server에서 그대로 가져옵니다.
- **ext** – 파일 확장자(또는):
 - **zip** – 압축이 활성화된 경우(기본값).
 - **sqlaudit** – 압축이 비활성화된 경우.

수동으로 SQL Server Audit에 대한 IAM 역할 생성

일반적으로 새 옵션을 만들면 AWS Management 콘솔이 IAM 역할 및 IAM 신뢰 정책을 생성합니다. 그러나 SQL Server Audit에서 사용할 새로운 IAM 역할을 수동으로 생성할 수 있으므로 추가 요구 사항에 맞게 사용자 지정할 수 있습니다. 그러면 IAM 역할을 생성하고 Amazon RDS 서비스가 Amazon S3 버킷을 사용할 수 있도록 권한을 위임합니다. IAM 역할을 만들 때 신뢰 및 권한 정책을 연결합니다. 신뢰 정책은 Amazon RDS가 이 역할을 맡도록 허용합니다. 권한 정책은 이 역할이 수행할 수 있는 작업을 정의합니다. 자세한 내용은 AWS Identity and Access Management 사용 설명서의 [AWS 서비스에 대한 권한을 위임할 역할 생성](#)을 참조하십시오.

이 단원의 예를 사용하여 필요한 신뢰 및 사용 권한 정책을 생성할 수 있습니다.

다음 예는 SQL Server Audit에 대한 신뢰 정책을 보여줍니다. 정책은 서비스 원칙 `rds.amazonaws.com`을 사용하여 RDS가 S3 버킷에 쓸 수 있도록 허용합니다. 서비스 보안 주체는 서비스에 권한을 부여하는 데 사용되는 식별자입니다. 이러한 방식으로 `rds.amazonaws.com`에 대한 액세스를 허용할 때마다 RDS가 사용자를 대신하여 작업을 수행하도록 허용합니다. 서비스 보안 주체에 대한 자세한 내용은 [AWS JSON 정책 요소: 보안 주체](#)를 참조하십시오.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "rds.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
]
```

SQL Server Audit에 대한 권한 정책의 다음 예제에서는 Amazon S3 버킷에 대한 Amazon 리소스 이름(ARN)을 지정합니다. ARN을 사용하여 액세스 권한을 부여할 특정 계정, 사용자 또는 역할을 식별할 수 있습니다. ARN 사용에 대한 자세한 내용은 [Amazon 리소스 이름\(ARN\)](#)을 참조하십시오.

```
{
    "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/*"
        }
    ]
}
```

Microsoft SQL Server에 대한 일반 DBA 작업

이 단원에서는 Microsoft SQL Server 데이터베이스 엔진을 실행하는 DB 인스턴스의 몇 가지 일반적인 DBA 작업을 Amazon RDS에 따라 구현하는 방법에 대해 살펴봅니다. 관리되는 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 셀 액세스를 제공하지 않으며, 고급 권한을 필요로 하는 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다.

Note

SQL Server DB 인스턴스를 이용한 작업에서는 스크립트를 실행하여 새롭게 생성한 데이터베이스를 수정할 수는 있지만 새로운 데이터베이스 모델로 사용되는 데이터베이스인 [모델] 데이터베이스는 수정할 수 없습니다.

주제

- Amazon RDS에서 실행되는 Microsoft SQL Server DB 인스턴스의 tempdb 데이터베이스에 액세스 (p. 555)
- SQL Server 튜닝 어드바이저를 사용한 Amazon RDS DB 인스턴스의 데이터베이스 워크로드 분석 (p. 557)
- Microsoft SQL Server의 데이터 정렬 및 문자 집합 (p. 560)
- Microsoft SQL Server 데이터베이스에 적용할 복구 모델 결정 (p. 560)
- 다중 AZ인 Microsoft SQL Server 데이터베이스 삭제 (p. 561)
- 변경 데이터 캡처 사용 (p. 561)
- 다중 AZ 배포의 Microsoft SQL Server 데이터베이스 이름 변경 (p. 563)
- db_owner 역할 암호 재설정 (p. 563)
- 라이선스가 종료된 DB 인스턴스 복원 (p. 564)
- 오프라인에서 온라인으로 Microsoft SQL Server 데이터베이스 전환 (p. 564)
- SQL Server 에이전트의 사용 (p. 564)
- Microsoft SQL Server 로그 작업 (p. 566)
- 추적 및 덤프 파일 작업 (p. 566)

Amazon RDS에서 실행되는 Microsoft SQL Server DB 인스턴스의 tempdb 데이터베이스에 액세스

Amazon RDS에서 실행되는 Microsoft SQL Server DB 인스턴스의 tempdb 데이터베이스에 액세스할 수 있습니다. Microsoft SQL Server Management Studio(SSMS)를 통한 Transact-SQL 또는 기타 표준 SQL 클라이언트 애플리케이션을 사용하여 tempdb에서 코드를 실행할 수 있습니다. DB 인스턴스 연결에 대한 자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 503\)](#) 단원을 참조하십시오.

DB 인스턴스의 마스터 사용자는 tempdb 데이터베이스 옵션을 수정할 수 있도록 tempdb에 대한 CONTROL 액세스 권한을 부여받습니다. 마스터 사용자는 tempdb 데이터베이스의 데이터베이스 소유자가 아닙니다. 필요한 경우 마스터 사용자는 다른 사용자가 tempdb 데이터베이스 옵션을 수정할 수 있도록 이들에게 CONTROL 액세스 권한을 부여할 수 있습니다.

Note

tempdb 데이터베이스에서 Database Console Commands(DBCC)를 실행할 수 없습니다.

tempdb 데이터베이스 옵션 수정

Amazon RDS DB 인스턴스에 있는 tempdb 데이터베이스의 데이터베이스 옵션을 수정할 수 있습니다. 수정할 수 있는 옵션에 대한 자세한 내용은 Microsoft 설명서의 [tempdb 데이터베이스](#)를 참조하십시오.

최대 파일 크기 옵션과 같은 데이터베이스 옵션은 DB 인스턴스를 다시 시작한 후에도 유지됩니다. 데이터를 가져올 때 성능을 최적화하고 스토리지 부족을 방지하도록 데이터베이스 옵션을 수정할 수 있습니다.

데이터를 가져올 때 성능 최적화

대량의 데이터를 DB 인스턴스로 가져올 때 성능을 최적화하려면 tempdb 데이터베이스의 SIZE 및 FILEGROWTH 속성을 큰 수로 설정합니다. tempdb를 최적화하는 방법에 대한 자세한 내용은 Microsoft 설명서의 [tempdb 성능 최적화](#)를 참조하십시오.

다음은 크기를 100GB로, 파일 증가를 10%로 설정한 예입니다.

```
alter database[tempdb] modify file (NAME = N'templog', SIZE=100GB, FILEGROWTH = 10%)
```

스토리지 문제 방지

tempdb 데이터베이스에서 사용 가능한 모든 디스크 공간을 사용하는 것을 방지하려면 MAXSIZE 속성을 설정합니다. 다음은 속성을 2048MB로 설정한 예입니다.

```
alter database [tempdb] modify file (NAME = N'templog', MAXSIZE = 2048MB)
```

tempdb 데이터베이스 축소

Amazon RDS DB 인스턴스의 tempdb 데이터베이스는 두 가지 방법으로 축소할 수 있습니다.
`rds_shrink_tempdbfile` 프로시저를 사용하거나, SIZE 속성을 설정하면 됩니다.

rds_shrink_tempdbfile 프로시저 사용

Amazon RDS 프로시저 `msdb.dbo.rds_shrink_tempdbfile`을 사용하여 tempdb 데이터베이스를 축소할 수 있습니다. tempdb에 CONTROL 액세스 할 수 있을 때만 `rds_shrink_tempdbfile`을 호출할 수 있습니다. `rds_shrink_tempdbfile`을 호출해도 DB 인스턴스에 가동 중지가 발생하지 않습니다.

`rds_shrink_tempdbfile` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	필수 여부	설명
<code>@temp_filename</code>	SYSNAME	—	필수	축소할 파일의 논리적 이름입니다.
<code>@target_size</code>	int	null	선택 사항	파일의 새로운 크기(MB)입니다.

다음은 tempdb 데이터베이스에 파일 이름을 가져오는 예제입니다.

```
use tempdb;
GO

select name, * from sys.sysfiles;
GO
```

다음은 `test_file`이라는 tempdb 데이터베이스 파일을 축소하고, 새로운 10MB 크기를 요청하는 예제입니다.

```
exec msdb.dbo.rds_shrink_tempdbfile @temp_filename = N'test_file', @target_size = 10;
```

SIZE 속성 설정

SIZE 속성을 설정한 다음 DB 인스턴스를 다시 시작하여 tempdb 데이터베이스를 축소할 수도 있습니다. DB 인스턴스를 다시 시작하는 방법에 대한 자세한 내용은 [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#) 단원을 참조하십시오.

다음은 SIZE 속성을 1024MB로 설정한 예입니다.

```
alter database [tempdb] modify file (NAME = N'templog', SIZE = 1024MB)
```

다중 AZ 배포에 대한 고려 사항

Amazon RDS DB 인스턴스가 데이터베이스 미러링(DBM) 또는 상시 작동 가용성 그룹(AG) 기능을 이용한 Microsoft SQL Server의 다중 AZ 배포에 있는 경우, 고려해야 할 몇 가지 사항이 있습니다.

tempdb 데이터베이스는 복제할 수 없습니다. 기본 인스턴스에 저장하는 어떠한 데이터도 보조 인스턴스에 복제되지 않습니다.

tempdb 데이터베이스의 데이터베이스 옵션을 수정하는 경우, 다음 방법 중 하나를 사용하여 보조 인스턴스에 그러한 변경 사항을 캡처할 수 있습니다.

- 먼저 DB 인스턴스를 수정하고 다중 AZ를 비활성화한 다음 tempdb를 수정하고 마지막으로 다중 AZ를 다시 활성화합니다. 이 방법을 사용하는 경우 가동 중지가 없습니다.

자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#) 단원을 참조하십시오.

- 먼저 원래의 기본 인스턴스에서 tempdb를 수정한 다음 수동으로 장애 조치를 수행한 다음 마지막으로 새 기본 인스턴스에서 tempdb를 수정합니다. 이 방법을 사용하는 경우 가동 중지가 있습니다.

자세한 내용은 [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#) 단원을 참조하십시오.

SQL Server 튜닝 어드바이저를 사용한 Amazon RDS DB 인스턴스의 데이터베이스 워크로드 분석

데이터베이스 엔진 튜닝 어드바이저는 데이터베이스 워크로드를 분석하고, 실행하는 쿼리 종류에 따라 최적의 Microsoft SQL Server 데이터베이스 인덱스를 권장하기 위해 Microsoft가 지원하는 클라이언트 애플리케이션입니다. SQL Server Management Studio와 마찬가지로 튜닝 어드바이저 역시 SQL Server 기반 Amazon RDS DB 인스턴스에 연결되어 있는 클라이언트 컴퓨터에서 실행됩니다. 여기서 클라이언트 컴퓨터는 기업 네트워크 내 은프레미스에서 실행되는 로컬 컴퓨터가 될 수도 있고, 혹은 Amazon RDS DB 인스턴스와 동일한 리전에서 실행되는 Amazon EC2 Windows 인스턴스가 될 수도 있습니다.

이 섹션에서는 튜닝 어드바이저의 분석 워크로드를 캡처하는 방법에 대해 살펴보겠습니다. Amazon RDS는 SQL Server 인스턴스에 대한 호스트 액세스를 제한하기 때문에 이 방법은 워크로드 캡처에 우선적으로 사용되는 프로세스입니다. 튜닝 어드바이저에 대한 전체 문서는 [MSDN](#)에서 확인할 수 있습니다.

튜닝 어드바이저를 사용하려면 먼저 워크로드를 어드바이저에게 제공해야 합니다. 여기서 워크로드란 튜닝하려는 데이터베이스(들)에 실행하는 Transact-SQL 문의 집합을 말합니다. 데이터베이스 엔진 튜닝 어드바이저는 데이터베이스를 튜닝할 때 트레이스 파일, 트레이스 테이블, Transact-SQL 스크립트 또는 XML 파일을 워크로드 입력 수단으로 사용합니다. Amazon RDS에서는 클라이언트 컴퓨터의 파일이나 클라이언트 컴퓨터에 액세스할 수 있는 Amazon RDS SQL Server DB의 데이터베이스 테이블을 워크로드로 사용할 수도 있습니다. 하지만 파일이든, 테이블이든 상관없이 튜닝하려는 데이터베이스에 대한 쿼리는 재실행에 적합한 형식에 따라 저장되어야 합니다.

튜닝 어드바이저의 효과를 극대화 하려면 워크로드가 최대한 사실적이어야 합니다. 워크로드 파일이나 테이블은 DB 인스턴스에 대한 트레이스를 수행하여 생성할 수 있습니다. 트레이스가 실행 중일 때도 DB 인스턴스에 가해지는 부하를 시뮬레이션하거나 정상적인 부하로 애플리케이션을 실행할 수 있습니다.

트레이스는 클라이언트 측과 서버 측, 두 가지 유형이 있습니다. 클라이언트 측 트레이스는 설치가 더욱 쉬울 뿐만 아니라 트레이스 이벤트가 캡처되는 것을 SQL Server 프로파일러에서 실시간으로 볼 수 있습니다. 반면 서버 측 트레이스는 설치가 더욱 복잡할 뿐만 아니라 몇 가지 Transact-SQL 스크립트를 작성해야 합니다. 그뿐만 아니라 트레이스가 Amazon RDS DB 인스턴스의 파일에 기록되기 때문에 스토리지 공간을 차지하게 됩니다. DB 인스턴스는 스토리지가 가득 찬 상태가 되어 스토리지 공간이 부족할 경우 더 이상 사용할 수 없기 때문에 실행 중인 서버 측 트레이스가 사용하는 스토리지 공간을 계속해서 주적해야 합니다.

클라이언트 측 트레이스의 경우, SQL Server 프로파일러에 트레이스 데이터가 충분히 캡처되면 트레이스를 로컬 컴퓨터의 파일이나 클라이언트 컴퓨터에 액세스할 수 있는 DB 인스턴스의 데이터베이스 테이블에 저

장하여 워크로드 파일을 생성할 수 있습니다. 하지만 클라이언트 측 트레이스를 사용할 때는 과도한 부하 시트레이스가 쿼리를 모두 캡처하지 못한다는 커다란 단점이 있습니다. 이로 인해 데이터베이스 엔진 투닝 어드바이저의 분석 효과가 약해질 수 있습니다. 따라서 과도한 부하에서 트레이스를 실행하면서 트레이스 세션 중 모든 쿼리를 캡처해야 하는 경우에는 서버 측 트레이스를 사용하는 것이 바람직합니다.

서버 측 트레이스에서는 DB 인스턴스의 트레이스 파일을 적합한 워크로드 파일로 가져오거나, 완료된 트레이스를 DB 인스턴스의 테이블에 저장할 수 있습니다. 또한 SQL Server 프로파일러를 사용해 트레이스를 로컬 컴퓨터의 파일에 저장하거나 투닝 어드바이저를 사용해 DB 인스턴스의 트레이스 테이블을 읽을 수 있습니다.

SQL Server DB 인스턴스의 클라이언트 측 트레이스 실행

SQL Server DB 인스턴스의 클라이언트 측 트레이스를 실행하는 방법

1. SQL Server 프로파일러를 시작합니다. 이 프로파일러는 SQL Server 인스턴스 폴더 내 Performance Tools 폴더에 설치되어 있습니다. 클라이언트 측 트레이스를 시작하려면 트레이스 정의 템플릿을 로드하거나 정의해야 합니다.
2. SQL Server 프로파일러 파일 메뉴에서 [New Trace]를 클릭합니다. [Connect to Server] 대화 상자에 DB 인스턴스 앤드포인트, 포트, 마스터 사용자 이름 그리고 트레이스를 실행할 데이터베이스의 암호를 입력합니다.
3. [Trace Properties] 대화 상자에 트레이스 이름을 입력한 다음 트레이스 정의 템플릿을 선택합니다. 기본 템플릿인 TSQL_Replay는 애플리케이션으로 제공됩니다. 이 템플릿을 편집하여 트레이스를 정의할 수 있습니다. [Trace Properties] 대화 상자의 [Events Selection] 탭 아래 있는 이벤트와 이벤트 정보를 편집합니다. 트레이스 정의 템플릿을 비롯해 SQL Server 프로파일러를 사용하여 클라이언트 측 트레이스를 지정하는 방법에 대한 자세한 내용은 [MSDN](#)을 참조하십시오.
4. 클라이언트 측 트레이스를 시작하여 DB 인스턴스에 대한 SQL 쿼리 실행을 실시간으로 모니터링합니다.
5. 트레이스가 완료되면 파일 메뉴에서 Stop Trace(트레이스 중지)를 선택합니다. 결과를 파일이나 DB 인스턴스의 트레이스 테이블로 저장합니다.

SQL Server DB 인스턴스의 서버 측 트레이스 실행

스크립트를 작성하여 서버 측 트레이스를 생성하는 것은 복잡할 뿐만 아니라 이 문서의 범위에서 벗어납니다. 따라서 여기서는 예제로 사용할 수 있는 샘플 스크립트만 다릅니다. 클라이언트 측 트레이스와 마찬가지로 이 트레이스의 목적 역시 데이터베이스 엔진 투닝 어드바이저를 사용해 열 수 있도록 워크로드 파일 또는 트레이스를 생성하는 데 있습니다.

다음은 서버 측 트레이스를 시작하여 세부 정보를 워크 로드에 캡처하기 위해 요약된 예제 스크립트입니다. 트레이스가 처음에는 D:\RDSDDBDATA\Log 디렉터리의 RDSTrace.trc 파일에 저장되지만 100MB마다 롤오버되어 이후부터는 RDSTrace_1.trc, RDSTrace_2.trc 등의 이름으로 트레이스 파일이 저장됩니다.

```
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
```

다음은 트레이스를 중단하는 스크립트 예제입니다. 단, 이전 스크립트에서 생성된 트레이스는 명시적으로 트레이스가 중단되거나, 프로세스의 디스크 공간이 부족해질 때까지 계속 실행됩니다.

```
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
```

서버 측 트레이스 결과는 데이터베이스 테이블로 저장한 후 fn_trace_gettable 함수를 이용하면 투닝 어드바이저의 워크로드로 사용할 수 있습니다. 다음 명령은 RDSTrace_1.trc 등의 모든 룰오버 파일을 포함해 D:\rdsdbdata\Log 디렉터리에서 RDSTrace.trc라는 이름의 모든 파일 결과를 현재 데이터베이스에서 RDSTrace라는 이름의 테이블로 로드합니다.

```
SELECT * INTO RDSTrace
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace.trc', default);
```

예를 들어 RDSTrace_1.trc 같이 특정 룰오버 파일을 테이블에 저장하려면 룰오버 파일의 이름을 지정한 다음 fn_trace_gettable의 마지막 파라미터를 default가 아닌 1로 변경합니다.

```
SELECT * INTO RDSTrace_1
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace_1.trc', 1);
```

트레이스를 이용한 투닝 어드바이저 실행

로컬 파일이든 데이터베이스 테이블이든 상관없이 트레이스가 생성되면 이제 DB 인스턴스에 대해 투닝 어드바이저를 실행할 수 있습니다. Microsoft는 데이터베이스 엔진 투닝 어드바이저의 사용법에 관한 문서를 [MSDN](#)에서 제공하고 있습니다. Amazon RDS와 함께 투닝 어드바이저를 사용하는 방법은 독립된 원격 SQL Server 인스턴스를 사용할 때와 프로세스가 동일합니다. 그 밖에 클라이언트 컴퓨터에서 투닝 어드바이저 UI를 사용하거나 명령줄에서 dta.exe 유ти리티를 사용할 수도 있습니다. 하지만 두 가지 경우 모두 DB 인스턴스 앤드포인트를 사용하여 Amazon RDS DB 인스턴스에 연결한 후 마스터 사용자 이름과 마스터 사용자 암호를 입력해야만 투닝 어드바이저를 사용할 수 있습니다.

다음 코드 예제는 dta.cnazcmklsdei.us-east-1.rds.amazonaws.com 앤드포인트에서 Amazon RDS DB 인스턴스에 대해 dta.exe 명령줄 유ти리티를 사용하는 방법을 나타냅니다. 예제에서 admin은 마스터 사용자 이름이고, test는 마스터 사용자 암호입니다. 예로써 투닝할 데이터베이스 이름은 RDSDTA이며, C:\RDSTrace.trc라는 이름의 로컬 컴퓨터에 저장된 트레이스 파일을 입력 워크로드로 사용하였습니다. 또한 예제 명령줄 코드를 보면 트레이스 세션 이름을 RDSTrace1로 지정하였으며, 로컬 컴퓨터에 출력되는 파일 이름을 SQL 출력 스크립트의 경우 RDSTrace.sql로, 결과 파일의 경우 RDSTrace.txt로, 그리고 분석한 XML 파일의 경우 RDSTrace.xml로 지정하였습니다. 마지막으로 RDSDTA 데이터베이스 이름을 RDSTraceErrors로 지정한 오류 테이블도 있습니다.

```
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
```

다음 예제 명령줄 코드는 입력 워크로드가 RDSDTA 데이터베이스에서 RDSTrace라는 이름의 원격 Amazon RDS 인스턴스에 저장된 테이블이라는 점을 제외하고 모두 동일합니다.

```
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
```

dta 유틸리티의 명령줄 파라미터 전체 목록은 [MSDN](#)에서 확인할 수 있습니다.

Microsoft SQL Server의 데이터 정렬 및 문자 집합

Amazon RDS는 Microsoft SQL Server DB 인스턴스를 생성할 때 문자 세트를 위한 기본 서버 데이터 정렬(collation)을 생성합니다. 이 기본 서버 데이터 정렬은 현재 영어(미국) 버전이지만 엄밀히 얘기해서 SQL_Latin1_General_CI_AS가 사용됩니다. 새로운 데이터베이스 또는 데이터베이스 객체를 생성하면서 데이터 정렬을 무시하면 데이터베이스, 테이블 또는 열을 기준으로 기본 데이터 정렬을 변경할 수도 있습니다. 예를 들어 기본 데이터 정렬인 SQL_Latin1_General_CI_AS를 Japanese_CI_AS로 변경하여 일본어 데이터 정렬을 지원하는 것도 가능합니다. 모든 쿼리 인수는 필요에 따라 타입캐스트를 통해 다른 데이터 정렬을 사용할 수 있습니다.

예를 들어 다음 쿼리는 AccountName 열의 기본 데이터 정렬을 Japanese_CI_AS로 변경합니다.

```
CREATE TABLE [dbo].[Account]
(
    [AccountID] [nvarchar](10) NOT NULL,
    [AccountName] [nvarchar](100) COLLATE Japanese_CI_AS NOT NULL
) ON [PRIMARY];
```

Microsoft SQL Server DB 엔진은 기본적인 NCHAR, NVARCHAR 및 NTEXT 데이터 유형에 따라 Unicode를 지원합니다. 이를 테면 CJK 지원이 필요할 경우 문자 스토리지에 이 세 가지 Unicode 데이터 유형을 사용하면 데이터베이스 및 테이블 생성 시 기본 서버 데이터 정렬을 무시할 수 있습니다. SQL Server에 대한 데이터 정렬 및 Unicode 지원을 설명한 Microsoft 링크를 몇 가지 소개합니다.

- [Working with Collations](#)
- [Collation and International Terminology](#)
- [Using SQL Server Collations](#)
- [International Considerations for Databases and Database Engine Applications](#)

Microsoft SQL Server 데이터베이스에 적용할 복구 모델 결정

Amazon RDS에서 복구 모델, 보존 기간 및 데이터베이스 상태는 서로 연결되어 있습니다.

이러한 설정 중 하나를 변경하기 전에 결과를 이해하는 것이 중요합니다. 각 설정은 다른 설정에 영향을 줄 수 있습니다. 예:

- 백업 보존이 활성화되어 있는 상태에서 데이터베이스의 복구 모델을 SIMPLE로 변경하면 Amazon RDS는 5분 이내에 복구 모델을 다시 FULL로 설정합니다. 또한 Amazon RDS가 DB 인스턴스의 스냅샷을 캡처합니다.
- 백업 보존을 "0"일로 설정하면 Amazon RDS가 복구 모드를 SIMPLE로 설정합니다.
- 백업 보존이 "0"일로 설정되어 있는 상태에서 데이터베이스의 복구 모델을 SIMPLE에서 다른 옵션으로 변경하면 Amazon RDS가 복구 모델을 다시 SIMPLE로 설정합니다.

Important

ALTER DATABASE 등을 사용하여 수행할 수 있을 것처럼 보이더라도 다중 AZ 인스턴스에서 복구 모델을 절대로 변경하지 마십시오. 백업 보존, 따라서 전체 복구 모드는 다중 AZ에 필요합니다. 복구 모델을 변경하면 RDS가 즉시 다시 이를 FULL로 변경합니다.

이 자동 재설정을 수행하면 RDS가 미러를 완전히 재구축합니다. 이 재구축 중에 미러가 장애 조치에 대비할 때까지 약 30-90분 동안 데이터베이스의 가용성이 저하됩니다. 또한 DB 인스턴스는 단일 AZ에서 다중 AZ로 변환하는 경우와 같은 방식으로 성능 저하를 경험합니다. 성능이 저하되는 기간

은 데이터베이스 스토리지 크기에 따라 달라지는데, 저장된 데이터베이스가 클수록 성능 저하가 길어집니다.

다중 AZ인 Microsoft SQL Server 데이터베이스 삭제

다중 AZ 배포에서 Microsoft SQL Server를 실행하는 Amazon RDS DB 인스턴스의 데이터베이스를 삭제할 수 있습니다. 데이터베이스를 삭제하려면 다음 명령을 사용합니다.

```
--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

이 절차를 사용하여 데이터베이스를 삭제하면, Amazon RDS가 데이터베이스와의 기존 연결을 모두 삭제하고 데이터베이스의 백업 기록을 제거합니다.

변경 데이터 캡처 사용

Amazon RDS는 Microsoft SQL Server를 실행하는 DB 인스턴스에 대해 CDC(변경 데이터 캡처)를 지원합니다. 테이블의 데이터에 대한 CDC 캡처 변형입니다. 각 변경 사항에 대한 메타데이터를 저장하고, 이후에 액세스할 수 있습니다. CDC 작동 방식에 대한 자세한 내용은 Microsoft 설명서의 [변경 데이터 캡처](#)를 참조하십시오.

Amazon RDS DB 인스턴스에 CDC를 사용하기 전에 `msdb.dbo.rds_cdc_enable_db`를 실행하여 데이터베이스에서 이를 활성화합니다. CDC가 활성화된 이후 데이터베이스의 `db_owner`인 모든 사용자는 데이터베이스의 테이블에서 CDC를 활성화 또는 비활성화할 수 있습니다.

Important

복원 도중 CDC가 비활성화됩니다. 모든 관련 메타데이터가 데이터베이스에서 자동으로 제거됩니다. 이는 스냅샷 복원, 특정 시점으로 복원 및 S3로부터의 SQL Server 기본 복원에 적용됩니다. 이러한 유형의 복원 중 하나를 수행한 이후 CDC를 다시 활성화하고 추적할 테이블을 다시 지정할 수 있습니다.

```
--Enable CDC for RDS DB Instance
exec msdb.dbo.rds_cdc_enable_db '<database name>'
```

CDC를 비활성화하려면 `msdb.dbo.rds_cdc_disable_db`를 실행합니다.

```
--Disable CDC for RDS DB Instance
exec msdb.dbo.rds_cdc_disable_db '<database name>'
```

주제

- [변경 데이터 캡처로 테이블 추적 \(p. 561\)](#)
- [데이터 캡처 변경 작업 \(p. 562\)](#)
- [다중 AZ 인스턴스에 대한 변경 데이터 캡처 \(p. 562\)](#)

변경 데이터 캡처로 테이블 추적

데이터베이스에서 CDC가 활성화된 이후 특정 테이블 추적을 시작할 수 있습니다. `sys.sp_cdc_enable_table`을 실행하여 추적할 테이블을 선택할 수 있습니다.

```
--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>'  
;
```

테이블에 대한 CDC 구성은 [sys.sp_cdc_help_change_data_capture](#)를 실행합니다.

```
--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>'  
;
```

SQL Server 설명서의 CDC 테이블, 함수 및 저장된 절차에 대한 자세한 내용은 다음을 참조하십시오.

- 변경 데이터 캡처 저장된 절차(Transact-SQL)
- 변경 데이터 캡처 함수(Transact-SQL)
- 변경 데이터 캡처 테이블(Transact-SQL)

데이터 캡처 변경 작업

CDC를 활성화할 때 SQL Server가 CDC 작업을 생성합니다. 데이터베이스 소유자(db_owner)는 CDC 작업의 보기, 생성, 수정 및 삭제가 가능합니다. 하지만 RDS 시스템 계정은 이들을 소유합니다. 따라서 기본 보기, 절차 또는 SQL Server Management Studio에서 작업이 보이지 않습니다.

데이터베이스의 CDC 동작을 제어하려면 [sp_cdc_enable_table](#) 및 [sp_cdc_start_job](#)과 같은 기본 SQL Server 절차를 사용합니다. maxtrans 및 maxscans와 같은 CDC 작업 파라미터를 변경하려면 [sp_cdc_change_jobs](#)를 사용할 수 있습니다.

CDC 작업에 관한 추가 정보를 얻으려면 다음 동적 관리 보기([dm_cdc_errors](#))를 쿼리할 수 있습니다.

- [sys.dm_cdc_errors](#)
- [sys.dm_cdc_log_scan_sessions](#)
- [sysjobs](#)
- [sysjobhistory](#)

다중 AZ 인스턴스에 대한 변경 데이터 캡처

다중 AZ 인스턴스에서 CDC를 사용하는 경우 미러의 CDC 작업 구성이 보안 주체에 있는 것과 일치해야 합니다. CDC 작업은 database_id로 매핑됩니다. 보조의 데이터베이스 ID가 보안 주체의 ID와 다른 경우 작업이 옮겨집니다. 장애 조치 이후 오류를 방지하기 위해 RDS는 작업을 취소하고 새 보안 주체에 작업을 다시 생성합니다. 다시 생성된 작업은 장애 조치 이전에 보안 주체가 기록한 파라미터를 사용합니다.

이 프로세스가 빠르게 실행되기는 하지만 CDC 작업은 RDS가 이를 수정하기 전에 실행될 수 있습니다. 기본 복제본과 보조 복제본 사이의 파라미터를 강제로 일관되게 하는 3가지 방법이 있습니다.

- CDC를 활성화한 모든 데이터베이스에 대해 동일한 작업 파라미터를 사용합니다.
- CDC 작업 구성을 변경하기 전에 다중 AZ 인스턴스를 단일 AZ로 변환합니다.
- 보안 주체에서 파라미터를 변경할 때마다 수동으로 전송합니다.

장애 조치 이후 CDC 작업을 다시 생성하는 데 사용되는 CDC 파라미터를 보고 정의하려면 `rds_show_configuration` 및 `rds_set_configuration`을 사용합니다.

다음 예제는 `cdc_capture_maxtrans`에 대한 값 세트를 반환합니다. RDS_DEFAULT로 설정된 모든 파라미터의 경우 RDS가 자동으로 값을 구성합니다.

```
-- Show configuration for each parameter on either primary and secondary replicas.  
exec rdsadmin.dbo.rds_show_configuration 'cdc_capture_maxtrans'
```

보조에서 구성을 설정하려면 `rdsadmin.dbo.rds_set_configuration`을 실행합니다. 이 절차는 부 서버의 모든 데이터베이스에 대한 파라미터 값을 설정합니다. 이러한 설정은 장애 조치 이후에만 사용됩니다. 다음 예제는 모든 CDC 캡처 작업에 대한 `maxtrans`을 **1000**으로 설정합니다.

```
--To set values on secondary. These are used after failover.  
exec rdsadmin..rds_set_configuration 'cdc_capture_maxtrans' , 1000
```

보안 주체에서 CDC 작업 파라미터를 설정하려면 `sys.sp_cdc_change_jobs`를 대신 사용합니다.

다중 AZ 배포의 Microsoft SQL Server 데이터베이스 이름 변경

다중 AZ를 사용하는 Microsoft SQL Server 데이터베이스 인스턴스의 이름을 변경하려면 다음 절차에 따릅니다.

1. 먼저 해당 DB 인스턴스에 대해 다중 AZ를 해제합니다.
2. `rdsadmin.dbo.rds_modify_db_name`을 실행하여 데이터베이스의 이름을 변경합니다.
3. 그 다음에 해당 DB 인스턴스에 대해 다중 AZ 미러링 또는 상시 작동 가용성 그룹을 활성화하여 원래 상태로 되돌립니다.

자세한 내용은 [다중 AZ를 Microsoft SQL Server DB 인스턴스에 추가 \(p. 538\)](#) 단원을 참조하십시오.

Note

해당 인스턴스에서 다중 AZ를 사용하지 않는 경우 `rdsadmin.dbo.rds_modify_db_name`을 실행하기 전 또는 후에 설정 값을 수정할 필요가 없습니다.

예시: 다음 예시에서 `rdsadmin.dbo.rds_modify_db_name` 저장 프로시저는 데이터베이스 이름을 **MOO**에서 **ZAR**로 바꿉니다. 이는 DDL `ALTER DATABASE [MOO] MODIFY NAME = [ZAR]` 문을 실행하는 것과 유사합니다.

```
EXEC rdsadmin.dbo.rds_modify_db_name N'MOO', N'ZAR'  
GO
```

db_owner 역할 암호 재설정

Microsoft SQL Server 데이터베이스의 `db_owner` 역할의 암호를 분실했을 경우 DB 인스턴스 마스터 암호를 수정하여 `db_owner` 역할 암호를 재설정할 수 있습니다. DB 인스턴스 마스터 암호를 변경하면 DB 인스턴스

에 다시 액세스할 수 있고, 수정된 db_owner 암호를 사용하여 데이터베이스에 액세스할 수 있으며, 실수로 취소되었을 수 있는 db_owner 역할에 대한 권한을 복원할 수 있습니다. Amazon RDS 콘솔, AWS CLI 명령 [modify-db-instance](#), 또는 [ModifyDBInstance](#) 작업을 사용하여 DB 인스턴스 암호를 변경할 수 있습니다. SQL Server DB 인스턴스 수정에 대한 자세한 내용은 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#) 단원을 참조하십시오.

라이선스가 종료된 DB 인스턴스 복원

Microsoft는 Microsoft License Mobility 정보를 보고하지 않은 일부 Amazon RDS 고객에게 DB 인스턴스를 종료하도록 요청한 바 있습니다. Amazon RDS는 이러한 DB 인스턴스의 스냅샷을 캡처하며, 이 스냅샷에서 라이선스 포함 서비스 모델이 있는 새 DB 인스턴스로 복원할 수 있습니다.

Standard Edition의 스냅샷에서 Standard Edition 또는 Enterprise Edition으로 복원할 수 있습니다.

Enterprise Edition의 스냅샷에서 Standard Edition 또는 Enterprise Edition으로 복원할 수 있습니다.

Amazon RDS가 인스턴스의 최종 스냅샷을 생성한 후 SQL Server 스냅샷에서 복원하는 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. SQL Server DB 인스턴스의 스냅샷을 선택합니다. Amazon RDS가 DB 인스턴스의 최종 스냅샷을 생성합니다. 종료된 인스턴스 스냅샷의 이름은 '<name of instance>-final-snapshot' 형식입니다. 예를 들어 DB 인스턴스 이름이 `mytest.cdxgahs1ksma.us-east-1.rds.com`일 경우, 최종 스냅샷은 `mytest-final-snapshot`이라는 이름으로 원본 DB 인스턴스와 동일한 AWS 리전에 있습니다.
4. 작업에서 스냅샷 복원을 선택합니다.
[Restore DB Instance] 창이 나타납니다.
5. 라이선스 모델에서 라이선스 포함을 선택합니다.
6. 사용할 SQL Server DB 엔진을 선택합니다.
7. DB 인스턴스 식별자에 복원된 DB 인스턴스의 이름을 입력합니다.
8. [Restore DB Instance]를 선택합니다.

스냅샷에서 복원하는 자세한 방법은 [DB 스냅샷에서 복원 \(p. 205\)](#) 단원을 참조하십시오.

오프라인에서 온라인으로 Microsoft SQL Server 데이터베이스 전환

OFFLINE에서 ONLINE으로 Amazon RDS DB 인스턴스의 Microsoft SQL Server 데이터베이스를 전환할 수 있습니다.

SQL Server 메소드	Amazon RDS 메서드
ALTER DATABASE ## SET ONLINE;	EXEC rdsadmin.dbo.rds_set_database_online ##

SQL Server 에이전트의 사용

Amazon RDS는 Microsoft SQL Server Standard, Web Edition 또는 Enterprise Edition을 기반으로 하는 DB 인스턴스에서 SQL Server 에이전트를 사용할 수 있습니다. SQL Server 에이전트는 관리 작업을 예약하여

실행할 수 있는 Microsoft Windows 서비스입니다. SQL Server 에이전트를 사용하면 T-SQL 작업을 통해 인덱스를 리빌드하고, 손상 검사를 실행하고, SQL Server DB 인스턴스의 데이터를 집계할 수 있습니다.

SQL Server 에이전트는 특정 이벤트 발생 시, 혹은 필요에 따라 예약 작업을 실행할 수 있습니다. 자세한 내용은 SQL Server 문서의 [SQL Server Agent](#) 단원을 참조하십시오. 단, DB 인스턴스의 유지 관리 기간이나 백업 기간에는 예약 작업을 피해야 합니다. AWS에서 실행하는 유지 관리 및 백업 프로세스로 인해 예약 작업이 중단되거나 취소될 수도 있기 때문입니다. Amazon RDS는 DB 인스턴스를 백업하기 때문에 SQL Server 에이전트를 사용해 백업 파일을 만들 필요는 없습니다.

SQL Server Management Studio에서 SQL Server 에이전트의 개별 작업 이력을 보려면 Object Explorer를 열고 작업을 마우스 오른쪽 버튼으로 클릭한 다음 [View History]를 클릭합니다.

SQL Server 에이전트는 DB 인스턴스의 관리형 호스트를 기반으로 하기 때문에 지원되지 않는 작업이 몇 가지 있습니다. 예를 들어 ActiveX, Windows cmdshell, 또는 Windows PowerShell을 이용한 복제 작업이나 명령줄 스크립트가 지원되지 않습니다. 또한 작업을 호스트가 관리하기 때문에 SQL Server 에이전트를 수동으로 시작, 종료 또는 재시작할 수 없습니다. 그 밖에 DB 인스턴스에서 SQL Server 에이전트를 통한 이메일 알림 메시지도 이용하지 못합니다.

SQL Server DB 인스턴스를 생성하면 마스터 사용자 이름이 SQLAgentUserRole 역할에 등록됩니다. 추가 로그인/사용자의 SQL Server 에이전트 사용을 허용하려면 마스터 사용자로 로그인한 후 다음과 같이 실행합니다.

1. CREATE LOGIN 명령을 사용하여 다른 서버 수준 로그인을 생성합니다.
2. CREATE USER 명령을 사용하여 msdb 사용자를 생성한 다음 이전 단계에서 생성한 로그인과 이 사용자를 연결합니다.
3. sp_addrolemember 시스템 저장 프로시저를 사용하여 사용자를 SQLAgentUserRole에 추가합니다.

예를 들어 마스터 사용자 이름은 **myawsmaster**이며, 이름이 **theirname**이고 암호가 **theirpassword**인 사용자에게 SQL Server 에이전트에 대한 액세스 권한을 부여한다고 가정하겠습니다. 먼저 마스터 사용자 이름을 사용해 로그인한 후 다음 명령을 실행합니다.

```
--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];
```

SQL Server 에이전트 작업을 삭제하려면 다음 T-SQL 문을 실행해야 합니다.

```
EXEC msdb..sp_delete_job @job_name = '<job-name>';
```

Note

SQL Server 관리 콘솔(SSMS)의 UI를 사용하여 SQL Server 에이전트 작업을 삭제하지 마십시오. 그렇게 한 경우 다음과 유사한 오류 메시지가 표시됩니다.

```
The EXECUTE permission was denied on the object 'xp_regrid', database
'mssqlsystemresource', schema 'sys'.
```

이 오류는 관리형 서비스로서 RDS가 Windows 레지스트리에 액세스하는 절차 실행이 제한되기 때문에 발생합니다. SSMS를 사용하여 작업을 삭제할 때 RDS에게 권한이 없는 프로세스를 실행하려 합니다.

Microsoft SQL Server 로그 작업

SQL Server 에이전트 로그와 Microsoft SQL Server 오류 로그는 Amazon RDS 콘솔을 사용해 확인하고, 모니터링하고, 다운로드할 수 있습니다.

로그 파일 조사

Amazon RDS 콘솔을 통해 로그를 보면 당시에 존재하는 내용을 확인할 수 있습니다. 콘솔에서 로그를 모니터링하면 로그가 동적 상태로 열리기 때문에 거의 실시간으로 업데이트가 보입니다.

따라서 최신 상태의 로그만 모니터링됩니다. 예를 들어 다음과 같은 로그가 있다고 가정하겠습니다.

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

가장 최근 로그인 log/ERROR만 업데이트가 활성화되어 있습니다. 다른 로그도 볼 수는 있지만 정적 상태로 열리기 때문에 업데이트되지 않습니다.

로그 파일 보관

Amazon RDS 콘솔은 지난 주부터 당일까지 로그를 표시합니다. 이 로그는 당일 이후 참조를 목적으로 다운로드하여 아카이브(보관)할 수 있습니다. Amazon S3 인스턴스에 로드하는 것도 로그를 아카이브할 수 있는 한 가지 방법입니다. Amazon S3 인스턴스 설치 및 파일 업로드 방법에 대한 자세한 내용은 Amazon Simple Storage Service 시작 안내서의 [Amazon S3 기본 사항](#)에서 시작하기를 클릭하면 확인할 수 있습니다.

rds_read_error_log 프로시저 사용

Microsoft SQL 서버 오류 및 에이전트 로그를 보려면 다음 파라미터와 함께 Amazon RDS 저장 프로시저 rds_read_error_log를 사용합니다.

- **@index** – 가져올 로그의 번호. 기본값은 0이며, 현재 오류 로그를 가져옵니다. 이전 로그를 가져오려면 1을 지정하고, 그보다 하나 이전의 로그를 가져오려면 2를 지정하는 식입니다.
- **@type** – 가져올 로그의 유형. 오류 로그를 가져오려면 1을 지정합니다. 에이전트 로그를 가져오려면 2를 지정합니다.

Example

다음 예시에서는 현재 오류 로그를 요청합니다.

```
EXEC rdsadmin.dbo.rds_read_error_log @index = 0, @type = 1;
```

추적 및 덤프 파일 작업

이 단원에서는 Microsoft SQL Server를 실행하는 Amazon RDS DB 인스턴스에 대한 추적 파일 및 덤프 파일 작업에 대해 설명합니다.

추적 SQL 쿼리 생성

```
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
```

열린 추적 보기

```
select * from ::fn_trace_getinfo(default)
```

추적 내용 보기

```
select * from ::fn_trace_gettable('D:\rdsdbdata\log\rdstest.trc', default)
```

추적 및 덤프 파일의 보존 기간 설정

추적 및 덤프 파일이 누적되면서 디스크 공간을 낭비할 수 있습니다. Amazon RDS는 7일 이상 지난 추적 및 덤프 파일은 제거하도록 기본 설정되어 있습니다.

다음 예시에서처럼 현재 추적 및 덤프 파일 보존 기간을 보려면 `rds_show_configuration` 프로시저를 사용합니다.

```
exec rdsadmin..rds_show_configuration;
```

추적 파일의 보존 기간을 수정하려면 `rds_set_configuration` 프로시저를 사용하여 `tracefile retention`을 분 단위로 설정합니다. 다음 예시에서는 추적 파일 보존 기간을 24시간으로 설정합니다.

```
exec rdsadmin..rds_set_configuration 'tracefile retention', 1440;
```

덤프 파일의 보존 기간을 수정하려면 `rds_set_configuration` 프로시저를 사용하여 `dumpfile retention`을 분 단위로 설정합니다. 다음 예시에서는 덤프 파일 보존 기간을 3일로 설정합니다.

```
exec rdsadmin..rds_set_configuration 'dumpfile retention', 4320;
```

보안상의 이유로 인해 SQL Server DB 인스턴스에서는 특정 추적 또는 덤프 파일을 삭제할 수 없습니다. 사용하지 않는 추적 또는 덤프 파일을 모두 삭제하려면 해당 파일에 대한 보존 기간을 0으로 설정합니다.

Microsoft SQL Server DB 인스턴스의 고급 관리 작업 및 개념

이 단원에서는 Amazon RDS 기반 Microsoft SQL Server DB 인스턴스의 고급 관리 작업과 개념에 대한 정보를 제공합니다.

주제

- Microsoft SQL Server DB 인스턴스를 통한 Windows 인증 사용 (p. 568)

Microsoft SQL Server DB 인스턴스를 통한 Windows 인증 사용

사용자가 Microsoft SQL Server 기반 Amazon RDS DB 인스턴스에 접속하려고 할 때 Windows 인증을 사용하여 사용자를 인증할 수 있습니다. DB 인스턴스는 AWS Managed Microsoft AD라고도 하는 Microsoft Active Directory용 AWS 디렉터리 서비스(Enterprise Edition)과 함께 작동하여 Windows 인증을 활성화합니다. 사용자가 신뢰할 수 있는 도메인에 가입된 SQL Server DB 인스턴스를 사용하여 인증할 경우 AWS Directory Service를 사용하여 만든 도메인 디렉터리에 인증 요청이 전달됩니다.

Amazon RDS는 다음을 제외한 모든 AWS 리전에서 SQL Server의 Windows 인증을 지원합니다.

- 미국 서부(캘리포니아 북부 지역)
- 아시아 태평양(뭄바이)
- 남아메리카(상파울루)
- AWS GovCloud(US-East)
- AWS GovCloud (US-West)

Amazon RDS에서는 Windows 인증을 위해 혼합 모드를 사용합니다. 이 방법에서 마스터 사용자(SQL Server DB 인스턴스를 만드는 데 사용된 이름과 암호)는 SQL 인증을 사용합니다. 마스터 사용자 계정은 권한 있는 자격 증명이므로 이 계정에 대한 액세스를 제한해야 합니다.

온프레미스 또는 자체 호스팅된 Microsoft Active Directory를 사용하여 Windows 인증을 수행하려면 포里斯트 트러스트를 만들어야 합니다. AWS Directory Service를 사용하여 포里斯트 트러스트를 설정하는 방법에 대한 자세한 정보는 [신뢰 관계 생성\(AWS Managed Microsoft AD\)](#)을 참조하십시오.

SQL Server DB 인스턴스에 대한 Windows 인증을 설정하려면 이 단원에 자세히 설명된 다음 단계를 수행하십시오.

1. AWS Managed Microsoft AD라고도 하는 Microsoft Active Directory용 AWS 디렉터리 서비스(Enterprise Edition)을 사용하여 AWS 콘솔 또는 AWS Directory Service API에서 AWS Managed Microsoft AD 디렉터리를 만듭니다.
2. AWS CLI 또는 Amazon RDS API를 사용하여 SQL Server DB 인스턴스를 생성하는 경우 관리형 IAM 정책 AmazonRDSDirectoryServiceAccess를 사용하는 AWS Identity and Access Management(IAM) 역할을 만들어야 합니다. 이 역할을 사용하여 Amazon RDS에서 디렉터리를 호출할 수 있습니다. AWS 콘솔을 사용하여 SQL Server DB 인스턴스를 만드는 경우 AWS에서 IAM 역할을 자동으로 생성합니다.

역할이 액세스를 허용하려면 AWS 리전에서 AWS 계정의 AWS Security Token Service(AWS STS) 엔드 포인트를 활성화해야 합니다. AWS STS 엔드포인트는 기본적으로 모든 AWS 리전에서 활성화되어 있으므로 더 이상의 조치 없이 사용할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 리전에서 AWS STS 활성화 및 비활성화](#)를 참조하십시오.

3. Microsoft Active Directory 도구를 사용하여 AWS Managed Microsoft AD 디렉터리에서 사용자 및 그룹을 만들고 구성합니다. Active Directory에서 사용자 및 그룹을 만드는 방법에 대한 자세한 정보는 AWS Directory Service 문서의 사용자 및 그룹 추가(Simple AD 및 AWS Managed Microsoft AD)를 참조하십시오. [사용자 및 그룹을 추가합니다\(Simple AD 및 AWS Managed Microsoft AD\)](#).
4. Amazon RDS를 사용하여 AWS 콘솔, AWS CLI 또는 Amazon RDS API에서 새 SQL Server DB 인스턴스를 만듭니다. 생성 요청에서 디렉터리를 만들 때 생성된 도메인 식별자("d-*" 식별자)와 생성된 역할의 이름을 제공합니다. DB 인스턴스에 대한 domain 및 IAM role 파라미터를 설정하고 도메인 디렉터리와 동일한 VPC에 DB 인스턴스를 배치하여 Windows 인증을 사용하도록 기존 SQL Server DB 인스턴스를 수정할 수도 있습니다.
5. 다른 DB 인스턴스의 경우와 같은 방법으로 Amazon RDS 마스터 사용자 자격 증명을 사용하여 SQL Server DB 인스턴스에 연결합니다. DB 인스턴스는 AWS Managed Microsoft AD 도메인에 조인되므로 도

메인에 있는 Active Directory 사용자 및 그룹의 SQL Server 로그인 및 사용자를 프로비저닝할 수 있습니다(SQL Server "Windows" 로그인이라고 함). 데이터베이스 권한은 이러한 Windows 로그인에 부여되거나 취소되는 표준 SQL Server 권한을 통해 관리됩니다.

Kerberos 인증에 대한 엔드포인트 만들기

Kerberos 기반 인증을 사용하려면 엔드포인트가 고객이 지정한 호스트 이름과 마침표(.) 및 정규화된 도메인 이름(FQDN) 순으로 구성되어야 합니다. Kerberos 기반 인증에 사용되는 엔드포인트의 예를 들면 다음과 같습니다. 이 예에서 SQL Server DB 인스턴스 호스트 이름은 ad-test이고 도메인 이름은 corp-ad.company.com입니다.

```
ad-test.corp-ad.company.com
```

현재 연결에서 Kerberos를 사용 중인지 확인하려면 다음 쿼리를 실행할 수 있습니다.

```
SELECT net_transport, auth_scheme
  FROM sys.dm_exec_connections
 WHERE session_id = @@SPID;
```

SQL Server DB 인스턴스에 대한 Windows 인증 설정

AWS Managed Microsoft AD라고도 하는 Microsoft Active Directory용 AWS 딕렉터리 서비스(Enterprise Edition)을 사용하여 SQL Server DB 인스턴스의 Windows 인증을 설정합니다. Windows 인증을 설정하려면 다음 단계를 수행하십시오.

1단계: Microsoft Active Directory용 AWS 딕렉터리 서비스(Enterprise Edition)을 사용하여 딕렉터리 만들기

AWS Directory Service에서는 AWS 클라우드에서 완벽하게 관리되는 Microsoft Active Directory를 만듭니다. AWS Managed Microsoft AD 딕렉터리를 만들면 AWS Directory Service에서 두 도메인 컨트롤러와 DNS 서버를 자동으로 만듭니다. 딕렉터리 서버는 VPC 내의 서로 다른 서브넷에 생성됩니다. 이 중복으로 인해 장애가 발생하더라도 딕렉터리에 액세스할 수 있습니다.

AWS Managed Microsoft AD 딕렉터리를 만드는 경우 AWS Directory Service에서 다음 작업이 자동으로 수행됩니다.

- VPC 내에서 Microsoft Active Directory를 설정합니다.
- 사용자 이름 Admin과 지정된 암호를 사용하여 딕렉터리 관리자 계정을 생성합니다. 이 계정을 사용하여 딕렉터리를 관리할 수 있습니다.

Note

반드시 이 암호를 저장해야 합니다. AWS Directory Service에서는 이 암호를 저장하지 않으므로 암호를 검색하거나 다시 설정할 수 없습니다.

- 딕렉터리 컨트롤러에 대한 보안 그룹을 만듭니다.

Microsoft Active Directory용 AWS 딕렉터리 서비스(Enterprise Edition)을 시작하면 AWS에서 모든 딕렉터리의 객체를 포함하는 OU(조직 단위)를 만듭니다. 딕렉터리를 만들 때 입력한 NetBIOS 이름을 가진 이 OU는 도메인 루트에 있습니다. 도메인 루트는 AWS에서 소유하고 관리합니다.

AWS Managed Microsoft AD 딕렉터리를 사용하여 만든 admin 계정은 OU의 가장 일반적인 관리 활동에 대한 권한을 갖습니다.

- 업데이트 생성 또는 사용자, 그룹, 컴퓨터 삭제

- 도메인(예: 파일 또는 인쇄 서버)에 리소스를 추가한 다음 OU 내의 사용자 및 그룹에 해당 리소스에 대한 권한 할당
- 추가 OU 및 컨테이너 생성
- 권한 위임
- 그룹 정책 생성 및 연결
- Active Directory 휴지통에서 삭제된 객체 복원
- Active Directory 웹 서비스에서 AD 및 DNS Windows PowerShell 모듈 실행

또한 admin 계정은 다음 도메인 차원 활동을 수행할 권리가 있습니다.

- DNS 구성 관리(레코드, 영역 및 전달자 추가, 제거 또는 업데이트)
- DNS 이벤트 로그 보기
- 보안 이벤트 로그 보기

Microsoft Active Directory용 AWS 디렉터리 서비스(Enterprise Edition)으로 디렉터리를 생성하려면

1. [AWS Directory Service 콘솔](#) 탐색 창에서 Directories(디렉터리)를 선택한 후 디렉터리 설정을 선택합니다.
2. AWS Managed Microsoft AD 생성을 선택합니다. AWS Managed Microsoft AD는 현재 Amazon RDS에서 사용하도록 지원되는 유일한 옵션입니다.
3. 다음 정보를 제공합니다.

디렉터리 DNS

디렉터리의 정규화된 이름(예: corp.example.com)입니다.

NetBIOS name

디렉터리의 짧은 이름(예: CORP)입니다.

관리자 암호

디렉터리 관리자의 암호입니다. 디렉터리 생성 프로세스에서는 사용자 이름 Admin과 이 암호를 사용하여 관리자 계정을 생성합니다.

디렉터리 관리자 암호는 단어 "admin"을 포함할 수 없습니다. 암호는 대소문자를 구분하며 길이가 8~64자 사이여야 합니다. 또한 다음 네 범주 중 세 개에 해당하는 문자를 1자 이상 포함해야 합니다.

- 소문자(a-z)
- 대문자(A-Z)
- 숫자(0-9)
- 영숫자 외의 특수 문자(~!@#\$%^&*_+=`|\(){}[],;"<>,.?/)

[Confirm Password]

관리자 암호를 다시 입력합니다.

설명

디렉터리에 대한 선택적 설명을 입력합니다.

4. [VPC Details] 섹션에 다음 정보를 제공하고 [Next Step]을 선택합니다.

VPC

디렉터리에 대한 VPC입니다. SQL Server DB 인스턴스가 동일한 VPC에 생성되어야 합니다.

서브넷

디렉터리 서버에 대한 서브넷을 선택합니다. 두 서브넷이 서로 다른 가용 영역에 있어야 합니다.

5. 디렉터리 정보를 검토하고 필요한 사항을 변경합니다. 정보가 올바르면 AWS Managed Microsoft AD 생성을 선택합니다.

디렉터리를 생성하는 데 몇 분 정도 걸립니다. 디렉터리가 생성된 경우 [Status] 값이 [Active]로 변경됩니다.

디렉터리에 대한 정보를 보려면 디렉터리 목록에서 해당 디렉터리를 선택합니다. SQL Server DB 인스턴스를 생성하거나 수정할 때 디렉터리 ID가 필요하므로 이 값을 기록해 두십시오.

2단계: Amazon RDS에 사용할 IAM 역할 만들기

AWS 콘솔을 사용하여 SQL Server DB 인스턴스를 만들 경우 이 단계를 건너뛸 수 있습니다. AWS CLI 또는 Amazon RDS API를 사용하여 SQL Server DB 인스턴스를 생성한 경우 관리형 IAM 정책 AmazonRDSDirectoryServiceAccess를 사용하는 IAM 역할을 생성해야 합니다. 이 역할을 사용하여 Amazon RDS에서 AWS Directory Service를 자동으로 호출할 수 있습니다.

다음 IAM 정책 AmazonRDSDirectoryServiceAccess를 사용하여 AWS Directory Service에 액세스할 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "ds:DescribeDirectories",  
                "ds:AuthorizeApplication",  
                "ds:UnauthorizeApplication"  
            ],  
            "Effect": "Allow",  
            "Resource": "*"  
        }  
    ]  
}
```

이 정책을 사용하여 IAM 역할을 만듭니다. IAM 역할을 만드는 방법에 대한 자세한 정보는 [고객 관리형 정책 만들기](#)를 참조하십시오.

3단계: 사용자 및 그룹 생성 및 구성

Active Directory 도메인 서비스 및 Active Directory Lightweight Directory Services 도구의 일부인 Active Directory 사용자 및 컴퓨터 도구를 사용하여 사용자 및 그룹을 생성할 수 있습니다. 사용자는 디렉터리에 액세스할 수 있는 개별 사용자 또는 개체를 나타냅니다. 그룹은 개별 사용자에게 권한을 적용할 필요 없이 사용자 그룹에 권한을 부여하거나 거부하는 데 매우 유용합니다.

AWS Directory Service 디렉터리에서 사용자 및 그룹을 생성하려면 AWS Directory Service 디렉터리의 구성 원인 Windows EC2 인스턴스에 연결하고 사용자 및 그룹 생성 권한이 있는 사용자로 로그인해야 합니다. 자세한 정보는 [사용자 및 그룹 추가\(Simple AD 및 AWS Managed Microsoft AD\)](#) 단원을 참조하십시오.

4단계: SQL Server DB 인스턴스 생성 또는 수정

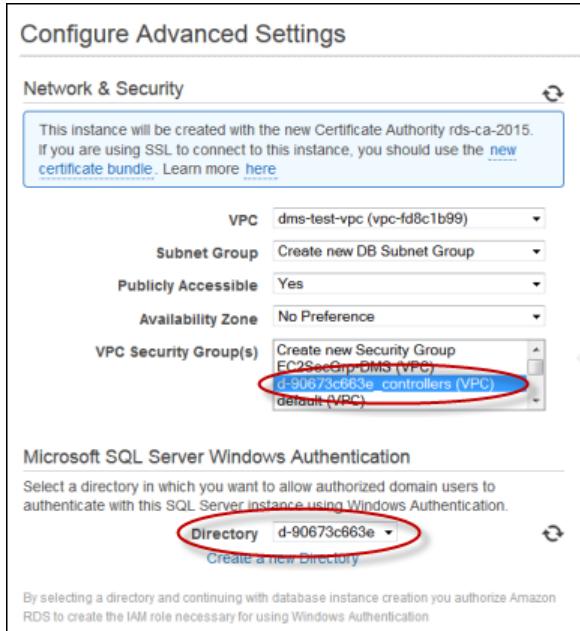
이제 디렉터리에 사용할 Microsoft SQL Server DB 인스턴스를 생성하거나 수정합니다. 이 작업을 다음 중 한 가지 방법으로 수행할 수 있습니다.

- 새 SQL Server DB 인스턴스 생성
- 기존 SQL Server DB 인스턴스 수정
- DB 스냅샷에서 SQL Server DB 인스턴스 복원
- 특정 시점 복원에서 SQL Server DB 인스턴스 복원

Windows 인증은 VPC의 SQL Server DB 인스턴스에 대해서만 지원되고, DB 인스턴스가 디렉터리와 동일한 VPC 내에 있어야 합니다.

DB 인스턴스에서 생성된 도메인 디렉터리를 사용하려면 여러 파라미터가 필요합니다.

- domain 파라미터의 경우 디렉터리를 만들 때 생성된 도메인 식별자("d-*" 식별자)를 입력해야 합니다.
- 디렉터리를 생성할 때 사용한 것과 동일한 VPC를 사용합니다.
- DB 인스턴스가 디렉터리와 통신할 수 있도록 VPC 내에서 송신할 수 있는 보안 그룹을 사용합니다.



5단계: Windows 인증 SQL Server 로그인 생성

다른 DB 인스턴스의 경우와 같은 방법으로 Amazon RDS 마스터 사용자 자격 증명을 사용하여 SQL Server DB 인스턴스에 연결합니다. DB 인스턴스는 AWS Managed Microsoft AD 도메인에 조인되므로 도메인에 있는 Active Directory 사용자 및 그룹의 SQL Server 로그인과 사용자를 프로비저닝할 수 있습니다. 데이터베이스 권한은 이러한 Windows 로그인에 부여되거나 최소되는 표준 SQL Server 권한을 통해 관리됩니다.

Active Directory 사용자가 SQL Server로 인증하도록 허용하려면 사용자가 구성원인 그룹 또는 사용자에 대한 SQL Server Windows 로그인이 있어야 합니다. 세분화된 액세스 제어는 이러한 SQL Server 로그인에 대한 권한을 부여하거나 최소하여 처리합니다. 사용자가 해당 SQL Server 로그인이 없고 해당 SQL Server 로그인을 가진 그룹의 구성원이 아닌 경우 해당 사용자는 SQL Server DB 인스턴스에 액세스할 수 없습니다.

Active Directory SQL Server 로그인을 생성하려면 ALTER ANY LOGIN 권한이 필요합니다. 이 권한을 가진 로그인을 아직 생성하지 않은 경우 SQL Server 인증을 사용하여 DB 인스턴스의 마스터 사용자로 연결합니다. 다음 데이터 조작 언어(DDL) 명령을 실행하여 Active Directory 사용자 또는 그룹에 대한 SQL Server 로그인을 생성합니다.

```
CREATE LOGIN [<user or group>] FROM WINDOWS WITH DEFAULT_DATABASE = [master],  
DEFAULT_LANGUAGE = [us_english];
```

domainName\login_name 형식의 Windows 2000 이전 로그인 이름을 사용하여 사용자 또는 그룹을 지정해야 합니다. login_name@DomainName 형식의 UPN(User Principle Name)을 사용할 수 없습니다. CREATE LOGIN에 대한 자세한 정보는 Microsoft Developer Network 문서의 <https://msdn.microsoft.com/en-us/library/ms189751.aspx>를 참조하십시오.

도메인의 사용자(사용자 및 애플리케이션)는 이제 Windows 인증을 사용하여 도메인이 조인된 클라이언트 컴퓨터의 RDS SQL Server 인스턴스에 연결할 수 있습니다.

도메인에서 DB 인스턴스 관리

AWS 콘솔, AWS CLI 또는 Amazon RDS API를 사용하여 DB 인스턴스 및 DB 인스턴스와 도메인의 관계를 관리할 수 있습니다(예: 도메인 내부, 외부 또는 사이에서 DB 인스턴스 이동).

예를 들어 Amazon RDS API를 사용하여 다음을 수행할 수 있습니다.

- 실패한 멤버십에 대한 도메인 조인을 다시 시도하려면 ModifyDBInstance API 작업을 사용하고 현재 멤버십의 디렉터리 ID를 지정합니다.
- 멤버십에 대한 IAM 역할 이름을 업데이트하려면 ModifyDBInstance API 작업을 사용하고 현재 멤버십의 디렉터리 ID와 새 IAM 역할을 지정합니다.
- 도메인에서 DB 인스턴스를 제거하려면 ModifyDBInstance API 작업을 사용하고 'none'을 도메인 파라미터로 지정합니다.
- 한 도메인에서 다른 도메인으로 DB 인스턴스를 이동하려면 ModifyDBInstance API 작업을 사용하고 새 도메인의 도메인 식별자를 도메인 파라미터로 지정합니다.
- 각 DB 인스턴스에 대한 멤버십을 나열하려면 DescribeDBInstances API 작업을 사용합니다.

도메인 멤버십 이해

DB 인스턴스를 생성하거나 수정한 경우 해당 인스턴스는 도메인의 구성원이 됩니다. AWS 콘솔은 DB 인스턴스에 대한 도메인 멤버십의 상태를 나타냅니다. DB 인스턴스의 상태는 다음 중 한 가지가 될 수 있습니다.

- joined - 인스턴스가 도메인의 구성원입니다.
- joining - 인스턴스가 도메인 구성원이 되기 위한 과정을 진행하고 있습니다.
- pending-join - 인스턴스 멤버십이 보류 중입니다.
- pending-maintenance-join - AWS에서 다음 예약된 유지 관리 기간 동안 인스턴스를 도메인의 구성원으로 만들려고 시도합니다.
- pending-removal - 도메인에서 인스턴스 제거 작업이 보류 중입니다.
- pending-maintenance-removal - AWS에서 다음 예약된 유지 관리 기간 동안 도메인에서 인스턴스를 제거하려고 시도합니다.
- failed - 구성 문제가 발생하여 인스턴스가 도메인에 조인되지 않았습니다. 구성을 확인하고 문제를 해결한 다음 인스턴스 수정 명령을 다시 실행하십시오.
- removing - 인스턴스를 도메인에서 제거하고 있습니다.

네트워크 연결 문제 또는 잘못된 IAM 역할로 인해 도메인 구성원 되기 요청이 실패할 수 있습니다. DB 인스턴스를 생성하거나 기존 인스턴스를 수정한 후 도메인 구성원 되기 시도가 실패할 경우 수정 명령을 다시 실행하거나 새로 생성된 인스턴스를 수정하여 도메인에 조인해야 합니다.

Windows 인증을 사용하여 SQL Server에 연결

Windows 인증을 사용하여 SQL Server에 연결하려면 도메인에 조인된 컴퓨터에 도메인 사용자로 로그인해야 합니다. 다음과 같이 SQL Server Management Studio를 시작한 후 [Windows 인증]을 인증 유형으로 선택합니다.



SQL Server DB 인스턴스를 복원한 후 도메인에 추가

DB 스냅샷을 복원하거나 SQL Server DB 인스턴스에 대한 특정 시점 복원을 수행한 후 도메인에 추가할 수 있습니다. DB 인스턴스가 복원된 후 [4단계: SQL Server DB 인스턴스 생성 또는 수정 \(p. 572\)](#) 단원에 설명된 프로세스를 사용하여 DB 인스턴스를 도메인에 추가하도록 인스턴스를 수정합니다.

관련 주제

- [Amazon RDS의 Microsoft SQL Server \(p. 477\)](#)
- [Amazon RDS의 보안 구성 \(p. 339\)](#)

Amazon RDS의 MySQL

Amazon RDS는 여러 MySQL 버전을 실행하는 DB 인스턴스를 지원합니다. 다음 메이저 버전을 사용할 수 있습니다.

- MySQL 8.0
- MySQL 5.7
- MySQL 5.6
- MySQL 5.5

마이너 버전 지원에 대한 자세한 내용은 [Amazon RDS의 MySQL 버전 \(p. 578\)](#) 단원을 참조하십시오.

사용자는 먼저 Amazon RDS 관리 도구 또는 인터페이스를 사용하여 Amazon RDS MySQL DB 인스턴스를 생성합니다. 그런 다음 DB 인스턴스 크기 조정, DB 인스턴스에 대한 연결 인증, 백업 또는 스냅샷 생성 및 복원, 다중 AZ 보조 생성, 읽기 전용 복제본 생성 및 DB 인스턴스의 성능 모니터링이 가능합니다. 표준 MySQL 유ти리티 및 애플리케이션을 사용하여 DB 인스턴스에서 데이터를 저장하고 데이터에 액세스할 수 있습니다.

MySQL용 Amazon RDS는 다수의 산업 표준을 준수합니다. 예를 들면, MySQL 데이터베이스용 Amazon RDS를 사용하여 HIPAA 준수 애플리케이션을 구축하고 AWS와 체결한 이행 중인 비즈니스 제휴 계약(BAA)에 따라 보호 대상 건강 정보(PHI)를 비롯한 의료 관련 정보를 저장할 수 있습니다. 또한 MySQL용 Amazon RDS는 FedRAMP(연방 위험 및 인증 관리 프로그램) 보안 요건을 충족하며 FedRAMP 공동 승인 위원회(JAB)로부터 AWS GovCloud(미국 서부) 리전 내에서 행사할 수 있는 FedRAMP HIGH Baseline 수준의 잠정적 운영 권한(P-ATO)을 취득했습니다. 지원되는 규정 준수 표준에 대한 자세한 내용은 [AWS 클라우드 규정 준수](#)를 참조하십시오.

MySQL의 각 버전의 기능에 대한 내용은 MySQL 설명서의 [MySQL의 기본 기능](#) 단원을 참조하십시오.

Amazon RDS MySQL을 위한 공통 관리 작업

다음은 Amazon RDS MySQL DB 인스턴스로 수행하는 일반적인 관리 작업과 각 작업에 해당하는 문서 링크입니다.

작업 영역	관련 문서
Amazon RDS 이해 DB 인스턴스, 리전, 사용 영역, 보안 그룹, 파라미터 그룹, 옵션 그룹 등 Amazon RDS의 핵심 구성 요소를 이해합니다.	Amazon Relational Database Service(Amazon RDS)란 무엇입니까? (p. 1)
처음 사용 시 Amazon RDS 설정 Amazon Web Services(AWS)에서 MySQL DB 인스턴스를 생성할 수 있도록 Amazon RDS를 설정합니다.	Amazon RDS 설정 (p. 5)
Amazon RDS DB 인스턴스 이해 AWS에서 실행되는 가상 MySQL 서버 인스턴스를 생성합니다. DB 인스턴스는 Amazon RDS의 기본 구성 요소이므로 해당 원칙을 이해하는 것이 좋습니다.	Amazon RDS DB 인스턴스 (p. 76)

작업 영역	관련 문서
<p>프로덕션용 DB 인스턴스 만들기</p> <p>프로덕션 목적으로 DB 인스턴스를 생성합니다. 인스턴스 생성 과정에서는 적절한 처리 기능과 메모리 용량을 가진 DB 인스턴스 클래스를 선택하고 데이터베이스 사용 방법에 맞는 스토리지 유형을 선택합니다.</p>	<p>DB 인스턴스 클래스 선택 (p. 77)</p> <p>Amazon RDS 스토리지 유형 (p. 101)</p> <p>MySQL 데이터베이스 엔진 기반 DB 인스턴스 생성 (p. 587)</p>
<p>DB 인스턴스의 보안 관리</p> <p>기본적으로, DB 인스턴스와 함께 인스턴스에 대한 액세스를 막는 방화벽도 생성됩니다. 따라서 DB 인스턴스에 액세스하기 위한 알맞은 IP 주소와 네트워크 구성으로 보안 그룹을 만들어야 합니다. RDS 리소스를 관리할 수 있는 사용자를 결정하는 권한을 배정하는 데 AWS Identity and Access Management(IAM) 정책을 사용할 수도 있습니다.</p>	<p>Amazon RDS의 보안 구성 (p. 339)</p> <p>Amazon RDS 리소스에 대한 액세스 권한 관리 개요 (p. 341)</p> <p>보안 그룹을 통한 액세스 제어 (p. 390)</p> <p>EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 (p. 407)</p>
<p>DB 인스턴스에 연결</p> <p>MySQL 명령줄 유ти리티 또는 MySQL Workbench와 같은 표준 SQL 클라이언트 애플리케이션을 사용하여 DB 인스턴스에 연결합니다.</p>	<p>MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 (p. 595)</p>
<p>프로덕션 DB 인스턴스에 대해 고가용성 구성</p> <p>다른 가용 영역의 동기식 예비 복제본, 자동 장애 조치, 다중 AZ 배포를 통한 DB 인스턴스의 내결함성, 읽기 전용 복제본 등을 통해 고가용성을 제공합니다.</p>	<p>Amazon RDS를 위한 고가용성(다중 AZ) (p. 109)</p>
<p>Amazon Virtual Private Cloud에 DB 인스턴스 구성</p> <p>Amazon VPC 서비스에서 Virtual Private Cloud(VPC)를 구성합니다. Amazon VPC는 AWS 내 다른 가상 네트워크에서 논리적으로 격리된 가상 네트워크입니다.</p>	<p>EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 (p. 407)</p> <p>VPC에서 Amazon RDS DB 인스턴스를 사용한 작업 (p. 415)</p>
<p>특정 MySQL 데이터베이스 파라미터 및 기능 구성</p> <p>많은 DB 인스턴스와 연결 가능한 파라미터 그룹을 사용하여 특정 MySQL 데이터베이스 파라미터를 구성합니다. 많은 DB 인스턴스와 연결 가능한 옵션 그룹을 사용하여 특정 MySQL 데이터베이스 기능을 구성할 수도 있습니다.</p>	<p>DB 파라미터 그룹 작업 (p. 161)</p> <p>옵션 그룹 작업 (p. 148)</p> <p>MySQL DB 인스턴스 옵션 (p. 663)</p>
<p>MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경</p> <p>추가 스토리지를 더하거나 DB 인스턴스 클래스를 변경하는 것과 같은 작업을 완수하기 위해 DB 인스턴스의 설정을 변경합니다.</p>	<p>MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경 (p. 600)</p> <p>Amazon RDS DB 인스턴스 수정 (p. 111)</p>

작업 영역	관련 문서
데이터베이스 백업 및 복원 구성 자동 백업을 수행하도록 DB 인스턴스를 구성합니다. 또한 전체 백업 파일을 사용하여 데이터베이스를 수동으로 백업 및 복원할 수도 있습니다.	백업 작업 (p. 196) Amazon RDS DB 인스턴스 백업 및 복원 (p. 196)
데이터 가져오기 및 내보내기 다른 RDS MySQL DB 인스턴스, Amazon RDS 외부에서 실행 중인 MySQL 인스턴스 및 기타 유형의 데이터 소스에서 데이터를 가져오고, Amazon RDS 외부에서 실행 중인 MySQL 인스턴스로 데이터를 내보냅니다.	Amazon RDS MySQL DB 인스턴스로 백업 복원 (p. 619)
MySQL DB 인스턴스 모니터링 Amazon CloudWatch RDS 측정치, 이벤트 및 향상된 모니터링 기능을 통해 RDS MySQL DB 인스턴스를 모니터링합니다. RDS MySQL DB 인스턴스에 대한 로그 파일을 봅니다.	Amazon RDS 모니터링 (p. 228) DB 인스턴스 측정치 보기 (p. 236) Amazon RDS 이벤트 보기 (p. 305) Amazon RDS 데이터베이스 로그 파일 (p. 306) MySQL 데이터베이스 로그 파일 (p. 318)
데이터 복제 동일한 AWS 리전 또는 다른 리전에서 MySQL 읽기 전용 복제본을 생성하십시오. 로드 밸런싱, 재해 복구, 읽기 중심의 데이터베이스 워크로드 처리 등을 위해(예: 분석 및 보고를 위해) 읽기 전용 복제본을 사용할 수 있습니다.	읽기 전용 복제본 작업 (p. 135) Amazon RDS 외부에서 실행 중인 MySQL 또는 MariaDB 인스턴스를 사용한 복제 (p. 655)

또한 Amazon RDS MySQL DB 인스턴스 작업에 대한 유용한 정보가 여러 섹션에 수록되어 있습니다.

- MySQL DB 인스턴스에 대한 공통 DBA 작업 (p. 669)
- MySQL DB 인스턴스 옵션 (p. 663)
- Amazon RDS SQL의 MySQL 참조 (p. 676)

Amazon RDS의 MySQL 버전

MySQL의 경우, 버전 번호는 버전 X.Y.Z로 구성됩니다. Amazon RDS 용어에서, X.Y는 메이저 버전을 나타내고 Z는 마이너 버전 번호를 나타냅니다. Amazon RDS 구현을 위해서, 메이저 버전 번호가 변경될 경우(예: 버전 5.6에서 5.7로) 이를 메이저 버전 변경으로 간주합니다. 단지 마이너 버전 번호가 변경된 경우(예: 버전 5.7.16에서 5.7.21로)에는 마이너 버전 변경으로 간주합니다.

Amazon RDS에서는 현재 다음과 같은 MySQL 버전을 지원합니다.

메이저 버전	마이너 버전
MySQL 8.0	<ul style="list-style-type: none"> • 8.0.15 • 8.0.13 • 8.0.11

메이저 버전	마이너 버전
MySQL 5.7	<ul style="list-style-type: none"> • 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"> • 5.6.41 • 5.6.40 • 5.6.39 • 5.6.37 • 5.6.35 • 5.6.34
MySQL 5.5	<ul style="list-style-type: none"> • 5.5.61 • 5.5.59 • 5.5.57 • 5.5.54 • 5.5.53 • 5.5.46

새 DB 인스턴스를 생성할 때는 현재 지원되는 모든 MySQL 버전을 지정할 수 있습니다. 메이저 버전(예: MySQL 5.7) 및 지정된 메이저 버전에 대해 지원되는 모든 마이너 버전을 지정할 수 있습니다. 버전이 지정되지 않은 경우 Amazon RDS는 지원되는 버전(보통 최신 버전)을 기본값으로 설정합니다. 메이저 버전이 지정되었지만 마이너 버전이 지정되지 않은 경우, Amazon RDS는 고객이 지정한 메이저 버전의 최근 릴리스를 기본값으로 설정합니다. 지원되는 버전 목록과 새로 만든 DB 인스턴스의 기본값을 보려면 [describe-db-engine-versions](#) AWS CLI 명령을 사용합니다.

Amazon RDS를 통해 사용자는 MySQL 인스턴스를 Amazon RDS가 지원하는 새 버전으로 언제 업그레이드 할지를 제어합니다. 특정 MySQL 버전과의 호환성을 유지하고, 프로덕션 환경에 배포하기 전에 애플리케이션으로 새 버전을 테스트하고, 가장 원하는 일정에 맞춰 버전 업그레이드를 수행할 수 있습니다.

사용자가 달리 지정하지 않는 한 Amazon RDS에서 새로운 MySQL 마이너 버전을 지원할 때 DB 인스턴스가 새 버전으로 자동 업그레이드됩니다. 이 패치는 예약 유지보수 중에 발생합니다. 마이너 버전 자동 업그레이드를 끄도록 DB 인스턴스를 수정할 수 있습니다.

자동으로 예약된 업그레이드를 사용하지 않기로 선택한 경우, 사용자는 메이저 버전 업데이트를 위해 선택한 것과 동일한 절차에 따라 지원되는 마이너 버전 릴리스로 수동으로 업그레이드할 수 있습니다. 자세한 내용은 [DB 인스턴스 엔진 버전 업그레이드 \(p. 118\)](#) 단원을 참조하십시오.

Amazon RDS는 현재 MySQL 버전 5.5에서 버전 5.6으로, MySQL 버전 5.6에서 버전 5.7로, MySQL 버전 5.7에서 버전 8.0으로의 메이저 버전 업그레이드를 지원합니다. 메이저 버전 업그레이드에 약간의 호환성 위험 이 수반되므로 업그레이드가 자동으로 이루어지지 않습니다. 즉, DB 인스턴스 수정을 요청해야 합니다. 프로덕션 인스턴스를 업그레이드하기 전에 모든 업그레이드를 철저하게 테스트해야 합니다. MySQL DB 인스턴스 업그레이드에 대한 자세한 내용은 [MySQL DB 엔진 업그레이드 \(p. 607\)](#)를 참조하십시오.

기존 DB 인스턴스의 DB 스냅샷을 만들고 DB 스냅샷에서 복구해 새 DB 인스턴스를 만든 다음 새로운 DB 인스턴스에 대한 버전 업그레이드를 시작함으로써, 업그레이드하기 전에 새 버전과 비교하여 DB 인스턴스를 테스트할 수 있습니다. 그런 다음 원래의 DB 인스턴스에 대한 업그레이드 여부를 결정하기 전에 DB 인스턴스의 업그레이드된 복제본에서 안전하게 실험을 진행할 수 있습니다.

MySQL에 대한 Amazon RDS 운영 중단 정책에 대한 자세한 내용은 [Amazon RDS FAQ](#) 단원을 참조하십시오.

Amazon RDS가 지원하지 않는 MySQL 기능

Amazon RDS는 현재 다음과 같은 MySQL 기능은 지원하지 않습니다.

- 인증 플러그인
- 시스템 로그에 오류 로깅
- 그룹 복제 플러그인
- InnoDB Tablespace 암호화
- MariaDB 감사 플러그인(Amazon RDS MySQL 버전 8.0에만 지원되지 않음)

MariaDB 감사 플러그인은 Amazon RDS MySQL 버전 5.5, 5.6, 5.7에서 지원됩니다.

- 암호 보안 수준 플러그인
- 지속된 시스템 변수
- 복제 필터
- 반동기식 복제
- 전송 가능한 테이블스페이스
- X 플러그인

Note

글로벌 트랜잭션 ID는 MySQL 5.7.23 및 MySQL 5.7 이후 버전에서 지원됩니다. 글로벌 트랜잭션 ID는 Amazon RDS MySQL 5.5, 5.6, 또는 8.0 버전에서 지원되지 않습니다.

IAM 데이터베이스 인증은 MySQL 5.6과 5.7에서 지원됩니다. IAM 데이터베이스 인증은 MySQL 5.5 또는 8.0에서 지원되지 않습니다.

Amazon RDS 성능 개선 도우미는 MySQL 5.6과 5.7에서 지원됩니다. Amazon RDS 성능 개선 도우미는 MySQL 5.5 또는 8.0에서 지원되지 않습니다.

관리형 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 shell 액세스를 제공하지 않습니다. 또한 고급 권한을 필요로 하는 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다. Amazon RDS는 표준 SQL 클라이언트 애플리케이션을 사용하여 DB 인스턴스의 데이터베이스에 대한 액세스를 지원합니다. Amazon RDS는 Telnet, Secure Shell(SSH), 또는 Windows 원격 데스크톱 연결을 사용하여 DB 인스턴스에 대한 직접 호스트 액세스를 허용하지 않습니다. DB 인스턴스를 생성할 때 사용자는 해당 인스턴스의 모든 데이터베이스에 대한 db_owner 역할을 할당받게 되며, 백업을 위해 사용된 권한을 제외한 모든 데이터베이스 수준의 권한을 갖게 됩니다. Amazon RDS가 백업을 자동으로 관리합니다.

Amazon RDS MySQL에 대해 지원되는 스토리지 엔진

MySQL은 다양한 기능으로 여러 스토리지 엔진을 지원하지만, 모든 기능이 복구와 데이터 내구성에 최적화되어 있지는 않습니다. Amazon RDS에서는 MySQL DB 인스턴스용 InnoDB 스토리지 엔진을 완벽히 지원합니다. 지정 시간 복원 및 스냅샷 복원과 같은 Amazon RDS 기능은 복구 가능 스토리지 엔진이 필요하며 InnoDB 스토리지 엔진에 대해서만 지원됩니다. InnoDB memcached 인터페이스를 사용하려면 MySQL 5.6 이상의 인스턴스를 실행하고 있어야 합니다. 자세한 내용은 [MySQL memcached 지원 \(p. 666\)](#) 단원을 참조하십시오.

MySQL을 위한 외부 스토리지 엔진은 현재 Amazon RDS에서 지원되지 않습니다.

사용자가 생성한 스키마의 경우 MyISAM 스토리지 엔진이 안정적인 복구를 지원하지 않으며 복구 후에 MySQL을 다시 시작할 때 데이터가 손실되거나 손상되어 특정 시점으로 복원 또는 스냅샷 복원 기능이 계획대로 작동하지 않을 수 있습니다. 그래도 Amazon RDS에서 MyISAM을 사용하기로 선택한 경우, 일부 조건에서는 스냅샷이 유용할 수 있습니다.

Note

`mysql` 스키마의 시스템 테이블은 MyISAM 스토리지에 있을 수 있습니다.

기존 MyISAM 테이블을 InnoDB 테이블로 변환하려는 경우 `ALTER TABLE` 명령(예, `alter table TABLE_NAME engine=innodb;`)을 사용하면 됩니다. MyISAM과 InnoDB는 각기 다른 장점과 단점을 갖고 있으므로 전환하기 전에 전환이 현재 애플리케이션에 미치는 영향을 충분히 평가해야 합니다.

MySQL 5.1은 이제 Amazon RDS에서 지원하지 않습니다. 그러나 기존의 MySQL 5.1 스냅샷을 복원할 수 있습니다. MySQL 5.1 스냅샷을 복원할 경우 인스턴스가 자동으로 MySQL 5.5로 업그레이드됩니다.

Amazon RDS MySQL 보안

Amazon RDS MySQL DB 인스턴스용 보안은 세 가지 수준에서 관리됩니다.

- AWS Identity and Access Management은 누가 DB 인스턴스에 대한 Amazon RDS 관리 작업을 수행할 수 있는지 제어합니다. IAM 자격 증명을 사용하여 AWS에 연결할 때, IAM 계정은 Amazon RDS 관리 작업을 수행하는 데 필요한 권한을 부여하는 IAM 정책을 보유하고 있어야 합니다. 자세한 내용은 [인증 및 액세스 제어 \(p. 339\)](#) 단원을 참조하십시오.
- DB 인스턴스를 생성할 때는 VPC 보안 그룹 또는 DB 보안 그룹을 사용하여 어떤 디바이스 및 Amazon EC2 인스턴스가 DB 인스턴스의 앤드포인트 및 포트에 대한 연결을 열 수 있는지를 제어합니다. 이러한 연결은 Secure Socket Layer(SSL)를 이용해 이루어질 수 있습니다. 또한 회사의 방화벽 규칙을 통해 회사에서 실행하는 디바이스가 DB 인스턴스에 대한 연결을 열 수 있는지를 제어할 수 있습니다.
- MySQL DB 인스턴스에 대한 로그인 및 권한을 인증하기 위해서는 다음 접근 방식 중 하나를 따르거나 두 방식을 조합할 수 있습니다.

독립형 MySQL 인스턴스와 동일한 접근법을 사용할 수 있습니다. `CREATE USER`, `RENAME USER`, `GRANT`, `REVOKE` 및 `SET PASSWORD` 등의 명령은 온프레미스 데이터베이스에서 작동하는 것과 마찬가지로 작동하며, 데이터베이스 스키마 테이블을 직접 수정할 때도 동일합니다. 자세한 내용은 MySQL 설명서의 [MySQL 사용자 계정 관리](#) 단원을 참조하십시오.

또한 IAM 데이터베이스 인증을 사용할 수도 있습니다. IAM 데이터베이스 인증의 경우, IAM 사용자 또는 IAM 역할 및 인증 토큰을 이용해 DB 인스턴스에 인증합니다. 인증 토큰은 서명 버전 4 서명 프로세스를 통해 생성하는 고유 값입니다. IAM 데이터베이스 인증을 사용하면 동일한 자격 증명을 사용해 AWS 리소스 및 데이터베이스에 대한 액세스를 제어할 수 있습니다. 자세한 내용은 [MySQL 및 PostgreSQL을 위한 IAM 데이터베이스 인증 \(p. 369\)](#) 단원을 참조하십시오.

Amazon RDS DB 인스턴스를 생성할 때 마스터 사용자는 다음과 같은 기본 권한을 갖습니다.

- `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

DB 인스턴스에서 마스터 사용자를 삭제할 수 있지만, 권장하지는 않습니다. 마스터 사용자를 다시 생성하려면 [ModifyDBInstance](#) RDS API 작업 또는 [modify-db-instance](#) AWS CLI 명령을 사용하고 적절한 파라미터와 함께 새 마스터 사용자 암호를 지정합니다. 마스터 사용자가 인스턴스에 존재하지 않는 경우 마스터 사용자가 지정된 암호와 함께 생성됩니다.

각 DB 인스턴스에 관리 서비스를 제공하기 위해 DB 인스턴스가 생성될 때 `rdsadmin` 사용자가 만들어집니다. `rdsadmin` 계정에 대한 암호를 삭제하거나 이름 바꾸기를 하거나 변경하려고 시도하면 또는 권한을 변경하려고 시도하면 오류가 발생합니다.

DB 인스턴스의 관리를 허용하기 위해 표준 `kill` 및 `kill_query` 명령이 제한되었습니다. DB 인스턴스에서 사용자 세션 또는 쿼리를 종료할 수 있도록 Amazon RDS 명령 `rds_kill` 및 `rds_kill_query`가 제공됩니다.

암호 확인 플러그인 사용

MySQL은 향상된 보안을 위한 `validate_password` 플러그인을 제공합니다. 이 플러그인은 MySQL DB 인스턴스의 DB 파라미터 그룹에 있는 파라미터를 사용하여 암호 정책을 적용합니다. 이 플러그인은 MySQL 버전 5.6, 5.7 및 8.0을 실행하는 DB 인스턴스에 지원됩니다. `validate_password` 플러그인에 대한 자세한 내용은 MySQL 설명서의 [암호 확인 플러그인](#)을 참조하십시오.

MySQL DB 인스턴스에 `validate_password` 플러그인을 활성화하려면

1. MySQL DB 인스턴스에 연결하고 다음 명령을 실행합니다.

```
INSTALL PLUGIN validate_password SONAME 'validate_password.so';
```

2. DB 인스턴스에서 사용하는 DB 파라미터 그룹에 플러그인에 대한 파라미터를 구성하면 됩니다.

파라미터에 대한 자세한 내용은 MySQL 설명서의 [암호 확인 플러그인 옵션 및 변수](#)를 참조하십시오.

DB 인스턴스 파라미터 수정에 대한 자세한 정보는 [DB 파라미터 그룹의 파라미터 수정 \(p. 164\)](#) 단원을 참조하십시오.

`password_validate` 플러그인을 설치하고 활성화한 후 새 확인 정책을 준수하도록 기존 암호를 재설정하십시오.

Amazon RDS는 암호를 확인하지 않습니다. MySQL DB 인스턴스는 암호 확인을 수행합니다. AWS Management 콘솔, `modify-db-instance` AWS CLI 명령 또는 `ModifyDBInstance` RDS API 작업을 사용하여 사용자 암호를 설정하면 새 암호가 암호 정책에 부합하지 않아도 변경 작업이 완료될 수 있습니다. 그러나 새로운 암호는 암호 정책에 부합하는 경우에만 MySQL DB 인스턴스에 설정됩니다. 이 경우, Amazon RDS는 다음 이벤트를 기록합니다.

```
"RDS-EVENT-0067" - An attempt to reset the master password for the DB instance has failed.
```

Amazon RDS 이벤트에 대한 자세한 내용은 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오.

MySQL DB 인스턴스와 함께 SSL 사용

Amazon RDS는 MySQL 데이터베이스 엔진을 실행하는 DB 인스턴스와의 Secure Sockets Layer(SSL) 연결을 지원합니다.

Amazon RDS가 SSL 인증서를 생성한 후 Amazon RDS가 인스턴스를 프로비저닝할 때 DB 인스턴스에 인증서를 설치합니다. 인증 기관이 서명하는 SSL 인증서에는 SSL 인증서에는 스포핑 공격으로부터 보호해주는 SSL 인증서를 위한 일반 이름(CN)으로 DB 인스턴스 엔드포인트가 포함되어 있습니다. 퍼블릭 키는 <https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem>에 저장되어 있습니다.

Amazon RDS에서 생성하는 SSL 인증서는 신뢰할 수 있는 루트 개체이므로 대부분의 경우에 작동하지만 애플리케이션에서 인증서 체인을 수락하지 않을 경우 실패할 수 있습니다. 애플리케이션에서 인증서 체인을 허용하지 않는 경우 중간 인증서를 사용하여 자신의 리전에 연결해야 할 수도 있습니다. 예를 들어 SSL을 사용하여 AWS GovCloud(미국 서부) 리전에 연결하려면 중간 인증서를 사용해야 합니다. 다운로드할 수 있는 리전별 중간 인증서 목록은 [중간 인증서 \(p. 389\)](#) 단원을 참조하십시오.

MySQL은 다음과 같은 버전에서 보안 연결을 위해 yaSSL을 사용합니다.

- MySQL 버전 5.7.19 및 5.7 이전 버전
- MySQL 버전 5.6.37 및 5.6 이전 버전
- MySQL 버전 5.5.57 및 5.5 이전 버전

MySQL은 다음과 같은 버전에서 보안 연결을 위해 OpenSSL을 사용합니다.

- MySQL 버전 8.0
- MySQL 버전 5.7.21 및 5.7 이후 버전
- MySQL 버전 5.6.39 및 5.6 이후 버전
- MySQL 버전 5.5.59 및 5.5 이후 버전

MySQL용 Amazon RDS는 TLS(전송 계층 보안) 버전 1.0, 1.1 및 1.2를 지원합니다. 다음 표는 MySQL 버전에 대한 TLS 지원을 보여줍니다.

MySQL 버전	TLS 1.0	TLS 1.1	TLS 1.2
MySQL 8.0	지원	지원	지원
MySQL 5.7	지원	지원	MySQL 5.7.21 및 이후 버전에서 지원됨
MySQL 5.6	지원	지원되지 않음	지원되지 않음
MySQL 5.5	지원	지원되지 않음	지원되지 않음

기본 mysql 클라이언트를 사용하여 연결을 암호화하려면 --ssl-ca 파라미터를 사용하여 mysql 클라이언트를 시작하고 다음 예제와 같이 퍼블릭 키를 참조하십시오.

다음 예제는 MySQL 5.7 이후 버전의 경우 --ssl-ca 파라미터를 사용하여 클라이언트를 시작하는 방법을 보여줍니다.

```
mysql -h myinstance.c9akciq32.rds-us-east-1.amazonaws.com  
--ssl-ca=[full path]rds-combined-ca-bundle.pem --ssl-mode=VERIFY_IDENTITY
```

다음 예제는 MySQL 5.6 이전 버전의 경우 --ssl-ca 파라미터를 사용하여 클라이언트를 시작하는 방법을 보여줍니다.

```
mysql -h myinstance.c9akciq32.rds-us-east-1.amazonaws.com  
--ssl-ca=[full path]rds-combined-ca-bundle.pem --ssl-verify-server-cert
```

특정 사용자 계정에 대한 SSL 연결을 요구할 수 있습니다. 예를 들면 MySQL 버전에 따라 다음 문 중 하나를 사용하여 사용자 계정 encrypted_user에 대한 SSL 연결을 요구할 수 있습니다.

MySQL 5.7 및 이후 버전에서는 다음 문을 사용하십시오.

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

MySQL 5.6 및 이전 버전에서는 다음 문을 사용하십시오.

```
GRANT USAGE ON *.* TO 'encrypted_user'@'%' REQUIRE SSL;
```

MySQL과의 SSL 연결에 대한 자세한 내용은 MySQL 설명서의 [Using Encrypted Connections](#)를 참조하십시오.

MySQL에서 memcached 및 기타 옵션 사용

대부분의 Amazon RDS DB 엔진은 DB 인스턴스 관련 추가 기능을 선택할 수 있도록 옵션 그룹을 지원합니다. MySQL 5.6 이상의 DB 인스턴스는 단순 키 기반 캐시인 memcached 옵션을 지원합니다. memcached 및 기타 옵션에 대한 자세한 내용은 [MySQL DB 인스턴스 옵션 \(p. 663\)](#)을 참조하십시오. 옵션 그룹 작업에 대한 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오.

InnoDB 캐시 워밍

InnoDB 캐시 워밍은 DB 인스턴스가 종료될 때 버퍼 풀의 현재 상태를 저장한 다음 DB 인스턴스가 시작될 때 저장된 정보에서 버퍼 풀을 다시 로드하여 MySQL DB 인스턴스의 성능 향상을 제공할 수 있습니다. 이렇게 하면 보통 데이터베이스 사용에서 "준비"까지의 버퍼 풀에 대한 필요를 무시하고, 대신 알려진 공용 쿼리에 대한 페이지와 함께 버퍼 풀을 미리 로드합니다. 저장된 버퍼 풀 정보를 저장하는 파일은 페이지 자체가 아니라 단지 버퍼 풀에 있는 페이지에 대한 메타데이터만 저장합니다. 따라서 파일은 많은 저장 공간을 요구하지 않습니다. 파일 크기는 캐시 크기의 약 0.2%입니다. 예를 들면, 64GiB의 경우 캐시 워밍 파일 크기는 128MiB입니다. InnoDB 캐시 워밍에 대한 자세한 내용은 MySQL 설명서의 [버퍼 풀 상태 저장 및 복원](#) 단원을 참조하십시오.

Amazon RDS의 MySQL은 MySQL 버전 5.6 이상에 대한 InnoDB 캐시 워밍을 지원합니다. InnoDB 캐시 워밍을 활성화하려면 `innodb_buffer_pool_dump_at_shutdown` 및 `innodb_buffer_pool_load_at_startup` 파라미터를 DB 인스턴스용 파라미터 그룹에서 1로 설정합니다. 파라미터 그룹에서 이들 파라미터 값을 변경하면 파라미터 그룹을 사용하는 모든 MySQL DB 인스턴스가 영향을 받습니다. 특정 MySQL DB 인스턴스에 대한 InnoDB 캐시 워밍을 활성화하려면, 이들 인스턴스에 대한 새 파라미터 그룹을 생성해야 할 수도 있습니다. 파라미터 그룹에 대한 자세한 내용은 [DB 파라미터 그룹 작업](#) (p. 161) 단원을 참조하십시오.

InnoDB 캐시 워밍은 주로 표준 스토리지를 사용하는 DB 인스턴스를 위해 성능 혜택을 제공합니다. PIOPS 스토리지를 사용하는 경우에는 통상적으로 중요한 성능 혜택이 제공되지 않습니다.

Important

MySQL DB 인스턴스가 정상적으로 종료되지 않는 경우(예: 장애 조치 도중), 버퍼 풀 상태가 디스크에 저장되지 않습니다. 이 경우 MySQL은 DB 인스턴스가 다시 시작될 때 이용 가능한 모든 버퍼 풀 파일을 로드합니다. 어떤 손상도 발생하지 않지만, 복원된 버퍼 풀은 대부분의 경우 다시 시작하기 이전의 버퍼 풀 최신 상태를 반영하지 못할 수도 있습니다. 시작 시 InnoDB 캐시를 워밍업하기 위해 버퍼 풀의 최신 상태를 이용할 수 있게 하려면, "요청 시" 버퍼 풀을 주기적으로 덤프하는 것이 좋습니다. DB 인스턴스가 MySQL 버전 5.6.19 이상을 실행하고 있을 경우 요청 시 버퍼 풀을 덤프하거나 로드할 수 있습니다. 이벤트를 생성하여 버퍼 풀을 자동으로 그리고 정기적으로 덤프할 수 있습니다. 예를 들면, 다음 문은 매 시간마다 버퍼 풀을 덤프하는 이름이 `periodic_buffer_pool_dump`인 이벤트를 생성합니다.

```
CREATE EVENT periodic_buffer_pool_dump
ON SCHEDULE EVERY 1 HOUR
DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

MySQL 이벤트에 대한 자세한 내용은 MySQL 설명서의 [이벤트 구문](#)을 참조하십시오.

요청 시 버퍼 풀 덤팅 및 로딩

MySQL 버전 5.6.19 이상의 경우, "요청 시" InnoDB 캐시를 저장 및 로드할 수 있습니다.

- 버퍼 풀의 현재 상태를 디스크에 덤프하려면 [mysql.rds_innodb_buffer_pool_dump_now](#) (p. 694) 저장 절차를 호출합니다.
- 디스크에서 저장된 버퍼 풀의 상태를 로드하려면 [mysql.rds_innodb_buffer_pool_load_now](#) (p. 695) 저장 절차를 호출합니다.
- 진행 중인 로드 작업을 취소하려면 [mysql.rds_innodb_buffer_pool_load_abort](#) (p. 695) 저장 절차를 호출합니다.

MySQL DB 인스턴스의 현지 시간대

기본적으로 RDS MySQL DB 인스턴스의 시간대는 협정 세계시(UTC)입니다. 대신 DB 인스턴스의 시간대를 애플리케이션의 현지 시간대로 설정할 수 있습니다.

DB 인스턴스의 현지 시간대를 설정하려면 DB 인스턴스의 파라미터 그룹에서 `time_zone` 파라미터를 이 섹션의 뒤에 나오는 지원되는 값 중 하나로 설정합니다. 파라미터 그룹에 대한 `time_zone` 파라미터를 설정하면 해당 파라미터 그룹을 사용 중인 모든 DB 인스턴스와 읽기 전용 복제본이 새로운 현지 시간대를 사용하도록 변경됩니다. 파라미터 그룹에서 파라미터를 설정하는 방법에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

현지 시간대를 설정하면 데이터베이스에 대한 모든 새 연결에 변경 사항이 반영됩니다. 현지 시간대를 변경할 때 데이터베이스에 대해 열린 연결이 있는 경우 연결을 닫고 새 연결을 열어야 현지 시간대 업데이트가 표시됩니다.

DB 인스턴스와 하나 이상의 읽기 전용 복제본에 대해 다른 현지 시간대를 설정할 수 있습니다. 이렇게 하려면 DB 인스턴스와 복제본에 대해 서로 다른 파라미터 그룹을 사용하고 각 파라미터 그룹에서 `time_zone` 파라미터를 다른 현지 시간대로 설정합니다.

리전 간 복제를 사용 중인 경우 복제 마스터 DB 인스턴스와 읽기 전용 복제본이 서로 다른 파라미터 그룹을 사용합니다. 파라미터 그룹은 리전에 고유합니다. 각 인스턴스에 대해 동일한 현지 시간대를 사용하려면 인스턴스의 파라미터 그룹과 읽기 전용 복제본의 파라미터 그룹에서 `time_zone` 파라미터를 설정해야 합니다.

DB 스냅샷에서 DB 인스턴스를 복원할 경우 현지 시간대가 UTC로 설정됩니다. 복원이 완료된 후 시간대를 현지 시간대로 업데이트할 수 있습니다. DB 인스턴스를 특정 시점으로 복원할 경우 복원된 DB 인스턴스의 현지 시간대는 복원된 DB 인스턴스의 파라미터 그룹에서 설정한 시간대입니다.

현지 시간대를 다음 값 중 하나로 설정할 수 있습니다.

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

Amazon RDS MySQL에 대해 알려진 문제 및 제한

Amazon RDS MySQL 사용 시 몇몇 알려진 문제 및 제한이 있습니다. 자세한 내용은 [Amazon RDS MySQL에 대해 알려진 문제 및 제한 \(p. 673\)](#) 단원을 참조하십시오.

MySQL 데이터베이스 엔진 기반 DB 인스턴스 생성

Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. DB 인스턴스란 MySQL 데이터베이스가 생성되는 곳을 말합니다.

Important

DB 인스턴스를 생성하거나 DB 인스턴스에 연결하려면 먼저 [Amazon RDS 설정 \(p. 5\)](#) 섹션의 작업을 완료해야 합니다.

사용자가 샘플 DB 인스턴스를 만들어 연결하는 절차를 실습하는 예제는 [MySQL DB 인스턴스를 만들고 MySQL DB 인스턴스의 데이터베이스에 연결 \(p. 26\)](#) 단원을 참조하십시오.

AWS Management 콘솔

MySQL DB 인스턴스를 시작하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. AWS Management 콘솔의 오른쪽 위 모서리에서 DB 인스턴스를 만들 AWS 리전을 선택합니다.
3. 탐색 창에서 데이터베이스를 선택합니다.

탐색 창이 닫혀 있는 경우 왼쪽 상단의 메뉴 아이콘을 선택하여 여십시오.

4. 데이터베이스 생성을 선택하여 엔진 선택 페이지를 여십시오.

Select engine

Engine options

Amazon Aurora
Amazon Aurora

MySQL


MariaDB


PostgreSQL


Oracle


Microsoft SQL Server


MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 16 TB.
- Instances offer up to 32 vCPUs and 244 GiB Memory.
- Supports automated backup and point-in-time recovery.
- Supports cross-region read replicas.

Only enable options eligible for RDS Free Usage Tier [info](#)

[Cancel](#) **Next**

- [Select engine] 창에서 [MySQL]을 선택한 후 [Next]를 선택합니다.
- [Choose use case] 페이지에 생성하려는 DB 인스턴스를 프로덕션에 사용할지 묻는 메시지가 나타납니다. 사용할 계획이라면 [Production - MySQL]을 선택합니다. 프로덕션 - MySQL을 선택하면 다음 단계에서 다음 설정이 모두 미리 선택됩니다.
 - 다중 AZ 장애 조치 옵션
 - 프로비저닝된 IOPS 스토리지 옵션
 - 삭제 방지 활성화 옵션
- 어떤 프로덕션 환경이든 이 기능을 권장합니다.
- 다음을 선택하여 계속 진행합니다. [Specify DB details] 페이지가 나타납니다.

DB 세부 정보 지정 페이지에서 DB 인스턴스 정보를 지정합니다. 각 설정에 대한 자세한 내용은 [MySQL DB 인스턴스 설정 \(p. 591\)](#) 단원을 참조하십시오.

Specify DB details

Instance specifications

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

DB engine

MySQL Community Edition

License model [Info](#)

general-public-license

DB engine version [Info](#)

MySQL 5.6.40



Known Issues/Limitations

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

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)

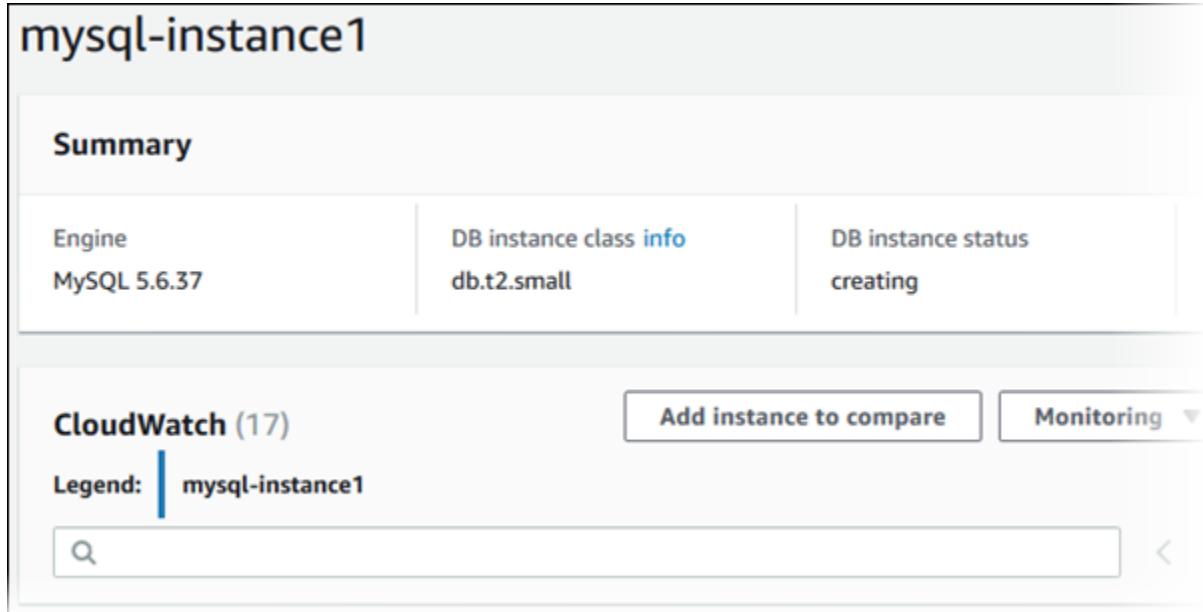
8. 다음을 선택하여 계속 진행합니다. [Configure advanced settings] 페이지가 나타납니다.

Configure Advanced Settings(고급 설정 구성) 페이지에서 Amazon RDS가 DB 인스턴스를 시작하는 데 필요한 추가 정보를 입력합니다. 각 설정에 대한 자세한 내용은 [MySQL DB 인스턴스 설정 \(p. 591\)](#) 단원을 참조하십시오.

9. 데이터베이스 생성을 선택합니다.

10. 마지막 페이지에서 DB 인스턴스 세부 정보 보기를 선택합니다.

RDS 콘솔에 새 DB 인스턴스의 세부 정보가 표시됩니다. DB 인스턴스를 만들고 사용할 준비가 될 때까지 DB 인스턴스의 상태는 [creating]입니다. 상태가 available로 변경되면 DB 인스턴스에 연결할 수 있습니다. DB 인스턴스 클래스와 할당된 스토리지에 따라 새 인스턴스를 사용할 수 있을 때까지 몇 분 정도 걸릴 수 있습니다.



CLI

AWS CLI를 사용해 MySQL DB 인스턴스를 생성하려면 다음 파라미터와 함께 [create-db-instance](#) 명령을 호출합니다. 각 설정에 대한 자세한 내용은 [MySQL DB 인스턴스 설정 \(p. 591\)](#) 단원을 참조하십시오.

- `--db-instance-identifier`
- `--db-instance-class`
- `--db-security-groups`
- `--db-subnet-group`
- `--engine`
- `--master-user-name`
- `--master-user-password`
- `--allocated-storage`
- `--backup-retention-period`

Example

다음은 mydbinstance라는 MySQL DB 인스턴스를 생성하는 예제입니다.

Linux, OS X, Unix의 경우:

```
aws rds create-db-instance \
  --db-instance-identifier mydbinstance \
  --db-instance-class db.m1.small \
  --engine MySQL \
  --allocated-storage 20 \
  --master-username masterawsuser \
  --master-user-password masteruserpassword \
  --backup-retention-period 3
```

Windows의 경우:

```
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 ^
--backup-retention-period 3
```

이 명령은 다음과 유사한 출력을 생성합니다.

```
DBINSTANCE mydbinstance db.m3.medium mysql 20 sa creating 3 **** n 5.6.40
SECGROUP default active
PARAMGRP default.mysql5.6 in-sync
```

API

Amazon RDS API를 사용해 MySQL DB 인스턴스를 생성하려면 다음 파라미터와 함께 [CreateDBInstance](#) 작업을 호출합니다. 각 설정에 대한 자세한 내용은 [MySQL DB 인스턴스 설정 \(p. 591\)](#) 단원을 참조하십시오.

- AllocatedStorage
- BackupRetentionPeriod
- DBInstanceClass
- DBInstanceIdentifier
- DBSecurityGroups
- DBSubnetGroup
- Engine
- MasterUsername
- MasterUserPassword

Example

다음은 mydbinstance라는 MySQL DB 인스턴스를 생성하는 예제입니다.

```
https://rds.us-west-2.amazonaws.com/
?Action=CreateDBInstance
&AllocatedStorage=20
&BackupRetentionPeriod=3
&DBInstanceClass=db.m3.medium
&DBInstanceIdentifier=mydbinstance
&DBName=mydatabase
&DBSecurityGroups.member.1=mysecuritygroup
&DBSubnetGroup=mydbsubnetgroup
&Engine=mysql
&MasterUserPassword=masteruserpassword
&MasterUsername=masterawsuser
&Version=2014-10-31
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140213/us-west-2/rds/aws4_request
&X-Amz-Date=20140213T162136Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=8052a76dfb18469393c5f0182cdab0ebc224a9c7c5c949155376c1c250fc7ec3
```

MySQL DB 인스턴스 설정

다음 표에는 MySQL DB 인스턴스를 생성할 때 선택하는 설정에 대한 세부 정보가 나와 있습니다.

설정	설정 설명
할당된 스토리지	DB 인스턴스용으로 할당할 스토리지(단위: GB)입니다. 경우에 따라 DB 인스턴스에 대해 데이터베이스의 크기보다 많은 양의 스토리지를 할당하면 I/O 성능을 개선할 수 있습니다. 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.
마이너 버전 자동 업그레이드	기본 마이너 DB 엔진 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다. Amazon RDS는 유지 관리 기간에 마이너 버전 자동 업그레이드를 수행합니다.
[Availability zone]	DB 인스턴스의 가용 영역입니다. 가용 영역을 지정하지 않으려면 기본값으로 No Preference를 사용합니다. 자세한 내용은 리전 및 가용 영역 (p. 99) 단원을 참조하십시오.
백업 보존 기간	DB 인스턴스의 자동 백업을 보존할 기간(단위: 일)입니다. 중요한 DB 인스턴스의 경우 이 값을 1 이상으로 설정해야 합니다. 자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.
백업 기간	Amazon RDS가 자동으로 DB 인스턴스를 백업하는 기간입니다. 데이터베이스를 백업할 특정 시간을 지정하지 않으려면 기본값으로 No Preference를 사용합니다. 자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.
스냅샷으로 태그 복사	스냅샷을 생성할 때 DB 인스턴스 태그를 DB 스냅샷에 복사하려면 이 옵션을 선택합니다. 자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.
데이터베이스 이름	DB 인스턴스에 있는 데이터베이스의 이름입니다. 데이터베이스 이름은 1~64자의 영숫자 문자로 구성되어야 합니다. 이름을 제공하지 않으면 생성 중인 DB 인스턴스에 Amazon RDS가 데이터베이스를 생성하지 않습니다. DB 인스턴스에서 추가 데이터베이스를 생성하려면, DB 인스턴스에 연결한 다음 SQL 명령 CREATE DATABASE를 사용하십시오. 자세한 내용은 MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 (p. 595) 단원을 참조하십시오.
데이터베이스 포트	DB 인스턴스에 액세스하는 데 사용할 포트입니다. MySQL 설치의 기본값은 포트 3306으로 설정됩니다. DB 보안 그룹을 DB 인스턴스에 사용할 경우 DB 보안 그룹을 생성할 때 제공한 포트 값과 같아야 합니다. 일부 기업에서는 방화벽이 기본 MySQL 포트 연결을 차단하는 경우도 있습니다. 이처럼 기업 방화벽이 기본 포트를 차단할 경우 DB 인스턴스에 다른 포트를 선택해야 합니다.
DB 엔진 버전	사용할 MySQL 버전입니다.

설정	설정 설명
DB 인스턴스 클래스	<p>DB 인스턴스에 대한 구성입니다. 예를 들어, [db.m1.small] 인스턴스 클래스는 1.7GiB 메모리, 1 ECU(1 ECU의 1 가상 코어), 64비트 플랫폼 및 보통 I/O 용량과 동일합니다.</p> <p>가능하면 일반 쿼리 작업 세트가 메모리에 상주할 수 있을 정도로 큰 인스턴스 클래스를 선택합니다. 작업 세트가 메모리에 상주할 경우 시스템의 디스크 쓰기가 불필요하여 성능이 향상됩니다.</p> <p>자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 단원을 참조하십시오.</p>
DB 인스턴스 식별자	<p>DB 인스턴스의 이름입니다. DB 인스턴스 식별자는 최대 63자 의 영숫자 문자를 포함할 수 있으며 선택한 AWS 리전의 계정에 대해 고유해야 합니다. 예를 들어 <code>mysql-instance1</code>와 같이 선택한 AWS 리전을 포함하는 등의 방식으로 알아보기 쉬운 이름을 지정할 수 있습니다.</p>
DB 파라미터 그룹	<p>DB 인스턴스의 파라미터 그룹입니다. 기본 파라미터 그룹을 사용하거나 사용자 지정 파라미터 그룹을 생성할 수 있습니다.</p> <p>자세한 내용은 DB 파라미터 그룹 작업 (p. 161) 단원을 참조하십시오.</p>
삭제 방지	<p>DB 인스턴스가 삭제되지 않도록 방지하려면, 삭제 방지를 활성화합니다. AWS Management 콘솔을 사용하여 프로덕션 DB 인스턴스를 생성할 경우 기본적으로 삭제 방지가 활성화됩니다. 자세한 내용은 DB 인스턴스 삭제 (p. 128) 단원을 참조하십시오.</p>
암호화	<p>이 DB 인스턴스에 대해 유튜 암호화를 활성화하기 위한 [Enable Encryption].</p> <p>자세한 내용은 Amazon RDS 리소스 암호화 (p. 385) 단원을 참조하십시오.</p>
확장 모니터링	<p>DB 인스턴스가 실행되는 운영 체제에 대한 실시간 측정치를 수집하려면 [Enable enhanced monitoring]을 선택합니다.</p> <p>자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.</p>
IAM DB 인증]	<p>이 DB 인스턴스에 대한 IAM 데이터베이스 인증을 활성화하기 위한 [Enable IAM DB authentication].</p> <p>자세한 내용은 MySQL 및 PostgreSQL을 위한 IAM 데이터베이스 인증 (p. 369) 단원을 참조하십시오.</p>
라이선스 모델	<p>MySQL에는 general-public-license MySQL에 대한 일반 라이선스 계약이라는 하나의 라이선스 모델만 있습니다.</p>
로그 내보내기	<p>생성할 MySQL 데이터베이스 로그 파일의 유형을 선택합니다.</p> <p>자세한 내용은 MySQL 데이터베이스 로그 파일 (p. 318) 단원을 참조하십시오.</p>

설정	설정 설명
유지 관리 기간	<p>대기 중인 DB 인스턴스 설정 변경이 적용되기 위해 경과해야 하는 기간(30분)입니다. 이 시간이 중요하지 않은 경우 [No Preference]를 선택합니다.</p> <p>자세한 내용은 Amazon RDS 유지 관리 기간 (p. 116) 단원을 참조하십시오.</p>
Master password	마스터 사용자 계정의 암호입니다. 암호는 8~16자 인쇄 가능한 ASCII 문자를 포함해야 합니다(/, ", 공백 및 @ 제외).
Master username	<p>DB 인스턴스에 로그인할 때 사용하는 마스터 사용자 이름입니다.</p> <p>자세한 내용과 마스터 사용자의 기본 권한 목록은 Amazon RDS MySQL 보안 (p. 581) 단원을 참조하십시오.</p>
다중 AZ 배포	<p>장애 조치를 위해 다른 가용 영역에 DB 인스턴스의 수동 보조 복제본을 생성하려면 다른 영역에 복제본 만들기를 선택합니다. 이때 고가용성을 유지하려면 프로덕션 워크로드를 위한 다중 AZ를 권장합니다. 개발 및 테스팅을 위해 No를 선택할 수 있습니다.</p> <p>자세한 내용은 Amazon RDS를 위한 고가용성(다중 AZ) (p. 109) 단원을 참조하십시오.</p>
옵션 그룹	<p>DB 인스턴스의 옵션 그룹입니다. 기본 옵션 그룹을 사용하거나 사용자 지정 옵션 그룹을 생성할 수 있습니다.</p> <p>자세한 내용은 옵션 그룹 작업 (p. 148) 단원을 참조하십시오.</p>
퍼블릭 액세스 가능성	<p>DB 인스턴스에 퍼블릭 IP 주소를 부여하려면 [Yes]를 선택합니다. 그러면 VPC 외부에서 DB 인스턴스에 액세스할 수 있습니다(DB 인스턴스도 VPC의 퍼블릭 서브넷에 있어야 함). VPC 내부에서만 DB 인스턴스에 액세스할 수 있게 하려면 [No]를 선택합니다.</p> <p>자세한 내용은 VPC에 있는 DB 인스턴스를 인터넷에서 습기 기 (p. 416) 단원을 참조하십시오.</p>
스토리지 유형	<p>DB 인스턴스의 스토리지 유형입니다.</p> <p>자세한 내용은 Amazon RDS 스토리지 유형 (p. 101) 단원을 참조하십시오.</p>
Subnet Group	이 설정은 현재 사용 중인 플랫폼에 따라 다릅니다. AWS를 처음 사용하는 고객은 [default]를 선택합니다. 이 DB 서브넷 그룹은 계정에 대해 생성된 기본 DB 서브넷 그룹입니다. 이전 E2-Classic 플랫폼에서 DB 인스턴스를 만들어서 특정 VPC에 배포 하려면 해당 VPC로 생성된 DB 서브넷 그룹을 선택합니다.
Virtual Private Cloud(VPC)	이 설정은 현재 사용 중인 플랫폼에 따라 다릅니다. AWS를 처음 사용하는 경우에는 기본 VPC를 선택합니다. 이전 E2-Classic 플랫폼에서 VPC를 사용하지 않는 DB 인스턴스를 만드는 경우에는 [Not in VPC]를 선택합니다.
	자세한 내용은 Amazon Virtual Private Cloud(VPC) 및 Amazon RDS (p. 406) 단원을 참조하십시오.

설정	설정 설명
VPC 보안 그룹	AWS를 처음 사용하는 경우에는 [Create new VPC security group]을 선택합니다. 처음 사용하는 경우가 아니라면 [Select existing VPC security groups]를 선택하고 앞서 생성한 보안 그룹을 선택합니다. RDS 콘솔에서 새 VPC 보안 그룹 생성을 선택하는 경우 브라우저에서 검색된 IP 주소에서 DB 인스턴스에 액세스하도록 허용하는 발신 규칙을 사용하여 새 보안 그룹이 생성됩니다. 자세한 내용은 DB 보안 그룹 작업(EC2-Classic 플랫폼) (p. 394) 단원을 참조하십시오.

관련 주제

- 자습서: [Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 \(p. 421\)](#)
- [MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 595\)](#)
- [MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 600\)](#)
- [DB 인스턴스 삭제 \(p. 128\)](#)

MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기

MySQL 데이터베이스 엔진을 실행하는 DB 인스턴스에 연결하려면 우선 DB 인스턴스를 생성해야 합니다. 자세한 내용은 [MySQL 데이터베이스 엔진 기반 DB 인스턴스 생성 \(p. 587\)](#) 단원을 참조하십시오. Amazon RDS가 DB 인스턴스를 프로비저닝한 후에는 표준 MySQL 클라이언트 애플리케이션 또는 유ти리티를 사용하여 인스턴스에 연결할 수 있습니다. 연결 문자열에는 호스트 파라미터로 DB 인스턴스 엔드포인트의 DNS 주소와 포트 파라미터로 DB 인스턴스 엔드포인트의 포트 번호를 지정합니다.

RDS DB 인스턴스에 인증하려면, MySQL과 IAM 데이터베이스 인증의 인증 방법 중 하나를 사용할 수 있습니다.

- MySQL의 인증 방법 중 하나를 사용하는 MySQL 인증 방법을 확인하려면, MySQL의 [인증 방법](#)을 참조하십시오.
- IAM 데이터베이스 인증을 사용하는 MySQL 인증 방법을 확인하려면 [MySQL 및 PostgreSQL을 위한 IAM 데이터베이스 인증 \(p. 369\)](#) 단원을 참조하십시오.

AWS Management 콘솔, AWS CLI `rds-describe-db-instances` 명령 또는 Amazon RDS API `DescribeDBInstances` 작업을 사용하여 엔드포인트를 포함한 Amazon RDS DB 인스턴스의 세부 정보를 나열할 수 있습니다.

AWS Management 콘솔에서 MySQL DB 인스턴스의 엔드포인트를 찾는 방법:

1. RDS 콘솔을 연 다음 데이터베이스를 선택해 DB 인스턴스의 목록을 표시합니다.
2. 세부 정보를 표시하고자 하는 MySQL DB 인스턴스 이름을 선택합니다.
3. 연결 탭에서 엔드포인트를 복사합니다. 또한 포트 번호를 적어둡니다. DB 인스턴스에 연결하려면 엔드포인트와 포트 번호가 모두 필요합니다.

mysql-instance1

Summary

DB Name	CPU
mysql-instance1	
Role	Curren
Master	

Connectivity Monitoring Logs & events Configuration Maintenance

Connectivity

Endpoint & port

Endpoint
mysql-instance1 [REDACTED] rds.amazonaws.com

Port
3306

엔드포인트 값이 mysql-instance1.123456789012.us-east-1.rds.amazonaws.com이고 포트 값이 3306이라면 MySQL 연결 문자열에 다음과 같이 값을 지정합니다.

- 호스트 또는 호스트 이름은 mysql-instance1.123456789012.us-east-1.rds.amazonaws.com으로 지정합니다.
- 포트는 3306으로 지정합니다.

Amazon RDS MySQL DB 인스턴스는 MySQL 명령줄 유ти리티 같은 도구를 사용하여 연결할 수 있습니다. MySQL 클라이언트 사용에 대한 자세한 내용은 MySQL 문서의 [mysql - The MySQL 명령줄 도구](#) 단원을 참조하십시오. 연결에 사용할 수 있는 GUI 기반 애플리케이션 중 하나는 MySQL Workbench입니다. 자세한 내용은 [MySQL Workbench 다운로드](#) 페이지 단원을 참조하십시오. MySQL 설치(MySQL 클라이언트 포함)에 대한 정보는 [MySQL 설치 및 업그레이드](#)를 참조하십시오.

새로운 DB 인스턴스의 연결 오류가 발생하는 두 가지 공통 원인은 다음과 같습니다.

- 보안 그룹을 사용하여 DB 인스턴스를 만들었지만, 이 보안 그룹이 MySQL 애플리케이션 또는 유ти리티를 실행 중인 디바이스나 Amazon EC2 인스턴스에서 연결할 수 있는 권한을 부여하지 않은 경우. DB 인스턴스가 VPC 내부에서 생성된 경우 이 인스턴스의 VPC 보안 그룹에서 연결 권한을 부여해야 합니다. DB 인스턴스가 VPC 외부에서 생성된 경우 이 인스턴스의 DB 보안 그룹에서 연결 권한을 부여해야 합니다. 자세한 내용은 [Amazon Virtual Private Cloud\(VPC\)](#) 및 [Amazon RDS \(p. 406\)](#) 단원을 참조하십시오.
- 포트 3306을 사용해 DB 인스턴스를 만들었는데 기업 방화벽 규칙에 따라 기업 네트워크의 디바이스에서 해당 포트에 연결하는 것이 차단된 경우. 이 오류를 수정하려면 인스턴스를 다른 포트로 다시 만들어야 합니다.

Amazon RDS MySQL DB 인스턴스에 연결할 때는 SSL 암호화를 사용할 수 있습니다. 자세한 내용은 [MySQL DB 인스턴스와 함께 SSL 사용 \(p. 583\)](#) 단원을 참조하십시오. IAM 데이터베이스 인증을 사용하고 있다면, SSL 연결을 사용해야 합니다. 자세한 내용은 [MySQL 및 PostgreSQL을 위한 IAM 데이터베이스 인증 \(p. 369\)](#) 단원을 참조하십시오.

SSL을 사용한 MariaDB DB 인스턴스 연결에 대한 자세한 내용은 [MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 \(p. 445\)](#) 단원을 참조하십시오.

MySQL 클라이언트를 통해 연결하기

MySQL 클라이언트를 사용하여 DB 인스턴스에 연결하려면 명령 프롬프트에 다음 명령을 입력하여 MySQL 클라이언트를 사용하여 DB 클러스터에 연결합니다. -h 파라미터의 경우 해당 DB 인스턴스의 DNS 이름(엔드포인트)으로 대체합니다. -p 파라미터에는 DB 인스턴스의 포트로 대체합니다. 입력 프롬프트가 표시되면 마스터 사용자 암호를 입력합니다.

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com -P 3306 -u mymasteruser -p
```

사용자에 대한 암호를 입력하면 다음과 유사한 출력이 나타납니다.

```
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>
```

SSL에 인스턴스 연결

Amazon RDS는 DB 인스턴스가 생성될 때 인스턴스의 SSL 인증서를 만듭니다. SSL 인증서 확인을 활성화하는 경우에는 SSL 인증서에 스푸핑 공격으로부터 보호해주는 SSL 인증서를 위한 일반 이름(CN)으로 DB 인스턴스 엔드포인트가 포함됩니다. SSL을 사용하여 DB 인스턴스에 연결하면, 기본 암호 인증 또는 IAM 데이터베이스 인증을 사용할 수 있습니다. IAM 데이터베이스 인증을 사용하여 DB 인스턴스에 연결하려면 [MySQL 및 PostgreSQL을 위한 IAM 데이터베이스 인증 \(p. 369\)](#) 단원을 참조하십시오. 기본 암호 인증을 사용하여 DB 인스턴스에 연결하려면 다음의 단계를 따릅니다.

MySQL 클라이언트를 사용하여 SSL에 DB 인스턴스 연결하는 방법

- 모든 리전에 적용되는 루트 인증서는 [여기](#)에서 다운로드할 수 있습니다.
- MySQL 클라이언트를 사용하여 SSL이 포함된 DB 인스턴스에 연결하려면 명령 프롬프트에 다음 명령을 입력합니다. -h 파라미터에는 DB 인스턴스의 DNS 이름으로 대체합니다. --ssl-ca 파라미터에는 해당하는 SSL 인증서 파일 이름으로 대체합니다.

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=rds-ca-2015-root.pem -p
```

- SSL 연결에서 SSL 인증서의 엔드포인트와 비교하여 DB 인스턴스 엔드포인트를 확인하도록 요구할 수 있습니다.

MySQL 5.7 이상인 경우:

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=rds-ca-2015-root.pem --ssl-mode=VERIFY_IDENTITY -p
```

MySQL 5.6 이전인 경우:

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=rds-ca-2015-root.pem --ssl-verify-server-cert -p
```

- 입력 프롬프트가 표시되면 마스터 사용자 암호를 입력합니다.

출력은 다음과 비슷합니다.

```
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>
```

MySQL Workbench에서 연결

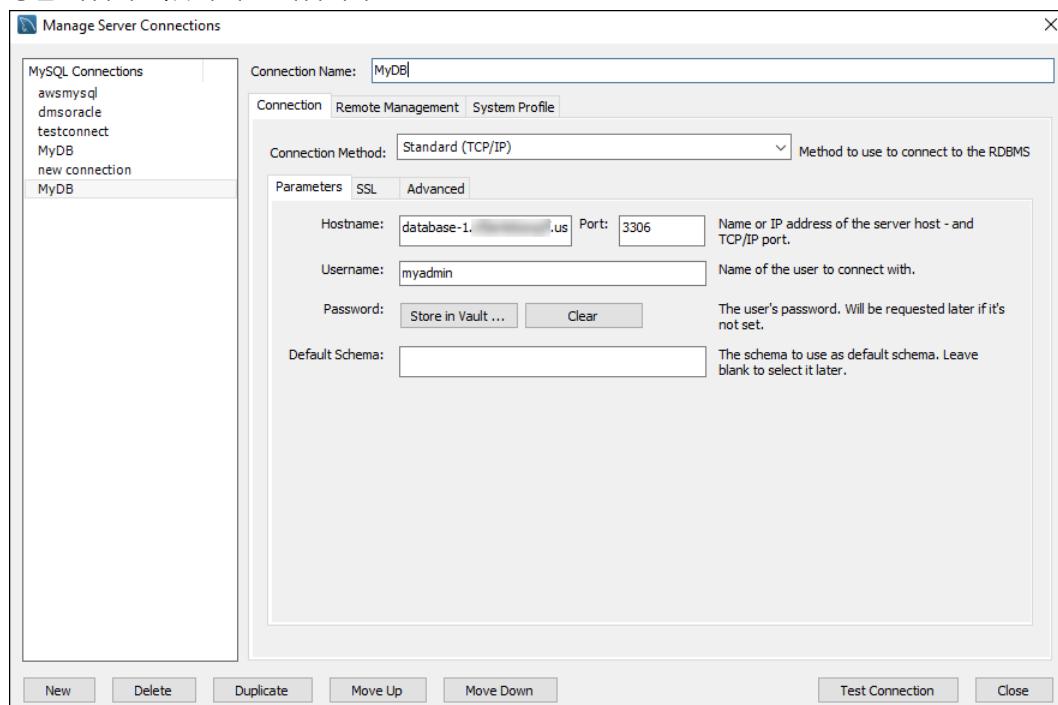
MySQL Workbench에서 연결하려면

- MySQL Workbench 다운로드에서 MySQL Workbench를 다운로드하고 설치합니다.
- MySQL Workbench를 엽니다.



3. 데이터베이스에서 Manage Connections(연결 관리)를 선택합니다.
4. Manage Server Connections(서버 연결 관리) 창에서 새로 생성을 선택합니다.
5. Connect to Database(데이터베이스에 연결) 창에 다음 정보를 입력합니다.
 - Stored Connection(저장된 연결) – 연결에 MyDB와 같은 이름을 입력합니다.
 - Hostname(호스트 이름) – DB 인스턴스 엔드포인트를 입력합니다.
 - 포트 – DB 인스턴스에서 사용한 포트를 입력합니다.
 - 사용자 이름 – 마스터 사용자와 같이 유효한 데이터베이스 사용자의 사용자 이름을 입력합니다.
 - 암호 – 선택적으로 Store in Vault(볼트에 저장)를 선택한 후 사용자의 암호를 입력하고 저장합니다.

창은 다음과 비슷하게 표시됩니다.



6. 선택적으로 연결 테스트를 선택하여 DB 인스턴스에 대한 연결이 성공적인지 확인합니다.
7. 닫기를 선택합니다.
8. 데이터베이스에서 Connect to Database(데이터베이스에 연결)를 선택합니다.
9. Stored Connection(저장된 연결)에서 연결을 선택합니다.
10. 확인을 선택합니다.

MySQL Workbench 사용에 대한 자세한 내용은 [MySQL Workbench 설명서](#)를 참조하십시오.

최대 MySQL 연결

Amazon RDS MySQL DB 인스턴스에 허용되는 최대 연결 수는 DB 인스턴스의 DB 인스턴스 클래스에 사용 가능한 메모리의 양에 따라 결정됩니다. 사용 가능한 메모리가 많은 DB 인스턴스 클래스는 가능한 연결 수가 더 많아집니다. DB 인스턴스 클래스에 대한 자세한 내용은 다음([DB 인스턴스 클래스 선택 \(p. 77\)](#)) 단원을 참조하십시오.

DB 인스턴스의 연결 제한은 기본적으로 DB 인스턴스의 DB 인스턴스 클래스 최대값으로 설정됩니다. DB 인스턴스의 파라미터 그룹에서 max_connections 파라미터를 사용하여 동시 연결 수를 허용되는 최대 연결 수까지로 제한할 수 있습니다. 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

Amazon RDS MySQL DB 인스턴스에 허용되는 최대 연결 수는 해당 DB 인스턴스에 대해 다음 쿼리를 실행하여 확인할 수 있습니다.

```
SELECT @@max_connections;
```

DB 인스턴스에 대해 다음 쿼리를 실행하여 Amazon RDS MySQL DB 인스턴스에 대한 활성 연결 수를 확인할 수 있습니다.

```
SHOW STATUS WHERE `variable_name` = 'Threads_connected';
```

MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경

추가 스토리지를 더하거나 DB 인스턴스 클래스를 변경하는 것과 같은 작업을 완수하기 위해 DB 인스턴스의 설정을 변경할 수 있습니다. 이 주제에서는 Amazon RDS MySQL DB 인스턴스를 변경하는 과정을 안내하고, MySQL 인스턴스에 대한 설정을 설명합니다.

프로덕션 인스턴스를 수정하기 전에 테스트 인스턴스에서 변경 사항을 테스트하면 각 변경 사항이 미칠 영향을 완전히 이해하는 데 도움이 됩니다. 이는 특히 데이터베이스 버전을 업그레이드할 때 중요합니다.

DB 인스턴스 설정을 수정한 후 변경 사항을 즉시 적용할 수도 있고, DB 인스턴스에 대한 다음 유지 관리 기간에 적용할 수도 있습니다. 일부 변경 사항의 경우 DB 인스턴스가 재시작될 수 있습니다.

Note

DB 인스턴스를 수정할 때 다음이 모두 맞는 경우 Amazon RDS가 인스턴스를 재부팅합니다.

- DB 인스턴스 클래스를 변경합니다.
- 사용자 지정 파라미터 그룹을 지정합니다.

AWS Management 콘솔

MySQL DB 인스턴스를 변경하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 변경하려는 DB 인스턴스를 선택합니다.
3. [Modify]를 선택합니다. [Modify DB Instance] 페이지가 나타납니다.
4. 원하는 설정을 모두 변경합니다. 각 설정에 대한 자세한 내용은 [MySQL DB 인스턴스 설정 \(p. 602\)](#) 단원을 참조하십시오.
5. 원하는 대로 모두 변경되었으면 [Continue]를 선택하고 수정 사항 요약을 확인합니다.
6. 변경 사항을 즉시 적용하려면 즉시 적용을 선택합니다. 일부의 경우 이 옵션을 선택하면 중단이 발생할 수 있습니다. 자세한 내용은 [즉시 적용 파라미터 사용 \(p. 112\)](#) 단원을 참조하십시오.
7. 확인 페이지에서 변경 내용을 검토합니다. 변경 내용이 정확할 경우 [Modify DB Instance]를 선택하여 변경 내용을 저장합니다.

그렇지 않으면 [Back]을 선택하여 변경 내용을 편집하거나 [Cancel]을 선택하여 변경 내용을 취소합니다.

CLI

AWS CLI를 사용해 MySQL DB 인스턴스를 수정하려면 [modify-db-instance](#) 명령을 호출합니다. DB 인스턴스 식별자와 수정하려는 설정의 파라미터를 지정합니다. 각 파라미터에 대한 자세한 내용은 [MySQL DB 인스턴스 설정 \(p. 602\)](#) 단원을 참조하십시오.

Example

다음 코드는 백업 보존 기간을 1주(7일)로 설정하여 mydbinstance를 수정합니다. 이 코드는 `--auto-minor-version-upgrade`를 사용하여 마이너 버전 자동 업그레이드를 활성화합니다. 마이너 버전 자동 업그레이드를 비활성화하려면 `--no-auto-minor-version-upgrade`를 사용합니다. 변경 사항은 `--no-apply-immediately`를 사용하여 다음 유지 관리 기간에 적용됩니다. 변경 사항을 바로 적용하려면 `--apply-immediately`를 사용합니다. 자세한 내용은 [즉시 적용 파라미터 사용 \(p. 112\)](#) 단원을 참조하십시오.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
  --db-instance-identifier mydbinstance \
  --backup-retention-period 7 \
  --auto-minor-version-upgrade \
  --no-apply-immediately
```

Windows의 경우:

```
aws rds modify-db-instance ^
  --db-instance-identifier mydbinstance ^
  --backup-retention-period 7 ^
  --auto-minor-version-upgrade ^
  --no-apply-immediately
```

API

Amazon RDS API를 사용해 MySQL 인스턴스를 수정하려면 [ModifyDBInstance](#) 작업을 호출합니다. DB 인스턴스 식별자와 수정하려는 설정의 파라미터를 지정합니다. 각 파라미터에 대한 자세한 내용은 [MySQL DB 인스턴스 설정 \(p. 602\)](#) 단원을 참조하십시오.

Example

다음은 백업 보존 기간을 1주일(7일)로 설정하고 마이너 버전 자동 업그레이드를 활성화하여 mydbinstance를 수정하는 코드입니다. 이 변경 사항은 다음 유지 관리 기간에 적용됩니다.

```
https://rds.amazonaws.com/
?Action=ModifyDBInstance
&ApplyImmediately=false
&AutoMinorVersionUpgrade=true
&BackupRetentionPeriod=7
&DBInstanceIdentifier=mydbinstance
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-10-31
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20131016/us-west-1/rds/aws4_request
&X-Amz-Date=20131016T233051Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=087a8eb41cb1ab0fc9ec1575f23e73757ffc6a1e42d7d2b30b9cc0be988cff97
```

MySQL DB 인스턴스 설정

다음 표에는 수정이 가능한 설정, 수정이 불가능한 설정, 변경 사항을 적용할 수 있는 시점, 변경 사항으로 인한 DB 인스턴스 가동 중지 여부에 대한 자세한 내용이 나와 있습니다.

설정	설정 설명	변경이 발생할 때	가동 중지 참고 사항
활당된 스토리지	DB 인스턴스에 활당할 스토리지(단위: GB). 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.	[Apply immediately]가 true로 설정되는 즉시 변경됩니다. [Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.	가동 중지 없음. 변경 도중 성능이 저하될 수 있습니다.
마이너 버전 자동 업그레이드	기본 마이너 DB 엔진 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다. Amazon RDS는 유지 관리 기간에 마이너 버전 자동 업그레이드를 수행합니다.	-	-
백업 보존 기간	자동 백업을 보존할 일수. 자동 백업을 비활성화하기 위해 백업 보존 기간을 0으로 설정합니다. 자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.	[Apply immediately]가 true로 설정되는 즉시 변경됩니다. [Apply immediately]가 false로 설정되고 이 설정을 0이 아닌 값에서 다른 0이 아닌 값으로 변경하면 비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이때 적용되지 않을 경우, 다음 유지 관리 기간에 변경 사항이 적용됩니다.	백업 보존 기간을 0에서 0이 아닌 값으로 또는 0이 아닌 값에서 0으로 변경할 경우 인스턴스가 종단됩니다.
백업 기간	데이터베이스의 자동 백업이 실행되는 기간. 백업 기간은 국제 표준시(UTC)의 시작 시간과 시간 단위의 지속 기간으로 구성됩니다. 자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다.	-
인증 기관	사용할 인증서.	-	-
스냅샷으로 태그 복사	DB 인스턴스 태그가 있는 경우 이 옵션은 DB 스냅샷을 만들 때 태그를 복사합니다. 자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
데이터베이스 포트	<p>데이터베이스에 액세스하는 데 사용할 포트.</p> <p>포트 값은 DB 인스턴스의 옵션 그룹에서 옵션에 대해 지정한 포트 값과 일치하지 않아야 합니다.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	DB 인스턴스가 즉시 재부팅됩니다.
DB 엔진 버전	<p>사용할 MySQL 데이터베이스 엔진의 버전. 프로덕션 DB 인스턴스를 업그레이드 하려면 먼저 테스트 인스턴스에서 업그레이드 프로세스를 테스트하여 업그레이드 시간을 확인하고 애플리케이션을 검증하는 것이 좋습니다.</p> <p>자세한 내용은 MySQL DB 엔진 업그레이드 (p. 607) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다.
DB 인스턴스 클래스	<p>사용할 DB 인스턴스 클래스.</p> <p>자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 단원을 참조하십시오.</p> <p>2014년 4월 23일 이후에 생성된 MySQL DB 인스턴스는 다른 수정과 마찬가지로 해당 DB 인스턴스를 수정하여 다른 DB 인스턴스 클래스로 변경할 수 있습니다. MySQL 버전 5.5를 실행하며 2014년 4월 23일 이전에 생성된 MySQL DB 인스턴스는 먼저 MySQL 버전 5.6으로 업그레이드해야 합니다. 자세한 내용은 MySQL DB 엔진 업그레이드 (p. 607) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다.
DB 인스턴스 식별자	<p>DB 인스턴스 식별자 이 값은 소문자 문자열로 저장됩니다.</p> <p>DB 인스턴스 이름 바꾸기의 영향에 대한 자세한 내용은 DB 인스턴스 이름 바꾸기 (p. 121) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다. DB 인스턴스가 재부팅됩니다.

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
DB 파라미터 그룹	DB 인스턴스와 연결할 파라미터 그룹. 자세한 내용은 DB 파라미터 그룹 작업 (p. 161) 단원을 참조하십시오.	파라미터 그룹 변경 사항은 즉시 적용됩니다.	이 변경 도중 인스턴스가 중단되지 않습니다. 파라미터 그룹을 변경하는 경우 일부 파라미터에 대한 변경 내용은 재부팅 없이 DB 인스턴스에 즉시 적용됩니다. 다른 파라미터에 대한 변경 내용은 DB 인스턴스를 재부팅한 후에만 적용됩니다. 자세한 내용은 DB 클러스터에서 DB 인스턴스 (p. 123) 단원을 참조하십시오.
삭제 방지	DB 인스턴스가 삭제되지 않도록 방지하려면, 삭제 방지를 활성화합니다. 자세한 내용은 DB 인스턴스 삭제 (p. 128) 단원을 참조하십시오.	-	-
확장 모니터링	DB 인스턴스가 실행되는 운영 체제에 대한 실시간 측정치 수집을 활성화하려면 [Enable enhanced monitoring]을 선택합니다. 자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.	-	-
IAM DB 인증	이 DB 인스턴스에 대한 IAM 데이터베이스 인증을 활성화하기 위한 [Enable IAM DB authentication]. 자세한 내용은 MySQL 및 PostgreSQL을 위한 IAM 데이터베이스 인증 (p. 369) 단원을 참조하십시오.	-	-
로그 내보내기	Amazon CloudWatch Logs에 게시할 MySQL 데이터베이스 로그 파일의 유형을 선택합니다. 자세한 내용은 MySQL 데이터베이스 로그 파일 (p. 318) 단원을 참조하십시오.	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
유지 관리 기간	<p>시스템 유지 관리를 실행하는 기간. 시스템 유지 관리는 업그레이드를 포함합니다(해당할 경우). 유지 관리 기간은 국제 표준시(UTC)의 시작 시간과 시간 단위의 지속 기간으로 구성됩니다.</p> <p>이 기간을 현재 시간으로 설정하려면 대기 중인 변경 사항이 모두 적용될 수 있도록 현재 시간과 기간 종료 시간 사이에 최소 30분 이상 필요합니다.</p> <p>자세한 내용은 Amazon RDS 유지 관리 기간 (p. 116) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	인스턴스가 종단될 수 있는 작업이 하나 이상 대기 중이고, 유지 관리 기간이 현재 시간을 포함하여 변경된 경우 대기 중인 작업들이 즉시 적용되고 인스턴스가 종단됩니다.
다중 AZ 배포	<p>다중 가용 영역에 DB 인스턴스를 배포하려면 [Yes], 그렇지 않으면 [No].</p> <p>자세한 내용은 리전 및 가용 영역 (p. 99) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	-
새 마스터 암호	마스터 사용자의 암호. 암호에는 8~41자의 영숫자 문자로 구성되어야 합니다.	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-
옵션 그룹	<p>DB 인스턴스와 연결할 옵션 그룹.</p> <p>자세한 내용은 옵션 그룹 작업 (p. 148) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	-
퍼블릭 액세스 가능성	<p>DB 인스턴스에 퍼블릭 IP 주소를 부여하려면(즉 VPC 외부에서 액세스할 수 있음) [Yes]를 선택합니다. 공개적으로 액세스가 가능하려면 DB 인스턴스도 VPC의 퍼블릭 서브넷에 있어야 합니다. VPC 내부에서만 DB 인스턴스에 액세스할 수 있게 하려면 [No].</p> <p>자세한 내용은 VPC에 있는 DB 인스턴스를 인터넷에서 습기기 (p. 416) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
보안 그룹	DB 인스턴스와 연결할 보안 그룹. 자세한 내용은 DB 보안 그룹 작업 (EC2-Classic 플랫폼) (p. 394) 단원을 참조하십시오.	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-
스토리지 유형	사용할 스토리지 유형. 자세한 내용은 Amazon RDS 스토리지 유형 (p. 101) 단원을 참조하십시오.	[Apply immediately]가 true로 설정되는 즉시 변경됩니다. [Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.	다음과 같이 변경하면 프로세스가 시작되는 동안 잠시 작동이 중단됩니다. 그 이후에 변경 사항이 적용되는 동안 데이터베이스를 정상적으로 사용 할 수 있습니다. <ul style="list-style-type: none"> • [General Purpose (SSD)] → [Magnetic]. • 범용(SSD)에서 프로비저닝된 IOPS(SSD)로. DB 인스턴스가 단일 AZ이고 사용자 지정 파라미터 그룹을 사용하는 경우에만 중단이 발생합니다. 다른 AZ DB 인스턴스가 중단되지는 않습니다. • [Magnetic] → [General Purpose (SSD)]. • [Magnetic] → [Provisioned IOPS (SSD)]. • [Provisioned IOPS (SSD)] → [Magnetic]. • 프로비저닝된 IOPS(SSD)에서 범용 (SSD)로. DB 인스턴스가 단일 AZ이고 사용자 지정 파라미터 그룹을 사용하는 경우에만 중단이 발생합니다. 다른 AZ DB 인스턴스가 중단되지는 않습니다.

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
Subnet Group	DB 인스턴스에 대한 서브넷 그룹. 이 설정을 사용하여 DB 인스턴스를 다른 VPC로 이동할 수 있습니다. DB 인스턴스가 VPC 내에 있지 않을 경우 이 설정을 사용하여 DB 인스턴스를 VPC 내로 이동할 수 있습니다. 자세한 내용은 VPC에 있지 않은 DB 인스턴스를 VPC로 이동 (p. 420) 단원을 참조하십시오.	[Apply Immediately]가 true로 설정되는 즉시 변경됩니다. [Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.	이 변경 도중 인스턴스가 중단됩니다. DB 인스턴스가 재부팅됩니다.

관련 주제

- DB 클러스터에서 DB 인스턴스 (p. 123)
- MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 (p. 595)
- MySQL DB 엔진 업그레이드 (p. 607)
- DB 인스턴스 삭제 (p. 128)

MySQL DB 엔진 업그레이드

Amazon RDS에서 새 데이터베이스 엔진 버전을 지원하는 경우, DB 인스턴스를 새 버전으로 업그레이드할 수 있습니다. 메이저 버전 업그레이드와 마이너 버전 업그레이드라는 두 가지 업그레이드가 있습니다. 일반적으로 메이저 엔진 버전 업그레이드는 기존 애플리케이션과 호환되지 않는 변경 사항을 가져올 수 있습니다. 반대로 마이너 버전 업그레이드에는 기존 애플리케이션과 호환되는 변경 사항만 포함됩니다.

메이저 버전 업그레이드를 수행하려면 DB 인스턴스를 수동으로 수정합니다. DB 인스턴스에서 마이너 버전 자동 업그레이드를 사용하도록 설정하면 마이너 버전 업그레이드가 자동으로 수행됩니다. 그 밖의 경우에는 DB 인스턴스를 수동으로 수정하여 마이너 버전 업그레이드를 수행합니다.

주제

- [업그레이드 개요 \(p. 607\)](#)
- [MySQL 메이저 버전 업그레이드 \(p. 608\)](#)
- [업그레이드 테스트 \(p. 610\)](#)
- [MySQL DB 인스턴스 업그레이드 \(p. 611\)](#)
- [자동 중지 시간을 최소화한 MySQL 데이터베이스 업그레이드 \(p. 611\)](#)

업그레이드 개요

Amazon RDS는 업그레이드 프로세스 중에 DB 스냅샷을 2개 캡처합니다. 첫 번째 DB 스냅샷은 업그레이드 변경 이전 DB 인스턴스의 스냅샷입니다. 업그레이드가 데이터베이스에 맞지 않는 경우에는 이 스냅샷을 복구하여 이전 버전의 DB 인스턴스를 생성할 수 있습니다. 두 번째 DB 스냅샷은 업그레이드 완료와 함께 캡처됩니다.

Note

DB 인스턴스에 대한 백업 보존 기간을 0보다 큰 수로 설정하면 Amazon RDS는 DB 스냅샷만 캡처합니다. 백업 보존 기간을 변경하려면 [MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 600\)](#) 단원을 참조하십시오.

업그레이드가 완료되면 이전 버전의 데이터베이스 엔진으로 되돌릴 수 없습니다. 이때 이전 버전으로 되돌리려면 첫 번째로 캡처한 DB 스냅샷을 복구하여 새로운 DB 인스턴스를 생성해야 합니다.

DB 인스턴스를 Amazon RDS가 지원하는 새 버전으로 업그레이드하는 시기는 사용자가 직접 관리합니다. 이러한 관리 수준은 특정 데이터베이스 버전과 호환성을 유지하거나 프로덕션 환경에 배포하기 전에 애플리케이션을 이용해 새 버전을 테스트하는 데 효과적입니다. 모든 준비를 마치면 일정에 가장 적합한 시기에 버전 업그레이드를 실행할 수 있습니다.

DB 인스턴스가 읽기 전용 복제본을 사용하는 경우 원본 인스턴스를 업그레이드하기 전에 읽기 전용 복제본부터 모두 업그레이드합니다.

DB 인스턴스를 다른 AZ 배포로 생성한 경우에는 기본 DB 인스턴스와 예비 DB 인스턴스가 모두 업그레이드됩니다. 기본 DB 인스턴스와 예비 DB 인스턴스가 모두 동시에 업그레이드되므로 업그레이드가 끝날 때까지 작동 중단을 겪게 됩니다. 중단 시간은 DB 인스턴스의 크기에 따라 다릅니다.

MySQL 메이저 버전 업그레이드

Amazon RDS는 MySQL 데이터베이스 엔진의 다음과 같은 메이저 버전 업그레이드를 지원합니다.

- MySQL 5.5에서 MySQL 5.6으로
- MySQL 5.6에서 MySQL 5.7로
- MySQL 5.7에서 MySQL 8.0으로

Note

최신 세대 및 현재 세대의 DB 인스턴스 클래스와 db.m3 이전 세대 DB 인스턴스 클래스로는 MySQL 버전 5.7 및 8.0 DB 인스턴스만 생성할 수 있습니다.

경우에 따라, 이전 세대의 DB 인스턴스 클래스(m3 이외)에서 실행되는 MySQL 버전 5.6 DB 인스턴스를 MySQL 버전 5.7 DB 인스턴스로 업그레이드해야 합니다. 이러한 경우 먼저 DB 인스턴스를 수정하여 최신 세대 또는 현재 세대 DB 인스턴스 클래스를 사용합니다. 그런 다음 DB 인스턴스를 수정하여 MySQL 버전 5.7 데이터베이스 엔진을 사용할 수 있습니다. Amazon RDS DB 인스턴스 클래스에 대한 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.

주제

- [MySQL 메이저 버전 업그레이드 개요 \(p. 608\)](#)
- [MySQL 버전 5.7로의 업그레이드가 느릴 수 있음 \(p. 609\)](#)
- [MySQL 5.7에서 8.0으로 업그레이드하기 위한 사전 점검 \(p. 609\)](#)

MySQL 메이저 버전 업그레이드 개요

메이저 버전 업그레이드에는 기존 애플리케이션과 호환되지 않는 데이터베이스 변경 사항이 포함될 수 있습니다. 결과적으로 Amazon RDS에서는 메이저 버전 업그레이드가 자동으로 적용되지 않으므로 DB 인스턴스를 수동으로 변경해야 합니다. 모든 업그레이드는 프로덕션 환경의 인스턴스에 적용하기 전에 반드시 철저하게 테스트하는 것이 좋습니다.

Amazon RDS의 MySQL 버전 5.5 DB 인스턴스를 MySQL 버전 5.6 이상으로 업그레이드하려면 먼저 사용 가능한 OS 업데이트를 수행합니다. OS 업데이트가 완료된 후 각 메이저 버전으로 업그레이드해야 합니다. 5.5를 5.6으로 업그레이드하고 5.6을 5.7로 업그레이드한 후 5.7을 8.0으로 업그레이드합니다. 2014년 4월 24일 이전에 생성한 MySQL DB 인스턴스에는 업데이트가 적용될 때까지 사용 가능한 OS 업데이트가 표시됩니다. OS 업데이트에 대한 자세한 내용은 [DB 인스턴스의 업데이트 적용 \(p. 114\)](#) 단원을 참조하십시오.

MySQL의 메이저 버전 업그레이드 중에는 필요하다면 Amazon RDS가 MySQL 바이너리 `mysql_upgrade`를 실행하여 테이블을 업그레이드합니다. 또한 Amazon RDS는 메이저 버전 업그레이드

도중 `slow_log` 및 `general_log` 테이블을 비웁니다. 로그 정보를 보존하려면 메이저 버전 업그레이드에 앞서 로그 내용을 저장하십시오.

MySQL 메이저 버전 업그레이드는 일반적으로 약 10분 정도 걸립니다. DB 인스턴스 클래스 크기 때문에 또는 인스턴스가 [Amazon RDS 모범 사례 \(p. 66\)](#)의 특정 작업 지침에 따르지 않는 탓에 일부 업그레이드는 시간이 더 걸릴 수도 있습니다. Amazon RDS 콘솔에서 DB 인스턴스를 업그레이드하는 경우, DB 인스턴스 상태를 보고 업그레이드 완료 시간을 알 수 있습니다. AWS Command Line Interface(AWS CLI)를 사용하여 업그레이드하는 경우, `describe-db-instances` 명령을 사용하여 `status` 값을 확인합니다.

사용자 지정 파라미터 그룹을 사용하는 경우 새 DB 엔진 버전에 대해 기본 파라미터 그룹을 지정하거나 사용자 지정 파라미터 그룹을 만듭니다. 새 파라미터 그룹을 DB 인스턴스에 연결하려면 업그레이드가 완료된 후 고객이 데이터베이스 재부팅을 시작해야 합니다. 파라미터 그룹 변경 사항을 적용하기 위해 DB 인스턴스를 재부팅해야 하는 경우, DB 인스턴스의 파라미터 그룹 상태가 `pending-reboot`로 표시됩니다. DB 인스턴스의 파라미터 그룹 상태는 콘솔에서 보거나 `describe-db-instances` 같은 호출을 사용하여 볼 수 있습니다.

MySQL 버전 5.7로의 업그레이드가 느릴 수 있음

MySQL 버전 5.6.4는 `datetime`, `time`, 및 `timestamp` 열에 대한 새로운 날짜와 시간 형식을 채택하여 날짜와 시간 값에 일부 구성 요소를 허용합니다. DB 인스턴스를 MySQL 버전 5.7로 업그레이드할 때, MySQL에서는 모든 날짜 및 시간 열 형식이 새 형식으로 강제로 변환됩니다.

이렇게 변환되면 테이블이 다시 작성되므로, DB 인스턴스 업그레이드를 완료하기까지 상당한 시간이 걸릴 수 있습니다. 강제 변환은 MySQL 버전 5.6.4 이전 버전을 실행 중인 모든 DB 인스턴스에 발생합니다. 또한 MySQL 버전 5.6.4 이전 버전에서 5.7 이외의 버전으로 업그레이드된 모든 DB 인스턴스에도 발생합니다.

DB 인스턴스가 MySQL 버전 5.6.4 이전 버전을 실행하거나 5.6.4 이전 버전에서 업그레이드된 경우 추가 조치를 수행하는 것이 좋습니다. 이 경우 DB 인스턴스를 MySQL 버전 5.7로 업그레이드하기 전에 데이터베이스의 `datetime`, `time` 및 `timestamp` 열을 변환하는 것이 좋습니다. 이렇게 변환하면 DB 인스턴스를 MySQL 버전 5.7로 업그레이드하는 데 필요한 시간이 상당히 줄어듭니다. 날짜 및 시간 열을 새로운 형식으로 업그레이드하려면 날짜 또는 시간 열이 포함된 각 테이블에 대해 `ALTER TABLE <table_name> FORCE;` 명령을 실행합니다. 테이블을 변경하면 테이블이 읽기 전용으로 잠기므로 유지 관리 기간 동안 이 업데이트를 수행하는 것이 좋습니다.

`datetime`, `time` 또는 `timestamp` 열이 있는 데이터베이스의 모든 테이블을 찾고, 각 테이블에 `ALTER TABLE <table_name> FORCE;` 명령을 생성하려면 다음 쿼리를 사용하십시오.

```
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);
```

MySQL 5.7에서 8.0으로 업그레이드하기 위한 사전 점검

MySQL 8.0에는 MySQL 5.7과 상당한 비호환성이 포함되어 있습니다. 이러한 비호환성으로 인해 MySQL 5.7에서 MySQL 8.0으로 업그레이드하는 동안 문제가 발생할 수 있습니다. 따라서 업그레이드가 성공하려

면 데이터베이스에 몇 가지 준비가 필요할 수 있습니다. MySQL 5.7에서 8.0으로 업그레이드를 시작하면 Amazon RDS가 자동으로 사전 점검을 실행하여 이러한 비호환성을 찾아냅니다. MySQL 8.0으로 업그레이드하는 방법은 [MySQL 업그레이드](#)를 참조하십시오.

이러한 사전 점검은 필수입니다. 건너뛸 수 없습니다. 사전 점검은 다음과 같은 이점을 제공합니다.

- 이를 통해 업그레이드 중 예기치 않은 가동 중단을 피할 수 있습니다.
- 비호환성이 있는 경우 사전 점검을 통해 이에 대해 알 수 있습니다. 그러면 비호환성을 제거하여 8.0으로 업그레이드하기 위한 데이터베이스 준비를 마칠 수 있습니다.

사전 점검에는 MySQL에 포함된 내용과 Amazon RDS 전용으로 생성된 내용이 포함됩니다. MySQL에서 제공하는 사전 점검에 대한 자세한 내용은 [업그레이드 확인 프로그램 유ти리티](#)를 참조하십시오.

사전 점검은 업그레이드를 위해 DB 인스턴스가 종지되기 전에 실행됩니다. 즉, 점검을 실행해도 가동 종지를 일으키지 않습니다. 사전 점검에서 비호환성이 발견되면 Amazon RDS는 DB 인스턴스가 종지되기 전에 자동으로 업그레이드를 취소합니다. Amazon RDS는 비호환성에 대한 이벤트도 생성합니다. Amazon RDS 이벤트에 대한 자세한 내용은 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오.

Amazon RDS는 각 비호환성에 대한 자세한 정보를 로그 파일 `PrePatchCompatibility.log`에 기록합니다. 대부분의 경우 로그 항목에는 비호환성 문제를 해결하기 위한 MySQL 설명서 링크가 포함되어 있습니다. 로그 파일 보기에 대한 자세한 내용은 [데이터베이스 로그 파일 보기 및 나열 \(p. 307\)](#) 단원을 참조하십시오.

Note

Amazon RDS는 MySQL 5.7에서 MySQL 8.0으로 업그레이드할 때만 사전 점검을 실행합니다. MySQL 8.0보다 낮은 릴리스로 업그레이드할 때는 실행되지 않습니다. 예를 들어, MySQL 5.6에서 MySQL 5.7로 업그레이드할 경우 사전 점검이 실행되지 않습니다.

업그레이드 테스트

DB 인스턴스에 대한 메이저 버전 업그레이드를 수행하기 전에 데이터베이스가 새 버전과 호환되는지 여부를 철저하게 테스트합니다. 또한 새 버전과의 호환성을 위해 데이터베이스에 액세스하는 모든 애플리케이션을 철저히 테스트합니다. 다음 절차를 참조하는 것이 좋습니다.

메이저 버전 업그레이드를 테스트하려면

1. 다음과 같이 새 버전의 데이터베이스 엔진에 대한 업그레이드 문서를 검토하여 데이터베이스나 애플리케이션에 영향을 끼칠 수도 있는 호환성 문제가 있는지 살펴봅니다.
 - [MySQL 5.6의 변경 사항](#)
 - [MySQL 5.7의 변경 사항](#)
 - [MySQL 8.0의 변경 사항](#)
2. DB 인스턴스가 사용자 정의 DB 파라미터 그룹의 구성원인 경우에는 기존 설정을 이용해 새로운 메이저 버전과 호환되는 새로운 DB 파라미터 그룹을 생성합니다. 테스트 인스턴스를 업그레이드 할 때는 새로운 DB 파라미터 그룹을 지정해야만 업그레이드 테스트가 올바로 진행될 수 있습니다. DB 파라미터 그룹을 생성하는 것에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.
3. 업그레이드 할 DB 인스턴스의 DB 스냅샷을 생성합니다. 자세한 내용은 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.
4. DB 스냅샷을 복구하여 새로운 테스트 DB 인스턴스를 생성합니다. 자세한 내용은 [DB 스냅샷에서 복원 \(p. 205\)](#) 단원을 참조하십시오.
5. 이후 이어지는 세부적인 방법 중 한 가지를 사용하여 이 새로운 테스트 DB 인스턴스를 변경하고 새로운 버전으로 업그레이드합니다. 2단계에서 새로운 파라미터 그룹을 생성하였다면 이 파라미터 그룹을 지정합니다.
6. 업그레이드한 인스턴스에서 사용할 스토리지를 평가하여 업그레이드 시 추가 스토리지의 필요 여부를 결정합니다.

7. 업그레이드한 DB 인스턴스와 관련하여 데이터베이스 및 애플리케이션과 새로운 버전의 호환성을 보장하는 데 필요하다면 최대한 많은 수의 품질 보증 테스트를 실행합니다. 또한 1단계에서 발견된 호환성 문제의 영향을 평가하는 데 필요한 새로운 테스트도 모두 실행합니다. 저장된 프로시저와 함수를 모두 테스트합니다. 업그레이드한 DB 인스턴스에 대해 애플리케이션의 테스트 버전을 실행합니다.
8. 모든 테스트가 통과되면 프로덕션 환경의 DB 인스턴스에도 업그레이드를 실행합니다. 단, 모든 기능이 정상 작동하는 것을 확인할 때까지 쓰기 연산은 DB 인스턴스에 실행하지 않는 것이 좋습니다.

MySQL DB 인스턴스 업그레이드

MySQL DB 인스턴스의 수동 또는 자동 업그레이드에 대한 자세한 내용은 [DB 인스턴스 엔진 버전 업그레이드 \(p. 118\)](#) 단원을 참조하십시오.

자동 중지 시간을 최소화한 MySQL 데이터베이스 업그레이드

현재 프로덕션 애플리케이션에 MySQL 데이터베이스 인스턴스를 사용 중인 경우, 다음 절차를 사용하여 DB 인스턴스의 데이터베이스 버전을 업그레이드할 수 있습니다. 이 절차는 애플리케이션의 가동 중지 시간을 줄일 수 있습니다.

이 절차는 MySQL 버전 5.5를 MySQL 버전 5.6으로 업그레이드하는 예를 보여줍니다. 동일한 일반 절차를 사용하여 다른 메이저 버전으로 업그레이드할 수 있습니다.

DB 인스턴스를 사용하면서 MySQL 데이터베이스를 업그레이드하는 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. MySQL 5.5 DB 인스턴스의 읽기 전용 복제본을 생성합니다. 이 프로세스에서 업그레이드 가능한 데이터베이스 사본이 만들어집니다.
 - a. 콘솔에서 데이터베이스와 업그레이드하려는 DB 인스턴스를 차례대로 선택합니다.
 - b. 작업에서 읽기 전용 복제본 만들기를 선택합니다.
 - c. 읽기 전용 복제본의 DB 인스턴스 식별자 값을 입력합니다. 이때 DB 인스턴스 클래스와 다른 설정이 MySQL 5.5 DB 인스턴스와 일치해야 합니다.
 - d. [Create read replica]를 선택합니다.
3. 읽기 전용 복제본이 생성되어 상태에 사용 가능이라고 표시되면 읽기 전용 복제본을 MySQL 5.6으로 업그레이드합니다.
 - a. 콘솔에서 데이터베이스와 방금 생성한 읽기 전용 복제본을 차례대로 선택합니다.
 - b. [Modify]를 선택합니다.
 - c. DB 엔진 버전에서 업그레이드 할 MySQL 5.6 버전을 선택한 후 계속을 선택합니다.
 - d. [Scheduling of Modifications]에 대해 [Apply Immediately]를 선택합니다.
 - e. [Modify DB instance]를 선택하여 업그레이드를 시작합니다.
4. 업그레이드가 끝나고 상태에 available이라고 표시되면 업그레이드한 읽기 전용 복제본이 마스터 MySQL 5.5 DB 인스턴스와 비교하여 최신 상태를 나타내는지 확인합니다. 읽기 전용 복제본에 연결하여 SHOW SLAVE STATUS 명령을 실행하면 확인할 수 있습니다. Seconds_Behind_Master 필드가 0이면 복제본이 최신 상태입니다.
5. MySQL 5.6 읽기 전용 복제본을 마스터 DB 인스턴스로 설정하기

Important

MySQL 5.6 읽기 전용 복제본이 독립된 단일 AZ DB 인스턴스로 승격되면 더 이상 MySQL 5.5 DB 인스턴스의 복제본이 아닙니다. 원본 MySQL 5.5 DB 인스턴스가 읽기 전용 모드이고, 모든 쓰기 연산이 중단되는 유지 관리 기간에 MySQL 5.6 읽기 전용 복제본을 승격시키는 것이 좋습니다.

니다. 승격이 완료되면 쓰기 연산을 MySQL 5.6 DB 인스턴스에서 실행하여 쓰기 연산이 순실되는 것을 막을 수 있습니다.

그 밖에도 MySQL 5.6 읽기 전용 복제본을 승격시키기 전에 필요한 모든 데이터 정의 언어 (DDL) 작업을 MySQL 5.6 읽기 전용 복제본에서 실행하는 것이 바람직합니다. 인덱스 생성을 예로 들 수 있습니다. 그러면 승격 후에도 MySQL 5.6 읽기 전용 복제본의 성능에 미치는 부정적인 영향을 방지할 수 있습니다. 읽기 전용 복제본을 승격하려면 다음 절차를 사용하십시오.

- a. 콘솔에서 데이터베이스와 방금 업그레이드한 읽기 전용 복제본을 차례대로 선택합니다.
- b. 작업에서 읽기 전용 복제본 승격을 선택합니다.
- c. [Yes]를 선택하여 읽기 전용 복제본 인스턴스의 자동 백업을 활성화합니다. 자세한 내용은 [백업 작업 \(p. 196\)](#) 단원을 참조하십시오.

[Continue]를 선택합니다.

- d. [Promote Read Replica]를 선택합니다.

6. 이제 MySQL 데이터베이스 버전이 업그레이드되었습니다. 지금부터는 애플리케이션을 새로운 MySQL 5.6 DB 인스턴스에서 실행하거나, 읽기 전용 복제본을 추가하거나, 다중 AZ 지원을 설정하는 등의 작업이 가능합니다.

MySQL DB 스냅샷 업그레이드

Amazon RDS를 사용하여 MySQL DB 인스턴스의 스토리지 볼륨 DB 스냅샷을 생성할 수 있습니다. 생성되는 DB 스냅샷은 Amazon RDS 인스턴스에서 사용하는 엔진 버전에 기반합니다. DB 인스턴스의 DB 엔진 버전 업그레이드뿐 아니라 DB 스냅샷의 엔진 버전도 업그레이드할 수 있습니다. 예를 들어 MySQL 5.1 엔진에서 생성된 DB 스냅샷을 MySQL 5.5 엔진의 DB 스냅샷으로 업그레이드할 수 있습니다. 새 엔진 버전으로 업그레이드된 DB 스냅샷을 복원한 후에는 업그레이드가 성공적이었는지 테스트해야 합니다. 메이저 버전 업그레이드를 테스트하는 방법은 [업그레이드 테스트 \(p. 610\)](#) 단원을 참조하십시오. DB 스냅샷을 복원하는 방법은 [DB 스냅샷에서 복원 \(p. 205\)](#) 단원을 참조하십시오.

Amazon RDS는 MySQL 5.1에서 MySQL 5.5로의 MySQL DB 스냅샷 업그레이드를 지원합니다.

MySQL DB 스냅샷 업그레이드

동일 AWS 리전 내에서 암호화되거나 암호화되지 않은 수동 DB 스냅샷을 MySQL 5.1에서 MySQL 5.5로 업그레이드할 수 있습니다. 자동 백업 과정에서 생성되는 자동 DB 스냅샷은 업그레이드 할 수 없습니다.

AWS Management 콘솔

DB 스냅샷을 업그레이드하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 작업에서 스냅샷 수정을 선택합니다. [Modify DB Snapshot] 페이지가 나타납니다.
4. [Modify Snapshot]를 선택하여 스냅샷을 업그레이드합니다. 업그레이드 중에는 모든 스냅샷 작업이 비활성화됩니다. 또한 DB 스냅샷 상태가 [available]에서 [upgrading]으로 바뀐 다음 완료되면 [active]로 바뀝니다. 스냅샷 손상 문제로 인해 DB 스냅샷을 업그레이드할 수 없는 경우, 상태가 [unavailable]로 바뀝니다. 이 상태로부터 스냅샷을 복구할 수는 없습니다.

AWS CLI

DB 스냅샷을 새 데이터베이스 엔진 버전으로 업그레이드하려면 AWS CLI `modify-db-snapshot` 명령을 사용합니다.

파라미터

- **--db-snapshot-identifier** – 업그레이드할 DB 스냅샷의 식별자입니다. 식별자는 고유의 Amazon 리소스 이름(ARN)이어야 합니다. 자세한 내용은 [Amazon RDS의 Amazon 리소스 이름\(ARN\)을 사용한 작업 \(p. 172\)](#) 단원을 참조하십시오.
- **--engine-version** – DB 스냅샷을 업그레이드할 엔진 버전입니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds modify-db-snapshot \  
    --db-snapshot-identifier <mydbsnapshot> \  
    --engine-version <new_version>
```

Windows의 경우:

```
aws rds modify-db-snapshot ^  
    --db-snapshot-identifier <mydbsnapshot> ^  
    --engine-version <new_version>
```

API

DB 스냅샷을 새 데이터베이스 엔진 버전으로 업그레이드하려면 Amazon RDS API [ModifyDBSnapshot](#) 작업을 사용합니다.

- **DBSnapshotIdentifier** – 업그레이드할 DB 스냅샷의 식별자입니다. 식별자는 고유의 Amazon 리소스 이름(ARN)이어야 합니다. 자세한 내용은 [Amazon RDS의 Amazon 리소스 이름\(ARN\)을 사용한 작업 \(p. 172\)](#) 단원을 참조하십시오.
- **EngineVersion** – DB 스냅샷을 업그레이드할 엔진 버전입니다.

Example

```
https://rds.us-west-2.amazonaws.com/  
?Action=ModifyDBSnapshot  
&DBSnapshotIdentifier=mydbsnapshot  
&EngineVersion=newversion  
&SignatureVersion=4  
&Version=2014-10-31  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Credential=AKIADQKE4SARGYLE/20161222/us-west-1/rds/aws4_request  
&X-Amz-Date=20161222T233051Z  
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date  
&X-Amz-Signature=8052a76dfb18469393c5f0182cdab0ebc224a9c7c5c949155376c1c250fc7ec3
```

MySQL DB 인스턴스로 데이터 가져오기

MySQL DB 인스턴스용 Amazon RDS로 데이터를 가져오는 기법에는 몇 가지가 있습니다. 데이터의 유형, 데이터의 양, 가져오기 작업이 일시적인지 지속적인지 등에 따라 바람직한 접근 방법이 달라집니다. 데이터와 함께 애플리케이션을 마이그레이션하는 경우라면 감당할 수 있는 작업 중단 시간도 고려해야 합니다.

개요

MySQL DB 인스턴스용 Amazon RDS로 데이터를 가져오는 기법을 다음 표에서 찾아보십시오.

소스	데이터 분량	일회성 혹은 지속적	애플리케이션 가동 중지	기술	추가 정보
온프레미스 또는 Amazon EC2에 있는 기존 MySQL 데이터베이스	모두 선택	한 번만	약간	온프레미스 데이터베이스의 백업을 만들어서 Amazon S3에 저장한 다음 MySQL을 실행하여 새로운 Amazon RDS DB 인스턴스에 백업 파일을 복원하십시오.	Amazon RDS MySQL DB 인스턴스로 백업 복원 (p. 619)
기존의 모든 데이터베이스	모두 선택	일회성 혹은 지속적	최소화	AWS Database Migration Service을 사용하면 가동 중지 시간을 최소화하면서 데이터베이스를 마이그레이션할 수 있으며 대부분의 DB 엔진에서는 지속적으로 복제를 계속할 수 있습니다.	AWS Database Migration Service 사용 설명서의 AWS Database Migration Service란 무엇입니까?
기존 Amazon RDS MySQL DB 인스턴스	모두 선택	한 번만	최소화	읽기 전용 복제본을 만들어 읽기 전용 복제본을 승격합니다.	읽기 전용 복제본 작업 (p. 135)
기존 MySQL 또는 MariaDB 데이터베이스	스몰	한 번만	약간	명령줄 유ти리티를 사용하여 Amazon RDS MySQL DB 인스턴스에 바로 데이터를 복제합니다.	MySQL 또는 MariaDB DB에서 Amazon RDS MySQL 또는 MariaDB DB 인스턴스로 데이터 가져오기 (p. 626)
기존 데이터베이스에 저장되지 않은 데이터	Medium	한 번만	약간	플랫 파일을 만들고 mysqlimport 유ти리티를 이용하여 가져옵니다.	임의의 소스에서 MySQL 또는 MariaDB DB 인스턴스로 데이터

소스	데이터 분량	일회성 혹은 지속적	애플리케이션 가동 중지	기술	추가 정보
					가져오기 (p. 640)
온프레미스 또는 Amazon EC2에 있는 기존 MySQL 또는 MariaDB 데이터베이스	모두 선택	지속적	최소화	기존 MySQL 데이터베이스가 복제 소스가 되도록 복제본을 구성합니다.	Amazon RDS 외부에서 실행 중인 MySQL 또는 MariaDB 인스턴스를 사용한 복제 (p. 655) 또는 가동 중지 시간을 단축하여 Amazon RDS MySQL MariaDB DB 인스턴스로 데이터 가져오기 (p. 628)

Note

'mysql' 시스템 데이터베이스에는 DB 인스턴스에 로그인하고 데이터에 액세스하는 데 필요한 인증 및 권한 부여 정보가 포함되어 있습니다. DB 인스턴스에 있는 'mysql' 데이터베이스의 각종 테이블, 데이터 또는 기타 콘텐츠를 삭제하거나 변경하거나 이름을 바꾸거나 자르면 오류가 발생하여 DB 인스턴스와 데이터에 액세스할 수 없게 될 수 있습니다. 이 문제가 발생할 경우 AWS CLI `restore-db-instance-from-db-snapshot` 명령을 사용하여 DB 인스턴스를 스냅샷에서 복원할 수 있습니다. AWS CLI `restore-db-instance-to-point-in-time` 명령을 사용하여 DB 인스턴스를 복원할 수 있습니다.

데이터 가져오기 고려 사항

다음은 MySQL로 데이터를 로드하는 것과 관련된 추가적인 기술 정보입니다. 이 정보는 MySQL 서버 아키텍처를 익히 잘 알고 있는 고급 사용자를 위한 것입니다. LOAD DATA LOCAL INFILE과 관련된 모든 설명은 `mysqlimport`에도 적용됩니다.

이진 로그

데이터 로드는 성능 저하를 초래하며, 이진 로깅을 비활성화한 상태에서 똑같은 데이터를 로드하는 데 비해 이진 로깅을 활성화하면 사용 가능한 디스크 공간이 추가로(최대 4배 더) 필요합니다. 성능 저하의 심각도와 사용 가능한 디스크 공간의 요구량은 데이터 로드에 사용되는 트랜잭션의 크기에 정비례합니다.

트랜잭션 크기

트랜잭션 크기는 MySQL 데이터 로드에서 중요한 역할을 합니다. 트랜잭션 크기는 리소스 소비, 디스크 공간 사용률, 재개 프로세스, 복구 시간 및 입력 형식(플랫 파일 또는 SQL)에 중대한 영향을 미칩니다. 이 섹션에서는 트랜잭션 크기가 이진 로깅에 미치는 영향을 설명하고 큰 데이터를 로드하는 중에 이진 로깅을 비활성화하는 이유를 논증합니다. 앞서 언급한 바와 같이, 이진 로깅은 Amazon RDS 자동 백업 보존 기간을 설정하여 활성화 및 비활성화합니다. 0이 아닌 값으로 설정하면 이진 로깅이 활성화되고 0으로 설정하면 비활성화됩니다. 대규모 트랜잭션이 InnoDB에 미치는 영향과 트랜잭션 크기를 작게 유지하는 것이 중요한 이유도 설명합니다.

작은 트랜잭션

작은 트랜잭션의 경우, 이진 로깅을 사용하면 데이터 로드에 필요한 디스크 쓰기 작업 수가 배가됩니다. 이 결과 다른 데이터베이스 세션의 성능이 심각하게 저하되고 데이터 로딩 시간이 증가할 수 있습니다. 부분적으로는 업로드 속도, 로드 중에 발생하는 다른 데이터베이스 작업, Amazon RDS DB 인스턴스의 용량에 따라 저하 수준이 좌우됩니다.

또한, 이진 로그는 백업 및 제거될 때까지 로드된 데이터의 양과 대략적으로 같은 양의 디스크 공간을 사용합니다. 다행히도, Amazon RDS는 이진 로그를 자주 백업하고 제거하는 방법으로 이 문제를 최소화합니다.

큰 트랜잭션

큰 트랜잭션의 경우, 이진 로깅이 활성화되어 있으면 IOPS와 디스크 사용량이 3배나 늘어납니다. 이는 이진 로그 캐시가 디스크로 유출되어 디스크 공간을 사용하고 각 쓰기 작업을 위해 IO가 추가로 발생하기 때문입니다. 트랜잭션이 커밋하거나 롤백해야 binlog에 캐시를 쓸 수 있으므로, 캐시는 로드되는 데이터의 양에 비례하여 디스크 공간을 사용합니다. 트랜잭션이 커밋할 때 캐시가 binlog로 복사되어야 하며, 이에 따라 디스크에 제3의 데이터 복사본이 생성됩니다.

이 때문에, 이진 로깅을 비활성화한 상태에서 로드할 때에 비해 데이터를 로드하기 위해 사용 가능한 디스크 공간이 3배 이상 많아야 합니다. 예를 들어 10GiB 데이터를 단일 트랜잭션으로 로드하면 로드 중에 최소 30GiB 디스크 공간을 소모합니다. 소모량은 테이블 10GiB + 이진 로그 캐시 10GiB + 이진 로그 자체 10GiB입니다. 캐시 파일을 만든 세션이 종료되거나 세션이 또 다른 트랜잭션 중에 이진 로그 캐시를 다시 채울 때 까지 캐시 파일은 디스크 상에 그대로 남습니다. 이진수 로그는 백업 시까지 디스크에 남아 있어야 하므로, 추가로 20GiB의 디스크 공간을 사용할 수 있게 되기까지 약간의 시간이 필요할 수 있습니다.

더구나 LOAD DATA LOCAL INFILE을 사용하여 데이터를 로드한 경우, 로드 이전에 만든 백업에서 데이터베이스를 복구해야 한다면 데이터의 또 다른 복사본이 생성됩니다. 복원 중에 MySQL이 이진 로그에서 플랫 파일로 데이터를 추출합니다. 그런 다음 MySQL이 원본 트랜잭션과 마찬가지로 LOAD DATA LOCAL INFILE을 실행합니다. 그러나 이번에는 입력 파일이 데이터베이스 서버의 로컬에 있습니다. 앞 예제를 계속 진행할 때, 40GiB 이상의 사용 가능한 디스크 공간이 없으면 복구에 실패합니다.

이진 로깅 비활성화

가능하다면 항상 큰 데이터 로드 중에는 이진 로깅을 비활성화하여 리소스 오버헤드와 추가 디스크 공간이 필요한 상황을 피하십시오. Amazon RDS에서는 백업 보존 기간을 0으로 설정하기만 하면 이진 로깅이 비활성화됩니다. 이 작업을 할 경우 데이터베이스 인스턴스의 DB 스냅샷을 생성한 직후에 로드하는 것이 좋습니다. 이렇게 하면 필요한 경우 로딩 중에 변경한 내용을 쉽고 빠르게 되돌릴 수 있습니다.

로드 후에는 백업 보존 기간을 다시 (0이 아닌) 적당한 값으로 설정합니다.

DB 인스턴스가 읽기 전용 복제본의 원본 DB 인스턴스인 경우에는 백업 보조 기간을 0으로 설정할 수 없습니다.

InnoDB

이 섹션에서는 InnoDB를 사용할 때 트랜잭션 크기를 작은 상태로 유지하기 위한 강력한 인수에 대해 설명합니다.

실행 취소

InnoDB는 트랜잭션 룰백 및 MVCC와 같은 기능을 지원하기 위해 실행 취소를 생성합니다. 실행 취소는 InnoDB 시스템 테이블스페이스(보통 ibdata1)에 저장되고 제거 스레드로 제거될 때까지는 보존됩니다. 제거 스레드가 가장 오래된 활성 트랜잭션의 실행 취소보다 앞설 수는 없으므로, 실제로는 트랜잭션이 룰백을 커밋하거나 완료할 때까지는 차단됩니다. 데이터베이스가 로드 중에 다른 트랜잭션을 처리 중인 경우, 이들 트랜잭션의 실행 취소 역시 시스템 테이블스페이스에 누적되며 트랜잭션이 커밋하고 MVCC에 대해 실행 취소 할 필요가 있는 다른 트랜잭션이 전혀 없더라도 누적된 실행 취소를 제거할 수 없습니다. 이런 상황에서는 (로드 트랜잭션뿐 아니라) 어떤 트랜잭션에서든 변경된 행에 액세스하는 모든 트랜잭션(읽기 전용 트랜잭션 포함)이 느려집니다. 장시간 실행 중인 로드 트랜잭션을 위한 것이 아니라면 제거되었을 수도 있는 실행 취소를 모두 검사하게 되므로 느려집니다.

실행 취소는 시스템 테이블스페이스에 저장되고 시스템 테이블스페이스는 크기가 결코 축소되지 않습니다. 따라서 큰 데이터 로드 트랜잭션은 시스템 테이블스페이스가 상당히 커지는 원인이 될 수 있고, 이때 사용되는 디스크 공간은 데이터베이스를 처음부터 다시 만들어야 회수할 수 있습니다.

룰백

InnoDB는 커밋에 최적화되어 있습니다. 큰 트랜잭션을 룰백하려면 시간이 엄청나게 오래 걸릴 수 있습니다. 경우에 따라서는 특정 시점으로 복구를 수행하거나 DB 스냅샷을 복원하는 것이 오히려 더 빠를 수도 있습니다.

입력 데이터 형식

MySQL은 플랫 파일과 SQL의 두 가지 형식 중 하나로 수신 데이터를 허용할 수 있습니다. 이 섹션에서는 각 형식의 몇몇 주요 장점과 단점을 지적합니다.

플랫 파일

트랜잭션이 비교적 작은 크기로 유지되는 한, LOAD DATA LOCAL INFILE로 플랫 파일을 로드하는 것이 가장 빠르고 최소의 비용으로 데이터를 로드하는 방법일 수 있습니다. SQL로 같은 데이터를 로드하는 것에 비해, 플랫 파일은 보통 네트워크 트래픽이 덜 소요되어 데이터베이스에서 오버헤드가 감소되므로 전송 비용이 절감되고 훨씬 더 빠르게 로드됩니다.

하나의 대형 트랜잭션

LOAD DATA LOCAL INFILE은 전체 플랫 파일을 하나의 트랜잭션으로 로드합니다. 이것이 꼭 나쁜 것은 아닙니다. 개별 파일의 크기를 작게 유지할 수 있다면, 이 방법은 다음과 같은 여러 가지 장점이 있습니다.

- 재개 기능 – 로드된 파일을 계속 손쉽게 추적할 수 있습니다. 로드 중에 문제가 발생하면 약간의 노력만으로 로드에 문제가 생긴 부분을 정확히 파악할 수 있습니다. 일부 데이터를 Amazon RDS로 다시 전송해야 할 수도 있지만, 파일이 작으면 재전송되는 양이 최소한으로 국한됩니다.
- 데이터 병렬 로드 – 단일 파일 로드와 함께 따로 남겨둘 IOPS와 네트워크 대역폭이 있는 경우 병렬로 로드하면 시간을 절약할 수 있습니다.
- 로드 속도 조절 – 데이터 로드가 다른 프로세스에 부정적인 영향을 미칩니다? 파일 간의 간격을 늘려 로드를 조절할 수 있습니다.

주의 사항

LOAD DATA LOCAL INFILE의 장점은 트랜잭션 크기가 증가함에 따라 빠르게 희석됩니다. 큰 데이터 집합을 여러 개의 작은 데이터 집합으로 나눌 수 없는 경우에는 SQL을 선택하는 것이 더 나을 수 있습니다.

SQL

SQL은 플랫 파일에 비해 한 가지 중요한 장점이 있는데, 그것은 바로 트랜잭션 크기를 작게 유지하기 쉽다는 점입니다. 하지만 SQL은 플랫 파일보다 로드하는 데 상당히 더 오랜 시간이 걸릴 수 있어 오류 발생 후로

드를 재개할 위치를 판단하기 어려울 수 있습니다. 예를 들어 mysqldump 파일은 다시 시작할 수 없습니다. mysqldump 파일을 로드하는 동안 오류가 발생하는 경우 이 파일을 수정하거나 바꾸어야 로드를 재개할 수 있습니다. 그 대안은 일단 오류의 원인이 수정되었으면 파일을 로드하고 재생하기 전의 특정 시점으로 복원하는 것입니다.

Amazon RDS 스냅샷을 사용하여 검사적 선택

여러 시간, 심지어 며칠씩 걸릴 정도의 로드 작업을 실행해야 할 경우, 주기적인 체크포인트를 선택할 수 없다면 이진 로깅 없이 로드하는 것은 그다지 좋은 방법이 아닙니다. 바로 이럴 때 매우 편리하게 이용할 수 있는 것이 Amazon RDS; DB 스냅샷 기능입니다. DB 스냅샷은 충돌이나 다른 사고 후의 특정 시점으로 데이터베이스를 복원하는 데 사용할 수 있는 데이터베이스 인스턴스의 특정 시점과 일치하는 복사본을 만듭니다.

체크포인트를 생성하려면 DB 스냅샷을 만들기만 하면 됩니다. 체크포인트를 위해 이전에 생성된 DB 스냅샷은 내구성이나 복원 시간에 영향을 주지 않고 제거할 수 있습니다.

스냅샷 역시 빠르므로, 체크포인트를 자주 사용해도 로드 시간이 크게 늘지는 않습니다.

로드 시간 감소

다음은 로드 시간을 단축하기 위한 몇 가지 추가 팁입니다.

- 모든 보조 인덱스를 만든 후에 로드하십시오. 이는 다른 데이터베이스에 익숙한 사용자에게는 직관에 반하는 팁입니다. 보조 인덱스를 추가하거나 수정하면 MySQL이 인덱스 변경에 따라 새 테이블을 만들고, 기존 테이블에서 새 테이블로 데이터를 복사하고, 원래 테이블은 삭제하게 됩니다.
- PK 순서대로 데이터를 로드하십시오. 이는 로드 시간을 75~80% 줄이고 데이터 파일 크기를 절반으로 줄일 수 있는 InnoDB 테이블에 특히 유용합니다.
- 외래 키 제약 조건 foreign_key_checks=0을 비활성화합니다. LOAD DATA LOCAL INFILE을 통해 로드되는 파일의 경우 이 단계는 많은 경우에 필수입니다. 어떤 로드에 대해서든, FK 검사를 비활성화하면 상당한 성능상의 이득이 생깁니다. 제약 조건을 반드시 활성화하고 로드 후 데이터를 확인하십시오.
- 이미 거의 리소스 제한에 근접하지 않았다면 병렬로 로드하십시오. 상황상 적절할 때는 분할된 테이블을 사용하십시오.
- SQL을 사용하여 로드할 때는 다중 값 삽입을 사용하여 문 실행 오버헤드를 최소화하십시오. mysqldump를 사용할 때는 이 작업이 자동으로 수행됩니다.
- InnoDB 로그 IO innodb_flush_log_at_trx_commit=0을 줄이십시오.
- 읽기 전용 복제본이 없는 DB 인스턴스로 데이터를 로딩하는 경우 데이터가 로딩되는 동안 sync_binlog 파라미터를 0으로 설정합니다. 데이터 로딩이 완료되면, sync_binlog 파라미터를 다시 1로 설정합니다.
- DB 인스턴스가 다중 AZ 배포로 전환되기 전에 데이터를 로딩합니다. 하지만 DB 인스턴스에서 이미 다중 AZ 배포를 사용하고 있는 경우, 데이터 로딩 동안 단일 AZ 배포로 전환하는 것을 권장하지 않습니다. 이렇게 할 경우 아주 적은 개선만 제공되기 때문입니다.

Note

innodb_flush_log_at_trx_commit=0을 사용하면 InnoDB가 매번 커밋할 때가 아니라 1초마다 로그를 폴러시합니다. 이는 상당한 속도상의 이점을 제공하지만, 충돌 중에 데이터 손실로 이어질 수 있습니다. 따라서 주의해서 사용하십시오.

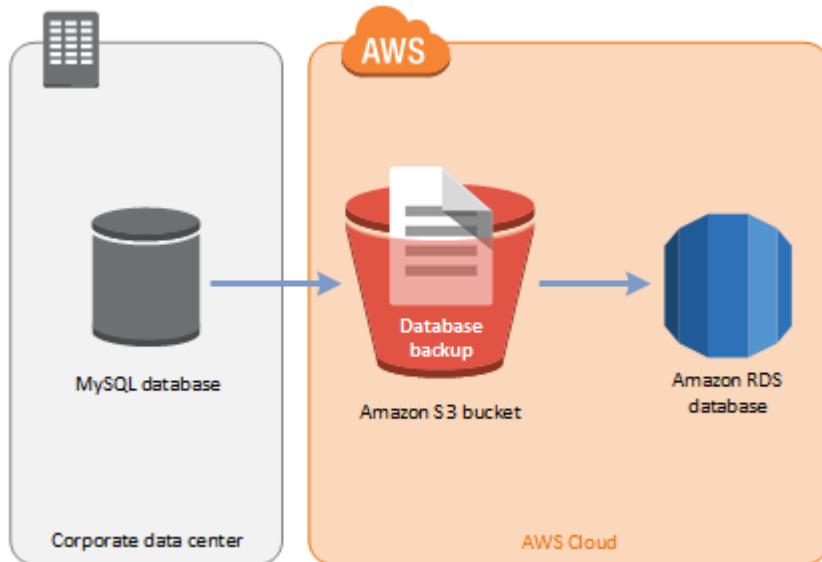
주제

- [Amazon RDS MySQL DB 인스턴스로 백업 복원 \(p. 619\)](#)
- [MySQL 또는 MariaDB DB에서 Amazon RDS MySQL 또는 MariaDB DB 인스턴스로 데이터 가져오기 \(p. 626\)](#)
- [자동 중지 시간을 단축하여 Amazon RDS MySQL MariaDB DB 인스턴스로 데이터 가져오기 \(p. 628\)](#)
- [임의의 소스에서 MySQL 또는 MariaDB DB 인스턴스로 데이터 가져오기 \(p. 640\)](#)

Amazon RDS MySQL DB 인스턴스로 백업 복원

백업 파일을 사용하여 Amazon RDS에서 MySQL 데이터베이스 가져오기를 지원합니다. 온프레미스 데이터베이스의 백업을 생성하여 Amazon S3에 저장한 다음 MySQL을 실행하는 새로운 Amazon RDS DB 인스턴스에 백업 파일을 복원할 수 있습니다.

다음 다이어그램에서 지원되는 시나리오를 찾을 수 있습니다.



Amazon S3에서 백업 파일을 가져오는 작업은 MySQL 버전 5.6 및 5.7에 지원됩니다. Amazon S3에서 백업 파일을 가져오는 작업은 모든 AWS 리전에서 사용할 수 있습니다.

백업 파일을 만들고, 복사하고, 복원하는 동안 데이터베이스를 오프라인 상태로 둘 수 있다면 백업 파일을 사용하여 데이터베이스를 Amazon RDS로 가져오는 것이 좋습니다. 온프레미스 데이터베이스를 오프라인 상태로 둘 수 없는 경우 이 주제에서 설명한 대로 Amazon S3를 통해 Amazon RDS로 마이그레이션한 후 데이터베이스를 업데이트하기 위해 binlog 복제를 사용할 수 있습니다. 자세한 내용은 [Amazon RDS 외부에서 실행 중인 MySQL 또는 MariaDB 인스턴스를 사용한 복제 \(p. 655\)](#) 단원을 참조하십시오. 또한 AWS Database Migration Service를 사용하여 데이터베이스를 Amazon RDS로 마이그레이션할 수도 있습니다. 자세한 내용은 [AWS Database Migration Service란 무엇입니까?](#) 단원을 참조하십시오.

Amazon S3에서 Amazon RDS로 백업 파일 가져오기에 대한 제한 및 권장 사항

다음은 Amazon S3에서 백업 파일을 가져올 때의 몇 가지 제한 및 권장 사항입니다.

- 기존 DB 인스턴스가 아닌 새 DB 인스턴스로만 데이터를 가져올 수 있습니다.
- 온프레미스 데이터베이스 백업을 생성할 때 Percona XtraBackup을 사용해야 합니다.
- 기본 MySQL 데이터 디렉토리 외부에서 정의된 테이블이 있는 원본 데이터베이스에서 마이그레이션할 수 없습니다.
- MySQL 5.5 또는 8.0 데이터베이스는 가져올 수 없습니다.
- 온프레미스 MySQL 5.6을 Amazon RDS MySQL 5.7 또는 8.0 데이터베이스로 가져올 수 없습니다. 가져오기가 완료되면 DB 인스턴스를 업그레이드할 수 있습니다.
- 크기가 6TB를 초과하는 데이터베이스는 복원할 수 없습니다.
- 암호화된 원본 데이터베이스에서 복원할 수 없지만 암호화된 Amazon RDS DB 인스턴스로 복원할 수 있습니다.

- Amazon RDS DB 인스턴스와 다른 AWS 리전에서는 Amazon S3 버킷으로 복원할 수 없습니다.
- Amazon S3에서 가져오기는 db.t2.micro DB 인스턴스 클래스에서 지원되지 않습니다. 그러나 다른 DB 인스턴스 클래스로 복원한 다음 나중에 인스턴스 클래스를 변경할 수 있습니다. 인스턴스 클래스에 대한 자세한 내용은 [사용 가능한 모든 DB 인스턴스 클래스의 하드웨어 사양 \(p. 79\)](#) 단원을 참조하십시오.
- Amazon S3는 Amazon S3 버킷에 업로드되는 파일 크기를 5TB로 제한합니다. 백업 파일이 5TB를 초과하면 해당 백업 파일을 더 작은 크기의 파일들로 나누어야 합니다.
- Amazon RDS는 Amazon S3 버킷에 업로드되는 파일을 100만 개로 제한합니다. 모든 전체 및 충분 백업을 포함하여 데이터베이스에 대한 백업 데이터가 100만 개의 파일을 초과하는 경우 tarball(.tar.gz) 파일을 사용하여 전체 및 충분 백업 파일을 Amazon S3 버킷에 저장하십시오.
- 사용자 계정을 자동으로 가져오지 않습니다. 원본 데이터베이스에서 사용자 계정을 저장하고 나중에 새 DB 인스턴스에 추가하십시오.
- 함수를 자동으로 가져오지 않습니다. 원본 데이터베이스에서 함수를 저장하고 나중에 새 DB 인스턴스에 추가하십시오.
- 저장 프로시저를 자동으로 가져오지 않습니다. 원본 데이터베이스에서 저장 프로시저를 저장하고 나중에 새 DB 인스턴스에 추가하십시오.
- 시간대 정보를 자동으로 가져오지 않습니다. 원본 데이터베이스에서 시간대 정보를 저장하고 나중에 새 DB 인스턴스의 시간대를 설정하십시오. 자세한 내용은 [MySQL DB 인스턴스의 현지 시간대 \(p. 585\)](#) 단원을 참조하십시오.
- 메이저 버전과 마이너 버전 모두 역방향 마이그레이션은 지원하지 않습니다. 예를 들어 버전 5.7에서 버전 5.6으로, 버전 5.6.39에서 버전 5.6.37로 마이그레이션할 수 없습니다.

Amazon S3에서 Amazon RDS로 백업 파일을 가져오는 설정 개요

Amazon S3에서 Amazon RDS로 백업 파일을 가져오는 설정에 필요한 구성 요소가 있습니다.

- 백업 파일을 저장할 Amazon S3 버킷.
- Percona XtraBackup이 생성한 온프레미스 데이터베이스 백업.
- Amazon RDS에서 버킷을 액세스할 수 있도록 허용하는 AWS Identity and Access Management(IAM) 역할.

이미 Amazon S3 버킷이 있으면 그 버킷을 사용하면 됩니다. Amazon S3 버킷이 없는 경우에는 새 버킷을 만들 수 있습니다. 새 버킷을 만드는 방법은 [버킷 생성](#)을 참조하십시오.

백업을 생성하려면 Percona XtraBackup 도구를 사용하십시오. 자세한 내용은 [데이터베이스 백업 생성 \(p. 620\)](#) 단원을 참조하십시오.

이미 IAM 역할이 있으면 그 역할을 사용하면 됩니다. IAM 역할이 없는 경우에는 수동으로 새 역할을 만들 수 있습니다. 또는 AWS Management 콘솔을 사용하여 데이터베이스를 복원할 때 마법사로 계정에서 새 IAM 역할이 생성되도록 선택할 수 있습니다. 수동으로 새 IAM 역할을 만들거나 기존 IAM 역할에 신뢰 및 권한 정책을 연결하려면 [수동으로 IAM 역할 만들기 \(p. 622\)](#) 단원을 참조하십시오. 새 IAM 역할을 생성하려면 [콘솔 \(p. 623\)](#) 단원의 절차를 따르십시오.

데이터베이스 백업 생성

백업을 생성하려면 Percona XtraBackup 소프트웨어를 사용하십시오. Percona XtraBackup은 [Download Percona XtraBackup](#)에서 설치할 수 있습니다.

Note

MySQL 5.7 마이그레이션을 위해서는 Percona XtraBackup 2.4를 사용해야 합니다. 그 이전에 출시된 MySQL 버전인 경우 Percona XtraBackup 2.3 또는 2.4를 사용하십시오.

Percona XtraBackup을 사용하여 MySQL 데이터베이스 파일의 전체 백업을 생성할 수 있습니다. 또는 이미 Percona XtraBackup을 사용하여 MySQL 데이터베이스 파일을 백업 중인 경우 기존의 전체 및 충분 백업 디렉터리 및 파일을 업로드할 수 있습니다.

Percona XtraBackup을 사용한 데이터베이스 백업에 대한 자세한 내용은 Percona 웹 사이트의 [Percona XtraBackup - Documentation](#) 및 [The xtrabackup Binary](#)를 참조하십시오.

Percona XtraBackup을 사용하여 전체 백업 만들기

Amazon S3에서 복원할 수 있는 MySQL 데이터베이스 파일의 전체 백업을 생성하려면 Percona XtraBackup 유틸리티(xtrabackup)를 사용하여 데이터베이스를 백업합니다.

예를 들어, 다음 명령을 실행하면 MySQL 데이터베이스 백업을 만들고 /on-premises/s3-restore/backup 폴더에 파일을 저장합니다.

```
xtrabackup --backup --user=<myuser> --password=<password> --target-dir=</on-premises/s3-restore/backup>
```

백업을 파일 하나로 압축하려면(필요하면 나중에 분할 가능) 다음 형식 중 하나로 백업을 저장하면 됩니다.

- Gzip(.gz)
- tar(.tar)
- Percona xbstream(.xbstream)

다음 명령을 실행하면 Gzip 파일 여러 개로 된 MySQL 데이터베이스 백업이 만들어집니다.

```
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
```

다음 명령을 실행하면 tar 파일 여러 개로 된 MySQL 데이터베이스 백업이 만들어집니다.

```
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
```

다음 명령을 실행하면 xbstream 파일 여러 개로 된 MySQL 데이터베이스 백업이 만들어집니다.

```
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
```

Percona XtraBackup을 사용한 충분 백업 사용

이미 Percona XtraBackup을 사용하여 MySQL 데이터베이스 파일의 전체 및 충분 백업을 수행 중인 경우 전체 백업을 만들고 백업 파일을 Amazon S3로 업로드할 필요가 없습니다. 대신, 기존 백업 디렉터리 및 파일을 Amazon S3 버킷으로 복사하여 시간을 크게 절약할 수 있습니다. Percona XtraBackup을 사용한 충분 백업 만들기에 대한 자세한 내용은 [Incremental Backup](#)을 참조하십시오.

기존의 전체 및 충분 백업 파일을 Amazon S3 버킷으로 복사할 때 기본 디렉터리의 콘텐츠를 반복적으로 복사해야 합니다. 이러한 콘텐츠에는 전체 백업이 포함되며 모든 충분 백업 디렉터리 및 파일도 포함됩니다. 이 복사본은 Amazon S3 버킷의 디렉터리 구조를 보존해야 합니다. Amazon RDS는 모든 파일과 디렉터리를 반복합니다. Amazon RDS는 각 충분 백업에 포함된 xtrabackup-checkpoints 파일을 사용하여 기본 디렉터리를 식별하고 로그 시퀀스 번호(LSN) 범위를 기준으로 충분 백업의 순서를 정렬합니다.

Percona XtraBackup의 백업 고려 사항

Amazon RDS은 파일 이름을 기준으로 백업 파일을 사용합니다. 파일 형식에 따라 백업 파일에 적절한 파일 확장명을 지정합니다. 예를 들어, Percona xbstream 형식을 사용하여 저장한 파일에는 .xbstream을 지정합니다.

Amazon RDS는 알파벳 순서뿐 아니라 자연수 순서로도 백업 파일을 사용합니다. xtrabackup 명령을 실행할 때는 split 옵션을 사용하여 적절한 순서로 백업 파일을 작성하고 이름을 붙여야 합니다.

Amazon RDS는 Percona XtraBackup을 사용하여 생성되는 부분 백업을 지원하지 않습니다. 데이터베이스의 소스 파일을 백업할 때 --tables, --tables-exclude, --tables-file, --databases, --databases-exclude 또는 --databases-file 옵션을 사용하여 부분 백업을 만들 수 없습니다.

Amazon RDS에서는 Percona XtraBackup을 사용한 충분 백업을 지원합니다. Percona XtraBackup을 사용한 충분 백업 만들기에 대한 자세한 내용은 [Incremental Backup](#)을 참조하십시오.

수동으로 IAM 역할 만들기

IAM 역할이 없는 경우에는 수동으로 새 역할을 만들 수 있습니다. 또는 AWS Management 콘솔을 사용하여 데이터베이스를 복원할 때 마법사로 새 IAM 역할이 생성되도록 할 수도 있습니다. 새 IAM 역할을 생성하려면 [콘솔 \(p. 623\)](#) 단원의 절차를 따르십시오.

Amazon S3에서 데이터베이스를 가져오기 위해 새 IAM 역할을 수동으로 생성하려면 Amazon RDS에서 Amazon S3 버킷으로 권한을 위임하기 위한 역할을 생성합니다. IAM 역할을 만들 때 신뢰 및 권한 정책을 연결합니다. Amazon S3에서 백업 파일을 가져오려는 경우 다음 예제와 비슷한 신뢰 및 권한 정책을 사용합니다. 역할을 만드는 방법에 대한 자세한 내용은 [AWS 서비스에 대한 권한을 위임할 역할 생성](#)을 참조하십시오.

또는 AWS Management 콘솔을 사용하여 데이터베이스를 복원할 때 마법사로 새 IAM 역할이 생성되도록 할 수도 있습니다. 새 IAM 역할을 생성하려면 [콘솔 \(p. 623\)](#)의 절차를 따르십시오.

신뢰 및 권한 정책에 따라 ARN(Amazon 리소스 이름)을 제공해야 합니다. ARN 형식에 대한 자세한 내용은 [ARN\(Amazon 리소스 이름\)](#) 및 [AWS 서비스 네임스페이스](#)를 참조하십시오.

Example Amazon S3에서 가져오기 위한 신뢰 정책

```
{  
    "Version": "2012-10-17",  
    "Statement":  
    [ {  
        "Effect": "Allow",  
        "Principal": {"Service": "rds.amazonaws.com"},  
        "Action": "sts:AssumeRole"  
    } ]  
}
```

Example Amazon S3에서 가져오기 위한 권한 정책 — IAM 사용자 권한

```
{  
    "Version": "2012-10-17",  
    "Statement":  
    [ {  
        "Sid": "AllowS3AccessRole",  
        "Effect": "Allow",  
        "Action": "iam:PassRole",  
        "Resource": "arn:aws:iam::IAM User ID:role/S3Access"  
    } ]  
}
```

```
        }
    ]  
}
```

Example Amazon S3에서 가져오기 위한 권한 정책 — 역할 권한

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Action":
            [
                "s3>ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::bucket_name"
        },
        {
            "Effect": "Allow",
            "Action":
            [
                "s3GetObject"
            ],
            "Resource": "arn:aws:s3:::bucket_name/prefix*"
        }
    ]
}
```

Note

파일 이름 접두사를 포함하는 경우 접두사 다음에 별표(*)를 포함합니다. 접두사를 지정하지 않으려면 별표만 지정하면 됩니다.

Amazon S3에서 새 MySQL DB 인스턴스로 데이터를 가져오려면

AWS Management 콘솔, AWS CLI 또는 RDS API를 사용하여 새로운 MySQL DB 인스턴스로 Amazon S3의 데이터를 가져올 수 있습니다.

콘솔

Amazon S3에서 새 MySQL DB 인스턴스로 데이터를 가져오려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. Amazon RDS 콘솔의 오른쪽 위 모서리에서 DB 인스턴스를 생성할 AWS 리전을 선택합니다. 데이터베이스 백업이 포함된 Amazon S3 버킷과 동일한 AWS 리전을 선택합니다.
3. 탐색 창에서 데이터베이스를 선택합니다.
4. [Restore from S3]을 선택하여 마법사를 실행합니다.
[Select engine] 페이지에서 마법사가 열립니다.
5. 엔진 선택 페이지에서 MySQL을 선택한 후 다음을 선택합니다.
[Specify source backup details] 페이지가 나타납니다.

Specify source backup details

Source database specifications

Source engine: mysql

Source engine version: 5.6

S3 bucket Refresh

S3 bucket: - Select one -

S3 folder path prefix (optional) Info

IAM role Refresh

Create a new role Yes

6. [Specify source backup details] 페이지에서 백업 정보를 지정합니다.
 - a. [Source engine]에서 [mysql]을 선택합니다.
 - b. [Source engine version]에서 원본 데이터베이스의 MySQL 버전을 선택합니다.
 - c. S3 bucket(S3 버킷)에서 Amazon S3 버킷을 선택합니다.
 - d. (선택 사항) S3 폴더 경로 접두사에 Amazon S3 버킷에 저장되는 파일의 파일 경로 접두사를 입력합니다. 접두사를 지정하지 않는 경우, RDS는 S3 버킷의 루트 폴더에 있는 모든 파일과 폴더를 사용하여 DB 인스턴스를 만들니다. 접두사를 지정하는 경우 RDS는 파일의 경로가 지정된 접두사로 시작하는 S3 버킷의 파일과 폴더를 사용하여 DB 인스턴스를 만듭니다. 예를 들어 이름이 backups인 하위 폴더의 S3에 백업 파일을 저장했고 여러 세트의 백업 파일이 각각 자체 디렉터리 (gzip_backup1, gzip_backup2 등)에 들어 있다고 가정해봅시다. 이 경우 gzip_backup1 폴더의 파일에서 복원하려면 backups/gzip_backup1의 접두사를 지정합니다.
 - e. Create a new role(새 역할 생성)에서 Yes(예)를 선택해 계정에서 새 IAM 역할을 생성하거나 No(아니요)를 선택해 기존 IAM 역할을 선택합니다.
 - f. IAM 역할에서 기존 IAM 역할을 선택하거나 IAM 역할 이름에 새 IAM 역할의 이름을 지정합니다. 새 역할 생성에서 예를 선택하여 새 IAM 역할을 생성하도록 할 수 있습니다.
7. 다음을 선택하여 계속 진행합니다. [Specify DB details] 페이지가 나타납니다.

DB 세부 정보 지정 페이지에서 DB 인스턴스 정보를 지정합니다. 각 설정에 대한 자세한 내용은 [MySQL DB 인스턴스 설정 \(p. 591\)](#) 단원을 참조하십시오.

Note

새 DB 인스턴스에 충분한 메모리를 할당해야 복원에 성공할 수 있습니다. 또한 나중을 대비하여 추가 메모리를 할당할 수도 있습니다.

8. 다음을 선택하여 계속 진행합니다. [Configure advanced settings] 페이지가 나타납니다.

Amazon RDS가 DB 인스턴스를 시작하는 데 필요한 추가 정보를 입력합니다. 각 설정에 대한 자세한 내용은 [MySQL DB 인스턴스 설정 \(p. 591\)](#) 단원을 참조하십시오.

9. 데이터베이스 생성을 선택하십시오.

AWS CLI

AWS CLI를 사용하여 Amazon S3에서 새 MySQL DB 인스턴스로 데이터를 가져오려면 다음 파라미터와 함께 `restore-db-instance-from-s3` 명령을 호출하십시오. 각 설정에 대한 자세한 내용은 [MySQL DB 인스턴스 설정 \(p. 591\)](#) 단원을 참조하십시오.

Note

새 DB 인스턴스에 충분한 메모리를 할당해야 복원에 성공할 수 있습니다. 또한 나중을 대비하여 추가 메모리를 할당할 수도 있습니다.

- `--allocated-storage`
- `--db-instance-identifier`
- `--db-instance-class`
- `--engine`
- `--master-user-name`
- `--master-user-password`
- `--s3-bucket-name`
- `--s3-ingestion-role-arn`
- `--s3-prefix`
- `--source-engine`
- `--source-engine-version`

Example

Linux, OS X, Unix의 경우:

```
aws rds restore-db-instance-from-s3 \
--allocated-storage 250 \
--db-instance-identifier myidentifier \
--db-instance-class db.m4.large \
--engine mysql \
--master-user-name masterawsuser \
--master-user-password masteruserpassword \
--s3-bucket-name mybucket \
--s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename \
--s3-prefix bucketprefix \
--source-engine mysql \
--source-engine-version 5.6.40
```

Windows의 경우:

```
aws rds restore-db-instance-from-s3 ^
--allocated-storage 250 ^
--db-instance-identifier myidentifier ^
--db-instance-class db.m4.large ^
--engine mysql ^
--master-user-name masterawsuser ^
--master-user-password masteruserpassword ^
--s3-bucket-name mybucket ^
--s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename ^
--s3-prefix bucketprefix ^
--source-engine mysql ^
--source-engine-version 5.6.40
```

RDS API

Amazon RDS API를 사용하여 Amazon S3에서 새 MySQL DB 인스턴스로 데이터를 가져오려면 [RestoreDBInstanceFromS3](#) 작업을 호출하십시오.

MySQL 또는 MariaDB DB에서 Amazon RDS MySQL 또는 MariaDB DB 인스턴스로 데이터 가져오기

시나리오에서 이 방법이 지원되는 경우 백업 파일 및 Amazon S3를 사용하여 Amazon RDS 내부 및 외부로 데이터를 더 쉽게 이동할 수 있습니다. 자세한 내용은 [Amazon RDS MySQL DB 인스턴스로 백업 복원 \(p. 619\)](#) 단원을 참조하십시오.

기존 MySQL 또는 MariaDB 데이터베이스에서 Amazon RDS MySQL 또는 MariaDB DB 인스턴스로 데이터를 가져올 수도 있습니다. 이렇게 하려면 [mysqldump](#)를 사용하여 데이터베이스를 복사하고 Amazon RDS MySQL 또는 MariaDB DB 인스턴스에 바로 파이프합니다. mysqldump 명령줄 유ти리티는 한 MySQL 또는 MariaDB 서버에서 다른 MySQL 또는 MariaDB 서버로 데이터를 전송하고 백업본을 만드는 데 흔히 사용됩니다. 이 유ти리티는 MySQL 및 MariaDB 클라이언트 소프트웨어와 함께 포함되어 있습니다.

외부 데이터베이스에서 Amazon RDS DB 인스턴스로 데이터를 이동하는 일반적인 mysqldump 명령은 다음과 같습니다.

```
mysqldump -u <local_user> \
--databases <database_name> \
--single-transaction \
--compress \
--order-by-primary \
-p<local_password> | mysql -u <RDS_user> \
--port=<port_number> \
--host=<host_name> \
-p<RDS_password>
```

Important

-p 옵션과 입력한 암호 사이에 공백이 없어야 합니다.

Note

- 덤프 파일에서 다음 스키마를 제외합니다. sys, performance_schema 및 information_schema. mysqldump 유ти리티는 기본적으로 이러한 스키마를 제외합니다.
- 사용자 및 권한을 마이그레이션해야 하는 경우 이를 다시 생성하는 데이터 제어 언어(DCL)를 생성하는 도구 사용을 고려합니다. 예를 들어 [pt-show-grants](#) 유ти리티가 있습니다.

사용되는 파라미터는 다음과 같습니다.

- **-u <local_user>** – 사용자 이름을 지정하기 위해 사용합니다. 이 파라미터를 처음 사용할 때, 로컬 MySQL 또는 MariaDB 데이터베이스에서 --databases 파라미터로 식별되는 사용자 계정 이름을 지정합니다.
- **--databases <database_name>** – 로컬 MySQL 또는 MariaDB 인스턴스에서 Amazon RDS로 가져오려는 데이터베이스 이름을 지정합니다.
- **--single-transaction** – 로컬 데이터베이스에서 로드한 모든 데이터가 단일 시점에서 일치하는지 확인하기 위해 사용합니다. mysqldump가 데이터를 읽는 동안 데이터를 변경하는 다른 프로세스가 있는 경우, 이 옵션을 사용하여 데이터 무결성을 유지합니다.
- **--compress** – 데이터를 Amazon RDS로 전송하기 전에 로컬 데이터베이스에서 데이터를 압축하여 네트워크 대역폭 사용을 줄이기 위해 사용합니다.
- **--order-by-primary** – 기본 키를 기준으로 각 테이블의 데이터를 정렬하여 로드 시간을 줄이기 위해 사용합니다.
- **-p<local_password>** – 암호를 지정하기 위해 사용합니다. 이 파라미터를 처음 사용할 때, 첫 번째 -u 파라미터로 식별되는 사용자 계정 암호를 지정합니다.
- **-u <RDS_user>** – 사용자 이름을 지정하기 위해 사용합니다. 이 파라미터를 두 번째로 사용할 때, Amazon RDS MySQL용 기본 데이터베이스 또는 MariaDB DB 인스턴스에서 --host 파라미터로 식별되는 사용자 계정 이름을 지정합니다.
- **--port <port_number>** – Amazon RDS MySQL 또는 MariaDB DB 인스턴스의 포트를 지정하기 위해 사용합니다. 인스턴스를 생성할 때 값을 변경하지 않는 한, 기본값은 3306입니다.
- **--host <host_name>** – Amazon RDS DB 인스턴스 엔드포인트의 DNS 이름을 지정하기 위해 사용합니다(예: myinstance.123456789012.us-east-1.rds.amazonaws.com). Amazon RDS Management Console의 인스턴스 세부 정보에서 엔드포인트 값을 찾을 수 있습니다.
- **-p<RDS_password>** – 암호를 지정하기 위해 사용합니다. 이 파라미터를 두 번째로 사용할 때, 두 번째 -u 파라미터로 식별되는 사용자 계정 암호를 지정합니다.

Amazon RDS 데이터베이스에서 저장 프로시저, 트리거, 함수 또는 이벤트를 수동으로 만들어야 합니다. 복사 종인 데이터베이스에 이런 객체가 하나라도 있는 경우에는 mysqldump 명령과 함께 --routines=0 --triggers=0 --events=0 파라미터를 포함시켜 mysqldump를 실행할 때 이런 객체를 제외합니다.

다음 예제에서는 로컬 호스트에 있는 world 샘플 데이터베이스를 Amazon RDS MySQL DB 인스턴스로 복사합니다.

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우, 다음 명령은 Windows 프로그램 메뉴에서 명령 프롬프트를 마우스 오른쪽 버튼으로 클릭하고 관리자 권한으로 실행을 선택하여 열린 명령 프롬프트 창에서 실행해야 합니다.

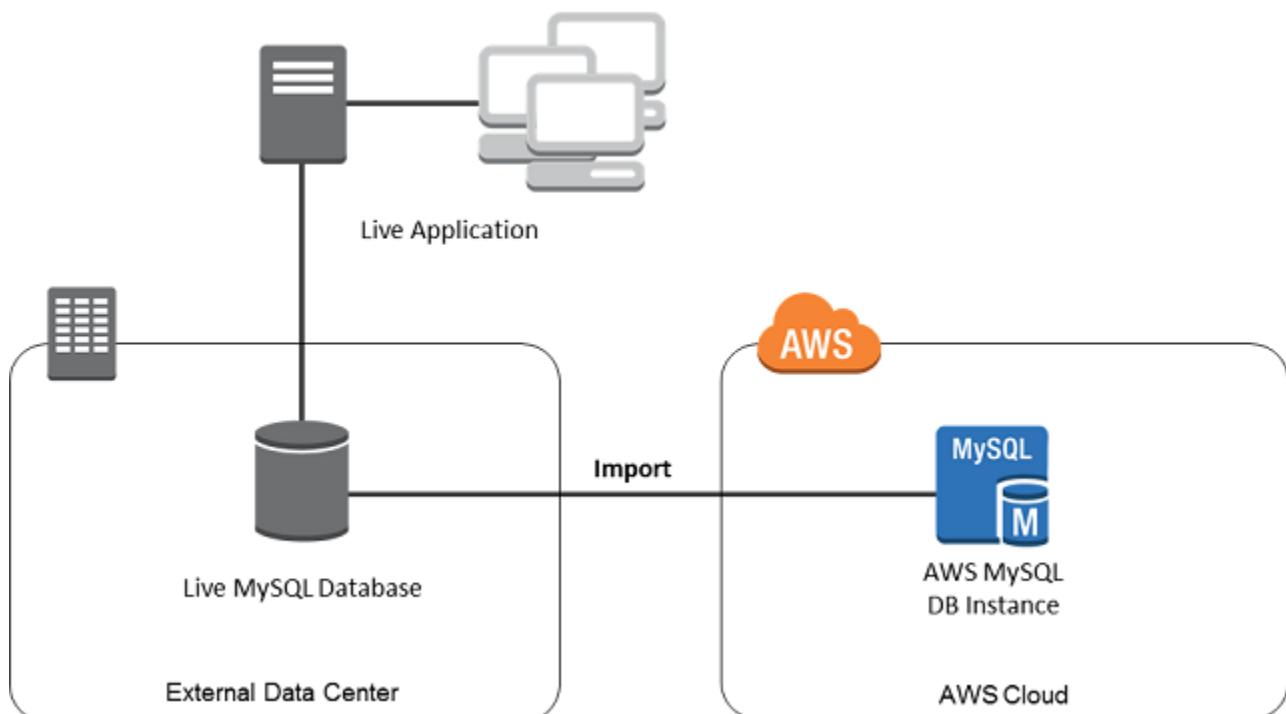
```
mysqldump -u localuser ^
    --databases world ^
    --single-transaction ^
    --compress ^
    --order-by-primary ^
    -plocalpassword | mysql -u rdsuser ^
        --port=3306 ^
        --host=myinstance.123456789012.us-east-1.rds.amazonaws.com ^
        -prdspassword
```

가동 중지 시간을 단축하여 Amazon RDS MySQL MariaDB DB 인스턴스로 데이터 가져오기

시나리오에서 이 방법이 지원되는 경우 백업 파일 및 Amazon S3를 사용하여 Amazon RDS 내부 및 외부로 데이터를 더 쉽게 이동할 수 있습니다. 자세한 내용은 [Amazon RDS MySQL DB 인스턴스로 백업 복원 \(p. 619\)](#) 단원을 참조하십시오.

라이브 애플리케이션을 지원하는 외부 MySQL 또는 MariaDB 데이터베이스에서 Amazon RDS MySQL 또는 MariaDB DB 인스턴스로 데이터를 가져와야 할 경우가 있습니다. 이런 경우 다음 절차를 통해 애플리케이션 가용성에 미치는 영향을 최소화할 수 있습니다. 이 절차는 대규모 데이터베이스로 작업하는 경우에도 유용합니다. 여기서 네트워크에서 AWS로 전달되는 데이터의 양을 줄여 가져오기 비용을 줄일 수 있으므로 이 절차가 도움이 됩니다.

이 절차에서는 데이터베이스 데이터의 복사본을 Amazon EC2 인스턴스로 전송하고 데이터를 새 Amazon RDS DB 인스턴스로 가져옵니다. 그런 다음, 애플리케이션을 Amazon RDS DB 인스턴스로 전달하기 전에 복제를 사용하여 Amazon RDS DB 인스턴스를 라이브 외부 인스턴스로 업데이트합니다. 외부 인스턴스가 MariaDB 10.0.2 이상이고 대상 인스턴스가 Amazon RDS MariaDB일 경우에는 전역 트랜잭션 ID(GTID)를 기반으로 MariaDB 복제를 구성하고, 그렇지 않은 경우에는 이진 로그 좌표를 기반으로 복제를 구성합니다. GTID 기반 복제는 충돌 안정성 기능을 개선하므로 외부 데이터베이스가 지원하는 경우에는 GTID 기반 복제를 사용하는 것이 좋습니다. 자세한 내용은 MariaDB 설명서에서 [Global Transaction ID](#) 단원을 참조하십시오.

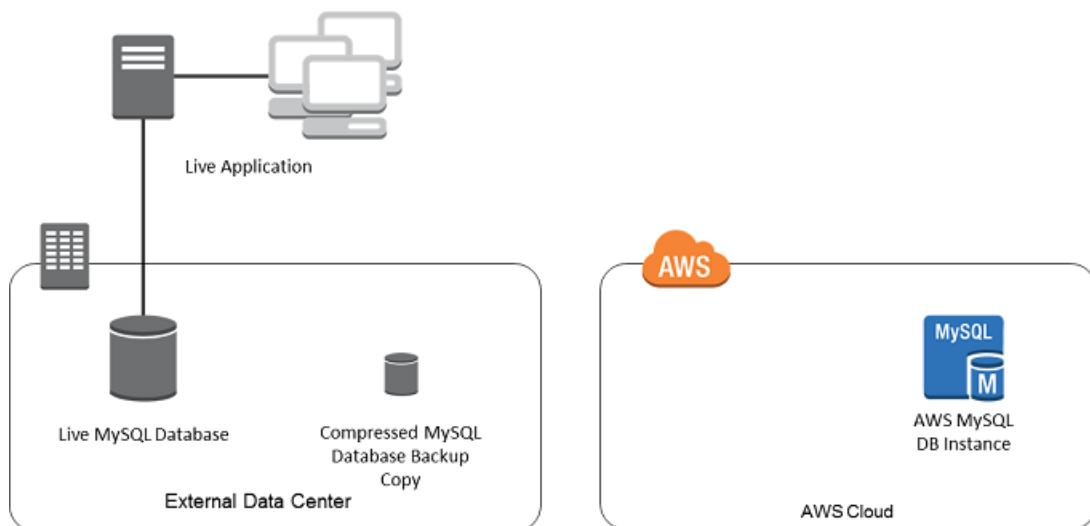


Note

버전 5.1 이전의 MySQL 버전에서 가져온 원본 MySQL 데이터베이스에 이 절차를 적용하면 복제 문제가 발생할 가능성이 있으므로, 이때는 이 절차를 사용하지 않는 것이 좋습니다. 자세한 내용은 MySQL 문서의 [Replication Compatibility Between MySQL Versions](#) 단원을 참조하십시오.

기존 데이터베이스의 복사본 만들기

최소한의 가동 중지 시간으로 대량의 데이터를 Amazon RDS MySQL 또는 MariaDB DB 인스턴스로 마이그레이션하는 프로세스에서 첫 번째 단계는 원본 데이터의 복사본을 만드는 것입니다.



`mysqldump` 유ти리티를 사용하여 SQL 또는 구분 기호로 분리된 텍스트 형식으로 데이터베이스 백업본을 만들 수 있습니다. 비프로덕션 환경에서 각각의 형식으로 테스트 실행을 통해 어떤 방법을 사용해야 `mysqldump` 실행 시간이 최소화되는지 확인해야 합니다.

또한, 로드를 위해 구분 기호로 분리된 텍스트 형식을 사용할 때의 이점에 대해 `mysqldump` 성능상의 이점을 비교 검토해야 합니다. 구분 기호로 분리된 텍스트 형식을 사용하는 백업에서는 덤프되는 각 테이블에 대해 텁으로 구분된 텍스트 파일이 생성됩니다. `LOAD DATA LOCAL INFILE` 명령을 사용하여 이런 파일을 병렬로 로드함으로써 데이터베이스를 가져오는 데 필요한 시간을 줄일 수 있습니다. `mysqldump` 형식을 선택한 다음 데이터를 로드하는 방법에 대한 자세한 내용은 MySQL 문서의 [Using mysqldump For Backups](#)를 참조 하십시오.

백업 작업을 시작하기 전에 Amazon RDS로 복사할 MySQL 또는 MariaDB 데이터베이스에서 복제 옵션을 설정해야 합니다. 복제 옵션에는 이진 로깅 활성화와 고유한 서버 ID 설정이 포함됩니다. 이런 옵션을 설정하면 서버가 데이터베이스 트랜잭션 로깅을 시작하고 이 프로세스의 후반부에 복제 마스터가 되도록 서버를 준비 시킵니다.

Note

- 백업 복사본이 생성되는 동안 복제 옵션을 설정하고 읽기 전용 모드가 되도록 하려면 데이터베이스를 중지해야 하므로, 이런 작업을 위한 유지 관리 기간을 예약할 필요가 있습니다.
- 덤프 파일에서 다음 스키마를 제외합니다. `sys`, `performance_schema` 및 `information_schema`. `mysqldump` 유ти리티는 기본적으로 이러한 스키마를 제외합니다.
- 사용자 및 권한을 마이그레이션해야 하는 경우 이를 다시 생성하는 데이터 제어 언어(DCL)를 생성하는 도구 사용을 고려합니다. 예를 들어 [pt-show-grants](#) 유ти리티가 있습니다.

복제 옵션을 설정하려면

- my.cnf 파일을 편집합니다(이 파일은 보통 /etc에 있음).

```
sudo vi /etc/my.cnf
```

`log_bin` 및 `server_id` 옵션을 [`mysqld`] 섹션에 추가합니다. `log_bin` 옵션은 이진 로그 파일에 대한 파일 이름 식별자를 제공합니다. `server_id` 옵션은 마스터-복제본 관계에서 서버의 고유 식별자를 제공합니다.

다음 예제에서는 my.cnf 파일의 업데이트된 [`mysqld`] 섹션을 보여줍니다.

```
[mysqld]
log-bin=mysql-bin
server-id=1
```

자세한 내용은 MySQL 문서의 [Setting the Replication Master Configuration](#) 단원을 참조하십시오.

- mysql 서비스를 다시 시작합니다.

```
sudo service mysqld restart
```

기존 데이터베이스의 백업 복사본을 만들려면

- SQL 또는 구분 기호로 분리된 텍스트 형식을 지정하는 mysqldump 유ти리티를 사용하여 데이터 백업을 만듭니다.

서버 간 복제를 시작할 때 사용할 수 있는 백업 파일을 만들려면 --master-data=2를 지정해야 합니다.
자세한 내용은 [mysqldump](#) 문서 단원을 참조하십시오.

성능을 개선하고 데이터 무결성을 보장하려면 mysqldump의 --order-by-primary 및 --single-transaction 옵션을 사용합니다.

백업에 MySQL 시스템 데이터베이스가 포함되지 않도록 하려면 mysqldump와 함께 --all-databases 옵션을 사용하지 마십시오. 자세한 내용은 MySQL 문서의 [Creating a Dump Snapshot Using mysqldump](#) 단원을 참조하십시오.

필요한 경우 chmod를 사용하여 백업 파일이 생성될 디렉터리가 쓰기 가능한 디렉터리가 되도록 하십시오.

Important

Windows에서는 명령 창을 관리자 권한으로 실행하십시오.

- SQL 출력을 표시하려면 다음 명령을 사용합니다.

Linux, OS X, Unix의 경우:

```
sudo mysqldump \
--databases <database_name> \
--master-data=2 \
--single-transaction \
--order-by-primary \
-r backup.sql \
-u <local_user> \
-p <password>
```

Windows의 경우:

```
mysqldump ^
--databases <database_name> ^
--master-data=2 ^
--single-transaction ^
--order-by-primary ^
-r backup.sql ^
-u <local_user> ^
-p <password>
```

- 구분 기호로 분리된 텍스트 출력을 표시하려면 다음 명령을 사용합니다.

Linux, OS X, Unix의 경우:

```
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>
```

Windows의 경우:

```
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

Amazon RDS 데이터베이스에서 저장 프로시저, 트리거, 함수 또는 이벤트를 수동으로 만들어야 합니다. 복사 중인 데이터베이스에 이런 객체가 하나라도 있는 경우에는 mysqldump 명령과 함께 `--routines=0 --triggers=0 --events=0` 인수를 포함시켜 mysqldump를 실행할 때 이런 객체를 제외합니다.

구분 기호로 분리된 텍스트 형식을 사용할 때 mysqldump를 실행하면 CHANGE MASTER TO 설명이 반환됩니다. 이 설명에는 마스터 로그 파일 이름과 위치가 포함됩니다. 외부 인스턴스가 MariaDB 버전 10.0.2 이상이 아닐 경우, MASTER_LOG_FILE 및 MASTER_LOG_POS 값을 메모하십시오. 복제를 설정할 때 필요합니다.

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
```

SQL 형식을 사용하는 경우 [외부 데이터베이스와 새 Amazon RDS DB 인스턴스 간의 복제 \(p. 636\)](#) 내 프로시저의 4단계에서 마스터 로그 파일 이름 및 위치를 가져올 수 있습니다. 외부 인스턴스가 MariaDB 버전 10.0.2 이상일 경우에는 GTID를 다음 단계에서 가져올 수 있습니다.

2. 사용하는 외부 인스턴스가 MariaDB 버전 10.0.2 이상일 경우 GTID 기반 복제를 사용합니다. 외부 MariaDB 인스턴스에서 SHOW MASTER STATUS를 실행하여 이전 로그 파일 이름 및 위치를 가져온 다음, 외부 MariaDB 인스턴스에서 BINLOG_GTID_POS를 실행하여 GTID로 변환합니다.

```
SELECT BINLOG_GTID_POS('<binary log file name>', <binary log file position>);
```

반환된 GTID를 메모하십시오. 복제를 구성하는 데 필요합니다.

3. 복사된 데이터를 압축하여 데이터를 Amazon RDS DB 인스턴스로 복사하는 데 필요한 네트워크 리소스의 양을 줄입니다. Amazon EC2 인스턴스를 얼마나 크게 만들지 결정할 때 백업 파일의 크기를 알아야 하므로 이 정보를 기록해 두십시오. 모두 완료되었으면, GZIP 또는 선호하는 압축 유ти리티를 사용하여 백업 파일을 압축합니다.

- SQL 출력을 압축하려면 다음 명령을 사용합니다.

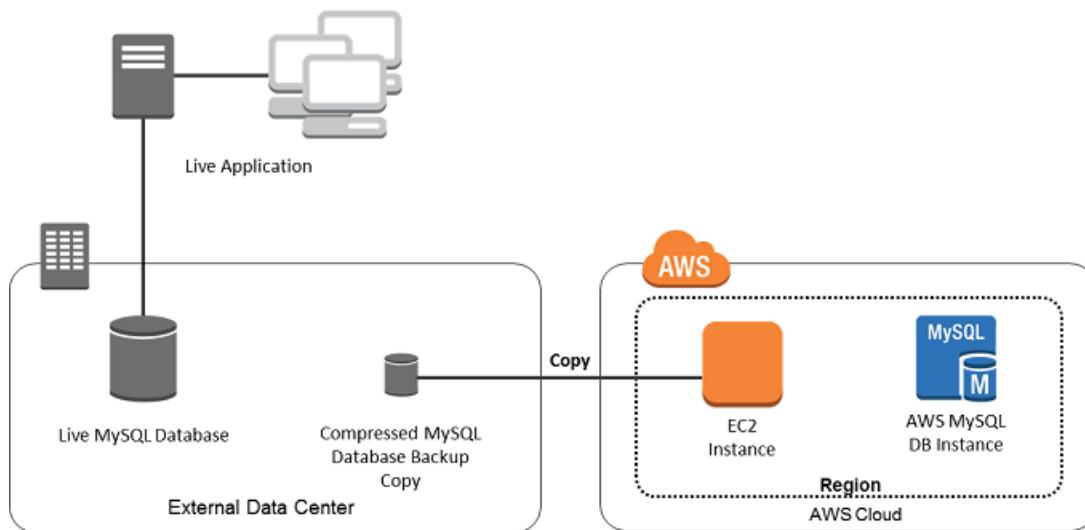
```
gzip backup.sql
```

- 구분 기호로 분리된 텍스트 출력을 압축하려면 다음 명령을 사용합니다.

```
tar -zcvf backup.tar.gz <target_directory>
```

Amazon EC2 인스턴스 만들기 및 압축된 데이터베이스 복사

압축된 데이터베이스 백업 파일을 Amazon EC2 인스턴스로 복사할 때는 데이터베이스 인스턴스 사이에서 압축되지 않은 데이터를 직접 복사할 때보다 네트워크 리소스를 덜 사용합니다. 데이터가 Amazon EC2에 있으면 Amazon RDS MySQL 또는 MariaDB DB 인스턴스로 바로 데이터를 복사할 수 있습니다. 네트워크 리소스의 비용을 절약하려면 Amazon EC2 인스턴스가 Amazon RDS DB 인스턴스와 같은 AWS 리전에 있어야 합니다. Amazon EC2 인스턴스를 Amazon RDS DB 인스턴스와 같은 AWS 리전에 두면 가져오기 도중의 네트워크 지연 시간도 줄어듭니다.



Amazon EC2 인스턴스를 만들고 데이터를 복사하려면

- MySQL 데이터베이스 엔진을 실행할 RDS DB 인스턴스를 생성하려는 AWS 리전에서 VPC, VPC 보안 그룹 및 VPC 서브넷을 만듭니다. VPC 보안 그룹의 인바운드 규칙에서 애플리케이션에 필요한 IP 주소가 AWS에 연결하도록 허용하는지 확인하십시오. 이는 IP 주소의 범위(예: 203.0.113.0/24) 또는 다른 VPC 보안 그룹일 수 있습니다. [Amazon VPC Management Console](#)을 사용하여 VPC, 서브넷 및 보안 그룹을 만들고 관리할 수 있습니다. 자세한 내용은 Amazon Virtual Private Cloud 시작 안내서의 [Amazon VPC 시작하기](#) 단원을 참조하십시오.

Note

이전의 AWS 계정은 Amazon EC2-Classic 모드에서 인스턴스를 시작할 수도 있습니다. 이 경우에는 Amazon RDS 인스턴스에 대한 DB 보안 그룹에서 인바운드 규칙이 Amazon EC2 프라이빗 IP 주소를 사용하여 EC2-Classic 인스턴스에 대한 액세스를 허용하는지 확인하십시오. 자세한 내용은 [DB 보안 그룹 작업\(EC2-Classic 플랫폼\) \(p. 394\)](#) 단원을 참조하십시오.

- Amazon EC2 관리 콘솔을 열고 Amazon EC2 인스턴스와 Amazon RDS DB 인스턴스를 모두 포함하는 AWS 리전을 선택합니다. 1단계에서 만든 VPC, 서브넷 및 보안 그룹을 사용하여 Amazon EC2 인스턴스를 시작합니다. 데이터베이스 백업 파일이 압축되어 있지 않을 때는 이 파일을 위한 충분한 스토리지 공간을 가진 인스턴스 유형을 선택해야 합니다. Amazon EC2 인스턴스에 관한 세부 정보는 Linux용 Amazon Elastic Compute Cloud 사용 설명서의 [Amazon EC2 Linux 인스턴스 시작하기](#)를 참조하십시오.

3. Amazon EC2 인스턴스에서 Amazon RDS DB 인스턴스로 연결하려면 VPC 보안 그룹을 편집하고, EC2 인스턴스의 프라이빗 IP 주소를 지정하는 인바운드 규칙을 추가해야 합니다. EC2 콘솔 창에 있는 [Instance] 창의 [Details] 탭에서 프라이빗 IP 주소를 찾을 수 있습니다. VPC 보안 그룹을 편집하고 인바운드 규칙을 추가하려면 EC2 콘솔 탐색 창에서 [Security Groups]를 선택하고 보안 그룹을 선택한 다음 EC2 인스턴스의 프라이빗 IP 주소를 지정하는 MySQL/Aurora에 대한 인바운드 규칙을 추가합니다. 인바운드 규칙을 VPC 보안 그룹에 추가하는 방법을 알아보려면 [규칙 추가 및 제거](#)를 참조하십시오.
4. 로컬 시스템에서 Amazon EC2 인스턴스로 압축된 데이터베이스 백업 파일을 복사합니다. 필요한 경우 chmod를 사용하여 Amazon EC2 인스턴스의 대상 디렉터리에 대해 쓰기 권한이 있는지 확인하십시오. scp 또는 SSH 클라이언트를 사용하여 파일을 복사할 수 있습니다. 다음은 그 한 예입니다.

```
$ scp -r -i <key pair>.pem backup.sql.gz ec2-user@<EC2 DNS>:</target_directory>/  
backup.sql.gz
```

Important

중요한 데이터는 반드시 보안 네트워크 전송 프로토콜을 사용하여 복사해야 합니다.

5. Amazon EC2 인스턴스에 연결하고 다음 명령을 사용하여 최신 업데이트와 MySQL 클라이언트 도구를 설치합니다.

```
sudo yum update -y  
sudo yum install mysql-server -y
```

자세한 내용은 Linux용 Amazon Elastic Compute Cloud 사용 설명서의 [인스턴스에 연결](#)을 참조하십시오.

6. Amazon EC2 인스턴스에 연결되어 있는 상태에서 데이터베이스 백업 파일의 압축을 풉니다. 예:

 - SQL 출력을 압축 해제하려면 다음 명령을 사용합니다.

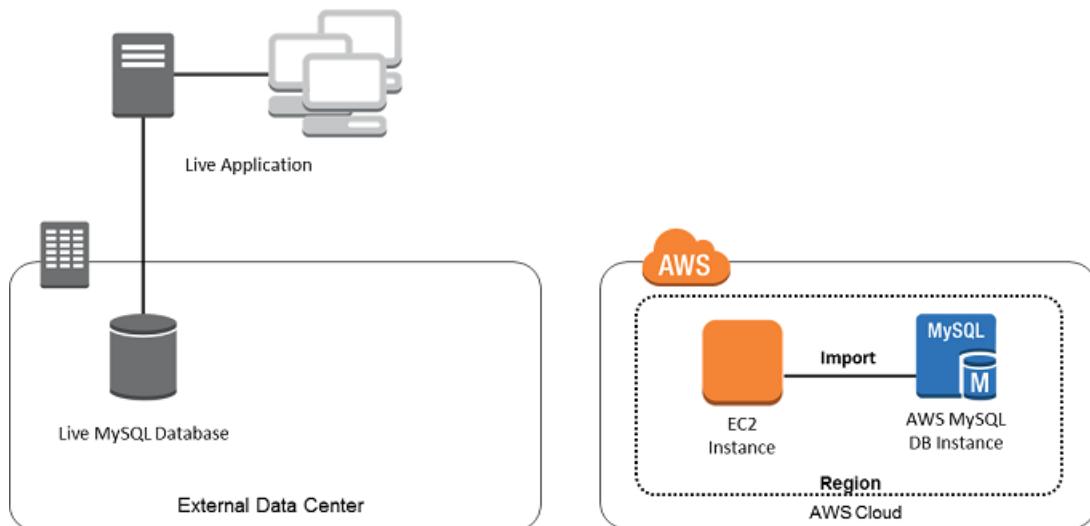
```
gzip backup.sql.gz -d
```

- 구분 기호로 분리된 텍스트 출력을 압축 해제하려면 다음 명령을 사용합니다.

```
tar xzvf backup.tar.gz
```

Amazon RDS MySQL 또는 MariaDB DB 인스턴스 만들기 및 Amazon EC2 인스턴스에서 데이터 가져오기

Amazon EC2 인스턴스와 같은 AWS 리전에 Amazon RDS MySQL 또는 MariaDB DB 인스턴스를 만들면 인터넷보다 빠른 속도로 EC2에서 데이터베이스 백업 파일을 가져올 수 있습니다.



Amazon RDS MySQL 또는 MariaDB DB 인스턴스를 만들고 데이터를 가져오려면

1. 이 Amazon RDS DB 인스턴스에 대해 예상되는 워크로드를 지원하기 위해 필요한 DB 인스턴스 클래스와 스토리지 공간의 양을 결정합니다. 이 프로세스에는 데이터 로드 절차를 위해 충분한 공간과 처리 용량이 어느 정도인지, 프로덕션 워크로드 처리에 필요한 것이 무엇인지 결정하는 과정이 포함되어야 합니다. 원본 MySQL 또는 MariaDB 데이터베이스의 크기와 리소스를 바탕으로 평가할 수 있습니다. 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.
2. 워크로드를 지원하기 위해 Amazon RDS Provisioned IOPS(초당 입/출력 작업 수)가 필요할지 결정합니다. 프로비저닝된 IOPS 스토리지는 I/O 집약적인 온라인 트랜잭션 처리(OLTP) 워크로드를 위한 높은 처리량을 제공합니다. 자세한 내용은 [프로비저닝된 IOPS SSD 스토리지 \(p. 103\)](#) 단원을 참조하십시오.
3. [Amazon RDS 콘솔](#)을 엽니다. 오른쪽 위 모서리에서 Amazon EC2 인스턴스를 포함한 AWS 리전을 선택합니다.
4. 탐색 창에서 데이터베이스를 선택합니다.
5. 데이터베이스 생성을 선택한 후 DB 인스턴스에 대한 옵션을 선택하는 단계를 진행합니다.
 - a. [Select engine] 페이지에서 적절하게 [MySQL] 또는 [MariaDB]를 선택한 후 [Next]를 선택합니다.
 - b. Choose use case(사용 사례 선택) 페이지에서 Dev/Test – MySQL(개발/테스트 - MySQL)을 선택하여 다중 AZ 배포 및 프로비저닝된 IOPS 스토리지 구성을 건너뜁니다.
 - c. [Specify DB details] 페이지의 [Instance specifications] 섹션에서 적당하다고 판단되는 DB 인스턴스 클래스와 할당된 스토리지 크기를 지정합니다. [Multi-AZ deployment]에 대해 [No]를 선택합니다. [Storage type]에 대해서는 2단계에서 결정한 대로 [Provisioned IOPS (SSD)]를 사용할지 지정합니다. DB 엔진 버전에서 원본 MySQL 인스턴스와 호환되는 버전을 다음과 같이 선택합니다.
 - 원본 인스턴스가 MySQL 5.1.x인 경우 Amazon RDS DB 인스턴스는 MySQL 5.5.x여야 합니다.
 - 원본 인스턴스가 MySQL 5.5.x인 경우 Amazon RDS DB 인스턴스는 MySQL 5.5.x 이상이어야 합니다.
 - 원본 인스턴스가 MySQL 5.6.x인 경우 Amazon RDS DB 인스턴스는 MySQL 5.6.x 또는 MariaDB여야 합니다.
 - 원본 인스턴스가 MySQL 5.7.x인 경우 Amazon RDS DB 인스턴스는 MySQL 5.7.x, 5.6.x 또는 MariaDB여야 합니다.
 - 원본 인스턴스가 MySQL 8.0.x인 경우 Amazon RDS DB 인스턴스는 MySQL 8.0.x여야 합니다.
 - 원본 인스턴스가 MariaDB 5.1, 5.2 또는 5.3인 경우 Amazon RDS DB 인스턴스는 MySQL 5.1.x여야 합니다.
 - 원본 인스턴스가 MariaDB 5.5 이상인 경우 Amazon RDS DB 인스턴스는 MariaDB여야 합니다.

이 섹션에서 다른 모든 상자에는 기본값을 적용합니다.

[Settings] 섹션에서 요청된 데이터베이스와 사용자 정보를 지정합니다. 완료되면 [Next]를 선택합니다.

- d. 고급 설정 구성 페이지의 네트워크 및 보안 섹션에서 Amazon EC2 인스턴스의 경우와 동일한 VPC 및 VPC 보안 그룹을 선택합니다. 이 접근 방식에서는 Amazon EC2 인스턴스와 Amazon RDS 인스턴스가 네트워크를 통해 서로를 볼 수 있습니다. [Public accessibility]를 [Yes]로 설정합니다. 이 항목의 뒷부분에 설명한 대로 소스 데이터베이스를 사용하여 복제를 설정하려면 DB 인스턴스에 공개적으로 액세스 할 수 있어야 합니다. 이 섹션에서 다른 모든 상자에는 기본값을 적용합니다.

[Database options] 섹션에서 데이터베이스 이름을 지정합니다. 이 섹션에서 다른 모든 상자에는 기본값을 적용합니다.

[Backup] 섹션에서 백업 보존 기간을 [0]으로 설정합니다. 이 섹션에서 다른 모든 상자에는 기본값을 적용합니다.

나머지 옵션에 대해서는 기본값을 수락합니다. 완료했으면 데이터베이스 생성을 선택합니다.

데이터베이스 백업을 가져온 후에야 여러 가용 영역, 백업 보존 또는 읽기 전용 복제본을 구성할 수 있습니다. 가져오기가 완료되면 프로덕션 인스턴스에 대해 원하는 방식으로 다중 AZ 및 백업 보존을 설정할 수 있습니다. Amazon RDS MySQL DB 인스턴스를 만드는 자세한 방법은 [MySQL 데이터베이스 엔진 기반 DB 인스턴스 생성 \(p. 587\)](#) 단원을 참조하십시오. Amazon RDS MariaDB DB 인스턴스를 만드는 자세한 방법은 [MariaDB 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 437\)](#) 단원을 참조하십시오.

6. Amazon RDS DB 인스턴스에 대한 기본 구성 옵션을 검토합니다. Amazon RDS Management Console의 왼쪽 탐색 창에서 Parameter Groups(파라미터 그룹)를 선택한 다음, default.mysqlx.x 또는 default.mariadbx.x 파라미터 그룹 옆에 있는 듣보기 아이콘을 선택합니다. 이 파라미터 그룹에 원하는 구성 옵션이 없는 경우 원하는 구성 옵션이 있는 다른 파라미터 그룹을 찾거나 새 파라미터 그룹을 만들니다. 파라미터 그룹 만들기에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오. 기본 설정과는 다른 파라미터 그룹을 사용하기로 했다면, 그 파라미터 그룹을 Amazon RDS DB 인스턴스와 연결합니다. 자세한 내용은 [MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 600\)](#) 또는 [MariaDB 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 449\)](#) 단원을 참조하십시오.
7. 마스터 사용자로 새 Amazon RDS DB 인스턴스에 연결하고, 인스턴스에 액세스하기 위해 필요한 서비스, 애플리케이션 및 관리자를 지원하는 데 필요한 사용자를 만듭니다. Amazon RDS DB 인스턴스의 호스트 이름은 포트 번호를 포함하지 않은 인스턴스의 Endpoint(엔드포인트) 값입니다(예: mysampled.db.claxc2oy9ak1.us-west-2.rds.amazonaws.com). Amazon RDS Management Console의 인스턴스 세부 정보에서 엔드포인트 값을 찾을 수 있습니다.
8. Amazon EC2 인스턴스에 연결합니다. 자세한 내용은 Linux용 Amazon Elastic Compute Cloud 사용 설명서의 [인스턴스에 연결](#)을 참조하십시오.
9. mysql 명령을 사용하여 Amazon EC2 인스턴스에서 원격 호스트로 Amazon RDS DB 인스턴스에 연결합니다. 다음은 그 한 예입니다.

```
mysql -h <host_name> -P 3306 -u <db_master_user> -p
```

호스트 이름은 Amazon RDS DB 인스턴스 엔드포인트의 DNS 이름입니다.

- 10 mysql 프롬프트에서 source 명령을 실행하고 데이터베이스 덤프 파일의 이름을 전달하여 데이터를 Amazon RDS DB 인스턴스로 로드합니다.

- SQL 형식인 경우 다음 명령을 사용합니다.

```
mysql> source backup.sql;
```

- 구분 기호로 분리된 텍스트 형식의 경우, 우선 데이터베이스를 만듭니다(Amazon RDS DB 인스턴스를 설정할 때 만든 기본 데이터베이스가 아닌 경우).

```
mysql> create database <database_name>;
$ mysql> use <database_name>;
```

그런 다음, 테이블을 생성합니다.

```
mysql> source <table1>.sql
$ mysql> source <table2>.sql
etc...
```

그런 다음, 데이터를 가져옵니다.

```
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...
```

성능을 개선하려면 여러 연결에서 이들 작업을 병렬로 수행하여 모든 테이블이 만들어진 다음에 동시에 로드되도록 할 수 있습니다.

Note

처음에 테이블을 덤프할 때 mysqldump와 함께 데이터 서식 옵션을 사용했다면, mysqlimport 또는 LOAD DATA LOCAL INFILE과 함께 같은 옵션을 사용하여 데이터 파일 내용을 올바로 해석해야 합니다.

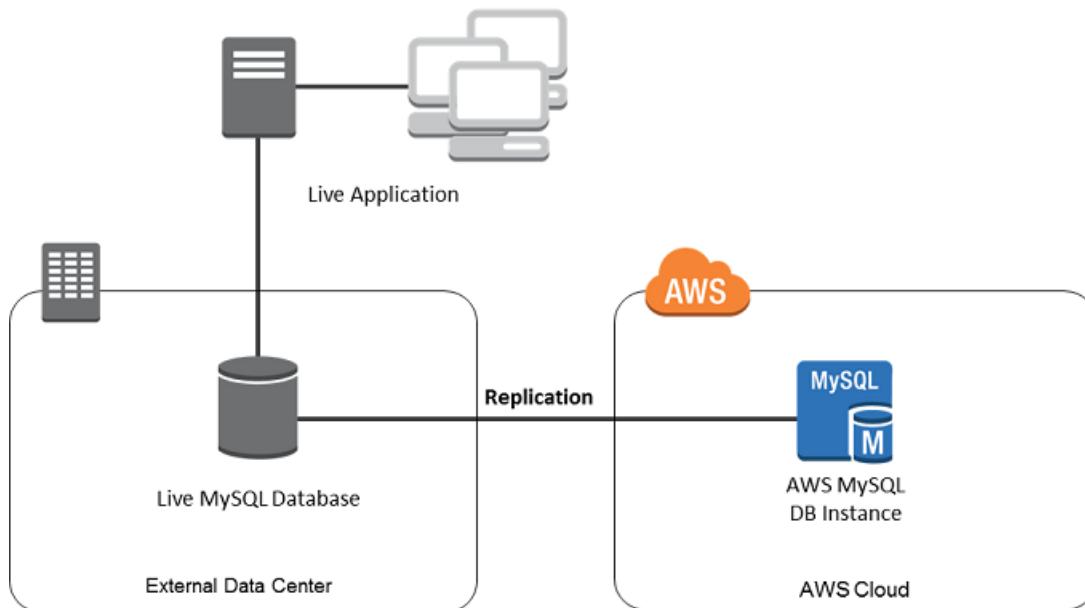
11가져온 데이터베이스에 있는 테이블 중 1개 또는 2개에 대해 간단한 SELECT 쿼리를 실행하여 가져오기에 성공했는지 확인합니다.

Note

이 절차에 사용되는 Amazon EC2 인스턴스가 더 이상 필요하지 않은 경우 EC2 인스턴스를 종료하여 Amazon AWS 리소스 사용을 줄여야 합니다. EC2 인스턴스를 종료하려면 [인스턴스 종료](#)를 참조하십시오.

외부 데이터베이스와 새 Amazon RDS DB 인스턴스 간의 복제

Amazon RDS MySQL 또는 MariaDB DB 인스턴스로 데이터를 복사하고 전송하는 데 걸린 시간 중에 원본 데이터베이스가 업데이트되었을 것입니다. 그런 경우 복제를 사용하여 복사한 데이터베이스를 원본 데이터베이스에 맞춰 최신 상태로 업데이트합니다.



Note

Amazon RDS DB 인스턴스에서 복제를 시작하는 데 필요한 권한은 제한되고 Amazon RDS 마스터 사용자는 사용할 수 없습니다. 따라서 Amazon RDS [mysql.rds_set_external_master \(p. 678\)](#) 명령 또는 [mysql.rds_set_external_master_gtid \(p. 474\)](#) 명령을 사용하여 복제를 구성한 다음, [mysql.rds_start_replication \(p. 688\)](#) 명령을 사용하여 라이브 데이터베이스와 Amazon RDS 데이터베이스 간에 복제를 시작해야 합니다.

복제를 시작하려면

앞에서 이진 로깅을 활성화하고 원본 데이터베이스에 대한 고유한 서버 ID를 설정했습니다. 이제는 라이브 데이터베이스를 복제 마스터로 포함하고 있는 복제본으로서 Amazon RDS DB 인스턴스를 설정할 수 있습니다.

1. Amazon RDS Management Console에서 Amazon RDS DB 인스턴스에 대한 VPC 보안 그룹에 원본 데이터베이스를 호스팅하는 서버의 IP 주소를 추가합니다. VPC 보안 그룹 수정에 대한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 [VPC용 보안 그룹](#)을 참조하십시오.

원본 인스턴스와 통신할 수 있도록, Amazon RDS DB 인스턴스 IP 주소에서의 연결을 허용하도록 로컬 네트워크를 구성해야 할 수도 있습니다. Amazon RDS DB 인스턴스의 IP 주소를 확인하려면 `host` 명령을 사용합니다.

```
host <RDS_MySQL_DB_host_name>
```

호스트 이름은 Amazon RDS DB 인스턴스 앤드포인트의 DNS 이름입니다(예: `myinstance.123456789012.us-east-1.rds.amazonaws.com`). Amazon RDS Management Console의 인스턴스 세부 정보에서 앤드포인트 값을 찾을 수 있습니다.

2. 선택한 클라이언트를 사용하여 원본 인스턴스에 연결하고 복제에 사용될 사용자를 만듭니다. 이 계정은 오직 복제용으로만 사용되며 보안 향상을 위해 사용자의 도메인으로 제한되어야 합니다. 다음은 예제입니다.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY '<password>';
```

3. 원본 인스턴스의 경우 복제 사용자에게 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여 합니다. 예를 들어 도메인의 'repl_user' 사용자를 위해 모든 데이터베이스에서 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여하려면 다음 명령을 실행합니다.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'  
IDENTIFIED BY '<password>';
```

4. SQL 형식을 사용하여 백업 파일을 만들었고 외부 인스턴스가 MariaDB 10.0.2 이상이 아닌 경우 그 파일의 내용을 살펴봅니다.

```
cat backup.sql
```

이 파일에는 마스터 로그 파일 이름과 위치를 포함한 CHANGE MASTER TO 설명이 포함됩니다. 이 설명은 `mysqldump`와 함께 `--master-data` 옵션을 사용할 때 백업 파일에 포함됩니다. `MASTER_LOG_FILE` 및 `MASTER_LOG_POS`의 값을 기록합니다.

```
--  
-- Position to start replication or point-in-time recovery from  
  
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
```

구분 기호로 분리된 텍스트 형식을 사용하여 백업 파일을 만들었고 외부 인스턴스가 MariaDB 10.0.2 이상이 아닌 경우 이미 [기존 데이터베이스의 백업 복사본을 만들려면 \(p. 630\)](#) 내 프로시저의 1단계에서 이진 로그 좌표를 가져옵니다.

외부 인스턴스가 MariaDB 10.0.2 이상인 경우 이미 [기존 데이터베이스의 백업 복사본을 만들려면 \(p. 630\)](#) 내 프로시저의 2단계에서 복제를 시작할 GTID를 가져옵니다.

5. Amazon RDS DB 인스턴스를 복제본으로 만들습니다. 외부 인스턴스가 MariaDB 10.0.2 이상이 아닌 경우 마스터 사용자로서 Amazon RDS DB 인스턴스에 연결하고 [mysql.rds_set_external_master \(p. 678\)](#) 명령을 사용하여 원본 데이터베이스를 복제 마스터로 식별합니다. SQL 형식 백업 파일이 있는 경우 이전 단계에서 확인한 마스터 로그 파일 이름과 마스터 로그 위치를 사용합니다. 혹은 구분 기호로 분리된 텍스트 형식을 사용한 경우 백업 파일을 만들 때 확인한 이름과 위치를 사용합니다. 다음은 예제입니다.

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306,  
'repl_user', '<password>', 'mysql-bin-changelog.000031', 107, 0);
```

외부 인스턴스가 MariaDB 10.0.2 이상인 경우 마스터 사용자로서 Amazon RDS DB 인스턴스에 연결하고 [mysql.rds_set_external_master_gtid \(p. 474\)](#) 명령을 사용하여 원본 데이터베이스를 복제 마스터로 식별합니다. [기존 데이터베이스의 백업 복사본을 만들려면 \(p. 630\)](#) 내 프로시저의 2단계에서 확인된 GTID를 사용하십시오. 다음은 예제입니다.

```
CALL mysql.rds_set_external_master_gtid ('<master_server_ip_address>', 3306,  
'ReplicationUser', '<password>', '<GTID>', 0);
```

`master_server_ip_address`는 마스터 MySQL 인스턴스의 IP 주소입니다. EC2 프라이빗 DNS 주소는 현재 지원되지 않습니다.

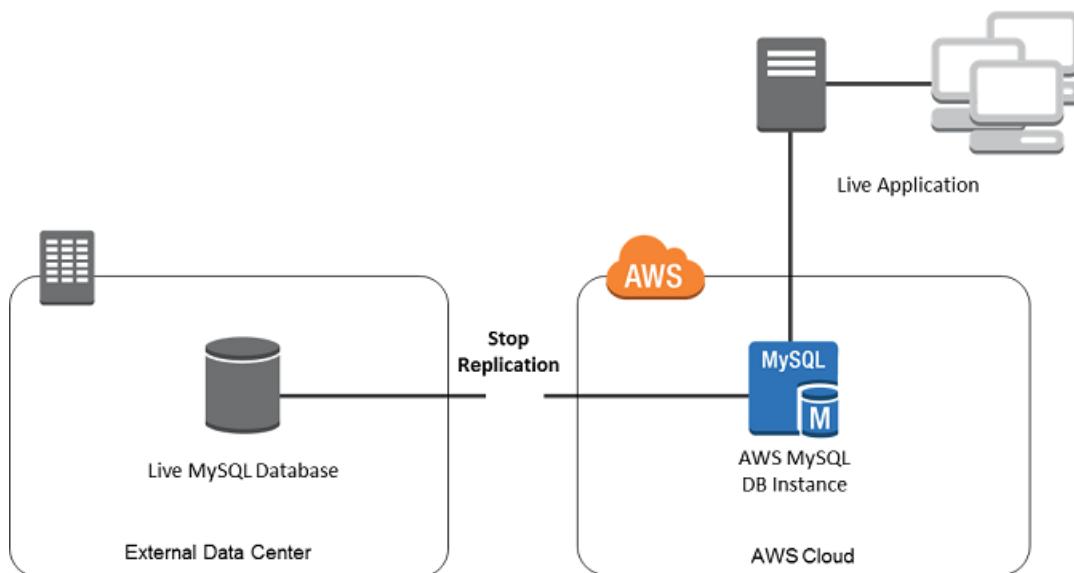
6. Amazon RDS DB 인스턴스에서 [mysql.rds_start_replication \(p. 688\)](#) 명령을 실행하여 복제를 시작합니다.

```
CALL mysql.rds_start_replication;
```

7. Amazon RDS DB 인스턴스에서 [SHOW SLAVE STATUS](#) 명령을 실행하여 복제본이 복제 마스터와 함께 최신 상태로 업데이트되는 시점을 결정합니다. `SHOW SLAVE STATUS` 명령의 결과에는 `Seconds_Behind_Master` 필드가 포함됩니다. `Seconds_Behind_Master` 필드에서 0을 반환하면 복제본이 마스터에 맞춰 최신 상태로 업데이트된 것입니다.
8. Amazon RDS DB 인스턴스가 최신 상태로 업데이트된 후, 필요할 경우 그 데이터베이스를 복원할 수 있도록 자동 백업을 활성화합니다. [Amazon RDS Management Console](#)을 사용하여 Amazon RDS DB 인스턴스에 대한 자동 백업을 활성화하거나 수정할 수 있습니다. 자세한 내용은 [백업 작업 \(p. 196\)](#) 단원을 참조하십시오.

라이브 애플리케이션을 Amazon RDS 인스턴스로 리디렉션

Amazon RDS MySQL 또는 MariaDB DB 인스턴스가 복제 마스터로 업데이트되었으면, 라이브 애플리케이션을 업데이트하여 Amazon RDS 인스턴스를 사용할 수 있습니다.



라이브 애플리케이션을 Amazon RDS MySQL 또는 MariaDB DB 인스턴스로 리디렉션하고 복제를 중지하려면

1. Amazon RDS DB 인스턴스에 대한 VPC 보안 그룹을 추가하려면 애플리케이션을 호스팅하는 서버의 IP 주소를 추가합니다. VPC 보안 그룹 수정에 대한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 [VPC용 보안 그룹](#)을 참조하십시오.
2. [SHOW SLAVE STATUS](#) 명령 결과에서 Seconds_Behind_Master 필드가 0인지 확인합니다. 이 필드가 0이라는 것은 복제본이 복제 마스터와 동일하게 최신 상태임을 나타냅니다.

```
SHOW SLAVE STATUS;
```

3. 트랜잭션이 완료되면 소스에 대한 모든 연결을 닫습니다.
4. Amazon RDS DB 인스턴스를 사용할 수 있도록 애플리케이션을 업데이트합니다. 이 업데이트에는 일반적으로 Amazon RDS DB 인스턴스의 호스트 이름과 포트, 인스턴스와 연결하기 위한 사용자 계정과 암호, 사용할 데이터베이스를 식별하기 위해 연결 설정을 변경하는 과정이 포함됩니다.
5. [mysql.rds_stop_replication \(p. 691\)](#) 명령을 사용하여 Amazon RDS 인스턴스에 대한 복제를 중지합니다.

```
CALL mysql.rds_stop_replication;
```

6. Amazon RDS DB 인스턴스가 더 이상 복제본과 같지 않도록 이 인스턴스에서 [mysql.rds_reset_external_master \(p. 685\)](#) 명령을 실행하여 복제 구성을 재설정합니다.

```
CALL mysql.rds_reset_external_master;
```

7. 다중 AZ 지원 및 읽기 전용 복제본과 같은 Amazon RDS 기능을 추가로 활성화합니다. 자세한 내용은 [Amazon RDS를 위한 고가용성\(다중 AZ\) \(p. 109\)](#) 및 [읽기 전용 복제본 작업 \(p. 135\)](#) 단원을 참조하십시오.

Note

이 절차에 사용되는 Amazon RDS 인스턴스가 더 이상 필요하지 않은 경우 RDS 인스턴스를 삭제하여 Amazon AWS 리소스 사용을 줄여야 합니다. RDS 인스턴스를 삭제하려면 [DB 인스턴스 삭제 \(p. 128\)](#) 단원을 참조하십시오.

임의의 소스에서 MySQL 또는 MariaDB DB 인스턴스로 데이터 가져오기

로드할 데이터가 1GiB보다 많거나 데이터가 MySQL 또는 MariaDB 데이터베이스 이외의 다른 위치에서 오는 경우 플랫 파일을 생성하고 mysqlimport를 사용하여 플랫 파일을 로드하는 것이 좋습니다. mysqlimport는 플랫 파일을 MySQL 또는 MariaDB로 로드할 목적으로 MySQL 및 MariaDB 클라이언트 소프트웨어와 함께 번들로 제공되는 또 하나의 명령줄 유ти리티입니다. mysqlimport에 대한 자세한 내용은 MySQL 설명서의 [mysqlimport - 데이터 가져오기 프로그램](#)을 참조하십시오.

또한 데이터 로드 전후에 대상 Amazon RDS DB 인스턴스의 DB 스냅샷을 만드는 것이 좋습니다. Amazon RDS DB 스냅샷은 DB 인스턴스를 알려진 상태로 복원하는 데 사용할 수 있는 DB 인스턴스의 완전한 백업입니다. DB 스냅샷을 시작하면 데이터베이스가 백업되는 동안 데이터베이스 인스턴스에 대한 I/O 작업이 일시 중단됩니다.

필요한 경우 로드 직전에 DB 스냅샷을 만들면 데이터베이스를 로드 이전의 상태로 복원할 수 있습니다. 로드 직후에 생성된 DB 스냅샷은 사고 발생 시 데이터를 꼭 로드하지 않아도 되도록 해주고, 새 데이터베이스 인스턴스를 시드하는 데 사용될 수도 있습니다.

아래 목록에 수행할 단계가 나와 있습니다. 각 단계에 대해서는 아래에서 자세히 설명합니다.

1. 로드할 데이터를 포함한 플랫 파일을 만듭니다.
2. 대상 DB 인스턴스에 액세스 중인 애플리케이션을 모두 종지합니다.
3. DB 스냅샷을 만듭니다.
4. Amazon RDS 자동 백업 비활성화를 고려하십시오.
5. mysqlimport를 사용하여 데이터를 로드합니다.
6. 자동 백업을 다시 활성화합니다.

1단계: 로드할 데이터를 포함한 플랫 파일 만들기

CSV(쉼표로 구분된 값)와 같은 일반적인 형식을 사용하여 로드할 데이터를 저장합니다. 각 테이블에는 자체 파일이 있어야 하며, 여러 테이블에 대한 데이터를 같은 파일에 결합할 수 없습니다. 각 파일의 이름은 그에 상응하는 테이블과 같은 이름으로 지정합니다. 파일 확장명은 원하는 대로 정할 수 있습니다. 예를 들어 테이블 이름이 "sales"라면, 파일 이름은 "sales.csv" 또는 "sales.txt"로 지정할 수 있지만 "sales_01.csv"로 지정할 수는 없습니다.

가능하면 항상 로드되는 테이블의 기본 키를 기준으로 데이터를 정렬하십시오. 그러면 로드 시간이 대폭 단축되고 데이터 스토리지 요구 사항이 최소화됩니다.

이 절차의 수행 속도와 효율성은 파일의 크기를 작게 유지하느냐에 좌우됩니다. 개별 파일의 압축되지 않은 크기가 1GiB보다 큰 경우에는 파일을 여러 개의 파일로 분할하고 각각 하나씩 따로 로드합니다.

Unix와 같은 시스템(Linux 포함)에서는 'split' 명령을 사용합니다. 예를 들어 다음 명령을 실행하면 sales.csv 파일이 1GiB 미만의 여러 파일로 분할되며, 줄 바꿈에서만 분할됩니다(-C 1024m). 새 파일의 이름은 sales.part_00, sales.part_01 등으로 지정됩니다.

```
split -C 1024m -d sales.csv sales.part_
```

다른 운영 체제에서는 이와 유사한 유ти리티를 사용할 수 있습니다.

2단계: 대상 DB 인스턴스에 액세스 중인 애플리케이션을 모두 종지

대량 로드를 시작하기 전에 로드하려는 대상 DB 인스턴스에 액세스하는 모든 애플리케이션의 활동을 중단합니다. 로드 중인 테이블 혹은 참조 테이블을 다른 세션이 수정하는 경우에는 더욱 더 중단해야 합니다. 그

그러면 로드 중에 발생하는 제약 조건 위반의 위험이 줄어들고 로드 성능이 향상됩니다. 또한 로드에 관여하지 않는 프로세스에 의한 변경 내용이 손실되지 않고 로드 직전의 시점으로 데이터베이스 인스턴스를 복원할 수 있습니다.

물론, 이것이 가능하지 않거나 유용하지 않을 수도 있습니다. 로드 전에 애플리케이션이 DB 인스턴스에 액세스하지 못하게 막을 수 없다면, 데이터의 가용성과 무결성을 보장하기 위한 일련의 단계를 수행하십시오. 필요한 구체적인 단계는 특정 사용 사례와 사이트 요구 사항에 따라 크게 달라집니다.

3단계: DB 스냅샷 만들기

아무런 데이터도 없는 새 DB 인스턴스로 데이터를 로드할 경우에는 이 단계를 건너뛰어도 됩니다. 그렇지 않을 경우, 필요하다면 DB 인스턴스의 DB 스냅샷을 만들면 DB 인스턴스를 로드 직전의 시점으로 복원할 수 있습니다. 앞서 언급한 바와 같이, DB 스냅샷을 시작하면 데이터베이스가 백업되는 동안 데이터베이스 인스턴스에 대한 I/O 작업이 몇 분간 일시 중단됩니다.

아래 예제에서는 AWS CLI `create-db-snapshot` 명령을 사용하여 AcmeRDS 인스턴스의 DB 스냅샷을 생성하고 DB 스냅샷에 "preload"라는 식별자를 지정합니다.

Linux, OS X, Unix의 경우:

```
aws rds create-db-snapshot \
--db-instance-identifier AcmeRDS \
--db-snapshot-identifier preload
```

Windows의 경우:

```
aws rds create-db-snapshot ^
--db-instance-identifier AcmeRDS ^
--db-snapshot-identifier preload
```

DB 스냅샷 기능에서 복원하는 기능을 사용하여 테스트 실행을 위한 테스트 데이터베이스 인스턴스를 만들거나 로드 중의 변경 사항을 "실행 취소"할 수도 있습니다.

DB 스냅샷에서 데이터베이스를 복원하면 모든 DB 인스턴스와 마찬가지로 고유 식별자와 엔드포인트가 있는 새 DB 인스턴스가 생성된다는 점을 꼭 명심해야 합니다. 엔드포인트를 변경하지 않고 데이터베이스 인스턴스를 복원해야 한다면, 우선 엔드포인트를 다시 사용할 수 있도록 DB 인스턴스를 삭제해야 합니다.

예를 들어 테스트 실행 또는 다른 테스트를 위한 DB 인스턴스를 만들려면 DB 인스턴스에 고유 식별자를 지정합니다. 이 예제에서는 "AcmeRDS-2"가 식별자이며 AcmeRDS-2와 연결된 엔드포인트를 사용하여 데이터베이스 인스턴스에 연결합니다.

Linux, OS X, Unix의 경우:

```
aws rds restore-db-instance-from-db-snapshot \
--db-instance-identifier AcmeRDS-2 \
--db-snapshot-identifier preload
```

Windows의 경우:

```
aws rds restore-db-instance-from-db-snapshot ^
--db-instance-identifier AcmeRDS-2 ^
--db-snapshot-identifier preload
```

기존 엔드포인트를 재사용하려면 우선 데이터베이스 인스턴스를 삭제한 다음, 복원된 데이터베이스에 같은 식별자를 지정해야 합니다.

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

예제에서는 데이터베이스 인스턴스의 최종 DB 스냅샷을 생성한 후에 삭제합니다. 이는 선택 사항이며, 권장 사항은 아닙니다.

4단계: Amazon RDS 자동 백업 비활성화 고려

Warning

특정 시점으로 복구를 수행하는 능력을 유지할 필요가 있는 경우에는 자동 백업을 비활성화하지 마십시오.

자동 백업을 비활성화하면 기존 백업이 전부 지워지므로, 자동 백업이 비활성화된 후에는 특정 시점으로 복구할 수 없습니다. 자동 백업 비활성화는 성능 최적화 수단으로서, 데이터 로드에 필수 사항은 아닙니다. 자동 백업을 비활성화해도 DB 스냅샷에는 영향을 주지 않습니다. 기존의 모든 DB 스냅샷을 계속 복원 작업에 사용할 수 있습니다.

자동 백업을 비활성화하면 로드 시간이 약 25% 단축되고 로드 중에 필요한 스토리지 공간의 양이 줄어듭니다. 아무런 데이터도 없는 새 DB 인스턴스로 데이터를 로드할 경우에는 백업을 비활성화하는 것이 로드 속도를 높이고 백업에 필요한 추가 스토리지의 사용을 피하는 손쉬운 방법입니다. 하지만 이미 데이터가 있는 DB 인스턴스로 로드할 경우에는 특정 시점으로 복구를 수행할 능력을 상실하는 데 따른 영향에 의해 백업 비활성화의 효과가 얼마나 클지 비교 검토합니다.

DB 인스턴스에서는 기본적으로 자동 백업이 활성화되어 있습니다(보존 기간은 하루). 자동 백업을 비활성화하려면 백업 보존 기간을 0으로 설정해야 합니다. 로드 후에는 백업 보존 기간을 0이 아닌 값으로 설정하여 백업을 다시 활성화할 수 있습니다. 백업을 활성화하거나 비활성화하려면 Amazon RDS가 DB 인스턴스를 종료했다가 다시 시작하여 MySQL 또는 MariaDB 로깅을 설정하거나 해제해야 합니다.

AWS CLI `modify-db-instance` 명령을 사용하여 백업 보존 기간을 0으로 설정하고 변경 사항을 즉시 적용합니다. 보존 기간을 0으로 설정하려면 DB 인스턴스를 다시 시작해야 하므로, 다시 시작 프로세스가 완료될 때까지 기다린 후 계속 진행합니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
--db-instance-identifier AcmeRDS \
--apply-immediately \
--backup-retention-period 0
```

Windows의 경우:

```
aws rds modify-db-instance ^
--db-instance-identifier AcmeRDS ^
```

```
--apply-immediately ^  
--backup-retention-period 0
```

AWS CLI `describe-db-instances` 명령으로 DB 인스턴스의 상태를 확인할 수 있습니다. 예제에는 AcmeRDS 데이터베이스 인스턴스의 상태가 표시되어 있고 열 머리글을 표시하기 위해 `--headers` 옵션이 포함되어 있습니다.

Linux, OS X, Unix의 경우:

```
aws rds describe-db-instances \  
  --db-instance-identifier AcmeRDS \  
  --headers
```

Windows의 경우:

```
aws rds describe-db-instances ^  
  --db-instance-identifier AcmeRDS ^  
  --headers
```

[Status] 열에 해당 데이터베이스를 사용할 수 있는 것으로 표시되면 계속 진행할 수 있습니다.

5단계: 데이터 로드

`mysqldump` 유ти리티를 사용하여 플랫 파일을 Amazon RDS로 로드합니다. 예제에서는 `mysqldump` 유ти리티에 이름이 "sales"이고 확장명이 "part_"로 시작되는 모든 파일을 로드하라고 명령합니다. 이는 "분할" 예제에서 만든 모든 파일을 편리하게 로드할 수 있는 방법입니다. `--compress` 옵션을 사용하여 네트워크 트래픽을 최소화합니다. `--fields-terminated-by=','` 옵션은 CSV 파일에 사용되고, `--local` 옵션은 수신 데이터가 클라우드 인스턴스에 있음을 지정합니다. `--local` 옵션이 없으면 Amazon RDS DB 인스턴스가 데이터베이스 호스트에서 데이터를 찾으므로, `--local` 옵션을 항상 지정하십시오.

Linux, OS X, Unix의 경우:

```
mysqldump --local \  
  --compress \  
  --user=username \  
  --password \  
  --host=hostname \  
  --fields-terminated-by=',' Acme sales.part_*
```

Windows의 경우:

```
mysqldump --local ^  
  --compress ^  
  --user=username ^  
  --password ^  
  --host=hostname ^  
  --fields-terminated-by=',' Acme sales.part_*
```

매우 큰 데이터 로드의 경우, 파일을 로드하는 사이에 추가 DB 스냅샷을 주기적으로 만들고 로드된 파일을 기록합니다. 문제가 발생할 경우 마지막 DB 스냅샷 시점에서 손쉽게 재개할 수 있으므로, 다시 로드하여 긴 시간을 낭비하는 상황을 피할 수 있습니다.

6단계: Amazon RDS 자동 백업 활성화

로드가 완료된 후에는 백업 보존 기간을 로드 이전의 값으로 다시 설정하여 Amazon RDS 자동 백업을 다시 활성화합니다. 앞서 언급한 것처럼, Amazon RDS가 DB 인스턴스를 다시 시작하므로 잠시 작동이 중단되는 상황에 대비하십시오.

예제에서는 AWS CLI `modify-db-instance` 명령을 사용하여 AcmeRDS DB 인스턴스에 대한 자동 백업을 활성화하고 보존 기간을 1일로 설정합니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
--db-instance-identifier AcmeRDS \
--backup-retention-period 1 \
--apply-immediately
```

Windows의 경우:

```
aws rds modify-db-instance ^
--db-instance-identifier AcmeRDS ^
--backup-retention-period 1 ^
--apply-immediately
```

MySQL 복제 작업

일반적으로 읽기 전용 복제본을 사용하여 Amazon RDS DB 인스턴스 간 복제를 구성합니다. 읽기 전용 복제본에 대한 일반적인 정보는 [읽기 전용 복제본 작업 \(p. 135\)](#) 단원을 참조하십시오. Amazon RDS MySQL의 읽기 전용 복제본 작업에 대한 자세한 내용은 [MySQL 읽기 전용 복제본 작업 \(p. 644\)](#) 단원을 참조하십시오.

Amazon RDS MySQL을 통한 복제에 GTID(전역 트랜잭션 ID)를 사용할 수 있습니다. 자세한 내용은 [Amazon RDS MySQL용 GTID 기반 복제 사용 \(p. 650\)](#) 단원을 참조하십시오.

Amazon RDS MySQL DB 인스턴스와 Amazon RDS 외부에 있는 MySQL 또는 MariaDB 인스턴스 간의 복제를 설정할 수도 있습니다. 외부 소스를 사용하여 복제를 구성하는 방법에 대한 자세한 내용은 [Amazon RDS 외부에서 실행 중인 MySQL 또는 MariaDB 인스턴스를 사용한 복제 \(p. 655\)](#) 단원을 참조하십시오.

이러한 복제 옵션의 경우 행 기반 복제, 문 기반 복제 또는 혼합 복제를 사용할 수 있습니다. 행 기반 복제는 SQL 문으로 인해 변경된 행만 복제합니다. 문 기반 복제는 전체 SQL 문을 복제합니다. 혼합 복제는 가능한 경우 문 기반 복제를 사용하지만, 문 기반 복제에 안전하지 않은 SQL 문이 실행될 경우 행 기반 복제로 전환합니다. 대부분의 경우 혼합 복제가 권장됩니다. DB 인스턴스의 이진 로그 형식은 복제가 행 기반인지, 문 기반인지, 혼합인지 결정합니다. 이진 로그 형식 설정에 대한 자세한 내용은 [이진 로깅 형식 \(p. 324\)](#) 단원을 참조하십시오.

Note

Amazon RDS 외부에 있는 MySQL 또는 MariaDB 인스턴스에서 데이터베이스를 가져오거나 그런 인스턴스로 데이터베이스를 내보내도록 복제를 구성할 수 있습니다. 자세한 내용은 [자동 중지 시간을 단축하여 Amazon RDS MySQL MariaDB DB 인스턴스로 데이터 가져오기 \(p. 628\)](#) 및 [복제를 사용하여 MySQL DB 인스턴스에서 데이터 내보내기 \(p. 660\)](#) 단원을 참조하십시오.

주제

- [MySQL 읽기 전용 복제본 작업 \(p. 644\)](#)
- [Amazon RDS MySQL용 GTID 기반 복제 사용 \(p. 650\)](#)
- [Amazon RDS 외부에서 실행 중인 MySQL 또는 MariaDB 인스턴스를 사용한 복제 \(p. 655\)](#)

MySQL 읽기 전용 복제본 작업

이 단원에는 Amazon RDS MySQL의 읽기 전용 복제본 작업에 대한 자세한 정보가 포함되어 있습니다. 읽기 전용 복제본에 대한 일반적인 정보와 사용 지침은 [읽기 전용 복제본 작업 \(p. 135\)](#) 단원을 참조하십시오.

주제

- MySQL을 사용한 읽기 전용 복제본 구성 (p. 645)
- MySQL을 사용한 지연 복제 구성 (p. 646)
- MySQL을 사용한 읽기 전용 복제본 업데이트 (p. 648)
- MySQL을 사용한 다중 AZ 읽기 전용 복제본 배포 (p. 648)
- MySQL 읽기 전용 복제본 모니터링 (p. 648)
- MySQL 읽기 전용 복제본을 사용한 복제 시작 및 중지 (p. 648)
- MySQL을 사용한 읽기 전용 복제본 삭제 (p. 649)
- MySQL 읽기 전용 복제본의 문제 해결 (p. 649)

MySQL을 사용한 읽기 전용 복제본 구성

임의의 MySQL DB 인스턴스를 복제 원본으로 사용하려면 백업 보존 기간을 0이 아닌 다른 값으로 설정하여 원본 DB 인스턴스의 자동 백업을 활성화해야 합니다. 이러한 요건은 다른 읽기 전용 복제본의 원본 DB 인스턴스인 읽기 전용 복제본에도 똑같이 적용됩니다. 자동 백업은 MySQL 5.6 이상 버전을 실행 중인 읽기 전용 복제본에 한해 지원됩니다. MySQL DB 인스턴스에 대해 바이너리 로그 좌표를 기반으로 복제를 구성할 수 있습니다.

Amazon RDS MySQL 버전 5.7.23 및 이후 MySQL 5.7 버전에서는 GTID(전역 트랜잭션 ID)를 사용하여 복제를 구성할 수 있습니다. 자세한 내용은 [Amazon RDS MySQL용 GTID 기반 복제 사용 \(p. 650\)](#) 단원을 참조 하십시오.

DB 인스턴스 하나에서 최대 5개까지 읽기 전용 복제본을 생성할 수 있습니다. 효과적인 복제를 위해서는 읽기 전용 복제본도 각각 원본 DB 인스턴스와 동일한 용량의 컴퓨팅 파워와 스토리지 리소스를 가져야 합니다. 원본 DB 인스턴스를 확장하면 읽기 전용 복제본도 확장해야 합니다.

읽기 전용 복제본이 MySQL 5.6 이상 버전을 실행 중인 경우에는 읽기 전용 복제본을 다른 읽기 전용 복제본의 원본 DB 인스턴스로 지정할 수 있습니다. 예를 들어 MyDBInstance에서 ReadReplica1을 생성한 후 다시 ReadReplica1에서 ReadReplica2를 생성할 수 있습니다. MyDBInstance의 업데이트는 ReadReplica1에도 복제되며, 다시 한 번 ReadReplica1에서 ReadReplica2로 복제됩니다. 하지만 복제 체인에 포함될 수 있는 인스턴스는 최대 4개로 제한됩니다. 예를 들어 MySourceDBInstance에서 ReadReplica1을 생성한 후 다시 ReadReplica1에서 ReadReplica2를 생성하고, ReadReplica2에서 ReadReplica3를 생성할 수 있지만 ReadReplica3에서 ReadReplica4를 생성할 수는 없습니다.

MySQL 읽기 전용 복제본을 다른 읽기 전용 복제본으로 복제할 수 있도록 승격한 경우 이 읽기 전용 복제본은 활성화 상태를 유지합니다. MyDBInstance1이 MyDBInstance2로, 그리고 MyDBInstance2가 다시 MyDBInstance3으로 복제된다고 예를 들어보겠습니다. 이때 MyDBInstance2를 승격하는 경우 MyDBInstance1에서 MyDBInstance2로는 더 이상 복제가 이루어지지 않습니다. 하지만 MyDBInstance2는 여전히 MyDBInstance3으로 복제됩니다.

Amazon RDS MySQL 5.6 이상 읽기 전용 복제본에서 자동 백업을 활성화하려면 먼저 읽기 전용 복제본을 생성한 다음 자동 백업이 활성화되도록 읽기 전용 복제본을 변경해야 합니다.

읽기 전용 복제본이 원본 인스턴스의 한계치인 5개를 벗어나지 않는 경우에 한해 다수의 읽기 전용 복제본을 동시에 실행하거나, 동일한 원본 DB 인스턴스를 참조하는 작업은 생성 또는 삭제할 수도 있습니다.

MyISAM을 사용하는 MySQL DB 인스턴스 준비

MySQL DB 인스턴스가 MyISAM 같은 비트랜잭션 엔진을 사용할 경우 다음과 같이 읽기 전용 복제본을 설정해야 합니다. 이 단계는 읽기 전용 복제본이 고객의 데이터와 동일한 사본이 되도록 하기 위해 필요합니다. 그러나 모든 테이블이 InnoDB와 같은 트랜잭션 엔진을 사용하는 경우에는 이 단계가 필요 없습니다.

1. 원본 DB 인스턴스에서 비트랜잭션 테이블의 모든 데이터 조작 언어(DML) 및 데이터 정의 언어(DDL) 작업을 중지하고 완료될 때까지 기다립니다. SELECT 문은 계속해서 실행할 수 있습니다.

2. 원본 DB 인스턴스의 테이블을 플러시한 후 잠금니다.
3. 아래 방법 중 한 가지를 사용하여 읽기 전용 복제본을 생성합니다.
4. 예를 들어 `DescribeDBInstances` API 작업 등을 사용하여 읽기 전용 복제본의 생성 프로세스를 확인합니다. 읽기 전용 복제본이 생성되면 원본 DB 인스턴스의 테이블 잠금을 해제하고 정상적인 데이터베이스 작업을 재개합니다.

MySQL을 사용한 지연 복제 구성

지연 복제를 재해 복구를 위한 전략으로 사용할 수 있습니다. 지연 복제를 사용하여 소스에서 읽기 전용 복제본으로의 복제를 지연할 최소 시간(초)을 지정합니다. 재해 발생 시(예: 실수로 테이블 삭제) 다음 단계를 완료하여 재해로부터 빠르게 복구할 수 있습니다.

- 재해를 일으킨 변경 사항이 읽기 전용 복제본으로 전송되기 이전에 읽기 전용 복제본에 대한 복제를 중지합니다.
`mysql.rds_stop_replication` (p. 691) 저장 프로시저를 사용하여 복제를 중지합니다.
- 복제를 시작하고 로그 파일 위치에서 복제가 자동으로 중지되도록 지정합니다.
`mysql.rds_start_replication_until` (p. 689) 저장 프로시저를 사용하여 재해 직전 위치를 지정합니다.
- 읽기 전용 복제본을 둑립 DB 인스턴스로 승격 (p. 139)의 지침에 따라 읽기 전용 복제본을 새 소스 DB 인스턴스로 승격합니다.

Note

- Amazon RDS MySQL 5.7에서는 지연 복제가 MySQL 5.7.22 이상에 대해 지원됩니다. Amazon RDS MySQL 5.6에서는 지연 복제가 MySQL 5.6.40 이상에 대해 지원됩니다. Amazon RDS MySQL 8.0에서는 지연 복제를 지원하지 않습니다.
- 저장 프로시저를 사용하여 지연 복제를 구성해야 합니다. AWS Management 콘솔, AWS CLI 또는 Amazon RDS API를 사용하여 지연 복제를 구성할 수 없습니다.
- Amazon RDS MySQL 5.7.23 및 이후 MySQL 5.7 버전에서는 지연 복제 구성에서 GTID 기반 복제를 사용할 수 있습니다. GTID 기반 복제를 사용하는 경우 `mysql.rds_start_replication_until` (p. 689) 저장 프로시저 대신 `mysql.rds_start_replication_until_gtid` (p. 690) 저장 프로시저를 사용하십시오. GTID 기반 복제에 대한 자세한 내용은 [Amazon RDS MySQL용 GTID 기반 복제 사용](#) (p. 650) 단원을 참조하십시오.

주제

- 읽기 전용 복제본 생성 중 지연 복제 구성 (p. 646)
- 기존 읽기 전용 복제본에 대한 지연 복제 수정 (p. 647)
- 읽기 전용 복제본에 대한 복제를 중지할 위치 설정 (p. 647)

읽기 전용 복제본 생성 중 지연 복제 구성

DB 인스턴스에서 향후에 생성되는 읽기 전용 복제본에 대한 지연 복제를 구성하려면 `mysql.rds_set_configuration` (p. 695) 파라미터와 함께 `target_delay` 저장 프로시저를 실행합니다.

읽기 전용 복제본 생성 중에 지연 복제를 구성하려면

1. MySQL 클라이언트를 사용하여 읽기 전용 복제본에 대한 원본 MySQL DB 인스턴스에 마스터 사용자로 연결합니다.
2. `mysql.rds_set_configuration` (p. 695) 파라미터와 함께 `target_delay` 저장 프로시저를 실행합니다.

예를 들어, 현재 DB 인스턴스에서 생성되는 모든 읽기 전용 복제본에 대해 1시간(3,600초) 이상 복제를 지연하도록 지정하려면 다음 저장 프로시저를 실행합니다.

```
call mysql.rds_set_configuration('target delay', 3600);
```

Note

이 저장 프로시저를 실행한 후 AWS CLI 또는 Amazon RDS API를 사용하여 생성하는 모든 읽기 전용 복제본은 지정된 시간(초)만큼 복제를 지연하도록 구성됩니다.

기존 읽기 전용 복제본에 대한 지연 복제 수정

기존 읽기 전용 복제본에 대한 지연 복제를 수정하려면 [mysql.rds_set_source_delay \(p. 687\)](#) 저장 프로시저를 실행합니다.

기존 읽기 전용 복제본에 대한 지연 복제를 수정하려면

1. MySQL 클라이언트를 사용하여 읽기 전용 복제본에 마스터 사용자로 연결합니다.
2. [mysql.rds_stop_replication \(p. 691\)](#) 저장 프로시저를 사용하여 복제를 중지합니다.
3. [mysql.rds_set_source_delay \(p. 687\)](#) 저장 프로시저를 실행합니다.

예를 들어, 읽기 전용 복제본에 대한 복제를 1시간(3,600초) 이상 복제를 지연하도록 지정하려면 다음 저장 프로시저를 실행합니다.

```
call mysql.rds_set_source_delay(3600);
```

4. [mysql.rds_start_replication \(p. 688\)](#) 저장 프로시저를 사용하여 복제를 시작합니다.

읽기 전용 복제본에 대한 복제를 중지할 위치 설정

읽기 전용 복제본에 대한 복제를 중지한 이후에 [mysql.rds_start_replication_until \(p. 689\)](#) 저장 프로시저를 사용하여 복제를 시작한 다음 지정된 바이너리 로그 파일 위치에서 복제를 중지할 수 있습니다.

읽기 전용 복제본에 대한 복제를 시작하고 지정된 위치에서 복제를 중지하려면

1. MySQL 클라이언트를 사용하여 원본 MySQL DB 인스턴스에 마스터 사용자로 연결합니다.
2. [mysql.rds_start_replication_until \(p. 689\)](#) 저장 프로시저를 실행합니다.

다음 예제에서는 복제를 시작하고 120 바이너리 로그 파일의 mysql-bin-changelog.000777 위치에 도달할 때까지 변경 사항을 복제합니다. 재해 복구 시나리오에서 120이 재해 직전 위치라고 가정합니다.

```
call mysql.rds_start_replication_until(
    'mysql-bin-changelog.000777',
    120);
```

중지 지점에 도달하면 복제가 자동으로 중지됩니다. Replication has been stopped since the replica reached the stop point specified by the rds_start_replication_until stored procedure RDS 이벤트가 생성됩니다.

복제가 중지된 후 재해 복구 시나리오에서 [읽기 전용 복제본을 드립 DB 인스턴스로 승격 \(p. 139\)](#) 읽기 전용 복제본을 새 소스 DB 인스턴스로 승격할 수 있습니다. 읽기 전용 복제본 승격에 대한 자세한 내용은 [읽기 전용 복제본을 드립 DB 인스턴스로 승격 \(p. 139\)](#) 단원을 참조하십시오.

MySQL을 사용한 읽기 전용 복제본 업데이트

읽기 전용 복제본은 읽기 쿼리를 지원하도록 설계되었지만 때때로 업데이트가 필요할 수 있습니다. 예를 들어 특정 유형의 쿼리가 복제본에 대한 액세스를 최저거화하기 위해 인덱스를 추가해야 할 경우가 있습니다. 업데이트를 활성화하려면 읽기 전용 복제본의 DB 파라미터 그룹에서 `read_only` 파라미터를 0으로 설정합니다. 읽기 복제본이 소스 DB 인스턴스와 호환되지 않으면 문제가 발생할 수 있으므로 읽기 복제본에서 읽기 전용을 비활성화할 때는 주의하십시오. 가능한 빨리 `read_only` 파라미터의 값을 다시 1로 변경하십시오.

MySQL을 사용한 다중 AZ 읽기 전용 복제본 배포

읽기 전용 복제본은 단일 AZ 또는 다중 AZ DB 인스턴스 배포를 통해서도 생성할 수 있습니다. 다중 AZ 배포는 중요 데이터의 내구성과 가용성을 개선하는데 효과적이지만 읽기 전용 쿼리를 실행하는 데 보조로 사용할 수는 없습니다. 그 대신 읽기 전용 쿼리 부하를 줄일 목적으로 트래픽이 많은 다중 AZ DB 인스턴스에서 읽기 전용 복제본을 생성할 수 있습니다. 다중 AZ 배포의 원본 인스턴스가 보조 인스턴스로 장애 조치된 경우에는 연결된 모든 읽기 전용 복제본이 자동으로 전환되어 보조(이제는 기본) 인스턴스를 복제 원본으로 사용합니다. 자세한 내용은 [Amazon RDS를 위한 고가용성\(다중 AZ\) \(p. 109\)](#) 단원을 참조하십시오.

읽기 전용 복제본을 Multi-AZ DB 인스턴스를 생성할 수 있습니다. Amazon RDS는 복제본에 대한 장애 조치 지원을 위해 다른 가용 영역에 복제본의 대기를 생성합니다. 읽기 전용 복제본을 다중 AZ DB 인스턴스로 생성하는 작업은 원본 데이터베이스가 다중 AZ DB 인스턴스인지 여부와는 독립적입니다.

MySQL 읽기 전용 복제본 모니터링

MySQL 읽기 전용 복제본의 경우 Amazon RDS ReplicaLag 지표를 보면서 Amazon CloudWatch의 복제 지연을 모니터링할 수 있습니다. ReplicaLag 지표는 `SHOW SLAVE STATUS` 명령의 `Seconds_Behind_Master` 필드의 값을 보고합니다.

이렇게 MySQL에서 복제 지연이 발생하는 공통 원인은 다음과 같습니다.

- 네트워크 중단.
- 읽기 전용 복제본에 관한 다양한 인덱스를 할당하여 테이블에 쓰기 작업 중일 때. 읽기 전용 복제본에 `read_only` 파라미터가 0으로 설정된 경우 읽기 전용 복제본이 소스 DB 인스턴스와 호환되지 않으면 복제가 중단될 수 있습니다. 읽기 전용 복제본에 유지 관리 작업을 수행 한 후에는 `read_only` 파라미터를 다시 1로 설정하는 것이 좋습니다.
- MyISAM과 같은 비트랜잭션 스토리지 엔진 사용. 복제는 MySQL의 InnoDB 스토리지 엔진에서만 지원됩니다.

ReplicaLag 지표가 0에 도달하면 복제본이 원본 DB 인스턴스를 따라잡은 것입니다. ReplicaLag 지표가 -1을 반환하는 경우 복제가 현재 활성이 아닙니다. `ReplicaLag = -1`은 `Seconds_Behind_Master = NULL`과 동등합니다.

MySQL 읽기 전용 복제본을 사용한 복제 시작 및 중지

Amazon RDS DB 인스턴스에서는 시스템에 저장된 프로시저인 [mysql.rds_stop_replication \(p. 691\)](#)과(와) [mysql.rds_start_replication \(p. 688\)](#)을 호출하여 복제 프로세스를 종료하거나 재시작할 수 있습니다. 대용량 인덱스를 생성하는 등 오랜 시간이 걸리는 작업에서 두 Amazon RDS 인스턴스를 서로 복제할 때도 이런 방법이 가능합니다. 또한 데이터베이스를 가져오거나 내보낼 때도 복제를 종료하거나 시작할 필요가 있습니다. 자세한 내용은 [자동 중지 시간을 단축하여 Amazon RDS MySQL MariaDB DB 인스턴스로 데이터 가져오기 \(p. 628\)](#) 및 [복제를 사용하여 MySQL DB 인스턴스에서 데이터 내보내기 \(p. 660\)](#) 단원을 참조하십시오.

수동으로, 혹은 복제 오류로 인해 연속하여 30일 이상 복제가 중단된 경우에는 Amazon RDS가 소스 DB 인스턴스와 모든 읽기 전용 복제본 사이의 복제를 종료합니다. 이렇게 하는 이유는 소스 DB 인스턴스에 대한 스토리지 요건 증가와 장애 조치의 장기화를 방지하기 위해서입니다. 이후에도 읽기 전용 복제본 DB 인스턴스는 여전히 사용할 수 있습니다. 하지만 읽기 전용 복제본에서 필요한 이진 로그가 복제 종료와 함께 소스 DB 인스턴스에서 삭제되어 복제를 재개할 수 없습니다. 소스 DB 인스턴스에서 복제를 재개하려면 새로운 읽기 전용 복제본을 생성해야 합니다.

MySQL을 사용한 읽기 전용 복제본 삭제

읽기 전용 복제본을 삭제할 때는 DB 인스턴스를 삭제할 때와 똑같은 메커니즘으로 명시적이어야 합니다. 원본 DB 인스턴스만 삭제하고 복제본을 삭제하지 않으면 각 복제본은 독립 실행형 DB 인스턴스로 승격됩니다.

MySQL 읽기 전용 복제본의 문제 해결

MySQL DB 인스턴스에서는 경우에 따라 읽기 전용 복제본을 보조 인스턴스로 전환하지 못할 수도 있습니다. 이는 일부 binlog 이벤트가 오류로 인해 플러시되지 않기 때문입니다. 이때는 수동으로 읽기 전용 복제본을 삭제한 후 다시 생성해야 합니다. 다음 동적 변수 값을 설정하여 발생하는 이러한 가능성을 줄일 수 있습니다: `sync_binlog=1`, `innodb_flush_log_at_trx_commit=1` 및 `innodb_support_xa=1`. 단, 이 설정은 성능 감소의 원인이 될 수도 있으므로 변경 사항을 프로덕션 환경에 적용하기 전에 그 효과를 테스트하는 것이 좋습니다. MySQL 5.5의 경우, `sync_binlog`은 기본적으로 0입니다. 하지만 MySQL 5.6 이상에서는 문제가 발생할 가능성이 적습니다. 이런 파라미터가 기본적으로 권장된 값으로 모두 설정되어 있기 때문입니다.

MySQL의 복제본 기술은 비동기적입니다. 간혹 원본 DB 인스턴스에서 `BinLogDiskUsage`이 증가하면 읽기 전용 복제본의 `ReplicaLag`가 예상되는 이유도 비동기식이기 때문입니다. 예를 들어 원본 DB 인스턴스에 대해 대량의 쓰기 작업이 동시에 발생할 수 있습니다. 반대로 읽기 전용 복제본에 대한 쓰기 작업은 단일 I/O 스레드를 사용하기 때문에 연이어 차례로 발생합니다. 이로 인해 원본 인스턴스와 읽기 전용 복제본 사이에 지연 시간이 있기 마련입니다. 읽기 전용 복제본에 대한 자세한 내용은 MySQL 문서에서 [Replication Implementation Details](#)를 참조하십시오.

원본 DB 인스턴스와 뒤이어 일어나는 읽기 전용 복제본의 업데이트 간 지연 시간을 줄일 수 있는 방법이 몇 가지 있으며, 다음과 같습니다.

- 읽기 전용 복제본의 크기를 조정하여 원본 DB 인스턴스에 버금가는 스토리지 크기와 DB 인스턴스 클래스를 할당합니다.
- 원본 DB 인스턴스와 읽기 전용 복제본에 사용되는 DB 파라미터 그룹의 파라미터 설정이 서로 호환되는지 확인합니다. 자세한 정보와 예는 이번 섹션 후반의 `max_allowed_packet` 파라미터 관련 설명을 참조하십시오.

Amazon RDS는 읽기 전용 복제본의 복제 상태를 모니터링하고, 어떤 이유로든 복제가 중지되는 경우 읽기 전용 복제본 인스턴스의 `Replication State` 필드를 `Error`로 업데이트합니다. 읽기 전용 복제본에 대한 DML 쿼리가 원본 DB 인스턴스의 업데이트와 충돌하는 경우가 한 예가 될 수 있습니다.

MySQL 엔진에서 발생하는 관련 오류에 대한 세부 정보는 `Replication Error` 필드에서 다시 확인할 수 있습니다. [RDS-EVENT-0045 \(p. 295\)](#), [RDS-EVENT-0046 \(p. 295\)](#) 및 [RDS-EVENT-0047 \(p. 294\)](#)을 포함하여 읽기 전용 복제본의 상태를 표시하는 이벤트도 생성됩니다. 이벤트와 이벤트 구독에 대한 자세한 내용은 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오. MySQL 오류 메시지가 반환되는 경우에는 [MySQL 오류 메시지 문서](#)에서 오류 번호를 확인하십시오.

복제 오류의 원인이 되는 공통적인 문제를 하나 꼽으라고 하면 읽기 전용 복제본의 `max_allowed_packet` 파라미터 값이 원본 DB 인스턴스의 `max_allowed_packet` 파라미터 값보다 작을 때입니다. `max_allowed_packet` 파라미터는 DB 파라미터 그룹에서 사용자가 직접 설정할 수 있는 파라미터로서 데이터베이스에서 실행할 수 있는 DML 코드의 최대 크기를 지정하는 데 사용됩니다. 경우에 따라 원본 DB 인스턴스와 연결된 DB 파라미터 그룹에서 `max_allowed_packet` 파라미터 값이 원본의 읽기 전용 복제본과 연결된 DB 파라미터 그룹에서 `max_allowed_packet` 파라미터 값보다 작습니다. 이러한 경우에는 복제 프로세스에서 오류(패킷이 'max_allowed_packet' 바이트보다 큼)가 발생하여 복제가 중단될 수도 있습니다. 원본 및 읽기 전용 복제본이 같은 `max_allowed_packet` 파라미터 값을 가진 DB 파라미터 그룹을 사용하도록 하여 이 오류를 수정할 수 있습니다.

이밖에 복제 오류의 원인이 되는 공통적인 상황은 다음과 같습니다.

- 읽기 전용 복제본의 테이블에 쓰기 작업을 실행하는 경우. 소스 DB 인스턴스의 인덱스와 다른 읽기 전용 복제본에 인덱스를 만드는 경우 인덱스를 만들려면 `read_only` 파라미터를 0으로 설정해야 합니다. 읽기

전용 복제본의 테이블에 쓰는 경우, 읽기 전용 복제본이 소스 DB 인스턴스와 호환되지 않으면 복제가 중단될 수 있습니다. 읽기 전용 복제본에 유지 관리 작업을 수행 한 후에는 `read_only` 파라미터를 다시 1로 설정하는 것이 좋습니다.

- MyISAM 같은 비트랜잭션 스토리지 엔진을 사용할 때. 읽기 전용 복제본에는 트랜잭션 스토리지 엔진이 필요합니다. 복제는 MySQL의 InnoDB 스토리지 엔진에서만 지원됩니다.
- `SYSDATE()`와 같이 안전하지 않은 비결정적 쿼리를 사용하는 경우. 자세한 내용은 [Determination of Safe and Unsafe Statements in Binary Logging](#)을 참조하십시오.

오류를 건너뛰어도 안전하다고 판단될 경우에는 [현재 복제 오류 넘어가기 \(p. 670\)](#) 섹션에 설명한 단계를 따르십시오. 그 밖에 읽기 전용 복제본을 삭제하고 엔드포인트가 이전 읽기 전용 복제본의 엔드포인트와 동일하게 유지되도록 같은 DB 인스턴스 식별자를 사용하여 인스턴스를 생성할 수도 있습니다. 복제 오류가 해결되면 `Replication State`가 `replicating`으로 변경됩니다.

Amazon RDS MySQL용 GTID 기반 복제 사용

아래에서는 Amazon RDS MySQL DB 인스턴스 간 이진 로그(binlog) 복제를 통해 전역 트랜잭션 식별자(GTID)를 사용하는 방법을 배울 수 있습니다.

binlog 복제를 사용하고 있지만 MySQL을 사용한 GTID 기반 복제에 대해 잘 알지 못하는 경우 MySQL 설명서의 [전역 트랜잭션 식별자를 사용한 복제](#)에서 배경 정보를 참조하십시오.

Note

GTID 기반 복제는 RDS MySQL 버전 5.7.23 및 이후 MySQL 5.7 버전에서 지원됩니다. 복제 구성의 모든 RDS MySQL DB 인스턴스는 이 요구 사항을 충족해야 합니다. RDS MySQL 5.5, 5.6 또는 8.0에서는 GTID 기반 복제가 지원되지 않습니다.

주제

- [전역 트랜잭션 식별자\(GTID\) 개요 \(p. 650\)](#)
- [GTID 기반 복제 파라미터 \(p. 651\)](#)
- [새 읽기 전용 복제본에 대한 GTID 기반 복제 구성 \(p. 652\)](#)
- [기존 읽기 전용 복제본에 대한 GTID 기반 복제 구성 \(p. 652\)](#)
- [읽기 전용 복제본을 사용하는 RDS MySQL DB 인스턴스에 대해 GTID 기반 복제 비활성화 \(p. 653\)](#)

Note

외부 데이터베이스로의 GTID 기반 복제를 구성하는 방법에 대한 자세한 내용은 [Amazon RDS 외부에서 실행 중인 MySQL 또는 MariaDB 인스턴스를 사용한 복제 \(p. 655\)](#) 단원을 참조하십시오.

전역 트랜잭션 식별자(GTID) 개요

전역 트랜잭션 ID(GTIDs) are unique identifiers generated for committed MySQL transactions. GTID를 사용해 binlog 복제 관련 문제를 더 간편하게 해결할 수 있습니다.

MySQL은 binlog 복제에 다음 두 가지 유형의 트랜잭션을 사용합니다.

- GTID 트랜잭션 – GTID로 식별되는 트랜잭션.
- 익명 트랜잭션 – GTID가 할당되지 않은 트랜잭션.

복제 구성의 GTID는 모든 DB 인스턴스에서 고유합니다. GTID를 사용하면 로그 파일 위치를 참조할 필요가 없기 때문에 복제 구성이 간편해집니다. 또한 GTID를 사용하면 복제된 트랜잭션을 추적하고 마스터 및 복제 본이 일치하는지를 쉽게 확인할 수 있습니다.

GTID 기반 복제를 사용하여 Amazon RDS MySQL 읽기 전용 복제본 또는 외부 MySQL 데이터베이스로 데이터를 복제할 수 있습니다. RDS MySQL 읽기 전용 복제본의 경우 읽기 전용 복제본을 새로 만들 때 GTID

기반 복제를 구성할 수 있습니다. 또는 기존 읽기 전용 복제본을 GTID 기반 복제를 사용하도록 변환할 수 있습니다.

RDS MySQL을 사용하여 지연된 복제 구성에서 GTID 기반 복제를 사용할 수도 있습니다. 자세한 내용은 [MySQL을 사용한 지연 복제 구성 \(p. 646\)](#) 단원을 참조하십시오.

GTID 기반 복제 파라미터

다음 파라미터를 사용하여 GTID 기반 복제를 구성할 수 있습니다.

파라미터	유효한 값	설명
gtid_mode	OFF, OFF_PERMISSIVE, ON_PERMISSIVE, ON	<p>OFF는 새 트랜잭션을 익명 트랜잭션(GTID가 없음)으로 지정하며, 트랜잭션을 복제하려면 익명이어야 합니다.</p> <p>OFF_PERMISSIVE는 새 트랜잭션을 익명 트랜잭션(GTID가 없음)으로 지정하지만, 모든 트랜잭션을 복제할 수 있습니다.</p> <p>ON_PERMISSIVE는 새 트랜잭션을 GTID 트랜잭션으로 지정하지만, 모든 트랜잭션을 복제할 수 있습니다.</p> <p>ON은 새 트랜잭션을 GTID 트랜잭션으로 지정하고, 트랜잭션을 복제하려면 GTID 트랜잭션이어야 합니다.</p>
enforce_gtid_consistency	OFF, ON, WARN	<p>OFF는 트랜잭션이 GTID 일관성을 위반하는 것을 허용합니다.</p> <p>ON은 트랜잭션이 GTID 일관성을 위반하지 않도록 합니다.</p> <p>WARN은 트랜잭션이 GTID 일관성을 위반하는 것을 허용하지만, 위반이 발생할 경우 경고를 생성합니다.</p>

GTID 기반 복제의 경우 DB 인스턴스 또는 읽기 전용 복제본의 파라미터 그룹에 대해 이 설정을 사용하십시오.

- ON 및 ON_PERMISSIVE는 RDS DB 인스턴스 또는 Aurora MySQL 클러스터에서 밖으로 복제하는 경우에만 적용됩니다. 이 두 값으로 인해 외부 데이터베이스로 복제되는 트랜잭션에 대해 RDS DB 인스턴스 또는 Aurora DB 클러스터가 GTID를 사용하게 됩니다. ON은 외부 데이터베이스에서도 GTID 기반 복제를 사용할 것을 요구합니다. ON_PERMISSIVE로 인해 외부 데이터베이스에서 GTID 기반 복제는 선택 사항이 됩니다.
- OFF_PERMISSIVE가 설정된 경우 이는 RDS DB 인스턴스 또는 Aurora DB 클러스터가 외부 데이터베이스에서 안으로 복제하는 것을 수락할 수 있음을 뜻합니다. 외부 데이터베이스가 GTID 기반 복제를 사용하는 사용하지 않는 이렇게 할 수 있습니다.
- OFF가 설정된 경우 이는 RDS DB 인스턴스 또는 Aurora DB 클러스터가 GTID 기반 복제를 사용하지 않는 외부 데이터베이스에서 안으로 복제하는 것만을 수락할 수 있음을 뜻합니다.

파라미터 그룹에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

Note

AWS Management 콘솔 콘솔에서 gtid_mode 파라미터는 gtid-mode로 표시됩니다.

새 읽기 전용 복제본에 대한 GTID 기반 복제 구성

RDS MySQL DB 인스턴스에 대해 GTID 기반 복제를 활성화하면 DB 인스턴스의 읽기 전용 복제본에 대해 GTID 기반 복제가 자동으로 구성됩니다.

새 읽기 전용 복제본에 대해 GTID 기반 복제를 활성화하려면

1. DB 인스턴스와 연결된 파라미터 그룹에서 다음과 같이 파라미터가 설정되어 있는지 확인합니다.

- `gtid_mode - ON` 또는 `ON_PERMISSIVE`
- `enforce_gtid_consistency - ON`

파라미터 그룹을 사용한 구성 파라미터 설정에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

2. DB 인스턴스의 파라미터 그룹을 변경한 경우 DB 인스턴스를 재부팅하십시오. 이 작업을 수행하는 방법에 대한 자세한 내용은 [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#) 단원을 참조하십시오.
3. DB 인스턴스의 읽기 전용 복제본을 한 개 이상 만듭니다. 이 작업을 수행하는 방법에 대한 자세한 내용은 [읽기 전용 복제본 만들기 \(p. 138\)](#) 단원을 참조하십시오.

Amazon RDS는 `MASTER_AUTO_POSITION`를 사용하여 MySQL DB 인스턴스와 읽기 전용 복제본 간에 GTID 기본 복제를 설정하려고 시도합니다. 이 시도가 실행할 경우 Amazon RDS는 읽기 전용 복제본으로의 복제에 로그 파일 위치를 사용합니다. `MASTER_AUTO_POSITION`에 대한 자세한 내용은 MySQL 설명서에서 [GTID 자동 배치](#)를 참조하십시오.

기존 읽기 전용 복제본에 대한 GTID 기반 복제 구성

읽기 전용 복제본이 포함되어 있고 GTID 기반 복제를 사용하지 않는 기존 RDS MySQL DB 인스턴스의 경우에는 DB 인스턴스와 읽기 전용 복제본 간에 GTID 기반 복제를 구성할 수 있습니다.

기존 읽기 전용 복제본에 대해 GTID 기반 복제를 활성화하려면

1. DB 인스턴스 또는 읽기 전용 복제본이 RDS MySQL 버전 5.7.22 이하를 사용하는 경우 DB 인스턴스 또는 읽기 전용 복제본을 업그레이드하십시오. RDS MySQL 5.7.23 또는 MySQL 5.7 후기 버전으로 업그레이드하십시오.

자세한 내용은 [MySQL DB 엔진 업그레이드 \(p. 607\)](#) 단원을 참조하십시오.

2. (선택 사항) GTID 파라미터를 재설정하고 DB 인스턴스와 읽기 전용 복제본의 동작을 테스트합니다.
 - a. DB 인스턴스 및 각 읽기 전용 복제본과 연결된 파라미터 그룹에서 `enforce_gtid_consistency` 파라미터가 `WARN`으로 설정되어 있는지 확인합니다.

파라미터 그룹을 사용한 구성 파라미터 설정에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

- b. DB 인스턴스의 파라미터 그룹을 변경한 경우 DB 인스턴스를 재부팅하십시오. 읽기 전용 복제본의 파라미터 그룹을 변경한 경우 읽기 전용 복제본을 재부팅하십시오.

자세한 내용은 [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#) 단원을 참조하십시오.

- c. 보통의 워크로드로 DB 인스턴스와 읽기 전용 복제본을 실행하고 로그 파일을 모니터링합니다.

GTID 비호환 트랜잭션에 대한 경고가 표시될 경우, GTID 호환 기능만 사용하도록 애플리케이션을 조정하십시오. 다음 단계로 진행하기 전에 DB 인스턴스에서 GTID 비호환 트랜잭션에 대한 경고가 표시되지 않는지 확인합니다.

3. 읽기 전용 복제본이 모든 트랜잭션을 처리할 때까지 익명 트랜잭션을 허용하는 GTID 기반 복제에 대한 GTID 파라미터를 재설정합니다.

- a. DB 인스턴스 및 각 읽기 전용 복제본과 연결된 파라미터 그룹에서 다음과 같이 파라미터가 설정되었는지 확인합니다.
 - `gtid_mode` – `ON_PERMISSIVE`
 - `enforce_gtid_consistency` – `ON`
- b. DB 인스턴스의 파라미터 그룹을 변경한 경우 DB 인스턴스를 재부팅하십시오. 읽기 전용 복제본의 파라미터 그룹을 변경한 경우 읽기 전용 복제본을 재부팅하십시오.
4. 익명 트랜잭션이 모두 복제될 때까지 기다립니다. 이러한 트랜잭션이 복제되었는지 확인하려면 다음과 같이 합니다.
 - a. 기본 DB 인스턴스에서 다음 명령문을 실행합니다.

```
SHOW MASTER STATUS;
```

`File` 및 `Position` 열의 값을 메모합니다.

- b. 각 읽기 전용 복제본에서 이전 단계에서 메모한 마스터의 파일 및 위치 정보를 사용하여 다음 쿼리를 실행합니다.

```
SELECT MASTER_POS_WAIT(file, position);
```

예를 들어, 파일 이름이 `mysql-bin-changelog.000031`이고 위치가 107일 경우 다음 명령문을 실행합니다.

```
SELECT MASTER_POS_WAIT(mysql-bin-changelog.000031, 107);
```

읽기 전용 복제본이 지정된 위치에 전달되면 쿼리가 즉시 반환합니다. 그렇지 않으면 함수가 대기합니다. 모든 읽기 전용 복제본에 대해 쿼리가 반환하면 다음 단계로 진행합니다.

5. GTID 기반 복제에 대해서만 GTID 파라미터를 재설정합니다.
 - a. DB 인스턴스 및 각 읽기 전용 복제본과 연결된 파라미터 그룹에서 다음과 같이 파라미터가 설정되었는지 확인합니다.
 - `gtid_mode` – `ON`
 - `enforce_gtid_consistency` – `ON`
 - b. DB 인스턴스와 각 읽기 전용 복제본을 재부팅합니다.
6. 각각의 읽기 전용 복제본에서 다음 프로시저를 실행합니다.

```
CALL mysql.rds_set_master_auto_position(1);
```

읽기 전용 복제본을 사용하는 RDS MySQL DB 인스턴스에 대해 GTID 기반 복제 비활성화

읽기 복제본이 포함된 RDS MySQL DB 인스턴스.

읽기 전용 복제본을 사용하는 RDS MySQL DB 인스턴스인에 대해 GTID 기반 복제를 비활성화하려면

- 각각의 읽기 전용 복제본에서 다음 프로시저를 실행합니다.

```
CALL mysql.rds_set_master_auto_position(0);
```

- gtid_mode를 ON_PERMISSIVE로 재설정합니다.

- RDS MySQL DB 인스턴스 및 각 읽기 전용 복제본과 연결된 파라미터 그룹에서 gtid_mode가 ON_PERMISSIVE로 설정되어 있는지 확인합니다.

파라미터 그룹을 사용한 구성 파라미터 설정에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

- RDS MySQL DB 인스턴스와 각 읽기 전용 복제본을 재부팅합니다. 재부팅에 대한 자세한 내용은 [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#) 단원을 참조하십시오.

- gtid_mode를 OFF_PERMISSIVE로 재설정합니다.

- RDS MySQL DB 인스턴스 및 각 읽기 전용 복제본과 연결된 파라미터 그룹에서 gtid_mode가 OFF_PERMISSIVE로 설정되어 있는지 확인합니다.

- RDS MySQL DB 인스턴스와 각 읽기 전용 복제본을 재부팅합니다.

- 모든 읽기 전용 복제본에서 모든 GTID 트랜잭션이 적용될 때까지 기다립니다. 이러한 트랜잭션이 적용되었는지 확인하려면 다음과 같이 합니다.

- Aurora 기본 인스턴스에서 모든 GTID 트랜잭션이 적용될 때까지 기다립니다. 이러한 트랜잭션이 적용되었는지 확인하려면 다음과 같이 합니다.
 - RDS MySQL DB 인스턴스에서 SHOW MASTER STATUS 명령을 실행합니다.

다음과 유사하게 출력되어야 합니다.

File	Position
mysql-bin-changelog.000031	107

출력에서 파일 및 위치를 메모합니다.

- 각 읽기 전용 복제본에서 이전 단계에서 메모한 마스터의 파일 및 위치 정보를 사용하여 다음 쿼리를 실행합니다.

```
SELECT MASTER_POS_WAIT(file, position);
```

예를 들어, 파일 이름이 mysql-bin-changelog.000031이고 위치가 107일 경우 다음 명령문을 실행합니다.

```
SELECT MASTER_POS_WAIT(mysql-bin-changelog.000031, 107);
```

읽기 전용 복제본이 지정된 위치에 전달되면 쿼리가 즉시 반환합니다. 그렇지 않으면 함수가 대기합니다. 쿼리가 모든 읽기 전용 복제본에 대해 반환을 완료하면 다음 단계를 진행합니다.

- GTID 기반 복제를 비활성화하도록 GTID 파라미터를 재설정합니다.

- a. RDS MySQL DB 인스턴스 및 각 읽기 전용 복제본과 연결된 파라미터 그룹에서 다음과 같이 파라미터가 설정되었는지 확인합니다.
 - `gtid_mode` – OFF
 - `enforce_gtid_consistency` – OFF
- b. RDS MySQL DB 인스턴스와 각 읽기 전용 복제본을 재부팅합니다.

Amazon RDS 외부에서 실행 중인 MySQL 또는 MariaDB 인스턴스를 사용한 복제

Amazon RDS MySQL 또는 MariaDB DB 인스턴스와 Amazon RDS 외부에 있는 MySQL 또는 MariaDB 인스턴스 간의 복제를 설정할 수 있습니다.

주제

- [시작하기 전에 \(p. 655\)](#)
- [외부 마스터 인스턴스를 사용하여 이진 로그 파일 위치 복제 구성 \(p. 655\)](#)
- [외부 마스터 인스턴스를 사용하여 GTID 기반 복제 구성 \(p. 658\)](#)

시작하기 전에

복제된 트랜잭션의 이진 로그 파일 위치를 사용하여 복제를 구성할 수 있습니다. Amazon RDS MySQL 5.7.23 및 이후 MySQL 5.7 버전에서는 GTID(글로벌 트랜잭션 식별자)를 사용하여 복제를 구성할 수도 있습니다.

Amazon RDS DB 인스턴스에서 복제를 시작하는 데 필요한 권한은 제한되고 Amazon RDS 마스터 사용자는 사용할 수 없습니다. 이 때문에, Amazon RDS [mysql.rds_set_external_master \(p. 678\)](#) 및 [mysql.rds_start_replication \(p. 688\)](#) 명령을 사용하여 라이브 데이터베이스와 Amazon RDS 데이터베이스 사이의 복제를 설정해야 합니다.

MySQL 또는 MariaDB 데이터베이스의 이진 로깅 형식을 설정하려면 `binlog_format` 파라미터를 업데이트합니다. DB 인스턴스가 기본 DB 인스턴스 파라미터 그룹을 사용하는 경우, `binlog_format` 설정을 수정하려면 새로운 DB 파라미터 그룹을 만들습니다. `binlog_format`의 기본 설정인 `MIXED`를 사용하는 것이 좋습니다. 그러나 특정한 `binlog` 형식이 필요하다면 `binlog_format`을 `ROW` 또는 `STATEMENT`로 설정할 수도 있습니다. 변경 사항을 적용하려면 DB 인스턴스를 재부팅합니다.

`binlog_format` 파라미터 설정에 대한 자세한 내용은 [이진 로깅 형식 \(p. 324\)](#) 단원을 참조하십시오. 다양한 MySQL 복제 유형의 영향에 대한 자세한 내용은 MySQL 설명서의 [Advantages and Disadvantages of Statement-Based and Row-Based Replication](#)을 참조하십시오.

Note

외부 인스턴스가 MariaDB 버전 10.0.2 이상이고 Amazon RDS 인스턴스가 MariaDB인 경우를 제외하고 항상 이 주제의 프로시저를 사용하여 복제를 구성합니다. 이 경우에는 [Amazon RDS MariaDB DB 인스턴스로의 GTID 기반 복제 구성 \(p. 464\)](#) 내 프로세서를 사용하여 GTID 기반 복제를 설정하십시오.

외부 마스터 인스턴스를 사용하여 이진 로그 파일 위치 복제 구성

Amazon RDS에서 외부 복제 마스터와 복제본을 설정할 때 다음 지침을 준수해야 합니다.

- 사용자의 복제본인 Amazon RDS DB 인스턴스에 대한 장애 조치 이벤트를 모니터링합니다. 장애 조치가 발생할 경우에는 사용자의 복제본인 DB 인스턴스가 다른 네트워크 주소를 가진 새 호스트에서 다시 생성

될 수도 있습니다. 장애 조치 이벤트를 모니터링하는 자세한 방법은 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오.

- 이진 로그(binlog)가 복제본에 적용되었음을 확인할 때까지는 마스터 인스턴스에서 binlog를 유지 관리하십시오. 이렇게 유지 관리해야 오류 발생 시 마스터 인스턴스를 복원할 수 있습니다.
- Amazon RDS DB 인스턴스에서 자동 백업을 활성화합니다. 자동 백업을 활성화하면 마스터 및 복제본을 다시 동기화할 필요가 있을 때 복제본을 특정 시점으로 복원할 수 있습니다. 백업 및 특정 시점으로 복원에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 백업 및 복원 \(p. 196\)](#) 단원을 참조하십시오.

외부 마스터 인스턴스를 사용하여 이진 로그 파일 위치 복제를 구성하려면

1. 원본 MySQL 또는 MariaDB 인스턴스를 읽기 전용으로 설정합니다.

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

2. 원본 MySQL 또는 MariaDB 인스턴스에서 SHOW MASTER STATUS 명령을 실행하여 binlog 위치를 확인합니다.

다음 예제와 비슷한 출력 결과를 얻습니다.

File	Position
mysql-bin-changelog.000031	107

3. mysqldump를 사용하여 외부 인스턴스에서 Amazon RDS DB 인스턴스로 데이터베이스를 복사합니다. 매우 큰 데이터베이스의 경우, [자동 중지 시간을 단축하여 Amazon RDS MySQL MariaDB DB 인스턴스로 데이터 가져오기 \(p. 628\)](#)에서 이 절차를 사용하고 싶을 것입니다.

Linux, OS X, Unix의 경우:

```
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>
```

Windows의 경우:

```
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>
```

Note

-p 옵션과 입력한 암호 사이에 공백이 없어야 합니다.

Amazon RDS DB 인스턴스에 연결하기 위해 호스트 이름, 사용자 이름, 포트 및 암호를 지정하려면 mysql 명령에 `--host`, `--user (-u)`, `--port`, `-p` 옵션을 사용합니다. 호스트 이름은 Amazon RDS DB 인스턴스 엔드포인트의 DNS(Domain Name Service) 이름입니다(예: `myinstance.123456789012.us-east-1.rds.amazonaws.com`). AWS Management 콘솔의 인스턴스 세부 정보에서 엔드포인트 값을 확인할 수 있습니다.

4. 원본 MySQL 또는 MariaDB 인스턴스를 다시 쓰기 가능한 상태로 만듭니다.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

복제와 함께 사용할 백업을 만드는 자세한 내용은 MySQL 문서의 [Backing Up a Master or Slave by Making It Read Only](#) 단원을 참조하십시오.

5. AWS Management 콘솔에서 Amazon RDS DB 인스턴스에 대한 VPC 보안 그룹에 외부 데이터베이스를 호스팅하는 서버의 IP 주소를 추가합니다. VPC 보안 그룹 수정에 대한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 [VPC용 보안 그룹](#)을 참조하십시오.

외부 MySQL 또는 MariaDB 인스턴스와 통신할 수 있도록, Amazon RDS DB 인스턴스 IP 주소에서의 연결을 허용하도록 로컬 네트워크를 구성해야 할 수도 있습니다. Amazon RDS DB 인스턴스의 IP 주소를 확인하려면 `host` 명령을 사용합니다.

```
host <RDS_SQL_DB_host_name>
```

호스트 이름은 Amazon RDS DB 인스턴스 엔드포인트의 DNS 이름입니다.

6. 선택한 클라이언트를 사용하여 외부 인스턴스에 연결하고 복제에 사용할 사용자를 만듭니다. 이 계정을 복제용으로만 사용하고, 보안을 강화하기 위해 해당 도메인으로만 제한하십시오. 다음은 예제입니다.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY '<password>';
```

7. 외부 인스턴스의 경우 복제 사용자에게 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여합니다. 예를 들어 도메인의 'repl_user' 사용자를 위해 모든 데이터베이스에서 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여하려면 다음 명령을 실행합니다.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
IDENTIFIED BY '<password>';
```

8. Amazon RDS DB 인스턴스를 복제본으로 만듭니다. 이렇게 하려면 마스터 사용자로 Amazon RDS DB 인스턴스에 연결한 후 [mysql.rds_set_external_master \(p. 678\)](#) 명령을 사용하여 외부 MySQL 또는 MariaDB 데이터베이스를 복제 마스터로 식별합니다. 2단계에서 확인한 마스터 로그 파일 이름과 마스터 로그 위치를 사용합니다. 다음은 예제입니다.

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306, 'repl_user',
'<password>', 'mysql-bin-changelog.000031', 107, 0);
```

Note

Amazon RDS MySQL에서 [mysql.rds_set_external_master_with_delay \(p. 680\)](#) 저장 프로시저를 대신 실행하여 지역 복제를 사용하도록 선택할 수 있습니다. 지역 복제를 사용하는 이유 중 하나는 [mysql.rds_start_replication_until \(p. 689\)](#) 저장 프로시저를 사용하여 재해를 복구할 수 있기 때문입니다. 현재는 지역 복제가 Amazon RDS MariaDB에서만 지원됩니다.

9. Amazon RDS DB 인스턴스에서 [mysql.rds_start_replication \(p. 688\)](#) 명령을 실행하여 복제를 시작합니다.

```
CALL mysql.rds_start_replication;
```

외부 마스터 인스턴스를 사용하여 GTID 기반 복제 구성

Amazon RDS에 외부 복제 마스터와 복제본을 설정한 경우, 복제본인 Amazon RDS DB 클러스터에 대한 장애 조치 이벤트를 모니터링하십시오. 장애 조치가 발생할 경우에는 사용자의 복제본인 DB 인스턴스가 다른 네트워크 주소를 가진 새 호스트에서 다시 생성될 수도 있습니다. 장애 조치 이벤트를 모니터링하는 자세한 방법은 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오.

Important

GTID 기반 복제는 Amazon RDS MySQL 버전 5.7.23 및 이후 MySQL 5.7 버전에서만 지원됩니다.
Amazon RDS MySQL 5.5, 5.6, 8.0에서는 GTID 기반 복제가 지원되지 않습니다.

외부 마스터 인스턴스를 사용하여 GTID 기반 복제를 구성하려면

1. GTID 기반 복제를 준비합니다.

- a. 외부 MySQL 또는 MariaDB 데이터베이스에서 GTID 기반 복제가 활성화되었는지 확인합니다. 이를 확인하려면 외부 데이터베이스의 다음 파라미터가 지정된 값으로 설정되었는지 확인합니다.

```
gtid_mode = ON
```

```
enforce_gtid_consistency = ON
```

자세한 내용은 MySQL 설명서의 [전역 트랜잭션 ID를 사용한 복제](#) 또는 MariaDB 설명서의 [전역 트랜잭션 ID](#)를 참조하십시오.

- b. DB 인스턴스와 연결된 파라미터 그룹에서 다음과 같이 파라미터가 설정되어 있는지 확인합니다.
- gtid_mode = ON, ON_PERMISSIVE 또는 OFF_PERMISSIVE
 - enforce_gtid_consistency = ON

파라미터 그룹에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

- c. DB 인스턴스의 파라미터 그룹을 변경한 경우 DB 인스턴스를 재부팅하십시오. 자세한 내용은 [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#) 단원을 참조하십시오.

2. 원본 MySQL 또는 MariaDB 인스턴스를 읽기 전용으로 설정합니다.

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

3. mysqldump를 사용하여 외부 인스턴스에서 Amazon RDS DB 인스턴스로 데이터베이스를 복사합니다. 매우 큰 데이터베이스의 경우, [자동 중지 시간을 단축하여 Amazon RDS MySQL MariaDB DB 인스턴스로 데이터 가져오기 \(p. 628\)](#)에서 이 절차를 사용하고 싶을 것입니다.

Linux, OS X, Unix의 경우:

```
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>

Windows의 경우:

```
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>
```

Note

-p 옵션과 입력한 암호 사이에 공백이 없어야 합니다.

Amazon RDS DB 인스턴스에 연결하기 위해 호스트 이름, 사용자 이름, 포트 및 암호를 지정하려면 mysql 명령에 --host, --user (-u), --port, -p 옵션을 사용합니다. 호스트 이름은 Amazon RDS DB 인스턴스 엔드포인트의 DNS 이름입니다(예: myinstance.123456789012.us-east-1.rds.amazonaws.com). AWS Management 콘솔의 인스턴스 세부 정보에서 엔드포인트 값을 확인할 수 있습니다.

4. 원본 MySQL 또는 MariaDB 인스턴스를 다시 쓰기 가능한 상태로 만듭니다.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

복제와 함께 사용할 백업을 만드는 자세한 내용은 MySQL 문서의 [Backing Up a Master or Slave by Making It Read Only](#) 단원을 참조하십시오.

5. AWS Management 콘솔에서 Amazon RDS DB 인스턴스에 대한 VPC 보안 그룹에 외부 데이터베이스를 호스팅하는 서버의 IP 주소를 추가합니다. VPC 보안 그룹 수정에 대한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 [VPC용 보안 그룹](#)을 참조하십시오.

외부 MySQL 또는 MariaDB 인스턴스와 통신할 수 있도록, Amazon RDS DB 인스턴스 IP 주소에서의 연결을 허용하도록 로컬 네트워크를 구성해야 할 수도 있습니다. Amazon RDS DB 인스턴스의 IP 주소를 확인하려면 host 명령을 사용합니다.

```
host <RDS_MySQL_DB_host_name>
```

호스트 이름은 Amazon RDS DB 인스턴스 엔드포인트의 DNS 이름입니다.

6. 선택한 클라이언트를 사용하여 외부 인스턴스에 연결하고 복제에 사용할 사용자를 만듭니다. 이 계정을 복제용으로만 사용하고, 보안을 강화하기 위해 해당 도메인으로만 제한하십시오. 다음은 예제입니다.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY '<password>';
```

7. 외부 인스턴스의 경우 복제 사용자에게 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여합니다. 예를 들어 도메인의 'repl_user' 사용자를 위해 모든 데이터베이스에서 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여하려면 다음 명령을 실행합니다.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
IDENTIFIED BY '<password>';
```

- Amazon RDS DB 인스턴스를 복제본으로 만듭니다. 이렇게 하려면 마스터 사용자로 Amazon RDS DB 인스턴스에 연결한 후 [mysql.rds_set_external_master_with_auto_position \(p. 683\)](#) 명령을 사용하여 외부 MySQL 또는 MariaDB 데이터베이스를 복제 마스터로 식별합니다. 다음은 예제입니다.

```
CALL mysql.rds_set_external_master_with_auto_position ('mymasterserver.mydomain.com',  
3306, 'repl_user', '<password>', 0, 0);
```

Note

Amazon RDS MySQL에서 [mysql.rds_set_external_master_with_delay \(p. 680\)](#) 저장 프로시저를 대신 실행하여 지연 복제를 사용하도록 선택할 수 있습니다. 지연 복제를 사용하는 이유 중 하나는 [mysql.rds_start_replication_until_gtid \(p. 690\)](#) 저장 프로시저를 사용하여 재해를 복구할 수 있기 때문입니다. 현재는 지연 복제가 Amazon RDS MariaDB에서만 지원됩니다.

- Amazon RDS DB 인스턴스에서 [mysql.rds_start_replication \(p. 688\)](#) 명령을 실행하여 복제를 시작합니다.

```
CALL mysql.rds_start_replication;
```

복제를 사용하여 MySQL DB 인스턴스에서 데이터 내보내기

복제를 사용하여 MySQL 5.6 이상 DB 인스턴스에서 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스로 데이터를 내보낼 수 있습니다. Amazon RDS 외부의 MySQL 인스턴스는 데이터 센터의 온프레미스 환경이나 Amazon EC2 인스턴스에서 실행 중일 수 있습니다. MySQL DB 인스턴스는 버전 5.6.13 이상을 실행 중이어야 합니다. Amazon RDS 외부의 MySQL 인스턴스는 Amazon RDS 인스턴스와 같은 버전이나 더 최신 버전을 실행 중이어야 합니다.

Amazon RDS 외부에서 실행 중인 MySQL의 인스턴스에 대한 복제는 MySQL DB 인스턴스에서 데이터베이스를 내보내는데 걸리는 시간 중에만 지원됩니다. 데이터 내보내기가 완료되어 애플리케이션이 외부 인스턴스에 대한 액세스를 시작할 수 있을 때 복제가 종료되어야 합니다.

아래 목록에 수행할 단계가 나와 있습니다. 각 단계에 대해서는 이후 섹션에서 자세히 설명합니다.

- Amazon RDS 외부에서 실행 중인 MySQL의 인스턴스를 준비합니다.
- MySQL DB 인스턴스가 복제 소스가 되도록 구성합니다.
- `mysqldump`를 사용하여 Amazon RDS 인스턴스에서 Amazon RDS 외부의 인스턴스로 데이터베이스를 전송합니다.
- Amazon RDS 외부에서 실행 중인 인스턴스에 대한 복제를 시작합니다.
- 내보내기가 완료된 후 복제를 중지합니다.

Amazon RDS 외부에 MySQL의 인스턴스 준비

Amazon RDS 외부에 MySQL의 인스턴스를 설치합니다.

마스터 사용자로 인스턴스에 연결하고, 인스턴스에 액세스하는 서비스, 애플리케이션 및 관리자를 지원하는데 필요한 사용자를 만듭니다.

MySQL 문서의 지침에 따라 Amazon RDS 외부에서 실행 중인 MySQL의 인스턴스를 복제본으로 준비합니다. 자세한 정보는 [Setting the Replication Slave Configuration](#)을 참조하십시오.

내보내기 중에 외부 인스턴스가 읽기 전용 복제본으로 작동하도록 송신 규칙을 구성합니다. 송신 규칙에 따라 MySQL 읽기 전용 복제본은 복제 중에 MySQL DB 인스턴스에 연결할 수 있습니다. 원본 Amazon RDS MySQL DB 인스턴스의 포트와 IP 주소에 대한 TCP 연결을 허용하는 송신 규칙을 지정합니다.

읽기 전용 복제본이 Amazon VPC의 Amazon EC2 인스턴스에서 실행 중인 경우 VPC 보안 그룹에서 송신 규칙을 지정합니다. 읽기 전용 복제본이 VPC에 있지 않은 Amazon EC2 인스턴스에서 실행 중인 경우 Amazon EC2 보안 그룹에서 송신 규칙을 지정합니다. 읽기 전용 복제본이 온프레미스에 설치되어 있는 경우 방화벽에서 송신 규칙을 지정합니다.

읽기 전용 복제본이 VPC에서 실행 중인 경우 보안 그룹 송신 규칙 이외에 VPC ACL 규칙을 구성합니다. Amazon VPC 네트워크 ACL에 대한 자세한 정보는 [네트워크 ACL](#)을 참조하십시오.

- 원본 MySQL DB 인스턴스의 IP 주소에서 포트 1024-65535로의 TCP 트래픽을 허용하는 ACL 송신 규칙
- ACL 송신 규칙: 원본 MySQL DB 인스턴스의 포트와 IP 주소로의 아웃바운드 TCP 트래픽 허용

복제 원본 준비

MySQL DB 인스턴스를 복제 소스로 준비합니다.

클라이언트 컴퓨터에서 복제를 설정하는 동안 이진 로그를 저장하기에 충분한 디스크 공간을 사용할 수 있는지 확인합니다.

[Creating a User For Replication](#) 섹션의 지침에 따라 복제 계정을 만듭니다.

복제 원본 MySQL DB 인스턴스를 실행 중인 시스템에서 복제 중에 외부 MySQL 읽기 전용 복제본의 연결을 허용하도록 송신 규칙을 구성합니다. Amazon RDS 외부에서 실행 중인 MySQL 읽기 전용 복제본의 IP 주소에서 Amazon RDS 인스턴스가 사용하는 포트에 대한 TCP 연결을 허용하는 수신 규칙을 지정합니다.

VPC에서 Amazon RDS 인스턴스가 실행 중인 경우 VPC 보안 그룹에서 수신 규칙을 지정합니다. Amazon RDS 인스턴스가 VPC에서 실행 중이지 않은 경우에는 데이터베이스 보안 그룹에서 수신 규칙을 지정합니다.

Amazon RDS 인스턴스가 VPC에서 실행 중인 경우 보안 그룹 수신 규칙 이외에 VPC ACL 규칙을 구성합니다. Amazon VPC 네트워크 ACL에 대한 자세한 정보는 [네트워크 ACL](#)을 참조하십시오.

- ACL 수신 규칙: 외부 MySQL 읽기 전용 복제본의 IP 주소에서 Amazon RDS 인스턴스가 사용하는 포트에 대한 TCP 연결을 허용합니다.
- ACL 송신 규칙: 포트 1024-65535에서 외부 MySQL 읽기 전용 복제본의 IP 주소로 TCP 연결을 허용합니다.

내보내기 중에 아무런 이진 로그도 제거되지 않을 만큼 백업 보존 기간이 충분히 길게 설정되어 있는지 확인합니다. 내보내기가 완료되기 전에 제거되는 로그가 있으면, 복제를 처음부터 다시 시작해야 합니다. 백업 보존 기간 설정에 대한 자세한 정보는 [백업 작업 \(p. 196\)](#) 단원을 참조하십시오.

`mysql.rds_set_configuration` 저장 프로시저를 사용하여 내보내기 중에 이진 로그가 제거되지 않을 만큼 충분히 길게 이진 로그 보존 기간을 설정합니다. 자세한 정보는 [MySQL 이진 로그 액세스 \(p. 324\)](#) 단원을 참조하십시오.

원본 인스턴스의 이진 로그가 제거되지 않도록 더욱 확실히 하기 위해, 원본 인스턴스에서 Amazon RDS 읽기 전용 복제본을 만듭니다. 자세한 정보는 [읽기 전용 복제본 만들기 \(p. 138\)](#) 단원을 참조하십시오. Amazon RDS 읽기 전용 복제본이 생성된 후, `mysql.rds_stop_replication` 저장 프로시저를 호출하여 복제 프로세스를 중지합니다. 원본 인스턴스가 더 이상 이진 로그 파일을 제거하지 않을 것이므로, 복제 프로세스에 이런 파일을 사용할 수 있습니다.

데이터베이스 복사

RDS 읽기 전용 복제본에서 MySQL SHOW SLAVE STATUS 문을 실행하고 다음 값을 기록합니다.

- master_host
- master_port
- master_log_file
- exec_master_log_pos

mysqldump 유ти리티를 사용하여 Amazon RDS에서 로컬 클라이언트 컴퓨터로 데이터를 복사하는 스냅샷을 만듭니다. 그런 다음, RDS 외부에서 실행 중인 MySQL 인스턴스로 데이터를 로드하는 또 다른 유ти리티를 실행합니다. 클라이언트에 복제할 데이터베이스의 mysqldump 파일을 보관하기에 충분한 공간이 있는지 확인합니다. 매우 큰 데이터베이스의 경우 이 프로세스는 여러 시간이 걸릴 수 있습니다. [Creating a Dump Snapshot Using mysqldump](#) 섹션의 지침을 따르십시오.

다음 예제에서는 클라이언트에서 mysqldump를 실행한 후 덤프를 mysql 클라이언트 유ти리티로 파이프하여 데이터를 외부 MySQL 인스턴스로 로드하는 방법을 보여줍니다.

Linux, OS X, Unix의 경우:

```
mysqldump -h RDS instance endpoint \
-u user \
-p password \
--port=3306 \
--single-transaction \
--routines \
--triggers \
--databases database database2 \
--compress \
--compact | mysql \
-h MySQL host \
-u master user \
-p password \
--port 3306
```

Windows의 경우:

```
mysqldump -h RDS instance endpoint ^
-u user ^
-p password ^
--port=3306 ^
--single-transaction ^
--routines ^
--triggers ^
--databases database database2 ^
--compress ^
--compact | mysql ^
-h MySQL host ^
-u master user ^
-p password ^
--port 3306
```

다음 예제에서는 클라이언트에서 mysqldump를 실행하고 덤프를 파일에 쓰는 방법을 보여줍니다.

Linux, OS X, Unix의 경우:

```
mysqldump -h RDS instance endpoint \
-u user \
-p password \
--port=3306 \
--single-transaction \
--routines \
--triggers \
```

```
--databases database database2 > path/rds-dump.sql
```

Windows의 경우:

```
mysqldump -h RDS instance endpoint ^
-u user ^
-p password ^
--port=3306 ^
--single-transaction ^
--routines ^
--triggers ^
--databases database database2 > path\rds-dump.sql
```

내보내기 완료

mysqldump 파일을 로드하여 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스에 데이터베이스를 만든 후, 원본 MySQL DB 인스턴스에서 복제를 시작하여 Amazon RDS 읽기 전용 복제본에서 복제를 중지한 후에 원본에 발생한 모든 변경 내용을 내보냅니다.

MySQL CHANGE MASTER 문을 사용하여 외부 MySQL 인스턴스를 구성합니다. 사용자에게 부여된 REPLICATION SLAVE 권한의 ID와 암호를 지정합니다. RDS 읽기 전용 복제본에서 실행한 MySQL SHOW SLAVE STATUS 문에서 얻은 master_host, master_port, relay_master_log_file 및 exec_master_log_pos 값을 지정합니다. 자세한 정보는 [Setting the Master Configuration on the Slave 단원](#)을 참조하십시오.

MySQL START SLAVE 명령을 사용하여 원본 MySQL DB 인스턴스와 MySQL 복제본에서 복제를 시작합니다.

Amazon RDS 인스턴스에서 MySQL SHOW SLAVE STATUS 명령을 실행하여 이 인스턴스가 읽기 전용 복제본으로 작동 중인지 확인합니다. 결과 해석에 대한 자세한 정보는 [SHOW SLAVE STATUS Syntax](#)를 참조하십시오.

MySQL 인스턴스에서의 복제가 Amazon RDS 원본을 따라잡은 후, MySQL STOP SLAVE 명령을 사용하여 원본 MySQL DB 인스턴스에서의 복제를 종료합니다.

Amazon RDS 읽기 전용 복제본에서 mysql.rds_start_replication 저장 프로시저를 호출합니다. 그러면 Amazon RDS가 원본 MySQL DB 인스턴스에서 이진로그 파일 제거를 시작할 수 있습니다.

관련 주제

- [Amazon RDS MySQL DB 인스턴스로 백업 복원 \(p. 619\)](#)
- [Amazon RDS DB 인스턴스 백업 및 복원 \(p. 196\)](#)

MySQL DB 인스턴스 옵션

이번 부록에서는 MySQL DB 엔진 기반 Amazon RDS 인스턴스의 옵션과 추가 기능에 대해 설명합니다. 이러한 옵션들을 활성화하려면 먼저 사용자 정의 옵션 그룹에 추가한 다음 옵션 그룹과 DB 인스턴스를 연동시켜야 합니다. 옵션 그룹 작업에 대한 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오.

Amazon RDS는 다음의 MySQL 옵션을 지원합니다.

옵션	옵션 ID	엔진 버전
MariaDB 감사 플러그인 지원 (p. 664)	MARIADB_AUDIT_PLUGIN	모든 MySQL 5.6 버전 MySQL 5.7.16 및 5.7 버전 이상

옵션	옵션 ID	엔진 버전
MySQL memcached 지원 (p. 666)	MEMCACHED	모든 MySQL 5.6, 5.7 및 8.0 버전

MariaDB 감사 플러그인 지원

Amazon RDS는 MySQL 데이터베이스 인스턴스에서의 MariaDB 감사 플러그인 사용을 지원합니다. MariaDB 감사 플러그인은 사용자의 데이터베이스 로그온, 데이터베이스에 대해 실행되는 쿼리 등의 데이터베이스 활동을 기록합니다. 데이터베이스 활동 기록은 로그 파일에 저장됩니다.

Note

현재 MariaDB 감사 플러그인은 다음 Amazon RDS MySQL 버전에만 지원됩니다.

- 모든 5.6 버전
- MySQL 5.7.16 및 5.7 버전 이상

감사 플러그인 옵션 설정

Amazon RDS는 MariaDB 감사 플러그인 옵션의 다음 설정을 지원합니다.

옵션 설정	유효한 값	기본값	설명
SERVER_AUDIT_FILEDATA/log/audit/	/rdsdbdata/log/audit/	/rdsdbdata/log/audit/	로그 파일의 위치. 로그 파일에는 SERVER_AUDIT_EVENTS에서 지정된 활동 기록이 포함되어 있습니다. 자세한 내용은 데이터베이스 로그 파일 보기 및 나열 (p. 307) 및 MySQL 데이터베이스 로그 파일 (p. 318) 단원을 참조하십시오.
SERVER_AUDIT_FILEROTATION_SIZE	1000000	1000000	도달 시 파일 로테이션을 초래하는 바이트 크기. 자세한 내용은 로그 파일 크기 (p. 323) 단원을 참조하십시오.
SERVER_AUDIT_FILE_ROTATION	9	9	저장할 로그 로테이션 수. 자세한 내용은 로그 파일 크기 (p. 323) 및 데이터베이스 로그 파일 다운로드 (p. 307) 단원을 참조하십시오.
SERVER_AUDIT_EVENT, QUERY	CONNECT, QUERY	CONNECT, QUERY	<p>로그에 기록할 활동 유형. MariaDB 감사 플러그인 설치 자체가 로깅됩니다.</p> <ul style="list-style-type: none">CONNECT: 성공/실패한 데이터베이스 연결과 데이터베이스 연결 해제를 로깅합니다.QUERY: 데이터베이스에 대해 실행된 모든 쿼리의 텍스트를 로깅합니다.TABLE: 데이터베이스에 대해 쿼리가 실행될 때 쿼리의 영향을 받는 테이블을 로깅합니다. <p>MariaDB에서는 CONNECT, QUERY 및 TABLE을 지원합니다.</p> <p>MySQL에서는 CONNECT 및 QUERY를 지원합니다.</p>

옵션 설정	유효한 값	기본값	설명
SERVER_AUDIT_ <small>별도의 허가로 분리된 값</small>		없음	지정된 사용자들의 활동만을 포함하십시오. 기본적으로 모든 사용자의 활동이 기록됩니다. 사용자가 SERVER_AUDIT_EXCL_USERS 및 SERVER_AUDIT_INCL_USERS에서 모두 지정된 경우, 해당 사용자에 대해 활동이 기록됩니다.
SERVER_AUDIT_ <small>별도의 허가로 분리된 값</small>		없음	지정된 사용자들의 활동을 제외하십시오. 기본적으로 모든 사용자의 활동이 기록됩니다. 사용자가 SERVER_AUDIT_EXCL_USERS 및 SERVER_AUDIT_INCL_USERS에서 모두 지정된 경우, 해당 사용자에 대해 활동이 기록됩니다.
SERVER_AUDIT_LOGGING	ON		<p>rdsadmin 사용자는 데이터베이스 상태를 확인하기 위해 1초마다 데이터베이스에 쿼리를 요청합니다. 다른 설정에 따라 이 활동은 로그 파일의 크기를 아주 빨리 대폭 증가시킬 수 있습니다. 이 활동을 기록할 필요가 없는 경우, rdsadmin 사용자를 SERVER_AUDIT_EXCL_USERS 목록에 추가하십시오.</p> <p>Note</p> <p>CONNECT 활동은 해당 사용자가 이 옵션 설정에 지정되었다 해도 모든 사용자에 대해 기록됩니다.</p>

MariaDB 감사 플러그인 추가하기

MariaDB 감사 플러그인을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

- 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
- 옵션을 옵션 그룹에 추가합니다.
- 옵션 그룹을 DB 인스턴스에 연동 시킵니다.

MariaDB 감사 플러그인을 추가한 후 DB 인스턴스를 재시작할 필요가 없습니다. 옵션 그룹이 활성화되자마자 감사가 즉시 시작됩니다.

Important

MariaDB 감사 플러그인을 DB 인스턴스에 추가하면 작동이 중단될 수 있습니다. 유지 관리 기간 또는 데이터베이스 워크로드가 적은 시간에 MariaDB 감사 플러그인을 추가하는 것이 좋습니다.

MariaDB 감사 플러그인을 추가하려면,

- 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않다면 사용자 지정 DB 옵션 그룹을 생성합니다. 엔진의 경우 mysql을 선택하고, 메이저 엔진 버전의 경우 5.6, 5.7을 선택하십시오. 자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.

2. MARIADB_AUDIT_PLUGIN 옵션을 옵션 그룹에 추가하고 옵션 설정을 구성하십시오. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#)을(를) 참조하십시오. 각 설정에 대한 자세한 내용은 [감사 플러그인 옵션 설정 \(p. 664\)](#) 단원을 참조하십시오.
3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다.
 - 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 정보는 [MySQL 데이터베이스 엔진 기반 DB 인스턴스 생성 \(p. 587\)](#) 단원을 참조하십시오.
 - 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 자세한 내용은 [MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 600\)](#) 단원을 참조하십시오.

MariaDB 감사 플러그인 로그 보기 및 다운로드

MariaDB 감사 플러그인을 활성화한 후 다른 텍스트 기반 로그 파일에 액세스하는 것과 동일한 방식으로 로그 파일의 결과에 액세스할 수 있습니다. 감사 로그 파일은 `/rdsdbdata/log/audit/`에 있습니다. 콘솔에서 로그 파일 보기에 대한 자세한 내용은 [데이터베이스 로그 파일 보기 및 나열 \(p. 307\)](#)을(를) 참조하십시오. 로그 파일 다운로드에 대한 자세한 내용은 [데이터베이스 로그 파일 다운로드 \(p. 307\)](#)을(를) 참조하십시오.

MariaDB 감사 플러그인 설정 수정

MariaDB 감사 플러그인을 활성화한 후 설정을 수정할 수 있습니다. 옵션 설정을 변경하는 방법에 대한 자세한 내용은 [옵션 설정 변경 \(p. 156\)](#)을(를) 참조하십시오. 각 설정에 대한 자세한 내용은 [감사 플러그인 옵션 설정 \(p. 664\)](#) 단원을 참조하십시오.

MariaDB 감사 플러그인 제거하기

Amazon RDS는 MariaDB 감사 플러그인에서의 로깅 끄기를 지원하지 않습니다. 다만 DB 인스턴스에서 플러그인을 제거할 수는 있습니다. MariaDB 감사 플러그인을 제거할 때 DB 인스턴스가 자동으로 재시작하여 감사가 중지됩니다.

MariaDB 감사 플러그인을 DB 인스턴스에서 제거하려면 다음 중 하나를 실행하십시오.

- MariaDB 감사 플러그인이 속한 옵션 그룹에서 MariaDB 감사 플러그인 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- DB 인스턴스를 수정하고, 플러그인이 포함되지 않은 다른 옵션 그룹을 지정합니다. 이 변경은 단일 DB 인스턴스에 영향을 미칩니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. 자세한 내용은 [MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 600\)](#) 단원을 참조하십시오.

MySQL memcached 지원

Amazon RDS는 MySQL 5.6에서 도입된 InnoDB 테이블에 대한 memcached 인터페이스 사용을 지원합니다. memcached API가 있으면 애플리케이션에서 NoSQL 키-값 데이터 스토어와 비슷한 방식으로 InnoDB 테이블을 사용할 수 있습니다.

memcached 인터페이스는 간단한 키 기반 캐시입니다. 애플리케이션은 이 memcached를 사용하여 키-값 데이터 쌍을 삽입하거나, 조작하거나, 캐시에서 가져옵니다. MySQL 5.6은 데몬 서비스를 이용한 플러그인이 추가되면서 memcached 프로토콜을 통해 InnoDB 테이블의 데이터를 노출시킵니다. MySQL memcached 플러그인에 대한 자세한 내용은 [InnoDB와 memcached의 통합](#)을 참조하십시오.

Amazon RDS MySQL 5.6 이상 인스턴스에 memcached 지원을 활성화하려면

1. 먼저 memcached 인터페이스에 대한 액세스를 제어하는 데 사용할 보안 그룹을 결정합니다. 기존에 SQL 인터페이스를 사용하는 애플리케이션과 앞으로 memcached 인터페이스에 액세스할 애플리케이션

이 동일한 경우에는 기존에 SQL 인터페이스에서 사용하는 VPC 또는 DB 보안 그룹을 그대로 사용할 수 있습니다. 하지만 memcached 인터페이스에 액세스하는 애플리케이션이 다른 경우에는 새로운 VPC 또는 DB 보안 그룹을 정의해야 합니다. 보안 그룹 관리에 대한 자세한 내용은 [보안 그룹을 통한 액세스 제어 \(p. 390\)](#) 단원을 참조하십시오.

2. 사용자 정의 DB 옵션 그룹을 생성한 후 엔진 유형으로 MySQL 5.6 이상을 선택합니다. 옵션 그룹의 생성에 대한 자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#)을(를) 참조하십시오.
3. [MEMCACHED] 옵션을 옵션 그룹에 추가합니다. memcached 인터페이스가 사용할 포트를 비롯해 인터페이스에 대한 액세스 제어에 사용할 보안 그룹을 지정합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#)을(를) 참조하십시오.
4. 필요하다면 옵션 설정을 변경하여 memcached 파라미터를 구성합니다. 옵션 설정을 변경하는 방법에 대한 자세한 내용은 [옵션 설정 변경 \(p. 156\)](#)을(를) 참조하십시오.
5. 옵션 그룹을 인스턴스에 적용합니다. Amazon RDS는 옵션 그룹이 적용될 때 해당 인스턴스에 대한 memcached 지원을 활성화합니다.
 - 인스턴스 실행 시 사용자 정의 옵션 그룹을 지정하여 새로운 인스턴스에 대한 memcached 지원을 활성화합니다. MySQL 인스턴스 실행에 대한 자세한 내용은 [MySQL 데이터베이스 엔진 기반 DB 인스턴스 생성 \(p. 587\)](#) 단원을 참조하십시오.
 - 인스턴스 설정 변경 시 사용자 정의 옵션 그룹을 지정하여 기존 인스턴스에 대한 memcached 지원을 활성화합니다. MySQL 인스턴스 설정 변경에 대한 자세한 내용은 [MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 600\)](#) 단원을 참조하십시오.
6. MySQL 테이블에서 memcached 인터페이스를 통해 액세스할 수 있는 열을 지정합니다. memcached 플러그인은 innodb_memcache라는 이름의 전용 데이터베이스에 containers라는 이름의 카탈로그 테이블을 생성합니다. 행을 containers 테이블에 삽입하여 memcached를 통해 액세스할 수 있도록 InnoDB 테이블을 매핑합니다. InnoDB 테이블에서 memcached 키 값을 저장하는 데 사용할 열 하나와, 이 키와 연동된 데이터 값을 저장하는 데 사용할 하나 이상의 열을 지정합니다. 또한 memcached 애플리케이션이 해당 열 세트를 참조하는데 사용할 이름도 지정합니다. 행을 containers 테이블에 삽입하는 방법에 대한 자세한 내용은 [InnoDB memcached 플러그인의 내부 요소](#)를 참조하십시오. InnoDB 테이블을 매핑하여 memcached를 통해 액세스하는 예제는 [InnoDB + memcached 애플리케이션을 위한 테이블 및 열 매핑 지정](#)을 참조하십시오.
7. memcached 인터페이스에 액세스하는 애플리케이션이 SQL 인터페이스를 사용하는 애플리케이션과 다른 컴퓨터 또는 EC2 인스턴스에 연결되어 있는 경우에는 이 컴퓨터에 대한 연결 정보를 MySQL 인스턴스와 연동되어 있는 VPC 또는 DB 보안 그룹에 추가합니다. 보안 그룹 관리에 대한 자세한 내용은 [보안 그룹을 통한 액세스 제어 \(p. 390\)](#)을(를) 참조하십시오.

인스턴스에 대한 memcached 지원을 비활성화하려면 인스턴스를 변경하여 MySQL 버전의 기본 옵션 그룹을 지정합니다. MySQL 인스턴스 설정 변경에 대한 자세한 내용은 [MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 600\)](#)을 참조하십시오.

MySQL memcached 보안 고려 사항

memcached 프로토콜은 사용자 인증을 지원하지 않습니다. MySQL memcached 보안 고려 사항에 대한 자세한 내용은 [memcached 배포 및 memcached를 MySQL 캐싱 레이어로 사용](#)을 참조하십시오.

다음은 memcached 인터페이스의 보안을 강화하는 데 효과적인 작업입니다.

- MEMCACHED 옵션을 옵션 그룹에 추가할 때 기본 11211이 아닌 다른 포트를 지정합니다.
- 기존에 신뢰할 수 있는 클라이언트 주소나 EC2 인스턴스에 대한 액세스를 제한하는 VPC 또는 DB 보안 그룹과 memcached 인터페이스가 연동되어 있는지 확인합니다. 보안 그룹 관리에 대한 자세한 내용은 [보안 그룹을 통한 액세스 제어 \(p. 390\)](#)을(를) 참조하십시오.

MySQL memcached 연결 정보

memcached 인터페이스에 연결하려면 애플리케이션이 Amazon RDS 인스턴스의 DNS 이름과 memcached 포트 번호를 모두 지정해야 합니다. 예를 들어 인스턴스의 DNS 이름이 my-cache-

instance.cg034hpkmmjt.region.rds.amazonaws.com이고, memcached 인스턴스가 포트 11212를 사용한다면 PHP에 지정되는 연결 정보는 다음과 같습니다.

```
<?php  
  
$cache = new Memcache;  
$cache->connect('my-cache-instance.cg034hpkmmjt.region.rds.amazonaws.com', 11212);  
?>
```

Amazon RDS MySQL 인스턴스의 DNS 이름과 memcached 포트를 찾으려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. AWS Management 콘솔 우측 상단 모서리에서 DB 인스턴스가 속한 리전을 선택합니다.
3. 탐색 창에서 데이터베이스를 선택합니다.
4. 세부 정보를 표시하고자 하는 MySQL DB 인스턴스 이름을 선택합니다.
5. [Connect] 섹션에 있는 [Endpoint] 필드의 값을 기록해둡니다. DNS 이름은 엔드포인트와 같습니다. 또한 [Connect] 섹션에 있는 포트는 memcached 인터페이스에 액세스하는 데 사용되지 않습니다.
6. 세부 정보 섹션의 옵션 그룹 필드에 나열된 이름을 기록해둡니다.
7. 탐색 창에서 옵션 그룹을 선택합니다.
8. MySQL DB 인스턴스가 사용하는 옵션 그룹의 이름을 선택하여 옵션 그룹 세부 정보를 표시합니다. [Options] 섹션에 있는 [MEMCACHED] 옵션의 [Port] 설정 값을 기록해둡니다.

MySQL memcached 옵션 설정

Amazon RDS는 MySQL memcached 파라미터를 Amazon RDS MEMCACHED 옵션의 옵션 설정으로 노출시킵니다.

MySQL memcached 파라미터

- DAEMON_MEMCACHED_R_BATCH_SIZE – 커밋으로 새로운 트랜잭션을 시작하기 전에 실행해야 할 memcached 읽기 연산(get) 수를 지정하는 정수입니다. 허용 값은 1~4294967295이고, 기본값은 1입니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다.
- DAEMON_MEMCACHED_W_BATCH_SIZE – 커밋으로 새로운 트랜잭션을 시작하기 전에 실행해야 할 memcached 쓰기 연산(add, set, incr 등) 수를 지정하는 정수입니다. 허용 값은 1~4294967295이고, 기본값은 1입니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다.
- INNODB_API_BK_COMMIT_INTERVAL – InnoDB memcached 인터페이스를 사용하는 자동 커밋 모드의 유회 연결(idle connection) 횟수를 지정하는 정수입니다. 허용 값은 1~1073741824이고, 기본값은 5입니다. 인스턴스를 다시 시작할 필요 없이 옵션이 바로 적용됩니다.
- INNODB_API_DISABLE_ROWLOCK – InnoDB memcached 인터페이스 사용 시 행 잠금 기능을 활성화(1, true) 또는 비활성화(0, false)하는 부울입니다. 기본값은 0(false)입니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다.
- INNODB_API_ENABLE_MDL – 0(false)으로 설정 시 InnoDB memcached 플러그인이 사용하는 테이블을 잠그는 부울입니다. 따라서 SQL 인터페이스에서 DDL을 통해 삭제 또는 변경이 불가능합니다. 기본값은 0(false)입니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다.
- INNODB_API_TRX_LEVEL – memcached 인터페이스에서 처리하는 쿼리의 트랜잭션 격리 수준을 지정하는 정수입니다. 허용 값은 0~3입니다. 기본값은 0입니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다.

Amazon RDS가 위 MySQL memcached 파라미터를 구성하지만 다음 파라미터는 변경할 수 없습니다.
DAEMON_MEMCACHED_LIB_NAME, DAEMON_MEMCACHED_LIB_PATH 및 INNODB_API_ENABLE_BINLOG.

MySQL 관리자가 `daemon_memcached_options`를 사용하여 설정하는 파라미터는 Amazon RDS의 개별 MEMCACHED 옵션 설정으로 이용할 수 있습니다.

MySQL `daemon_memcached_options` 파라미터

- `BINDING_PROTOCOL` – 사용할 바인딩 프로토콜을 지정하는 문자열입니다. 허용 값은 `auto`, `ascii` 또는 `binary`입니다. 기본 값은 `auto`이고, 이 경우 서버가 자동으로 클라이언트와 프로토콜을 협상합니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다.
- `BACKLOG_QUEUE_LIMIT` – memcached의 처리를 기다리는 네트워크 연결 수를 지정하는 정수입니다. 이 파라미터의 최대 값을 높이면 memcached 인스턴스에 연결할 수 없다는 클라이언트의 오류 메시지가 줄어들 수 있지만 그렇다고 서버 성능이 향상되는 것은 아닙니다. 허용 값은 1~2048이고, 기본값은 1024입니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다.
- `CAS_DISABLED` – Compare and Swap(CAS) 기능을 활성화(1. `true`)하거나 비활성화(0. `false`)하는 부울로서, 이 기능을 활성화하면 항목 1개의 크기가 8바이트까지 줄어듭니다. 기본 값은 0(`false`)입니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다.
- `CHUNK_SIZE` – 가장 작은 항목의 키, 값 및 플래그에 할당할 최소 청크 크기(바이트)를 지정하는 정수입니다. 허용 값은 1~48입니다. 기본 값은 48이며, 값이 작을수록 메모리 효율이 크게 개선됩니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다.
- `CHUNK_SIZE_GROWTH_FACTOR` – 새로운 청크 크기를 조절하는 부동 소수점입니다. 이전 청크 크기와 `CHUNK_SIZE_GROWTH_FACTOR`를 곱한 값이 새로운 청크의 크기가 됩니다. 허용되는 값은 1~2이고, 기본 값은 1.25입니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다.
- `ERROR_ON_MEMORY_EXHAUSTED` – 1(`true`)로 설정할 경우, 항목을 저장할 메모리가 없으면 memcached가 항목을 제거하지 않고 오류를 반환하는 부울입니다. 0(`false`)으로 설정할 경우 메모리가 없으면 memcached가 항목을 제거합니다. 기본 값은 0(`false`)입니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다.
- `MAX_SIMULTANEOUS_CONNECTIONS` – 동시에 접속할 수 있는 최대 수를 지정하는 정수입니다. 이 값을 10보다 작게 설정하면 MySQL을 시작하지 못합니다. 허용 값은 10~1024이고, 기본값은 1024입니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다.
- `VERBOSITY` – MySQL 오류 로그에서 memcached 서비스가 기록할 정보 수준을 지정하는 문자열입니다. 기본값은 `v`입니다. 옵션을 적용하려면 인스턴스를 다시 시작해야 합니다. 허용 값은 다음과 같습니다.
 - `v` – 주요 이벤트 루프 실행 중 발생하는 오류와 경고를 기록합니다.
 - `vv` – `v`에서 기록하는 정보에 더하여 각 클라이언트 명령과 응답을 기록합니다.
 - `vvv` – `vv`에서 기록하는 정보에 더하여 내부의 상태 전환을 기록합니다.

Amazon RDS가 위 MySQL `DAEMON_MEMCACHED_OPTIONS` 파라미터를 구성하지만 다음 파라미터는 변경할 수 없습니다. `DAEMON_PROCESS`, `LARGE_MEMORY_PAGES`, `MAXIMUM_CORE_FILE_LIMIT`, `MAX_ITEM_SIZE`, `LOCK_DOWN_PAGE_MEMORY`, `MASK`, `IDFILE`, `REQUESTS_PER_EVENT`, `SOCKET` 및 `USER`.

MySQL DB 인스턴스에 대한 공통 DBA 작업

이번 섹션에서는 MySQL 데이터베이스 엔진을 기반으로 한 몇 가지 DBA의 공통 DB 인스턴스 작업을 Amazon RDS에 따라 구현하는 방법에 대해 살펴보겠습니다. 관리되는 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 셀 액세스를 제공하지 않으며, 고급 권한을 필요로 하는 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다.

Amazon RDS의 MySQL 로그 파일 작업 방법에 대한 자세한 내용은 [MySQL 데이터베이스 로그 파일 \(p. 318\)](#) 단원을 참조하십시오.

주제

- [세션 또는 쿼리 종료 \(p. 670\)](#)

- 현재 복제 오류 넘어가기 (p. 670)
- 충돌 복구 시간 개선을 위한 InnoDB 테이블스페이스 작업 (p. 671)
- 전역적 상태 이력 관리 (p. 672)

세션 또는 쿼리 종료

`rds_kill` 및 `rds_kill_query` 명령을 사용하여 DB 인스턴스의 사용자 세션이나 쿼리를 종료할 수 있습니다. 먼저 MySQL 데이터베이스 인스턴스에 연결한 후 다음과 같이 해당 명령을 실행합니다. 자세한 내용은 [MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 595\)](#) 단원을 참조하십시오.

```
CALL mysql.rds_kill(thread-ID)
CALL mysql.rds_kill_query(thread-ID)
```

예를 들어, 스레드 99에서 실행 중인 세션을 종료하려면 다음과 같이 입력합니다.

```
CALL mysql.rds_kill(99);
```

또한 스레드 99에서 실행 중인 쿼리를 종료하려면 다음과 같이 입력합니다.

```
CALL mysql.rds_kill_query(99);
```

현재 복제 오류 넘어가기

Amazon RDS는 오류로 인해 읽기 전용 복제본이 중단되는 경우 읽기 전용 복제본의 오류를 건너뛸 수 있는 메커니즘을 제공하기 때문에 오류가 발생하더라도 데이터 무결성은 어떠한 영향도 받지 않습니다. 먼저 MySQL 데이터베이스 인스턴스에 연결한 후 다음과 같이 해당 명령을 실행합니다. 자세한 내용은 [MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 595\)](#) 단원을 참조하십시오.

Note

먼저 안전하게 건너뛸 수 있는 오류인지 확인해야 합니다. MySQL 유ти리티에서 읽기 전용 복제본에 연결한 후 다음 MySQL 명령을 실행합니다.

```
SHOW SLAVE STATUS\G
```

반환 값에 대한 자세한 내용은 MySQL 문서에서 [SHOW SLAVE STATUS Syntax](#)을(를) 참조하십시오.

오류를 건너뛰려면 다음 명령을 실행합니다.

```
CALL mysql.rds_skip_repl_error;
```

이 명령은 원본 DB 인스턴스에서 실행하거나, 혹은 복제 오류가 발생하지 않은 읽기 전용 복제본에서 실행하는 경우 아무런 효과도 없습니다.

`mysql.rds_skip_repl_error`가 지원되는 MySQL 버전 등에 대한 자세한 내용은 [mysql.rds_skip_repl_error \(p. 692\)](#) 단원을 참조하십시오.

Important

`mysql.rds_skip_repl_error`를 호출하려고 할 때 `ERROR 1305 (42000): PROCEDURE mysql.rds_skip_repl_error does not exist`과 같은 오류가 발생한 경우에는 MySQL DB

인스턴스를 최신 마이너 버전이나 [mysql.rds_skip_repl_error \(p. 692\)](#)에 등록된 최소 마이너 버전 중 하나로 업그레이드해야 합니다.

충돌 복구 시간 개선을 위한 InnoDB 테이블스페이스 작업

MySQL의 모든 테이블은 테이블 정의, 데이터 및 인덱스로 구성되어 있습니다. InnoDB는 MySQL 스토리지 엔진으로서 테이블 데이터와 인덱스를 테이블스페이스에 저장하는 역할을 합니다. 이 스토리지 엔진은 전역적 공유 테이블스페이스를 생성하여 데이터 사전을 비롯한 기타 관련 메타데이터, 그리고 테이블 데이터와 인덱스도 저장합니다. 또한 테이블 및 파티션마다 별도의 테이블스페이스를 생성할 수도 있습니다. 이렇게 별도로 생성된 테이블스페이스는 확장자가 .ibd인 파일에 저장되며, 각 테이블스페이스 헤더에는 식별할 수 있도록 고유 번호가 포함됩니다.

Amazon RDS는 MySQL 파라미터 그룹을 통해 `innodb_file_per_table`이라고 하는 파라미터를 하나 제공합니다. 이 파라미터는 InnoDB가 새로운 테이블 데이터와 인덱스를 공유 테이블스페이스에 추가할지 (파라미터 값을 0으로 설정), 혹은 개별 테이블스페이스에 추가할지(파라미터 값을 1로 설정) 결정합니다. Amazon RDS에서는 `innodb_file_per_table` 파라미터 값이 1로 기본 설정되어 있습니다. 이 설정에서 개별 InnoDB 테이블을 삭제하고 이 테이블에서 DB 인스턴스로 사용되었던 스토리지를 회수할 수 있습니다. 대부분 사용 사례에서 `innodb_file_per_table` 파라미터는 1로 설정하는 것이 바람직합니다.

하지만 표준 스토리지(마그네틱)나 일반 SSD 스토리지를 사용하여 테이블 수가 1,000개를 넘거나, 혹은 프로비저닝된 IOPS 스토리지를 사용하여 테이블 수가 10,000개를 넘는 등 테이블 수가 많을 때는 `innodb_file_per_table` 파라미터를 0으로 설정해야 합니다. 이 파라미터를 0으로 설정하면 테이블스페이스가 개별적으로 생성되지 않기 때문에 데이터베이스 충돌 복구에 걸리는 시간을 개선할 수 있습니다.

MySQL은 충돌 복구 주기에서 테이블스페이스가 저장된 메타데이터 파일을 각각 처리합니다. MySQL이 공유 테이블스페이스에 저장된 메타데이터 정보를 처리하는 데 걸리는 시간은 다수의 테이블스페이스로 인해 수천 개의 테이블스페이스 파일을 처리하는 데 걸리는 시간에 비하면 무시해도 될 정도입니다. 테이블스페이스 번호는 각 파일의 헤더에 저장되기 때문에 모든 테이블스페이스 파일을 읽으려면 최대 몇 시간까지 걸릴 수 있습니다. 예를 들어 표준 스토리지에 InnoDB 테이블스페이스가 수백만 개 저장되어 있다면 충돌 복구 주기에서 처리하는 데만 5~8시간이 소요됩니다. 경우에 따라 충돌 복구 주기가 끝나더라도 InnoDB가 추가 정리가 필요하다고 판단할 경우에는 또 다른 충돌 복구 주기가 시작되면서 복구 시간이 연장됩니다. 또 한 가지, 충돌 복구 주기는 테이블스페이스 정보 처리 외에도 롤링백 트랜잭션, 손상된 페이지 복구, 그리고 그 밖의 작업까지 수반한다는 점도 잊어서는 안 됩니다.

`innodb_file_per_table` 파라미터는 파라미터 그룹에 속하기 때문에 DB 인스턴스에 사용되는 파라미터 그룹만 편집하면 파라미터 값이 변경됩니다. 따라서 DB 인스턴스를 재부팅할 필요가 없습니다. 예를 들어 설정을 1(개별 테이블 생성)에서 0(공유 테이블스페이스 사용)으로 변경하면 새로운 InnoDB 테이블이 공유 테이블스페이스에 추가되는 반면 기존 테이블은 개별 테이블스페이스를 그대로 유지합니다. InnoDB 테이블을 공유 테이블스페이스로 이동하려면 `ALTER TABLE` 명령을 사용해야 합니다.

여러 테이블스페이스를 공유 테이블스페이스로 마이그레이션

InnoDB 테이블의 메타데이터를 자체 테이블스페이스에서 공유 테이블스페이스로 이동할 수 있습니다. 이렇게 하면 `innodb_file_per_table` 파라미터 설정에 따라 테이블 메타데이터가 다시 작성됩니다. 먼저 MySQL 데이터베이스 인스턴스에 연결한 후 다음과 같이 해당 명령을 실행합니다. 자세한 내용은 [MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 595\)](#) 단원을 참조하십시오.

```
ALTER TABLE table_name ENGINE = InnoDB, ALGORITHM=COPY;
```

예를 들어 다음 쿼리는 공유 테이블스페이스에 없는 모든 InnoDB 테이블에 대해 `ALTER TABLE` 문을 반환합니다.

```
SELECT CONCAT('ALTER TABLE `',
```

```
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '`', ``), `.` , `` , `` ENGINE=InnoDB,  
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '`, ``), `` AS Query  
ALGORITHM=COPY;')  
FROM INFORMATION_SCHEMA.INNODB_SYS_TABLES  
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

Note

이 쿼리는 MySQL 5.6 이상에서 지원됩니다.

MySQL 테이블을 리빌드하여 테이블의 메타데이터를 공유 테이블스페이스로 이동하려면 테이블을 리빌드 할 수 있는 스토리지 공간이 임시로 필요합니다. DB 인스턴스에 여유 스토리지 공간이 있어야 하는 이유도 바로 이 때문입니다. 리빌드 단계에서는 테이블이 잠겨서 쿼리에 액세스하지 못합니다. 작은 용량의 테이블이나 자주 액세스하지 않는 테이블의 경우 이것이 문제가 되지는 않습니다. 하지만 대용량의 테이블이나 동시에 접속자 수가 많은 환경에서 자주 액세스하는 테이블이라면 테이블을 읽기 전용 복제본에 리빌드할 수도 있습니다.

읽기 전용 복제본을 생성한 후 테이블 메타데이터를 읽기 전용 복제본의 공유 테이블스페이스로 마이그레이션하는 방법도 있습니다. ALTER TABLE 문이 읽기 전용 복제본에 대한 액세스를 차단하더라도 원본 DB 인스턴스는 영향을 받지 않습니다. 따라서 테이블 리빌딩 프로세스로 인해 읽기 전용 복제본이 지연되더라도 원본 DB 인스턴스는 계속해서 이진 로그를 생성할 수 있습니다. 리빌딩 프로세스에는 스토리지 공간이 추가로 필요할 뿐만 아니라 릴레이 로그 파일이 커질 수도 있기 때문에 원본 DB 인스턴스보다 큰 용량의 스토리지를 할당하여 읽기 전용 복제본을 생성해야 합니다.

읽기 전용 복제본을 생성하여 InnoDB 테이블을 리빌드한 후 공유 테이블스페이스를 사용하려면 다음 단계를 따라야 합니다.

1. 이진 로그 생성을 계속 할 수 있도록 원본 DB 인스턴스에 백업 보존 기간이 활성화되어 있는지 확인합니다.
2. AWS 콘솔 또는 AWS CLI를 사용하여 원본 DB 인스턴스의 읽기 전용 복제본을 생성합니다. 읽기 전용 복제본을 생성하려면 충돌 복구와 같이 다수의 동일한 프로세스를 거쳐야 하기 때문에 InnoDB 테이블스페이스가 많을 경우에는 일정 시간이 걸릴 수도 있습니다. 이때 읽기 전용 복제본에 할당하는 스토리지 공간은 현재 원본 DB 인스턴스에 사용 중인 스토리지 공간보다 많아야 합니다.
3. 읽기 전용 복제본이 생성된 후에는 파라미터 설정 `read_only = 0` 및 `innodb_file_per_table = 0`으로 구성된 파라미터 그룹을 생성한 다음 이 그룹을 읽기 전용 복제본과 연동 시킵니다.
4. 복제본에서 마이그레이션하고자 하는 모든 테이블에 대해 `ALTER TABLE <name> ENGINE = InnoDB`를 실행합니다.
5. 읽기 전용 복제본에서 `ALTER TABLE` 문을 모두 실행한 후에는 읽기 전용 복제본이 원본 DB 인스턴스에 연결되어 있고, 두 인스턴스가 동기화되어 있는지 확인합니다.
6. 준비가 끝났으면 AWS 콘솔 또는 AWS CLI를 사용해 읽기 전용 복제본을 마스터 인스턴스로 승격합니다. 새로운 마스터에 사용한 파라미터 그룹에서 `innodb_file_per_table` 파라미터가 0으로 설정되어 있는지 확인합니다. 새로운 마스터 이름을 변경하고 모든 애플리케이션이 새로운 마스터 인스턴스를 가리키도록 합니다.

전역적 상태 이력 관리

MySQL은 작업 관련 정보를 알 수 있는 다수의 상태 변수를 유지하고 있습니다. 이 변수 값은 DB 인스턴스에서 잠금 또는 메모리 문제를 파악하는 데 효과적입니다. DB 인스턴스를 마지막으로 시작한 때부터 계속해서 누적되기 때문입니다. 대부분 상태 변수는 `FLUSH STATUS` 명령을 사용해 0으로 재설정할 수 있습니다.

Amazon RDS는 시간이 지나면서 이 변수 값의 스냅샷을 캡처하거나, 마지막 스냅샷 이후 모든 변경 사항과 함께 변수 값을 테이블에 기록하는 등 시간 경과에 따른 상태 변수 값을 모니터링할 수 있는 프로시저를 지원합니다. 이러한 인프라를 전역적 상태 이력(GoSH)이라고 부릅니다. GoSH는 버전 5.5.23부터 모든 MySQL DB 인스턴스에 설치되기 시작했지만 기본적으로 비활성화되어 있습니다.

GoSH를 활성화하려면 먼저 파라미터 event_scheduler를 ON으로 설정하여 DB 파라미터 그룹의 이벤트 스케줄러를 활성화해야 합니다. DB 파라미터 그룹의 생성 및 변경에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#)을(를) 참조하십시오.

그런 다음 아래 표의 프로시저에 따라 GoSH를 활성화 및 구성할 수 있습니다. 먼저 MySQL 데이터베이스 인스턴스에 연결한 후 다음과 같이 해당 명령을 실행합니다. 자세한 정보는 [MySQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 595\)](#) 단원을 참조하십시오. 각 절차에 대해 다음을 입력합니다.

```
CALL procedure-name;
```

여기서 procedure-name에는 표에 보이는 프로시저 중 하나가 들어갑니다.

프로시저	설명
rds_enable_gsh_collector	rds_set_gsh_collector에서 설정한 주기에 따라 기본 스냅샷을 캡처하도록 GoSH를 활성화합니다.
rds_set_gsh_collector	스냅샷 캡처 주기(분)를 지정합니다. 기본 값은 5입니다.
rds_disable_gsh_collector	스냅샷을 비활성화합니다.
rds_collect_global_status_history	필요할 경우에만 스냅샷을 캡처합니다.
rds_enable_gsh_rotation	mysql.rds_global_status_history 테이블의 내용이 rds_set_gsh_rotation에서 설정한 주기에 따라 mysql.rds_global_status_history_old로 로테이션됩니다.
rds_set_gsh_rotation	테이블 로테이션 주기(일)를 지정합니다. 기본 값은 7입니다.
rds_disable_gsh_rotation	테이블 로테이션을 비활성화합니다.
rds_rotate_global_status_history	필요에 따라 mysql.rds_global_status_history 테이블의 내용을 mysql.rds_global_status_history_old로 로테이션합니다.

GoSH가 활성화되어 있을 때는 쓰기가 가능한 테이블에 쿼리를 요청할 수 있습니다. 예를 들어 InnoDB 버퍼풀의 적중률에 대한 쿼리를 요청하려면 다음과 같이 쿼리를 실행합니다.

```
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'
```

Amazon RDS MySQL에 대해 알려진 문제 및 제한

Amazon RDS MySQL 사용 시 알려진 문제 및 제한은 다음과 같습니다.

주제

- [일관되지 않은 InnoDB 버퍼풀 크기 \(p. 674\)](#)
- [인덱스 병합 최적화가 잘못된 결과를 반환 \(p. 674\)](#)
- [로그 파일 크기 \(p. 675\)](#)

- Amazon RDS DB 인스턴스에 대한 MySQL 파라미터 예외 (p. 675)
- MySQL 파일 크기 제한 (p. 675)

일관되지 않은 InnoDB 버퍼 풀 크기

MySQL 5.7에는 현재 InnoDB 버퍼 풀 크기가 관리되지 않는 버그가 있습니다. MySQL 5.7에서 innodb_buffer_pool_size 파라미터 값을 InnoDB 버퍼 풀 크기를 너무 많이 늘려 너무 많은 메모리를 소모하도록 하는 큰 값으로 조정할 수 있습니다. 이에 따라 MySQL 데이터베이스 엔진이 실행을 중단하거나 MySQL 데이터베이스 엔진이 시작하지 못할 수 있습니다. 이 문제는 사용 가능한 메모리가 적은 DB 인스턴스 클래스 문제보다 더 일반적으로 발생합니다.

이 문제를 해결하려면 innodb_buffer_pool_size 파라미터 값을 innodb_buffer_pool_instances 파라미터 값과 innodb_buffer_pool_chunk_size 파라미터 값의 곱의 배수로 설정해야 합니다. 예를 들어, 다음 예제와 같이 innodb_buffer_pool_size 파라미터 값을 innodb_buffer_pool_instances와 innodb_buffer_pool_chunk_size 파라미터 값의 곱의 8배로 설정할 수 있습니다.

```
innodb_buffer_pool_chunk_size = 536870912
innodb_buffer_pool_instances = 4
innodb_buffer_pool_size = (536870912 * 4) * 8 = 17179869184
```

MySQL 5.7 버그에 대한 자세한 내용은 MySQL 문서의 <https://bugs.mysql.com/bug.php?id=79379>를 참조하십시오.

인덱스 병합 최적화가 잘못된 결과를 반환

인덱스 병합 최적화를 사용하는 쿼리는 MySQL 5.5.37에서 도입된 MySQL 쿼리 옵티마이저의 버그로 인해 잘못된 결과를 반환할 수 있습니다. 여러 개의 인덱스가 있는 테이블에 대해 쿼리를 실행하면 옵티마이저가 여러 개의 인덱스를 기반으로 여러 범위의 행을 스캔하지만 결과를 올바르게 함께 병합하지 않습니다. 쿼리 최적화 프로그램 버그에 대한 자세한 내용은 MySQL 버그 데이터베이스의 <http://bugs.mysql.com/bug.php?id=72745> 및 <http://bugs.mysql.com/bug.php?id=68194>을(를) 참조하십시오.

예를 들면, 검색 인수가 인덱싱된 열을 참조하는 2개의 인덱스가 있는 테이블에 대한 쿼리를 고려합니다.

```
SELECT * FROM table1
WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

이 경우 검색 엔진이 두 인덱스를 모두 검색합니다. 그러나 버그로 인해 병합 결과가 정확하지 않습니다.

이 문제를 해결하려면 다음 중 한 가지 방법을 시도하면 됩니다.

- MySQL DB 인스턴스용 DB 파라미터 그룹에서 optimizer_switch 파라미터를 index_merge=off로 설정합니다. DB 파라미터 그룹 파라미터 설정에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.
- MySQL DB 인스턴스를 MySQL 버전 5.6, 5.7 또는 8.0로 업그레이드하십시오. 자세한 내용은 [MySQL DB 스냅샷 업그레이드 \(p. 612\)](#)을(를) 참조하십시오.
- 인스턴스를 업그레이드하거나 optimizer_switch 파라미터를 변경할 수 없는 경우, 쿼리에 대한 인덱스를 명시적으로 확인하여 버그를 해결할 수 있습니다. 예:

```
SELECT * FROM table1
USE INDEX covering_index
WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

자세한 내용은 [인덱스 병합 최적화](#)을(를) 참조하십시오.

로그 파일 크기

MySQL의 경우 다시 실행 로그에 기록되는 BLOB에 대한 크기 제한이 있습니다. 이러한 제한을 감안하려면 MySQL DB 인스턴스에 대한 `innodb_log_file_size` 파라미터가 테이블에서 발견된 BLOB 데이터의 최대 크기 및 동일한 테이블에서의 다른 가변 길이 필드(VARCHAR, VARBINARY, TEXT)의 길이 보다 10배 커야 합니다. 파라미터 값 설정 방법에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오. 다시 실행 로그 BLOB 크기 제한에 대한 자세한 내용은 [MySQL 5.6.20의 변경 내용](#)(를) 참조하십시오.

Amazon RDS DB 인스턴스에 대한 MySQL 파라미터 예외

일부 MySQL 파라미터의 경우 Amazon RDS DB 인스턴스와 함께 사용할 때 특별히 고려해야 할 사항이 있습니다.

lower_case_table_names

Amazon RDS는 대/소문자 구분 파일 시스템을 사용하므로 `lower_case_table_names` 서버 파라미터의 값을 2("이름은 지정된 대로 저장되지만 소문자로 비교됨")으로 설정하는 것은 지원하지 않습니다. Amazon RDS DB 인스턴스에 지원되는 값은 기본값인 0("이름은 지정된 대로 저장되며 비교는 대/소문자로 구분하여 진행됨") 또는 1("이름은 소문자로 저장되며 비교는 대/소문자를 구분하지 않고 진행됨")입니다.

`lower_case_table_names` 파라미터는 DB 인스턴스를 생성하기 전에 사용자 지정 DB 파라미터 그룹의 일부로 설정해야 합니다. 지정 시간 복구 백업과 읽기 전용 복제본 DB 인스턴스와 볼일치를 유발할 수 있으므로 기존 데이터베이스 인스턴스에 대한 `lower_case_table_names` 파라미터를 변경하는 것을 피해야 합니다.

읽기 전용 복제본은 마스터 DB 인스턴스로 항상 동일한 `lower_case_table_names` 파라미터 값을 사용해야 합니다.

long_query_time

マイクロ초 해상도(microsecond resolution)로 느린 쿼리를 MySQL의 느린 쿼리 로그에 기록할 수 있도록 `long_query_time` 파라미터를 부동 소수점 값으로 설정합니다. 100밀리초에 해당하는 0.1초와 같은 값을 설정하여 1초 이내의 시간이 걸리는 느린 트랜잭션을 디버깅할 때 도움을 받을 수 있습니다.

MySQL 파일 크기 제한

Amazon RDS MySQL DB 인스턴스의 경우 최대 프로비저닝 스토리지 제한으로 인해 InnoDB 테이블당 파일 테이블스페이스를 사용하여 각 테이블의 크기가 최대 16TB로 제한됩니다. 또한 이 제한은 시스템 테이블스페이스를 최대 16TB의 크기로 제한합니다. Amazon RDS MySQL DB 인스턴스에서는 테이블이 각각 자체 테이블스페이스에 들어 있는 InnoDB 테이블당 파일 테이블스페이스가 기본적으로 설정됩니다.

Note

일부 기존 DB 인스턴스에는 하한이 있습니다. 예를 들어 2014년 4월 이전에 생성된 MySQL DB 인스턴스는 파일과 테이블 크기 제한이 2TB입니다. 마찬가지로 DB 인스턴스의 생성 시기와 상관없이 2014년 4월 이전에 생성된 DB 스냅샷에서 생성한 DB 인스턴스나 읽기 복제 역시 파일 크기가 2TB로 제한됩니다.

애플리케이션에 따라 InnoDB 테이블당 파일 테이블스페이스 사용에 대한 장점과 단점은 서로 다릅니다. 애플리케이션에 가장 적합한 접근 방식을 확인하려면 MySQL 문서의 [InnoDB 테이블당 파일 모드](#)(를) 참조하십시오.

테이블을 최대 파일 크기로 늘리도록 허용하는 것은 권장하지 않습니다. 일반적으로 모범 사례는 성능 및 복구 시간을 향상할 수 있도록 데이터를 더 작은 테이블로 분할하는 것입니다.

라지 테이블을 여러 개의 스몰 테이블로 분할하는 데 사용할 수 있는 한 가지 옵션으로 파티셔닝이 있습니다. 파티셔닝을 수행하면 사용자가 지정하는 규칙에 따라 라지 테이블의 일부가 개별 파일로 배포됩니다. 예를 들어, 트랜잭션을 날짜별로 저장하는 경우 파티셔닝을 사용하여 이전 트랜잭션을 개별 파일로 배포하는 파티셔닝 규칙을 생성할 수 있습니다. 이렇게 하면 애플리케이션에서 즉시 사용할 필요가 없는 이전 트랜잭션 데이터를 주기적으로 보관할 수 있습니다. 자세한 내용은 MySQL 문서의 <https://dev.mysql.com/doc/refman/5.6/en/partitioning.html>을(를) 참조하십시오.

테이블의 파일 크기를 확인하는 방법

- 다음 SQL 명령을 사용하여 크기가 너무 커서 파티셔닝을 수행해야 하는 테이블이 있는지 확인합니다.

```
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');
```

InnoDB 테이블당 파일 테이블스페이스를 활성화하는 방법

- InnoDB 테이블당 파일 테이블스페이스를 활성화하려면 DB 인스턴스에 대한 파라미터 그룹에서 innodb_file_per_table 파라미터를 1로 설정합니다.

InnoDB 테이블당 파일 테이블스페이스를 비활성화하는 방법

- InnoDB 테이블당 파일 테이블스페이스를 비활성화하려면 DB 인스턴스에 대한 파라미터 그룹에서 innodb_file_per_table 파라미터를 0으로 설정합니다.

파라미터 그룹 업데이트에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#)을(를) 참조하십시오.

InnoDB 테이블당 파일 테이블스페이스를 활성화하거나 비활성화한 경우 ALTER TABLE 명령을 실행하여 아래의 예와 같이 테이블을 전역 테이블스페이스에서 자체 테이블스페이스로 이동하거나 자체 테이블스페이스에서 전역 테이블스페이스로 이동할 수 있습니다.

```
ALTER TABLE table_name ENGINE=InnoDB;
```

Amazon RDS SQL의 MySQL 참조

이 부록에서는 MySQL DB 엔진을 실행 중인 Amazon RDS 인스턴스에 사용할 수 있는 시스템 저장 프로시저를 설명합니다.

개요

다음은 MySQL을 실행 중인 Amazon RDS 인스턴스에 대해 지원되는 시스템 저장 프로시저입니다.

복제

- [mysql.rds_set_master_auto_position \(p. 678\)](#)
- [mysql.rds_set_external_master \(p. 678\)](#)
- [mysql.rds_set_external_master_with_delay \(p. 680\)](#)
- [mysql.rds_set_external_master_with_auto_position \(p. 683\)](#)
- [mysql.rds_reset_external_master \(p. 685\)](#)

- [mysql.rds_import_binlog_ssl_material \(p. 685\)](#)
- [mysql.rds_remove_binlog_ssl_material \(p. 687\)](#)
- [mysql.rds_set_source_delay \(p. 687\)](#)
- [mysql.rds_start_replication \(p. 688\)](#)
- [mysql.rds_start_replication_until \(p. 689\)](#)
- [mysql.rds_start_replication_until_gtid \(p. 690\)](#)
- [mysql.rds_stop_replication \(p. 691\)](#)
- [mysql.rds_skip_transaction_with_gtid \(p. 691\)](#)
- [mysql.rds_skip_repl_error \(p. 692\)](#)
- [mysql.rds_next_master_log \(p. 693\)](#)

InnoDB 캐시 위밍

- [mysql.rds_innodb_buffer_pool_dump_now \(p. 694\)](#)
- [mysql.rds_innodb_buffer_pool_load_now \(p. 695\)](#)
- [mysql.rds_innodb_buffer_pool_load_abort \(p. 695\)](#)

추가 구성 관리(예: binlog 파일 보존)

- [mysql.rds_set_configuration \(p. 695\)](#)
- [mysql.rds_show_configuration \(p. 697\)](#)

세션 또는 쿼리 종료

- [mysql.rds_kill \(p. 698\)](#)
- [mysql.rds_kill_query \(p. 698\)](#)

로깅

- [mysql.rds_rotate_general_log \(p. 699\)](#)
- [mysql.rds_rotate_slow_log \(p. 699\)](#)

전역적 상태 이력 관리

- [mysql.rds_enable_gsh_collector \(p. 699\)](#)
- [mysql.rds_set_gsh_collector \(p. 699\)](#)
- [mysql.rds_disable_gsh_collector \(p. 700\)](#)
- [mysql.rds_collect_global_status_history \(p. 700\)](#)
- [mysql.rds_enable_gsh_rotation \(p. 700\)](#)
- [mysql.rds_set_gsh_rotation \(p. 700\)](#)
- [mysql.rds_disable_gsh_rotation \(p. 701\)](#)
- [mysql.rds_rotate_global_status_history \(p. 701\)](#)

SQL 참조 규칙

아래에서 SQL 참조 섹션에 설명된 시스템 저장 절차와 테이블의 구문을 설명하는 데 사용되는 규칙을 확인 할 수 있습니다.

문자	설명
대문자	대문자로 표시된 단어는 키워드입니다.
[]	대괄호는 선택적 인수를 나타냅니다.
{}	중괄호는 중괄호 안의 인수 중 하나를 선택해야 함을 나타냅니다.
	세로줄은 선택할 수 있는 인수를 구분합니다.
기울임꼴	기울임꼴 단어는 자리 표시자를 나타냅니다. 기울임꼴 단어 자리에 적절한 값을 넣어야 합니다.
...	줄임표는 앞의 요소를 반복할 수 있음을 나타냅니다.
'	작은따옴표로 둑인 단어는 따옴표를 입력해야 함을 나타냅니다.

mysql.rds_set_master_auto_position

복제 모드를 바이너리 로그 파일 위치 또는 전역 트랜잭션 식별자(GTID)를 기반으로 설정합니다.

구문

```
CALL mysql.rds_set_master_auto_position (
    auto_position_mode
);
```

파라미터

auto_position_mode

로그 파일 위치 복제 또는 GTID를 기반으로 하는 복제를 사용할지 여부를 나타내는 값:

- 0 – 바이너리 로그 파일 위치를 기반으로 한 복제 방법을 사용합니다. 기본값은 0입니다.
- 1 – GTID 기반 복제 방법을 사용합니다.

사용 시 주의사항

마스터 사용자는 mysql.rds_set_master_auto_position 절차를 실행해야 합니다.

Amazon RDS MySQL 5.7의 경우 이 절차는 MySQL 5.7.23 및 MySQL 5.7 이상 버전에서 지원합니다.
Amazon RDS MySQL 5.5, 5.6 또는 8.0의 경우 이 절차는 지원하지 않습니다.

mysql.rds_set_external_master

MySQL DB 인스턴스를 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스의 읽기 전용 복제본으로 구성합니다.

Note

[mysql.rds_set_external_master_with_delay \(p. 680\)](#) 저장 프로시저를 사용하여 외부 마스터와 지연 복제를 구성할 수 있습니다.

구문

```
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  
);
```

파라미터

host_name

복제 마스터가 될 Amazon RDS의 외부에서 실행 중인 MySQL 인스턴스의 호스트 이름 또는 IP 주소입니다.

host_port

복제 마스터로 구성될 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스에서 사용하는 포트입니다. 네트워크 구성에 포트 번호를 변환하는 SSH 포트 복제가 포함되는 경우 SSH(Secure Shell)에 의해 공개되는 포트 이름을 지정하십시오.

replication_user_name

Amazon RDS 외부에서 실행하는 MySQL 인스턴스에서 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 보유한 사용자의 ID입니다. 외부 인스턴스를 사용한 복제에만 사용되는 계정을 제공하는 것이 좋습니다.

replication_user_password

replication_user_name에 지정된 사용자 ID의 암호입니다.

mysql_binary_log_file_name

복제 정보가 들어 있는 복제 마스터에 있는 이진수 로그의 이름입니다.

mysql_binary_log_file_location

복제 시 복제 정보를 읽기 시작하는 mysql_binary_log_file_name 이진수 로그 내 위치입니다.

ssl_encryption

복제 연결에 SSL(Secure Socket Layer) 암호화를 사용할지 여부를 지정하는 값입니다. 1은 SSL 암호화 사용, 0은 암호화 사용 안 함입니다. 기본값은 0입니다.

Note

MySQL과 호환되는 Amazon Aurora에서만 사용할 수 있는 파라미터입니다. MySQL DB 인스턴스에서는 기본값만 사용할 수 있습니다.

사용 시 주의사항

마스터 사용자는 mysql.rds_set_external_master 절차를 실행해야 합니다. 이 절차는 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스의 읽기 전용 복제본으로 구성될 MySQL DB 인스턴스에서 실행해야 합니다.

mysql.rds_set_external_master를 실행하기 전에 먼저 Amazon RDS 외부에서 실행하는 MySQL 인스턴스를 복제 마스터로 구성해야 합니다. Amazon RDS 외부에서 실행하는 MySQL 인스턴스에 연결하려면 MySQL의 외부 인스턴스에서 REPLICATION CLIENT 및 REPLICATION SLAVE 권한이 있는 복제 사용자를 나타내는 replication_user_name 및 replication_user_password 값을 지정해야 합니다.

MySQL의 외부 인스턴스를 복제 마스터로 구성하려면

- 선택한 MySQL 클라이언트를 사용하여 MySQL의 외부 인스턴스에 연결하고 복제에 사용될 사용자 계정을 생성합니다. 다음은 예제입니다.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password'
```

- MySQL의 외부 인스턴스에서 복제 사용자에게 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여합니다. 다음 예제에서는 도메인의 'repl_user' 사용자에게 모든 데이터베이스에 대한 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여합니다.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'  
IDENTIFIED BY 'password'
```

암호화된 복제를 사용하려면, 마스터가 SSL 연결을 사용하도록 구성해야 합니다. 또한 [mysql.rds_import_binlog_ssl_material \(p. 685\)](#) 프로시저를 사용하여 인증 기관(CA) 인증서, 클라이언트 인증서, 클라이언트 키를 DB 인스턴스나 DB 클러스터로 가져옵니다.

Note

가능하면 읽기 전용 복제본을 사용하여 Amazon RDS DB 인스턴스 간 복제를 관리하는 것이 좋습니다. 그렇게 하면 이 복제 및 다른 복제 관련 저장 절차만 사용하는 것이 좋습니다. 이러한 방법을 사용하면 Amazon RDS DB 인스턴스 간 더욱 복잡한 복제 토플로지를 사용할 수 있습니다. AWS는 주로 Amazon RDS 외부에서 실행하는 MySQL 인스턴스를 사용하여 복제가 가능한 이러한 저장 절차를 제공합니다. Amazon RDS DB 인스턴스 간 복제 관리에 대한 자세한 내용은 [읽기 전용 복제본 작업 \(p. 135\)](#) 주제를 참조하십시오.

`mysql.rds_set_external_master`를 호출하여 Amazon RDS DB 인스턴스를 읽기 전용 복제본으로 구성한 후 읽기 전용 복제본에서 [mysql.rds_start_replication \(p. 688\)](#)을 호출하여 복제 프로세스를 시작할 수 있습니다. [mysql.rds_reset_external_master \(p. 685\)](#)를 호출하여 읽기 전용 복제본 구성을 제거할 수 있습니다.

`mysql.rds_set_external_master`가 호출되면 Amazon RDS에서 `mysql.rds_history` 및 `mysql.rds_replication_status` 테이블에 시간, 사용자 및 "set master" 작업을 기록합니다.

예제

MySQL DB 인스턴스에서 다음 예제를 실행하면 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스의 읽기 전용 복제본이 되도록 DB 인스턴스가 구성됩니다.

```
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

Amazon RDS 외부에서 실행 중인 MySQL 인스턴스의 읽기 전용 복제본이 되도록 Amazon RDS MySQL DB 인스턴스를 구성하고 지연 복제를 구성합니다.

구문

```
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  
);
```

파라미터

host_name

복제 마스터가 될 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스의 호스트 이름 또는 IP 주소입니다.

host_port

복제 마스터로 구성될 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스에서 사용하는 포트입니다. 네트워크 구성에 포트 번호를 변환하는 SSH 포트 복제가 포함되는 경우 SSH에 의해 공개되는 포트 이름을 지정하십시오.

replication_user_name

Amazon RDS 외부에서 실행하는 MySQL 인스턴스에서 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 보유한 사용자의 ID입니다. 외부 인스턴스를 사용한 복제에만 사용되는 계정을 제공하는 것이 좋습니다.

replication_user_password

replication_user_name에 지정된 사용자 ID의 암호입니다.

mysql_binary_log_file_name

복제 정보를 포함하는 복제 마스터에 있는 이진 로그의 이름입니다.

mysql_binary_log_file_location

복제 시 복제 정보를 읽기 시작할 mysql_binary_log_file_name 이진 로그 내 위치입니다.

ssl_encryption

이 옵션은 현재 구현되지 않습니다. 기본값은 0입니다.

delay

마스터에서 복제를 지연할 최소 시간(초)입니다.

이 파라미터에 대한 제한은 1일(86,400초)입니다.

사용 시 주의사항

마스터 사용자는 mysql.rds_set_external_master_with_delay 절차를 실행해야 합니다. 이 절차는 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스의 읽기 전용 복제본으로 구성될 MySQL DB 인스턴스에서 실행해야 합니다.

mysql.rds_set_external_master_with_delay를 실행하기 전에 먼저 Amazon RDS 외부에서 실행하는 MySQL 인스턴스를 복제 마스터로 구성해야 합니다. Amazon RDS 외부에서 실행하는 MySQL 인스턴스에 연결하려면 replication_user_name 및 replication_user_password의 값을 지정해야 합니다. 이러한 값은 MySQL의 외부 인스턴스에 대해 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 가진 복제 사용자를 나타내야 합니다.

MySQL의 외부 인스턴스를 복제 마스터로 구성하려면

- 선택한 MySQL 클라이언트를 사용하여 MySQL의 외부 인스턴스에 연결하고 복제에 사용될 사용자 계정을 생성합니다. 다음은 예제입니다.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
```

- MySQL의 외부 인스턴스에서 복제 사용자에게 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여합니다. 다음 예제에서는 도메인의 'repl_user' 사용자에게 모든 데이터베이스에 대한 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여합니다.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'  
IDENTIFIED BY 'SomePassWord'
```

자세한 내용은 [Amazon RDS 외부에서 실행 중인 MySQL 또는 MariaDB 인스턴스를 사용한 복제 \(p. 655\)](#) 단원을 참조하십시오.

Note

가능하면 읽기 전용 복제본을 사용하여 Amazon RDS DB 인스턴스 간 복제를 관리하는 것이 좋습니다. 그렇게 하면 이 복제 및 다른 복제 관련 저장 절차만 사용하는 것이 좋습니다. 이러한 방법을 사용하면 Amazon RDS DB 인스턴스 간 더욱 복잡한 복제 토플로지를 사용할 수 있습니다. AWS는 주로 Amazon RDS 외부에서 실행하는 MySQL 인스턴스를 사용하여 복제가 가능한 이러한 저장 절차를 제공합니다. Amazon RDS DB 인스턴스 간 복제 관리에 대한 자세한 내용은 [읽기 전용 복제본 작업 \(p. 135\)](#) 주제를 참조하십시오.

`mysql.rds_set_external_master_with_delay`를 호출하여 Amazon RDS DB 인스턴스를 읽기 전용 복제본으로 구성한 후 읽기 전용 복제본에서 [mysql.rds_start_replication \(p. 688\)](#)을 호출하여 복제 프로세스를 시작할 수 있습니다. [mysql.rds_reset_external_master \(p. 685\)](#)를 호출하여 읽기 전용 복제본 구성을 제거할 수 있습니다.

`mysql.rds_set_external_master_with_delay`를 호출하면 Amazon RDS는 "set master"의 시간, 사용자 및 작업을 `mysql.rds_history` 및 `mysql.rds_replication_status` 테이블에 기록합니다.

재해 복구의 경우 이 절차를 [mysql.rds_start_replication_until \(p. 689\)](#) 또는 [mysql.rds_start_replication_until_gtid \(p. 690\)](#) 저장 절차와 함께 사용할 수 있습니다. 절차를 실행하여 지연된 읽기 전용 복제본에 대한 변경 사항을 재해 직전 시간으로 끌어오려면 `mysql.rds_set_external_master_with_delay` 절차를 실행할 수 있습니다. `mysql.rds_start_replication_until` 프로시저에서 복제를 중지한 이후에 [읽기 전용 복제본을 드립 DB 인스턴스로 승격 \(p. 139\)](#)의 지침에 따라 읽기 전용 복제본을 새 마스터 DB 인스턴스로 승격할 수 있습니다.

`mysql.rds_start_replication_until_gtid` 절차를 사용하려면 GTID를 기반으로 한 복제를 활성화해야 합니다. 재해 원인으로 알려진 특정 GTID 기반 트랜잭션을 건너뛰려면 `mysql.rds_skip_transaction_with_gtid (p. 691)` 저장 절차를 사용할 수 있습니다. GTID 기반 복제 작업에 대한 자세한 내용은 [Amazon RDS MySQL용 GTID 기반 복제 사용 \(p. 650\)](#) 단원을 참조하십시오.

`mysql.rds_set_external_master_with_delay` 프로시저는 다음 버전의 Amazon RDS MySQL에서 사용할 수 있습니다.

- MySQL 5.6.40 및 5.6 버전 이상
- MySQL 5.7.22 및 5.7 버전 이상

예제

MySQL DB 인스턴스에서 다음 예제를 실행하면 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스의 읽기 전용 복제본이 되도록 DB 인스턴스가 구성됩니다. MySQL DB 인스턴스에서 최소 복제 지연을 1시간

(3,600초)으로 설정합니다. Amazon RDS 외부에서 실행 중인 MySQL 마스터의 변경 사항은 최소 1시간 동안 MySQL DB 인스턴스 읽기 전용 복제본에 적용되지 않습니다.

```
call mysql.rds_set_external_master_with_delay(
    'Externaldb.some.com',
    3306,
    'repl_user',
    'SomePassW0rd',
    'mysql-bin-changelog.000777',
    120,
    0,
    3600);
```

mysql.rds_set_external_master_with_auto_position

Amazon RDS MySQL DB 인스턴스를 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스의 읽기 전용 복제본으로 구성합니다. 또한 이 절차는 전역 트랜잭션 식별자(GTID)를 기반으로 한 지연된 복제 및 복제를 구성합니다.

구문

```
CALL mysql.rds_set_external_master_with_auto_position (
    host_name
    , host_port
    , replication_user_name
    , replication_user_password
    , ssl_encryption
    , delay
);
```

파라미터

host_name

복제 마스터가 될 Amazon RDS의 외부에서 실행 중인 MySQL 인스턴스의 호스트 이름 또는 IP 주소입니다.

host_port

복제 마스터로 구성될 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스에서 사용하는 포트입니다. 네트워크 구성에 포트 번호를 변환하는 SSH 포트 복제가 포함되는 경우 SSH(Secure Shell)에 의해 공개되는 포트 이름을 지정하십시오.

replication_user_name

Amazon RDS 외부에서 실행하는 MySQL 인스턴스에서 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 보유한 사용자의 ID입니다. 외부 인스턴스를 사용한 복제에만 사용되는 계정을 제공하는 것이 좋습니다.

replication_user_password

replication_user_name에 지정된 사용자 ID의 암호입니다.

ssl_encryption

이 옵션은 현재 구현되지 않습니다. 기본값은 0입니다.

delay

마스터에서 복제를 지연할 최소 시간(초)입니다.

이 파라미터에 대한 제한은 1일(86,400초)입니다.

사용 시 주의사항

마스터 사용자는 `mysql.rds_set_external_master_with_auto_position` 절차를 실행해야 합니다. 이 절차는 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스의 읽기 전용 복제본으로 구성될 MySQL DB 인스턴스에서 실행해야 합니다.

Amazon RDS MySQL 5.7의 경우 이 절차는 MySQL 5.7.23 및 MySQL 5.7 이상 버전에서 지원합니다. Amazon RDS MySQL 5.5, 5.6 또는 8.0의 경우 이 절차는 지원하지 않습니다.

`mysql.rds_set_external_master_with_auto_position`을 실행하기 전에 먼저 Amazon RDS 외부에서 실행하는 MySQL 인스턴스를 복제 마스터로 구성해야 합니다. Amazon RDS 외부에서 실행하는 MySQL 인스턴스에 연결하려면 `replication_user_name` 및 `replication_user_password`의 값을 지정해야 합니다. 이러한 값은 MySQL의 외부 인스턴스에 대해 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 가진 복제 사용자를 나타내야 합니다.

MySQL의 외부 인스턴스를 복제 마스터로 구성하려면

- 선택한 MySQL 클라이언트를 사용하여 MySQL의 외부 인스턴스에 연결하고 복제에 사용될 사용자 계정을 생성합니다. 다음은 예제입니다.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassWOrd'
```

- MySQL의 외부 인스턴스에서 복제 사용자에게 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여합니다. 다음 예제에서는 도메인의 'repl_user' 사용자에게 모든 데이터베이스에 대한 REPLICATION CLIENT 및 REPLICATION SLAVE 권한을 부여합니다.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'  
IDENTIFIED BY 'SomePassWOrd'
```

자세한 내용은 [Amazon RDS 외부에서 실행 중인 MySQL 또는 MariaDB 인스턴스를 사용한 복제 \(p. 655\)](#) 단원을 참조하십시오.

Note

가능하면 읽기 전용 복제본을 사용하여 Amazon RDS DB 인스턴스 간 복제를 관리하는 것이 좋습니다. 그렇게 하면 이 복제 및 다른 복제 관련 저장 절차만 사용하는 것이 좋습니다. 이러한 방법을 사용하면 Amazon RDS DB 인스턴스 간 더욱 복잡한 복제 토플로지를 사용할 수 있습니다. AWS는 주로 Amazon RDS 외부에서 실행하는 MySQL 인스턴스를 사용하여 복제가 가능한 이러한 저장 절차를 제공합니다. Amazon RDS DB 인스턴스 간 복제 관리에 대한 자세한 내용은 [읽기 전용 복제본 작업 \(p. 135\)](#) 주제를 참조하십시오.

`mysql.rds_set_external_master_with_auto_position`을 호출하여 Amazon RDS DB 인스턴스를 읽기 전용 복제본으로 구성한 후 읽기 전용 복제본에서 [mysql.rds_start_replication \(p. 688\)](#)을 호출하여 복제 프로세스를 시작할 수 있습니다. [mysql.rds_reset_external_master \(p. 685\)](#)를 호출하여 읽기 전용 복제본 구성을 제거할 수 있습니다.

`mysql.rds_set_external_master_with_auto_position`을 호출하면 Amazon RDS는 "set master"의 시간, 사용자 및 작업을 `mysql.rds_history` 및 `mysql.rds_replication_status` 테이블에 기록합니다.

재해 복구의 경우 이 절차를 [mysql.rds_start_replication_until \(p. 689\)](#) 또는 [mysql.rds_start_replication_until_gtid \(p. 690\)](#) 저장 절차와 함께 사용할 수 있습니다. 절차를 실행하여 지연된 읽기 전용 복제본에 대한 변경 사항을 재해 직전 시간으로 둘 포워드하려면 `mysql.rds_set_external_master_with_auto_position` 절차를 실행할 수 있습니다.

mysql.rds_start_replication_until_gtid 프로시저에서 복제를 중지한 이후에 [읽기 전용 복제본을 득립 DB 인스턴스로 승격 \(p. 139\)](#)의 지침에 따라 읽기 전용 복제본을 새 마스터 DB 인스턴스로 승격할 수 있습니다.

mysql.rds_start_replication_until_gtid 절차를 사용하려면 GTID를 기반으로 한 복제를 활성화해야 합니다. 재해 원인으로 알려진 특정 GTID 기반 트랜잭션을 건너뛰려면 [mysql.rds_skip_transaction_with_gtid \(p. 691\)](#) 저장 절차를 사용할 수 있습니다. GTID 기반 복제 작업에 대한 자세한 내용은 [Amazon RDS MySQL용 GTID 기반 복제 사용 \(p. 650\)](#) 단원을 참조하십시오.

예제

MySQL DB 인스턴스에서 다음 예제를 실행하면 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스의 읽기 전용 복제본이 되도록 DB 인스턴스가 구성됩니다. MySQL DB 인스턴스에서 최소 복제 지연을 1시간 (3,600초)으로 설정합니다. Amazon RDS 외부에서 실행 중인 MySQL 마스터의 변경 사항은 최소 1시간 동안 MySQL DB 인스턴스 읽기 전용 복제본에 적용되지 않습니다.

```
call mysql.rds_set_external_master_with_auto_position(
    'Externaldb.some.com',
    3306,
    'repl_user',
    'SomePassW0rd',
    0,
    3600);
```

mysql.rds_reset_external_master

이제 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스의 읽기 전용 복제본이 되지 않도록 MySQL DB 인스턴스를 다시 구성합니다.

구문

```
CALL mysql.rds_reset_external_master;
```

사용 시 주의사항

마스터 사용자는 mysql.rds_reset_external_master 절차를 실행해야 합니다. 이 절차는 Amazon RDS 외부에서 실행하는 MySQL 인스턴스의 읽기 전용 복제본으로 제거될 MySQL DB 인스턴스에서 실행해야 합니다.

Note

가능하면 읽기 전용 복제본을 사용하여 Amazon RDS DB 인스턴스 간 복제를 관리하는 것이 좋습니다. 그렇게 하면 이 복제 및 다른 복제 관련 저장 절차만 사용하는 것이 좋습니다. 이러한 방법을 사용하면 Amazon RDS DB 인스턴스 간 더욱 복잡한 복제 토플로지를 사용할 수 있습니다. AWS는 주로 Amazon RDS 외부에서 실행하는 MySQL 인스턴스를 사용하여 복제가 가능한 이러한 저장 절차를 제공합니다. Amazon RDS DB 인스턴스 간 복제 관리에 대한 자세한 내용은 [읽기 전용 복제본 작업 \(p. 135\)](#) 주제를 참조하십시오.

복제를 사용하여 Amazon RDS 외부에서 실행 중인 MySQL 인스턴스에서 데이터 가져오기에 대한 자세한 내용은 [Amazon RDS MySQL DB 인스턴스로 백업 복원 \(p. 619\)](#) 주제를 참조하십시오.

mysql.rds_import_binlog_ssl_material

인증 기관(CA) 인증서, 클라이언트 인증서, 클라이언트 키를 Aurora MySQL DB 클러스터로 가져옵니다. SSL 통신과 암호화된 복제에 필요한 정보입니다.

Note

현재 이 절차는 Aurora MySQL 버전 5.6에서만 지원합니다.

구문

```
CALL mysql.rds_import_binlog_ssl_material (
    ssl_material
);
```

파라미터

ssl_material

MySQL 클라이언트를 위한 다음 .pem 형식 파일 내용이 포함된 JSON 페이로드.

- "ssl_ca": "*Certificate authority certificate*"
- "ssl_cert": "*Client certificate*"
- "ssl_key": "*Client key*"

사용 시 주의사항

이 프로시저를 실행하기 전에 암호화된 복제를 준비합니다.

- 외부 MySQL 마스터 데이터베이스에서 SSL을 활성화 시키지 않았고 클라이언트 키와 인증서가 준비되지 않았다면, MySQL 데이터베이스 서버의 SSL을 활성화하고, 필요한 클라이언트 키와 인증서를 생성합니다.
- 외부 마스터에 SSL이 활성화되어 있다면, Aurora MySQL DB 클러스터에 클라이언트 키와 인증서를 제공합니다. 인증서와 키가 없다면, Aurora MySQL DB 클러스터에 대해 새 키와 인증서를 생성합니다. 클라이언트 인증서에 서명하려면, 외부 MySQL 마스터 데이터베이스의 SSL을 구성할 때 사용하는 인증 기관(CA) 키가 있어야 합니다.

자세한 내용은 MySQL 문서의 [Creating SSL Certificates and Keys Using openssl](#)을 참조하십시오.

Important

암호화된 복제를 준비한 후, SSL 연결을 사용해 이 프로시저를 실행합니다. 클라이언트 키를 보안성이 낮은 연결을 이용해 전송할 수 없습니다.

이 프로시저는 외부 MySQL 데이터베이스의 SSL 정보를 Aurora MySQL DB 클러스터로 가져옵니다. SSL 정보는 DB 클러스터에 대한 SSL 정보가 포함되어 있는 .pem 형식의 파일입니다. 암호화 복제 중, DB 클러스터는 MySQL 데이터베이스 서버의 클라이언트 역할을 합니다. Aurora MySQL 클라이언트 인증서와 키는 .pem 형식의 파일입니다.

이들 파일에서 올바른 JSON 페이로드의 `ssl_material` 파라미터로 정보를 복사할 수 있습니다. 암호화된 복제를 지원하려면 DB 클러스터로 이 SSL 정보를 가져옵니다.

JSON 페이로드는 다음 형식이어야 합니다.

```
'{"ssl_ca":"-----BEGIN CERTIFICATE-----\nssl_ca_pem_body_code\n-----END CERTIFICATE-----\n","ssl_cert":"-----BEGIN CERTIFICATE-----\nssl_cert_pem_body_code\n-----END CERTIFICATE-----\n","ssl_key":"-----BEGIN RSA PRIVATE KEY-----\n'
```

```
ssl_key_pem_body_code
-----END RSA PRIVATE KEY-----\n"}'
```

예제

다음은 Aurora MySQL DB 클러스터에 SSL 정보를 가져오는 예제입니다. .pem 형식 파일은 본문 코드의 길이가 일반적으로 예제의 본문 코드 길이보다 깁니다.

```
call mysql.rds_import_binlog_ssl_material(
'{"ssl_ca":"-----BEGIN CERTIFICATE-----
AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jeZoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnB1tntckij7FbtJMxLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUzofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwi3f05p6KLxEXAMPLE
-----END CERTIFICATE-----\n","ssl_cert":"-----BEGIN CERTIFICATE-----
AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jeZoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnB1tntckij7FbtJMxLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUzofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwi3f05p6KLxEXAMPLE
-----END CERTIFICATE-----\n","ssl_key":"-----BEGIN RSA PRIVATE KEY-----
AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jeZoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnB1tntckij7FbtJMxLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUzofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwi3f05p6KLxEXAMPLE
-----END RSA PRIVATE KEY-----\n"}');
```

Note

Amazon Aurora 사용에 대한 자세한 내용은 [Amazon Aurora 사용 설명서](#)를 참조하십시오.

mysql.rds_remove_binlog_ssl_material

SSL 통신과 암호화된 복제용 인증 기관(CA) 인증서, 클라이언트 인증서, 클라이언트 키를 제거합니다.
[mysql.rds_import_binlog_ssl_material \(p. 685\)](#)를 사용하여, 이 정보를 가져옵니다.

Note

현재 이 절차는 Aurora MySQL 버전 5.6에서만 지원합니다.

구문

```
CALL mysql.rds_remove_binlog_ssl_material;
```

mysql.rds_set_source_delay

마스터에서 현재 읽기 전용 복제본으로의 복제를 지연할 최소 시간(초)을 설정합니다. 읽기 전용 복제본에 연결된 경우에 이 절차를 사용하여 마스터의 복제를 지연합니다.

구문

```
CALL mysql.rds_set_source_delay(
delay
```

) ;

파라미터

delay

마스터에서 복제를 지연할 최소 시간(초)입니다.

이 파라미터에 대한 제한은 1일(86,400초)입니다.

사용 시 주의사항

마스터 사용자는 `mysql.rds_set_source_delay` 절차를 실행해야 합니다.

재해 복구의 경우 이 절차를 [mysql.rds_start_replication_until \(p. 689\)](#) 저장 절차 또는 [mysql.rds_start_replication_until_gtid \(p. 690\)](#) 저장 절차와 함께 사용할 수 있습니다. 절차를 실행하여 지연된 읽기 전용 복제본에 대한 변경 사항을 재해 직전 시간으로 롤 포워드하려면 `mysql.rds_set_source_delay` 절차를 실행할 수 있습니다. `mysql.rds_start_replication_until` 또는 `mysql.rds_start_replication_until_gtid` 절차에서 복제를 중지한 경우 [읽기 전용 복제본을 드립 DB 인스턴스로 승격 \(p. 139\)](#)의 지침에 따라 읽기 전용 복제본을 새 마스터 DB 인스턴스로 승격할 수 있습니다.

`mysql.rds_start_replication_until_gtid` 절차를 사용하려면 GTID를 기반으로 한 복제를 활성화해야 합니다. 재해 원인으로 알려진 특정 GTID 기반 트랜잭션을 건너뛰려면 `mysql.rds_skip_transaction_with_gtid (p. 691)` 저장 절차를 사용할 수 있습니다. GTID 기반 복제에 대한 자세한 내용은 [Amazon RDS MySQL용 GTID 기반 복제 사용 \(p. 650\)](#) 단원을 참조하십시오.

`mysql.rds_set_source_delay` 프로시저는 다음 버전의 Amazon RDS MySQL에서 사용할 수 있습니다.

- MySQL 5.6.40 및 5.6 버전 이상
- MySQL 5.7.22 및 5.7 버전 이상

예제

마스터에서 현재 읽기 전용 복제본으로의 복제를 최소 1시간(3,600초) 동안 지연하려면 다음 파라미터를 사용하여 `mysql.rds_set_source_delay`를 호출할 수 있습니다.

```
CALL mysql.rds_set_source_delay(3600);
```

mysql.rds_start_replication

MySQL DB 인스턴스에서 복제를 시작합니다.

Note

[mysql.rds_start_replication_until \(p. 689\)](#) 또는 [mysql.rds_start_replication_until_gtid \(p. 690\)](#) 저장 절차를 사용하여 Amazon RDS MySQL DB 인스턴스에서 복제를 시작하고 지정된 바이너리 로그 파일 위치에서 복제를 중지할 수 있습니다.

구문

```
CALL mysql.rds_start_replication;
```

사용 시 주의사항

마스터 사용자는 `mysql.rds_start_replication` 절차를 실행해야 합니다.

Amazon RDS 외부에서 실행 중인 MySQL 인스턴스에서 데이터를 가져오도록 복제를 구성할 경우 읽기 전용 복제본에서 `mysql.rds_start_replication`을 호출하여 복제 프로세스를 시작하기 전에 `mysql.rds_set_external_master` (p. 678)를 호출하여 복제 구성을 빌드합니다. 자세한 내용은 [Amazon RDS MySQL DB 인스턴스로 백업 복원](#) (p. 619) 단원을 참조하십시오.

Amazon RDS 외부의 MySQL 인스턴스로 데이터를 내보내도록 복제를 구성할 경우 읽기 전용 복제본에서 `mysql.rds_start_replication` 및 `mysql.rds_stop_replication`을 호출하여 이진 로그 삭제 등 의 몇 가지 복제 작업을 제어합니다. 자세한 내용은 [복제를 사용하여 MySQL DB 인스턴스에서 데이터 내보내기](#) (p. 660) 단원을 참조하십시오.

읽기 전용 복제본에서 `mysql.rds_start_replication`을 호출하여 이전에 `mysql.rds_stop_replication` (p. 691)을 호출하여 중지한 복제 프로세스를 다시 시작할 수도 있습니다. 자세한 내용은 [읽기 전용 복제본 작업](#) (p. 135) 단원을 참조하십시오.

mysql.rds_start_replication_until

Amazon RDS MySQL DB 인스턴스에서 복제를 시작하고 지정된 바이너리 로그 파일 위치에서 복제를 중지합니다.

구문

```
CALL mysql.rds_start_replication_until (
    replication_log_file
    , replication_stop_point
);
```

파라미터

`replication_log_file`

복제 정보를 포함하는 복제 마스터에 있는 이진 로그의 이름입니다.

`replication_stop_point`

복제를 중지할 `replication_log_file` 바이너리 로그 내 위치입니다.

사용 시 주의사항

마스터 사용자는 `mysql.rds_start_replication_until` 절차를 실행해야 합니다.

이 절차는 지연 복제에서 재해 복구를 위해 사용할 있습니다. 지연 복제를 구성한 경우 이 절차를 사용하여 지연된 읽기 전용 복제본에 대한 변경 사항을 재해 직전 시간으로 롤포워드 할 수 있습니다. 이 프로시저에서 복제를 중지한 이후에 [읽기 전용 복제본을 드립 DB 인스턴스로 승격](#) (p. 139)의 지침에 따라 읽기 전용 복제본을 새 마스터 DB 인스턴스로 승격할 수 있습니다.

다음 저장 프로시저를 사용하여 지연 복제를 구성할 수 있습니다.

- [mysql.rds_set_configuration](#) (p. 695)
- [mysql.rds_set_external_master_with_delay](#) (p. 680)
- [mysql.rds_set_source_delay](#) (p. 687)

`replication_log_file` 파라미터에 대해 지정된 파일 이름이 마스터 binlog 파일 이름과 일치해야 합니다.

`replication_stop_point` 파라미터가 과거에 해당하는 중지 위치를 지정하는 경우 복제가 즉시 중지됩니다.

`mysql.rds_start_replication_until` 프로시저는 다음 버전의 Amazon RDS MySQL에서 사용할 수 있습니다.

- MySQL 5.6.40 및 5.6 버전 이상
- MySQL 5.7.22 및 5.7 버전 이상

예제

다음 예제에서는 복제를 시작하고 120 바이너리 로그 파일의 `mysql-bin-changelog.000777` 위치에 도달할 때까지 변경 사항을 복제합니다.

```
call mysql.rds_start_replication_until(
    'mysql-bin-changelog.000777',
    120);
```

mysql.rds_start_replication_until_gtid

Amazon RDS MySQL DB 인스턴스에서 복제를 시작하고 지정된 전역 트랜잭션 식별자(GTID) 바로 다음에 복제를 중지합니다.

구문

```
CALL mysql.rds_start_replication_until_gtid (
gtid
);
```

파라미터

gtid

GTID 이후 복제를 중지해야 합니다.

사용 시 주의사항

마스터 사용자는 `mysql.rds_start_replication_until_gtid` 절차를 실행해야 합니다.

Amazon RDS MySQL 5.7의 경우 이 절차는 MySQL 5.7.23 및 MySQL 5.7 이상 버전에서 지원합니다. Amazon RDS MySQL 5.5, 5.6 또는 8.0의 경우 이 절차는 지원하지 않습니다.

이 절차는 지역 복제에서 재해 복구를 위해 사용할 있습니다. 지역 복제를 구성한 경우 이 절차를 사용하여 지역된 읽기 전용 복제본에 대한 변경 사항을 재해 직전 시간으로 롤포워드 할 수 있습니다. 이 프로시저에서 복제를 중지한 이후에 [읽기 전용 복제본을 드립 DB 인스턴스로 승격 \(p. 139\)](#)의 지침에 따라 읽기 전용 복제본을 새 마스터 DB 인스턴스로 승격할 수 있습니다.

다음 저장 프로시저를 사용하여 지역 복제를 구성할 수 있습니다.

- [mysql.rds_set_configuration \(p. 695\)](#)

- [mysql.rds_set_external_master_with_auto_position \(p. 683\)](#)
- [mysql.rds_set_source_delay \(p. 687\)](#)

gtid 파라미터가 복제본으로 이미 실행한 트랜잭션을 지정하는 경우 복제가 즉시 중지됩니다.

예제

다음 예제에서는 복제를 시작하고 GTID 3E11FA47-71CA-11E1-9E33-C80AA9429562:23에 도달할 때 까지 변경 사항을 복제합니다.

```
call mysql.rds_start_replication_until_gtid(
  '3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

mysql.rds_stop_replication

MySQL DB 인스턴스에서 복제를 종료합니다.

구문

```
CALL mysql.rds_stop_replication;
```

사용 시 주의사항

마스터 사용자는 `mysql.rds_stop_replication` 절차를 실행해야 합니다.

Amazon RDS 외부에서 실행 중인 MySQL 인스턴스에서 데이터를 가져오도록 복제를 구성하는 경우 가져오기가 완료된 후 읽기 전용 복제본에서 `mysql.rds_stop_replication`을 호출하여 복제 프로세스를 중지합니다. 자세한 내용은 [Amazon RDS MySQL DB 인스턴스로 백업 복원 \(p. 619\)](#) 단원을 참조하십시오.

Amazon RDS 외부의 MySQL 인스턴스로 데이터를 내보내도록 복제를 구성할 경우 읽기 전용 복제본에서 `mysql.rds_start_replication` 및 `mysql.rds_stop_replication`을 호출하여 이진 로그 삭제 등 의 몇 가지 복제 작업을 제어합니다. 자세한 내용은 [복제를 사용하여 MySQL DB 인스턴스에서 데이터 내보내기 \(p. 660\)](#) 단원을 참조하십시오.

`mysql.rds_stop_replication`을 사용하여 두 Amazon RDS DB 인스턴스 간 복제를 중지할 수도 있습니다. 대개 읽기 전용 복제본에 대해 대규모 인덱스 생성과 같은 장기 실행 작업을 수행하려는 경우 복제를 중지합니다. 읽기 전용 복제본에서 `mysql.rds_start_replication (p. 688)`을 호출하여 중지한 복제 프로세스를 다시 시작할 수 있습니다. 자세한 내용은 [읽기 전용 복제본 작업 \(p. 135\)](#) 단원을 참조하십시오.

mysql.rds_skip_transaction_with_gtid

MySQL DB 인스턴스에서 지정된 전역 트랜잭션 식별자(GTID)를 사용하여 트랜잭션 복제를 건너뜁니다.

특정 GTID 트랜잭션이 문제의 원인으로 알려진 경우 재해 복구를 위해 이 절차를 사용할 수 있습니다. 이 저장 절차를 사용하여 문제의 트랜잭션을 건너뛰십시오. 문제의 트랜잭션의 예로는 복제를 비활성화하거나 중요한 데이터를 삭제하거나 DB 인스턴스를 사용할 수 없도록 하는 트랜잭션이 포함됩니다.

구문

```
CALL mysql.rds_skip_transaction_with_gtid (
  gtid_to_skip
```

):

파라미터

gtid_to_skip

건너뛸 복제 트랜잭션의 GTID입니다.

사용 시 주의사항

마스터 사용자는 `mysql.rds_skip_transaction_with_gtid` 절차를 실행해야 합니다.

Amazon RDS MySQL 5.7의 경우 이 절차는 MySQL 5.7.23 및 MySQL 5.7 이상 버전에서 지원합니다.
Amazon RDS MySQL 5.5, 5.6 또는 8.0의 경우 이 절차는 지원하지 않습니다.

mysql.rds_skip_repl_error

MySQL DB 인스턴스의 복제 오류를 건너뛰고 삭제합니다.

구문

```
CALL mysql.rds_skip_repl_error;
```

사용 시 주의사항

마스터 사용자는 `mysql.rds_skip_repl_error` 절차를 실행해야 합니다.

오류가 있는지 여부를 판별하려면 MySQL `show slave status\G` 명령을 실행하십시오. 중대한 복제 오류가 아닌 경우 `mysql.rds_skip_repl_error`를 사용하여 오류를 건너뛸 수 있습니다. 오류가 여러 개인 경우 `mysql.rds_skip_repl_error`는 첫 번째 오류를 삭제한 후 다른 오류가 있음을 경고합니다. `show slave status\G`를 사용하여 다음 오류에 대한 적합한 조치를 결정할 수 있습니다. 반환된 값에 대한 자세한 내용은 MySQL 설명서의 [SHOW SLAVE STATUS Syntax](#)를 참조하십시오.

Amazon RDS의 복제 오류 해결에 대한 자세한 내용은 [MySQL 읽기 전용 복제본의 문제 해결 \(p. 649\)](#) 주제를 참조하십시오.

Important

`mysql.rds_skip_repl_error`를 호출하려는 경우 `ERROR 1305 (42000): PROCEDURE mysql.rds_skip_repl_error does not exist` 오류가 발생할 수 있습니다. 그러한 경우 MySQL DB 인스턴스를 최신 마이너 버전이나 이 주제에 나열된 최소 마이너 버전 중 하나로 업그레이드하십시오.

Slave Down 또는 Disabled 오류

`mysql.rds_skip_repl_error` 명령을 호출하면 다음 오류 메시지가 표시될 수 있습니다. `Slave is down or disabled.`

이 오류 메시지는 복제가 중지되었고 재시작할 수 없기 때문에 표시됩니다.

많은 수의 오류를 건너뛰어야 하는 경우, 복제 지연이 바이너리 로그(binlog) 파일의 기본값 보관 기간 이상으로 늘어날 수 있습니다. 이 경우, binlog 파일이 읽기 전용 복제본에서 재실행되기 전에 지워지기 때문에 치명적 오류가 발생할 수 있습니다. 이 제거는 복제를 중지시키며, 복제 오류를 건너뛰기 위해 더 이상 `mysql.rds_skip_repl_error` 명령을 호출할 수 없습니다.

이 문제는 복제 마스터에서 binlog 파일이 보관되는 시간을 늘림으로써 완화할 수 있습니다. binlog 보관 시간을 늘린 후에 복제를 재시작하고 필요에 따라 mysql.rds_skip_repl_error 명령을 호출할 수 있습니다.

binlog 보관 기간을 설정하려면 [mysql.rds_set_configuration \(p. 695\)](#) 절차를 사용하여 'binlog retention hours' 구성 파라미터와 DB 클러스터에 binlog 파일을 보관할 시간을 함께 지정하십시오. 다음 예제에서는 binlog 파일의 보관 기간을 48시간으로 설정합니다.

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

mysql.rds_next_master_log

마스터에 있는 다음 이진 로그의 시작으로 복제 마스터 로그 위치를 변경합니다. 읽기 전용 복제본에 대한 복제 I/O 오류 1236을 수신 중일 경우에만 이 프로시저를 사용하십시오.

구문

```
CALL mysql.rds_next_master_log(  
curr_master_log  
);
```

파라미터

curr_master_log

현재 마스터 로그 파일의 인덱스입니다. 예를 들어, 현재 파일의 이름이 mysql-bin-changelog.012345라면 인덱스는 12345입니다. 현재 마스터 로그 파일 이름을 확인하려면 SHOW SLAVE STATUS 명령을 실행하고 Master_Log_File 필드를 봅니다.

사용 시 주의사항

마스터 사용자는 mysql.rds_next_master_log 절차를 실행해야 합니다.

Warning

복제 원본인 다른 AZ DB 인스턴스의 장애 조치 후 복제가 실패하고 SHOW SLAVE STATUS의 Last_IO_Error 필드에서 I/O 오류 1236을 보고하는 경우에만 mysql.rds_next_master_log를 호출합니다.

장애 조치 이벤트가 발생하기 전에 이진 로그에 원본 인스턴스의 트랜잭션을 기록되지 않은 경우 mysql.rds_next_master_log를 호출하면 읽기 전용 복제본에서 데이터가 손실됩니다. 원본 인스턴스 파라미터 sync_binlog = 1 및 innodb_support_xa = 1을 구성하여 데이터 손실 가능성을 줄일 수 있습니다. 그러나 성능이 저하될 수 있습니다. 자세한 내용은 [읽기 전용 복제본 작업 \(p. 135\)](#) 단원을 참조하십시오.

예제

Amazon RDS 읽기 전용 복제본에서 복제에 실패한 경우 읽기 전용 복제본에서 SHOW SLAVE STATUS\G를 실행하면 다음 결과가 반환됩니다.

```
***** 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_Errno: 1236
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_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 67285976
```

Last_IO_Errno 필드가 인스턴스에서 I/O 오류 1236을 수신하고 있음을 보여 줍니다. Master_Log_File 필드는 파일 이름이 mysql-bin-changelog.012345임을 보여 줍니다. 이는 로그 파일 인덱스가 12345임을 의미합니다. 오류를 해결하려면 다음 파라미터와 함께 mysql.rds_next_master_log를 호출합니다.

```
CALL mysql.rds_next_master_log(12345);
```

mysql.rds_innodb_buffer_pool_dump_now

버퍼 풀의 현재 상태를 디스크에 덤프합니다. 자세한 내용은 [InnoDB 캐시 워밍 \(p. 585\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_innodb_buffer_pool_dump_now();
```

사용 시 주의사항

마스터 사용자는 mysql.rds_innodb_buffer_pool_dump_now 절차를 실행해야 합니다.

mysql.rds_innodb_buffer_pool_dump_now 프로시저는 다음 버전의 Amazon RDS MySQL에서 사용할 수 있습니다.

- MySQL 5.6
- MySQL 5.7
- MySQL 8.0

mysql.rds_innodb_buffer_pool_load_now

디스크에서 버퍼 풀의 저장된 상태를 디스크에 로드합니다. 자세한 내용은 [InnoDB 캐시 워밍 \(p. 585\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_innodb_buffer_pool_load_now();
```

사용 시 주의사항

마스터 사용자는 mysql.rds_innodb_buffer_pool_load_now 절차를 실행해야 합니다.

mysql.rds_innodb_buffer_pool_load_now 프로시저는 다음 버전의 Amazon RDS MySQL에서 사용할 수 있습니다.

- MySQL 5.6
- MySQL 5.7
- MySQL 8.0

mysql.rds_innodb_buffer_pool_load_abort

진행 중인 저장된 버퍼 풀 상태 로드를 취소합니다. 자세한 내용은 [InnoDB 캐시 워밍 \(p. 585\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_innodb_buffer_pool_load_abort();
```

사용 시 주의사항

마스터 사용자는 mysql.rds_innodb_buffer_pool_load_abort 절차를 실행해야 합니다.

mysql.rds_innodb_buffer_pool_load_abort 프로시저는 다음 버전의 Amazon RDS MySQL에서 사용할 수 있습니다.

- MySQL 5.6
- MySQL 5.7
- MySQL 8.0

mysql.rds_set_configuration

바이너리 로그를 보관할 기간(시간) 또는 복제를 지연할 시간(초)을 지정합니다.

구문

```
CALL mysql.rds_set_configuration(name,value);
```

파라미터

name

설정할 구성 파라미터의 이름입니다.

USD 상당

구성 파라미터의 값입니다.

사용 시 주의사항

`mysql.rds_set_configuration` 저장 프로시저는 다음 버전의 Amazon RDS MySQL에서 사용할 수 있습니다.

- MySQL 5.6
- MySQL 5.7
- MySQL 8.0

`mysql.rds_set_configuration` 프로시저는 다음 구성 파라미터를 지원합니다.

- [binlog retention hours \(p. 696\)](#)
- [target delay \(p. 696\)](#)

binlog retention hours

`binlog retention hours` 파라미터는 바이너리 로그 파일을 보관할 기간(시간)을 지정하는 데 사용됩니다. Amazon RDS는 일반적으로 가능한 한 빨리 바이너리 로그를 삭제하지만 Amazon RDS 외부의 MySQL 데이터베이스 복제에는 바이너리 로그가 필요할 수 있습니다. `binlog retention hours`의 기본값은 NULL입니다(바이너리 로그 미보관).

Amazon RDS에서 DB 인스턴스에 대한 바이너리 로그를 보관할 기간(시간)을 지정하려면 다음 예에 나와 있는 것처럼 `mysql.rds_set_configuration` 저장 프로시저를 사용하여 복제 수행에 충분한 기간을 지정합니다.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

MySQL DB 인스턴스의 경우 최대 `binlog retention hours` 값은 168(7일)입니다.

보존 기간을 설정한 후, DB 인스턴스 스토리지의 사용량을 모니터링하여 보존된 바이너리 로그가 너무 많은 스토리지를 차지하지 않도록 합니다.

target delay

`target delay` 파라미터를 사용하여 마스터에서 읽기 전용 복제본으로의 복제를 지연할 시간(초)을 지정하십시오. 지정된 지연은 현재 DB 인스턴스에서 생성되는 새 복제본에 적용됩니다. Amazon RDS에서는 일반적으로 변경 사항을 최대한 빨리 복제하지만, 경우에 따라 복제를 지연해야 할 수 있습니다. 예를 들어, 복제가 지연될 경우 지연된 읽기 전용 복제본을 재해 직전 시간으로 롤포워드할 수 있습니다. 실수로 테이블이 삭제된 경우 지연된 복제를 사용하여 테이블을 빠르게 복구할 수 있습니다. `target delay`의 기본값은 0입니다(복제를 지연하지 않음).

재해 복구의 경우 이 구성 파라미터를 [mysql.rds_start_replication_until \(p. 689\)](#) 저장 절차 또는 [mysql.rds_start_replication_until_gtid \(p. 690\)](#) 저장 절차와 함께 사용할 수 있습니다. 지연된 읽기 전용 복제본에 대한 변경 사항을 재해 직전 시간으로 룰포워드하려면 이 파라미터를 설정한 상태에서 [mysql.rds_set_configuration](#) 절차를 실행할 수 있습니다. [mysql.rds_start_replication_until](#) 또는 [mysql.rds_start_replication_until_gtid](#) 절차에서 복제를 중지한 경우 [읽기 전용 복제본을 듀립 DB 인스턴스로 승격 \(p. 139\)](#)의 지침에 따라 읽기 전용 복제본을 새 마스터 DB 인스턴스로 승격할 수 있습니다.

[mysql.rds_start_replication_until_gtid](#) 절차를 사용하려면 GTID를 기반으로 한 복제를 활성화해야 합니다. 재해 원인으로 알려진 특정 GTID 기반 트랜잭션을 건너뛰려면 [mysql.rds_skip_transaction_with_gtid \(p. 691\)](#) 저장 절차를 사용할 수 있습니다. GTID 기반 복제 작업에 대한 자세한 내용은 [Amazon RDS MySQL용 GTID 기반 복제 사용 \(p. 650\)](#) 단원을 참조하십시오.

Amazon RDS에서 읽기 전용 복제본에 대한 복제를 지연할 시간(초)을 지정하려면 [mysql.rds_set_configuration](#) 저장 프로시저를 사용하고 복제를 지연할 시간(초)을 지정합니다. 다음 예제에서는 복제를 1시간(3,600초) 이상 지연하도록 지정합니다.

```
call mysql.rds_set_configuration('target delay', 3600);
```

target delay 파라미터에 대한 제한은 1일(86,400초)입니다.

Note

target delay 파라미터는 Amazon RDS MySQL에서만 지원됩니다.

target delay 파라미터는 Amazon RDS MySQL 버전 8.0에서 지원되지 않습니다.

mysql.rds_show_configuration

바이너리 로그가 유지되는 시간입니다.

구문

```
CALL mysql.rds_show_configuration;
```

사용 시 주의사항

Amazon RDS의 이진수 로그 보관 시간을 확인하려면 [mysql.rds_show_configuration](#) 저장 프로시저를 사용합니다.

[mysql.rds_show_configuration](#) 프로시저는 다음 버전의 Amazon RDS MySQL에서 사용할 수 있습니다.

- MySQL 5.6
- MySQL 5.7
- MySQL 8.0

예제

다음 예제는 보존 기간을 표시합니다.

```
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

MySQL Server와의 연결을 종료합니다.

구문

```
CALL mysql.rds_kill(processID);
```

파라미터

processID

종료할 연결 스레드의 ID입니다.

사용 시 주의사항

MySQL Server에 대한 각 연결은 별개의 스레드로 실행됩니다. 연결을 종료하려면 `mysql.rds_kill` 프로시저를 사용하여 해당 연결의 스레드 ID를 전달합니다. 스레드 ID를 확인하려면 MySQL [SHOW PROCESSLIST](#) 명령을 사용합니다.

예제

다음 예는 스레드 ID가 4243인 연결을 종료합니다.

```
call mysql.rds_kill(4243);
```

mysql.rds_kill_query

MySQL Server에서 실행 중인 쿼리를 종료합니다.

구문

```
CALL mysql.rds_kill_query(queryID);
```

파라미터

queryID

종료할 쿼리의 ID입니다.

사용 시 주의사항

MySQL Server에서 실행 중인 쿼리를 종료하려면 `mysql_rds_kill_query` 프로시저를 사용하여 해당 쿼리의 ID를 전달합니다. 쿼리 ID를 확인하려면 MySQL [INFORMATION_SCHEMA PROCESSLIST](#) 명령을 사용합니다. MySQL 서버에 대한 연결은 유지됩니다.

예제

다음 예는 스레드 ID가 230040인 쿼리를 종료합니다.

```
call mysql.rds_kill_query(230040);
```

mysql.rds_rotate_general_log

mysql.general_log 테이블을 백업 테이블로 로테이션합니다. 자세한 내용은 [MySQL 데이터베이스 로그 파일 \(p. 318\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_rotate_general_log;
```

사용 시 주의사항

mysql.rds_rotate_general_log 프로시저를 호출하여 mysql.general_log 테이블을 백업 테이블로 로테이션할 수 있습니다. 로그 테이블이 순환되면 현재 로그 테이블은 백업 로그 테이블에 복사되며 현재 로그 테이블의 해당 항목들은 제거됩니다. 백업 로그 테이블이 이미 존재할 경우, 현재 로그 테이블이 백업으로 복사되기 전에 백업 로그 테이블이 삭제됩니다. 필요하다면 백업 로그 테이블을 쿼리할 수 있습니다. mysql.general_log 테이블에 대한 백업 로그 테이블 이름은 mysql.general_log_backup으로 지정됩니다.

mysql.rds_rotate_slow_log

mysql.slow_log 테이블을 백업 테이블로 로테이션합니다. 자세한 내용은 [MySQL 데이터베이스 로그 파일 \(p. 318\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_rotate_slow_log;
```

사용 시 주의사항

mysql.rds_rotate_slow_log 프로시저를 호출하여 mysql.slow_log 테이블을 백업 테이블로 로테이션할 수 있습니다. 로그 테이블이 순환되면 현재 로그 테이블은 백업 로그 테이블에 복사되며 현재 로그 테이블의 해당 항목들은 제거됩니다. 백업 로그 테이블이 이미 존재할 경우, 현재 로그 테이블이 백업으로 복사되기 전에 백업 로그 테이블이 삭제됩니다.

필요하다면 백업 로그 테이블을 쿼리할 수 있습니다. mysql.slow_log 테이블에 대한 백업 로그 테이블 이름은 mysql.slow_log_backup으로 지정됩니다.

mysql.rds_enable_gsh_collector

전역적 상태 이력(GoSH)을 사용하도록 설정하여 rds_set_gsh_collector로 지정한 주기에 따라 기본 스냅샷을 생성합니다. 자세한 내용은 [전역적 상태 이력 관리 \(p. 672\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_enable_gsh_collector;
```

mysql.rds_set_gsh_collector

전역적 상태 이력(GoSH)에서 생성하는 스냅샷 주기(분)를 지정합니다. 기본값에 대한 자세한 내용은 [전역적 상태 이력 관리 \(p. 672\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_set_gsh_collector(intervalPeriod);
```

파라미터

intervalPeriod

스냅샷 주기(분)입니다. 기본 값은

mysql.rds_disable_gsh_collector

전역적 상태 이력(GoSH)의 스냅샷 생성을 비활성화합니다. 자세한 내용은 [전역적 상태 이력 관리 \(p. 672\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_disable_gsh_collector;
```

mysql.rds_collect_global_status_history

전역적 상태 이력(GoSH)에 대해 요청 시 스냅샷을 생성합니다. 자세한 내용은 [전역적 상태 이력 관리 \(p. 672\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_collect_global_status_history;
```

mysql.rds_enable_gsh_rotation

mysql.global_status_history 테이블의 내용이 rds_set_gsh_rotation에서 설정한 주기에 따라 mysql.global_status_history_old로 로테이션됩니다. 자세한 내용은 [전역적 상태 이력 관리 \(p. 672\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_enable_gsh_rotation;
```

mysql.rds_set_gsh_rotation

mysql.global_status_history 테이블의 로테이션 주기(일)을 지정합니다. 기본 값은 7입니다. 자세한 내용은 [전역적 상태 이력 관리 \(p. 672\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_set_gsh_rotation(intervalPeriod);
```

파라미터

intervalPeriod

테이블 로테이션 주기(일)입니다. 기본 값은 7입니다.

mysql.rds_disable_gsh_rotation

mysql.global_status_history 테이블의 로테이션을 비활성화합니다. 자세한 내용은 [전역적 상태 이력 관리 \(p. 672\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_disable_gsh_rotation;
```

mysql.rds_rotate_global_status_history

필요에 따라 mysql.global_status_history 테이블의 내용을 mysql.global_status_history_old로 로테이션합니다. 자세한 내용은 [전역적 상태 이력 관리 \(p. 672\)](#) 단원을 참조하십시오.

구문

```
CALL mysql.rds_rotate_global_status_history;
```

Amazon RDS의 Oracle

Amazon RDS는 Oracle Database의 여러 버전과 에디션을 실행하는 DB 인스턴스를 지원합니다. 다음 버전과 에디션을 사용할 수 있습니다.

- Oracle 12c, 버전 12.2.0.1
- Oracle 12c, 버전 12.1.0.2
- Oracle 11g, 버전 11.2.0.4

현재 Amazon RDS는 앞으로 사용 중단될 예정인 다음 버전도 지원하고 있습니다. Oracle에서는 더 이상 이를 버전에 대한 패치를 제공하지 않기 때문입니다.

- Oracle 12c, 버전 12.1.0.1([Oracle 12.1.0.1 운영 중단 \(p. 721\)](#))
- Oracle 11g, 버전 11.2.0.3([Oracle 11.2.0.3 운영 중단 \(p. 721\)](#))
- Oracle 11g, 버전 11.2.0.2([Oracle 11.2.0.2 운영 중단 \(p. 720\)](#))

DB 인스턴스 및 DB 스냅샷, 특정 시점으로 복원 및 자동 또는 수동 백업을 만들 수 있습니다. Oracle을 실행 중인 DB 인스턴스를 VPC 내에서 사용할 수 있습니다. 또한 Oracle DB 인스턴스에 다른 기능을 추가할 수 있는 다양한 옵션을 활성화할 수 있습니다. Amazon RDS는 현재 고가용성, 장애 조치 솔루션으로서 Oracle용 다중 AZ 배포를 지원합니다.

관리되는 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 셀 액세스를 제공하지 않으며, 고급 권한을 필요로 하는 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다. Amazon RDS는 Oracle SQL Plus와 같은 표준 SQL 클라이언트 애플리케이션을 사용하여 DB 인스턴스의 데이터베이스에 대한 액세스를 지원합니다. Amazon RDS는 Telnet 또는 Secure Shell(SSH)을 사용해 DB 인스턴스에 직접 호스트 액세스하는 것을 허용하지 않습니다.

DB 인스턴스를 생성할 경우 인스턴스를 생성하는 데 사용하는 마스터 계정은 DBA 사용자 권한을 받습니다 (일부 제한 있음). 이 계정을 사용하여 데이터베이스에서 추가 사용자 계정 생성 등과 같은 관리 작업을 수행 합니다. SYS 사용자, SYSTEM 사용자 및 기타 관리 계정은 사용할 수 없습니다.

DB 인스턴스를 생성하기 전에 이 안내서의 [Amazon RDS 설정 \(p. 5\)](#) 섹션에 있는 단계를 완료해야 합니다.

Amazon RDS의 Oracle을 위한 공통 관리 작업

다음은 Amazon RDS Oracle DB 인스턴스로 수행하는 일반적인 관리 작업과 각 작업에 해당하는 문서 링크입니다.

작업 영역	관련 문서
인스턴스 클래스, 스토리지 및 PIOPS 프로덕션 목적으로 DB 인스턴스를 만들 경우에는 Amazon RDS에서 인스턴스 클래스, 스토리지 유형 및 프로비저닝된 IOPS이 작동하는 방식을 이해해야 합니다.	Oracle을 위한 DB 인스턴스 클래스 지원 (p. 705) Amazon RDS 스토리지 유형 (p. 101)
다중 AZ 배포 프로덕션 DB 인스턴스에서는 다중 AZ 배포를 사용해야 합니다. 다중 AZ 배포는 DB 인스턴스를 위해 향상된 가용성, 데이터 내구성 및 내결함성을 제공합니다.	Amazon RDS를 위한 고가용성(다중 AZ) (p. 109)

작업 영역	관련 문서
Amazon Virtual Private Cloud(VPC) AWS 계정에 기본 VPC가 있는 경우에는 DB 인스턴스가 기본 VPC 내부에 자동으로 생성됩니다. 계정에 기본 VPC가 없는데 VPC 안에 DB 인스턴스를 만들려면 VPC와 서브넷 그룹을 만든 후 DB 인스턴스를 만들립니다.	EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 (p. 407) VPC에서 Amazon RDS DB 인스턴스를 사용한 작업 (p. 415)
보안 그룹 기본적으로, DB 인스턴스와 함께 인스턴스에 대한 액세스를 막는 방화벽도 생성됩니다. 따라서 DB 인스턴스에 액세스하기 위한 알맞은 IP 주소와 네트워크 구성으로 보안 그룹을 만들어야 합니다. 만들어야 할 보안 그룹은 DB 인스턴스가 어떤 Amazon EC2 플랫폼에 있고, Amazon EC2 인스턴스에서 DB 인스턴스에 액세스하는지 여부에 따라 결정됩니다. 일반적으로 DB 인스턴스가 EC2-Classic 플랫폼에 있으면 DB 보안 그룹을 생성해야 하고, DB 인스턴스가 EC2-VPC 플랫폼에 있으면 VPC 보안 그룹을 생성해야 합니다.	EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 (p. 407) 보안 그룹을 통한 액세스 제어 (p. 390)
파라미터 그룹 DB 인스턴스에 특정 데이터베이스 파라미터가 필요할 경우, 파라미터 그룹을 만든 후 DB 인스턴스를 만들어야 합니다.	DB 파라미터 그룹 작업 (p. 161)
옵션 그룹 DB 인스턴스에 특정 데이터베이스 옵션이 필요할 경우, 옵션 그룹을 만든 후 DB 인스턴스를 만들어야 합니다.	Oracle DB 인스턴스 옵션 (p. 772)
DB 인스턴스에 연결 보안 그룹을 만들고 이를 DB 인스턴스에 연결한 후, Oracle SQL Plus 와 같은 스탠다드 SQL 클라이언트 애플리케이션을 사용하여 DB 인스턴스에 연결할 수 있습니다.	Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 (p. 736)
백업 및 복원 DB 인스턴스를 구성하여 자동 백업을 생성하거나 수동 스냅샷을 생성한 다음 백업 또는 스냅샷에서 인스턴스를 복원할 수 있습니다.	Amazon RDS DB 인스턴스 백업 및 복원 (p. 196)
모니터링 CloudWatch Amazon RDS 측정치, 이벤트 및 향상된 모니터링 기능을 통해 Oracle DB 인스턴스를 모니터링할 수 있습니다.	DB 인스턴스 측정치 보기 (p. 236) Amazon RDS 이벤트 보기 (p. 305)
로그 파일 Oracle DB 인스턴스의 로그 파일에 액세스할 수 있습니다.	Amazon RDS 데이터베이스 로그 파일 (p. 306)

Oracle DB 인스턴스 작업을 위한 고급 작업과 선택적 기능도 있습니다. 자세한 내용은 다음 설명서 단원을 참조하십시오.

- Amazon RDS의 Oracle과 관련된 일반 DBA 작업에 대한 자세한 내용은 [Oracle DB 인스턴스에 대한 일반 DBA 작업 \(p. 832\)](#) 단원을 참조하십시오.
- Oracle GoldenGate 지원에 대한 자세한 내용은 [Amazon RDS와 함께 Oracle GoldenGate 사용하기 \(p. 885\)](#) 단원을 참조하십시오.

- Siebel 고객 관계 관리(CRM) 지원에 대한 자세한 내용은 [Amazon RDS에서 Oracle에 Siebel Database 설치 \(p. 899\)](#) 단원을 참조하십시오.

Oracle 라이선싱

Oracle용 Amazon RDS에 사용 가능한 라이선싱 옵션은 라이선스 포함 및 기존 보유 라이선스 사용(BYOL)의 2가지입니다. Amazon RDS에 Oracle DB 인스턴스를 만든 후 [AWS Management 콘솔](#), Amazon RDS API [ModifyDBInstance](#) 작업 또는 AWS CLI [modify-db-instance](#) 명령을 사용하여 라이선싱 모델을 변경할 수 있습니다.

라이선스 포함

라이선스 포함 모델에서는 Oracle 라이선스를 별도로 구입할 필요가 없습니다. AWS는 Oracle 데이터베이스 소프트웨어 라이선스를 보유합니다. 이 모델에서는 케이스 지원이 포함된 AWS Support 계정이 있는 경우, Amazon RDS 및 Oracle Database 서비스 요청은 모두 AWS Support에 문의해야 합니다.

Amazon RDS의 License Included 모델은 다음 Oracle 데이터베이스 에디션에 지원됩니다.

- Oracle Database Standard Edition One(SE1)
- Oracle Database Standard Edition Two(SE2)

Bring Your Own License(BYOL)

기존 보유 라이선스 사용 모델에서는 기존의 Oracle Database 라이선스를 사용하여 Amazon RDS에서 Oracle 배포를 실행할 수 있습니다. 실행할 DB 인스턴스 클래스와 Oracle Database 에디션에 적합한 Oracle 데이터베이스 라이선스(소프트웨어 업데이트 라이선스 및 지원 포함)가 있어야 합니다. 또한 클라우드 컴퓨팅 환경에 대한 Oracle의 Oracle Database 소프트웨어 라이선스 부여 정책을 따라야 합니다. Oracle의 Amazon EC2 라이선스 정책에 대한 자세한 내용은 [Licensing Oracle Software in the Cloud Computing Environment](#)를 참조하십시오.

이 모델에서는 활성 Oracle 지원 계정을 계속 사용할 수 있습니다. Oracle Database 서비스 요청은 Oracle에 직접 문의하십시오. 케이스 지원이 포함된 AWS Support 계정이 있는 경우 Amazon RDS 관련 문제는 AWS Support에 문의할 수 있습니다. Amazon Web Services와 Oracle은 두 조직의 지원이 필요한 경우를 위해 다른 공급업체 지원 프로세스를 갖추고 있습니다.

Amazon RDS의 Bring Your Own License 모델은 다음 Oracle 데이터베이스 에디션에 지원됩니다.

- Oracle Database Enterprise Edition(EE)
- Oracle Database Standard Edition(SE)
- Oracle Database Standard Edition One(SE1)
- Oracle Database Standard Edition Two(SE2)

Oracle 다중 AZ 배포 라이선스

Amazon RDS는 고가용성 장애 조치 솔루션으로서 Oracle용 다중 AZ 배포를 지원합니다. 프로덕션 워크로드에는 다중 AZ를 권장합니다. 자세한 내용은 [Amazon RDS를 위한 고가용성\(다중 AZ\) \(p. 109\)](#) 단원을 참조하십시오.

기본 보유 라이선스 사용 모델을 사용하는 경우, 다중 AZ 배포에 기본 DB 인스턴스와 보조 DB 인스턴스 모두에 대한 라이선스가 있어야 합니다.

Oracle 버전 간 마이그레이션

BYOL 모델의 경우, 실행하려는 DB 인스턴스의 에디션과 클래스에 적합한 미사용 Oracle 라이선스가 있다고 가정하면, 스탠다드 에디션(SE, SE1 또는 SE2)에서 엔터프라이즈 에디션(EE)으로 마이그레이션할 수 있습니다. 엔터프라이즈 에디션에서 다른 에디션으로 마이그레이션할 수는 없습니다.

에디션을 변경하고 데이터를 유지하려면

1. DB 인스턴스의 스냅샷을 생성합니다.

자세한 내용은 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.

2. 스냅샷을 새 DB 인스턴스로 복원하고 사용하려는 Oracle 데이터베이스 에디션을 선택합니다.

자세한 내용은 [DB 스냅샷에서 복원 \(p. 205\)](#) 단원을 참조하십시오.

3. (선택 사항) 이전 DB 인스턴스를 계속 실행하고 적절한 Oracle 데이터베이스 라이선스를 갖고 싶지 않으면 이전 DB 인스턴스를 삭제하십시오.

자세한 내용은 [DB 인스턴스 삭제 \(p. 128\)](#) 단원을 참조하십시오.

Oracle을 위한 DB 인스턴스 클래스 지원

DB 인스턴스의 계산 및 메모리 용량은 해당 DB 인스턴스 클래스에 의해 결정됩니다. 필요한 DB 인스턴스 클래스는 DB 인스턴스의 처리력 및 메모리 요구 사항에 따라 다릅니다. 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.

다음은 Oracle을 지원하는 DB 인스턴스 클래스입니다.

Oracle Edition	버전 12.2.0.1 지원	버전 12.1.0.2 지원	버전 11.2.0.4 지원
Enterprise Edition(EE) Bring Your Own License(BYOL)	db.m5.large–db.m5.24xlarge	db.m5.large–db.m5.24xlarge	db.m5.large–db.m5.24xlarge
	db.m4.large–db.m4.16xlarge	db.m4.large–db.m4.16xlarge	db.m4.large–db.m4.16xlarge
	db.m3.medium–db.m3.2xlarge	db.m3.medium–db.m3.2xlarge	db.m3.medium–db.m3.2xlarge
	db.x1e.xlarge–db.x1e.32xlarge	db.x1e.xlarge–db.x1e.32xlarge	db.x1e.xlarge–db.x1e.32xlarge
	db.x1.16xlarge–db.x1.32xlarge	db.x1.16xlarge–db.x1.32xlarge	db.x1.16xlarge–db.x1.32xlarge
	db.r5.large–db.r5.24xlarge	db.r5.large–db.r5.24xlarge	db.r5.large–db.r5.24xlarge
	db.r4.large–db.r4.16xlarge	db.r4.large–db.r4.16xlarge	db.r4.large–db.r4.16xlarge
	db.r3.large–db.r3.8xlarge	db.r3.large–db.r3.8xlarge	db.r3.large–db.r3.8xlarge
	db.t3.small–db.t3.2xlarge	db.t3.micro–db.t3.2xlarge	db.t3.micro–db.t3.2xlarge
	db.t2.small–db.t2.2xlarge	db.t2.micro–db.t2.2xlarge	db.t2.micro–db.t2.2xlarge
Standard Edition 2(SE2)	db.m5.large–db.m5.4xlarge	db.m5.large–db.m5.4xlarge	—
	db.m4.large–db.m4.4xlarge	db.m4.large–db.m4.4xlarge	

Oracle Edition	버전 12.2.0.1 지원	버전 12.1.0.2 지원	버전 11.2.0.4 지원
Bring Your Own License(BYOL)	db.m3.medium– db.m3.2xlarge db.x1e.xlarge– db.x1e.4xlarge db.r5.large–db.r5.4xlarge db.r4.large–db.r4.4xlarge db.r3.large–db.r3.4xlarge db.t3.small–db.t3.2xlarge db.t2.small–db.t2.2xlarge	db.m3.medium– db.m3.2xlarge db.x1e.xlarge– db.x1e.4xlarge db.r5.large–db.r5.4xlarge db.r4.large–db.r4.4xlarge db.r3.large–db.r3.4xlarge db.t3.micro–db.t3.2xlarge db.t2.micro–db.t2.2xlarge	
Standard Edition 2(SE2) 라이선스 포함	db.m5.large–db.m5.4xlarge db.m4.large–db.m4.4xlarge db.m3.medium– db.m3.2xlarge db.r5.large–db.r5.4xlarge db.r4.large–db.r4.4xlarge db.r3.large–db.r3.4xlarge db.t3.small–db.t3.2xlarge db.t2.small–db.t2.2xlarge	db.m5.large–db.m5.4xlarge db.m4.large–db.m4.4xlarge db.m3.medium– db.m3.2xlarge db.r5.large–db.r5.4xlarge db.r4.large–db.r4.4xlarge db.r3.large–db.r3.4xlarge db.t3.micro–db.t3.2xlarge db.t2.micro–db.t2.2xlarge	—
Standard Edition 1(SE1) Bring Your Own License(BYOL)	—	—	db.m5.large–db.m5.4xlarge db.m4.large–db.m4.4xlarge db.m3.medium– db.m3.2xlarge db.x1e.xlarge– db.x1e.4xlarge db.r5.large–db.r5.4xlarge db.r4.large–db.r4.4xlarge db.r3.large–db.r3.4xlarge db.t3.micro–db.t3.2xlarge db.t2.micro–db.t2.2xlarge

Oracle Edition	버전 12.2.0.1 지원	버전 12.1.0.2 지원	버전 11.2.0.4 지원
Standard Edition 1(SE1) 라이선스 포함	—	—	db.m5.large–db.m5.4xlarge db.m4.large–db.m4.4xlarge db.m3.medium–db.m3.2xlarge db.r5.large–db.r5.4xlarge db.r3.large–db.r3.4xlarge db.t3.micro–db.t3.2xlarge db.t2.micro–db.t2.large
Standard Edition(SE) Bring Your Own License(BYOL)	—	—	db.m5.large–db.m5.4xlarge db.m4.large–db.m4.4xlarge db.m3.medium–db.m3.2xlarge db.x1e.xlarge–db.x1e.8xlarge db.r5.large–db.r5.4xlarge db.r4.large–db.r4.8xlarge db.r3.large–db.r3.8xlarge db.t3.micro–db.t3.2xlarge db.t2.micro–db.t2.2xlarge

Oracle용 db.m3 및 db.r3 DB 인스턴스 클래스 지원 중단

db.m3 및 db.r3 DB 인스턴스 클래스는 Oracle용 Amazon RDS에 대해 더 이상 지원되지 않습니다. 해당 DB 인스턴스 클래스는 성능이 더 좋고 일반적으로 낮은 가격으로 구할 수 있는 DB 인스턴스로 교체되었습니다. 2019년 9월 30일부터 Oracle용 Amazon RDS는 DB 인스턴스를 만료되지 않은 DB 인스턴스 클래스로 자동으로 확장합니다.

db.m3 및 db.r3 DB 인스턴스 클래스를 사용하는 DB 인스턴스를 보유한 경우 Amazon RDS에서는 만료되지 않은 유사한 DB 인스턴스 클래스를 사용하도록 각 인스턴스를 자동으로 수정합니다. DB 인스턴스를 수정하여 DB 인스턴스에 대한 DB 인스턴스 클래스를 직접 변경할 수 있습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

db.m3 또는 db.r3 DB 인스턴스 클래스를 사용 중인 DB 인스턴스의 DB 스냅샷이 있는 경우 DB 스냅샷을 복원할 때 만료되지 않은 DB 인스턴스 클래스를 선택할 수 있습니다. 자세한 내용은 [DB 스냅샷에서 복원 \(p. 205\)](#) 단원을 참조하십시오.

Oracle용 db.m1 및 db.m2 DB 인스턴스 클래스 지원 종단

db.m1 및 db.m2 DB 인스턴스 클래스는 Oracle용 Amazon RDS에 대해 더 이상 지원되지 않습니다. 해당 DB 인스턴스 클래스는 성능이 더 좋고 일반적으로 낮은 가격으로 구할 수 있는 DB 인스턴스로 교체되었습니다. 2018년 9월 12일부터 Oracle용 Amazon RDS는 DB 인스턴스를 만료되지 않은 DB 인스턴스 클래스로 자동으로 확장합니다.

db.m1 및 db.m2 DB 인스턴스 클래스를 사용하는 DB 인스턴스를 보유한 경우 Amazon RDS에서는 만료되지 않은 유사한 DB 인스턴스 클래스를 사용하도록 각 인스턴스를 자동으로 수정합니다. DB 인스턴스를 수정하여 DB 인스턴스에 대한 DB 인스턴스 클래스를 직접 변경할 수 있습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

db.m1 또는 db.m2 DB 인스턴스 클래스를 사용 중인 DB 인스턴스의 DB 스냅샷이 있는 경우 DB 스냅샷을 복원할 때 만료되지 않은 DB 인스턴스 클래스를 선택할 수 있습니다. 자세한 내용은 [DB 스냅샷에서 복원 \(p. 205\)](#) 단원을 참조하십시오.

Oracle 보안

Oracle 데이터베이스 엔진에는 역할 기반 보안이 사용됩니다. 역할이란 사용자에게 부여되거나 사용자로부터 해지될 수 있는 권한의 모음입니다. 일반적으로 DBA라는 사전 정의된 역할은 Oracle 데이터베이스 엔진에 대한 모든 관리 권한을 허용합니다. 아래의 권한들은 Oracle 엔진을 사용하는 Amazon RDS DB 인스턴스의 DBA 역할에는 사용할 수 없습니다.

- 데이터베이스 변경
- 시스템 변경
- 디렉터리 생성
- 디렉터리 삭제
- 권한 부여
- 역할 부여

DB 인스턴스를 생성할 경우 인스턴스를 생성하는 데 사용하는 마스터 계정은 DBA 사용자 권한을 받습니다 (일부 제한 있음). 이 계정을 사용하여 데이터베이스에서 추가 사용자 계정 생성 등과 같은 관리 작업을 수행합니다. SYS 사용자, SYSTEM 사용자 및 기타 관리 계정은 사용할 수 없습니다.

Amazon RDS Oracle에서는 SSL/TLS 암호화 연결과 Oracle NNE(기본 네트워크 암호화) 옵션을 사용하여 애플리케이션과 Oracle DB 인스턴스 간의 연결을 암호화할 수 있습니다. Amazon RDS의 Oracle에서 SSL을 사용하는 방법에 대한 자세한 내용은 [Oracle DB 인스턴스에 SSL 사용 \(p. 708\)](#) 단원을 참조하십시오. Oracle 기본 네트워크 암호화 옵션에 대한 자세한 내용은 [Oracle 기본 네트워크 암호화 \(p. 807\)](#) 단원을 참조하십시오.

Oracle DB 인스턴스에 SSL 사용

SSL(Secure Sockets Layer)은 클라이언트와 서버 간의 네트워크 연결을 보호하는 데 사용되는 업계 표준 프로토콜입니다. SSL 버전 3.0 이후에 이름이 TLS(전송 계층 보안)로 변경되었지만 여전히 SSL로 지칭되는 경우가 많으므로 여기서도 이 프로토콜을 SSL로 지칭하겠습니다. Amazon RDS에서는 Oracle DB 인스턴스를 위한 SSL 암호화를 지원합니다. SSL을 사용하여 애플리케이션 클라이언트와 Oracle DB 인스턴스 간의 연결을 암호화할 수 있습니다. Oracle의 모든 AWS 리전에서 SSL 지원 기능을 사용할 수 있습니다.

DB 인스턴스와 연결된 옵션 그룹에 Oracle SSL 옵션을 추가하여 Oracle DB 인스턴스에 대해 SSL 암호화를 활성화할 수 있습니다. Amazon RDS에서는 Oracle의 요구 사항에 따라 두 번째 포트를 SSL 연결에 사용합니다. 이로 인해 DB 인스턴스와 Oracle 클라이언트 간에 클리어 텍스트 통신과 SSL로 암호화된 통신이 동시

에 발생할 수 있습니다. 예를 들어 이 포트를 클리어 텍스트 통신에 사용하여 VPC 내의 다른 리소스와 통신하면서 동일한 포트를 SSL로 암호화된 통신에 사용하여 VPC 외부의 리소스와 통신할 수 있습니다.

자세한 내용은 [Oracle 보안 소켓 Layer \(p. 810\)](#) 단원을 참조하십시오.

Note

동일한 DB 인스턴스에서는 SSL과 Oracle NNE(기본 네트워크 암호화)를 모두 사용할 수 없습니다.
SSL 암호화를 사용하려면 먼저 다른 연결 암호화를 모두 비활성화해야 합니다.

Amazon RDS 기반 Oracle 12c

Amazon RDS는 Oracle Enterprise Edition과 Oracle Standard Edition Two를 포함하는 Oracle 버전 12c를 지원합니다. Oracle 버전 12c에는 메이저 버전 두 개가 포함되어 있습니다.

- [Oracle 12c 버전 12.2.0.1\(Amazon RDS 포함\) \(p. 709\)](#)
- [Oracle 12c 버전 12.1.0.2\(Amazon RDS 포함\) \(p. 712\)](#)

Oracle 12c 버전 12.2.0.1(Amazon RDS 포함)

Oracle 12c 버전 12.2.0.1에는 이전 버전에 비해 새로운 기능과 업데이트가 많이 포함되어 있습니다. 이 단원에서는 Amazon RDS에서 Oracle 12c 버전 12.2.0.1를 사용하는 데 중요한 기능과 변경 내용을 확인할 수 있습니다. 변경 내용의 전체 목록은 [Oracle 12c 버전 12.2 설명서](#)를 참조하십시오. 각 Oracle 12c 에디션에서 지원하는 전체 기능 목록은 Oracle 설명서의 [Permitted Features, Options, and Management Packs by Oracle Database Offering](#)을 참조하십시오.

Oracle 12c 버전 12.1.0.2에는 Amazon RDS DB 인스턴스에 영향을 주는 16개의 새로운 파라미터와 18개의 새로운 시스템 권한을 비롯하여, 더 이상 지원되지 않는 여러 패키지와 다양한 새 옵션 그룹 설정이 포함되어 있습니다. 이러한 변경 내용에 대한 자세한 내용은 다음 단원을 참조하십시오.

Oracle 12c 버전 12.2.0.1의 Amazon RDS 파라미터 변경

Oracle 12c 버전 12.2.0.1에는 범위와 기본값이 새로 변경된 다양한 파라미터와 20개의 새로운 파라미터가 포함되었습니다.

다음 표에는 Oracle 12c 버전 12.2.0.1에 대한 새로운 Amazon RDS 파라미터가 나와 있습니다.

이름	값	수정 가능	설명
allow_global_dblinks	TRUE, FALSE(기본 값)	Y	데이터베이스에 대해 데이터베이스 링크의 LDAP 조회가 허용되는지 여부를 지정합니다.
approx_for_aggregation	TRUE, FALSE(기본 값)	Y	집계 쿼리의 정확한 쿼리 처리를 대략적인 쿼리 처리로 바꿉니다.
approx_for_count_distinct	TRUE, FALSE(기본 값)	Y	COUNT (DISTINCT <i>expr</i>) 쿼리를 APPROX_COUNT_DISTINCT 쿼리로 자동으로 바꿉니다.
approx_for_percentile	없음(기본 값), PERCENTILE_CONT, PERCENTILE_CONT DETERMINISTIC, PERCENTILE_DISC,	Y	정확한 percentile 함수를 대략적인 percentile 함수로 변환합니다.

이름	값	수정 가능	설명
	PERCENTILE_DISC DETERMINISTIC, ALL, ALL DETERMINISTIC		
cursor_invalidation	DEFERRED, IMMEDIATE(기본값)	Y	DDL 문에 기본적으로 연기된 커서 무효화를 사용할지, 즉각적인 커서 무효화를 사용할지 제어합니다.
data_guard_sync_latency	0(기본값) - LOG_ARCHIVE_DEST_n 파라미터의 NET_TIMEOUT 속성으로 지정된 시간(초)	Y	로그 기록기(LGWR) 프로세스가 일련의 Oracle Data Guard SYNC redo transport 모드 연결에서 첫 번째 응답 이후를 기다리는 시간(초)을 제어합니다.
data_transfer_cache_size	0 – 512M, 다음 그레뉼(granule) 크기로 반올림	Y	RMAN RECOVER ... NONLOGGED BLOCK 명령을 실행하는 동안 인스턴스가 소비할 데이터 블록(일반적으로 Oracle Data Guard 환경의 기본 데이터베이스에 있음)을 수신하는 데 사용되는 데이터 전송 캐시의 크기(바이트)를 설정합니다.
inmemory_adg_enabled	TRUE(기본값), FALSE	Y	인 메모리 캐시 크기 외에도 Active Data Guard의 인 메모리가 활성화되어 있는지 나타냅니다.
inmemory_expressions_usage	STATIC_ONLY, DYNAMIC_ONLY, ENABLE(기본값), DISABLE	Y	In-Memory Column Store(IM 열 저장소)에 채워지고 쿼리에 사용할 수 있는 인 메모리 표현식(IM 표현식)을 제어합니다.
inmemory_virtual_columns	ENABLE, MANUAL(기본값), DISABLE	Y	In-Memory Column Store(IM 열 저장소)에 채워지고 쿼리에 사용할 수 있는 인 메모리 표현식(IM 표현식)을 제어합니다.
instance_abort_delay_time	0(기본값) 이상	Y	중대한 프로세스가 멈추거나 복구 불가능한 인스턴스 오류가 발생한 경우와 같이 내부에서 시작된 인스턴스 중단을 지연할 시간(초)을 지정합니다.
instance_mode	READ-WRITE(기본값), READ-ONLY, READ-MOSTLY	N	인스턴스가 읽기-쓰기인지, 읽기 전용인지, 읽기 위주인지 나타냅니다.
long_module_action	TRUE(기본값), FALSE	Y	모듈 및 작업에 더 긴 길이 사용을 활성화합니다.
max_idle_time	0(기본값)~최대 정수. 값 0은 제한이 없음을 나타냅니다.	Y	세션이 유휴 상태일 수 있는 최대 시간(분)을 지정합니다. 그 시점 후에 세션이 자동으로 종료됩니다.

이름	값	수정 가능	설명
optimizer_adaptive_plans	TRUE(기본값), FALSE	Y	적응형 계획을 제어합니다. 적응형 계획은 쿼리를 실행할 때 수집한 통계를 기반으로 실행 시 결정하는 대체 선택으로 구축된 실행 계획입니다.
optimizer_adaptive_statistics	TRUE, FALSE(기본값)	Y	적응형 통계를 제어합니다. 일부 쿼리 형태는 기본 테이블 통계에만 의존하기에는 너무 복잡하므로, 최적화 프로그램이 적응형 통계로 이러한 통계를 보완합니다.
outbound_dblink_protocols	ALL(기본값), NONE, TCP, TCPS, IPC	N	데이터베이스의 아웃바운드 데이터베이스 링크를 위해 통신이 허용되는 네트워크 프로토콜을 지정합니다.
resource_manage_goldengate	TRUE, FALSE(기본값)	Y	데이터베이스의 Oracle GoldenGate 적용 프로세스가 리소스 관리형인지 여부를 결정합니다.
standby_db_preserve_states	NONE(기본값), SESSION, ALL	N	읽을 수 있는 물리적 스탠바이 데이터베이스가 기본 데이터베이스로 변환될 때 인스턴스의 사용자 세션 및 기타 내부 상태를 보존할지 여부를 제어합니다.
uniform_log_timestamp_format	TRUE(기본값), FALSE	Y	Oracle 데이터베이스 추적(.trc) 파일과 로그 파일(예: 알림 로그)에서 균일한 타임스탬프 형식을 사용하도록 지정합니다.

`compatible` 파라미터는 Amazon RDS 기반 Oracle 12c 버전 12.2.0.1에 대한 기본값이 변경되었습니다. 다음 표에는 새 기본값이 나와 있습니다.

파라미터 이름	Oracle 12c 버전 12.2.0.1 기본값	Oracle 12c 버전 12.1.0.2 기본값
<code>compatible</code>	12.2.0	12.0.0

`optimizer_features_enable` 파라미터는 Amazon RDS 기반 Oracle 12c 버전 12.2.0.1에 대한 값 범위가 변경되었습니다. 이전 및 새 값 범위는 다음 표를 참조하십시오.

파라미터 이름	12c 버전 12.2.0.1 범위	12c 버전 12.1.0.2 범위
<code>optimizer_features_enable</code>	8.0.0 ~ 12.2.0.1	8.0.0 ~ 12.1.0.2

Oracle 12c 버전 12.2.0.1에서는 다음 파라미터가 제거되었습니다.

- `global_context_pool_size`
- `max_enabled_roles`
- `optimizer_adaptive_features`
- `parallel_automatic_tuning`
- `parallel_degree_level`

- use_indirect_data_buffers

Oracle 12c 버전 12.2.0.1에서는 다음 파라미터가 지원되지 않습니다.

- sec_case_sensitive_logon

Oracle 12c 버전 12.2.0.1의 Amazon RDS 보안 변경

Oracle 12c 버전 12.2.0.1에서는 데이터베이스 수준 트리거 소유자에게 ADMINISTER DATABASE TRIGGER 권한을 직접 부여해야 합니다. Oracle 12c 버전 12.2.0.1로의 메이저 버전 업그레이드 중에 트리거 소유자가 필요한 권한을 가질 수 있도록 Amazon RDS가 트리거를 소유한 모든 사용자에게 이 권한을 부여합니다. 자세한 내용은 My Oracle Support 문서 [2275535.1](#)을 참조하십시오.

Oracle 12c 버전 12.1.0.2(Amazon RDS 포함)

Oracle 12c 버전 12.1.0.2는 이전 버전에 비해 500개 이상의 새로운 기능과 업데이트를 제공합니다. 이 단원에서는 Amazon RDS에서 Oracle 12c 버전 12.1.0.2를 사용하는 데 중요한 기능과 변경 내용을 확인할 수 있습니다. 변경 내용의 전체 목록은 [Oracle 12c 버전 12.1 설명서](#)를 참조하십시오. 각 Oracle 12c 에디션에서 지원하는 전체 기능 목록은 Oracle 설명서의 [Permitted Features, Options, and Management Packs by Oracle Database Edition](#)을 참조하십시오.

Oracle 12c 버전 12.1.0.2에는 Amazon RDS DB 인스턴스에 영향을 주는 16개의 새로운 파라미터와 18개의 새로운 시스템 권한을 비롯하여, 더 이상 지원되지 않는 여러 패키지와 다양한 새 옵션 그룹 설정이 포함되어 있습니다. 이러한 변경 사항에 대한 자세한 내용은 다음 단원을 참조하십시오.

Oracle 12c 버전 12.1.0.2의 Amazon RDS 파라미터 변경

Oracle 12c 버전 12.1.0.2에는 범위와 기본값이 새로 변경된 다양한 파라미터와 16개의 새로운 파라미터가 포함되었습니다.

다음 표에는 Oracle 12c 버전 12.1.0.2에 대한 새로운 Amazon RDS 파라미터가 나와 있습니다.

이름	값	수정 가능	설명
connection_brokers	CONNECTION_BROKERS = broker_description[...]	N	연결 브로커 유형, 유형별 연결 브로커 수, 브로커당 최대 연결 수를 지정합니다.
db_index_compression_inheritance	TABLESPACE, TABLE, ALL, NONE	Y	테이블 또는 테이블스페이스 수준 압축 상속에 대해 설정되는 옵션을 표시합니다.
db_big_table_cache_percent_target	0~90	Y	자동 대형 테이블 캐싱에 대한 캐시 셙션 대상 크기를 버퍼 캐시의 비율로 지정합니다.
heat_map	ON, OFF	Y	데이터베이스에서 데이터 조작 언어 (DML) 및 데이터 정의 언어 (DDL) 문에 따른 모든 세그먼트의 읽기 및 쓰기 액세스와 데이터베이스 블록 수정을 추적 할 수 있습니다.
inmemory_clause_default	INMEMORY, NO INMEMORY	Y	INMEMORY_CLAUSE_DEFAULT를 사용하면 새 테이블과 구체화된 보기에 대

이름	값	수정 가능	설명
			한 기본 In-Memory Column Store(IM 열 저장소) 절을 지정할 수 있습니다.
inmemory_clause_default_memcompress	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	INMEMORY_CLAUSE_DEFAULT를 참조하십시오.
inmemory_clause_default_priority	PRIORITY LOW, PRIORITY MEDIUM, PRIORITY HIGH, PRIORITY CRITICAL, PRIORITY NONE	Y	INMEMORY_CLAUSE_DEFAULT를 참조하십시오.
inmemory_force	DEFAULT, OFF	Y	INMEMORY_FORCE를 사용하면 INMEMORY로 지정된 테이블과 구체화된 보기가 In-Memory Column Store(IM 열 저장소)에 채워지는지 여부를 지정할 수 있습니다.
inmemory_max_populate_servers	Null	N	INMEMORY_MAX_POPULATE_SERVERS는 백그라운드 채우기 서버에서 나머지 시스템을 오버로드하지 않도록 In-Memory Column Store(IM 열 저장소) 채우기에 사용할 백그라운드 채우기 서버의 최대 수를 지정합니다.
inmemory_query	ENABLE(기본값), DISABLE	Y	INMEMORY_QUERY는 세션 또는 시스템 수준에서 전체 데이터베이스에 대한 인 메모리 쿼리를 활성화하거나 비활성화하는 데 사용됩니다.
inmemory_size	0, 104857600-274877906944	Y	INMEMORY_SIZE는 데이터베이스 인스턴스에 대한 In-Memory Column Store(IM 열 저장소) 크기를 설정합니다.

이름	값	수정 가능	설명
inmemory_trickle_repopulate_servers_percent	0 to 50 percent	Y	INMEMORY_TRICKLE_REPOPULATE_SERVERS_PERCENT는 In-Memory Column Store(IM 열 저장소) 다시 채우기에 사용되는 백그라운드 채우기 서버의 최대 수를 제한합니다. 서서히 다시 채우기는 채우기 서버의 작은 부분만 사용하도록 설계되었으므로 이 제한이 적용됩니다.
max_string_size	STANDARD(기본값), EXTENDED	N	VARCHAR2, NVARCHAR2 및 RAW의 최대 크기를 제어합니다. 자세한 내용은 확장 데이터 형식 사용 (p. 726) 단원을 참조하십시오.
optimizer_adaptive_features	TRUE(기본값), FALSE	Y	모든 가변 최적화 프로그램 기능을 활성화 또는 비활성화합니다.
optimizer_adaptive_reporting_only	TRUE, FALSE(기본값)	Y	가변 최적화에 대한 보고 전용 모드를 제어합니다.
pdb_file_name_convert	기본값이 없습니다.	N	기존 파일의 이름을 새 파일 이름에 매핑합니다.
pga_aggregate_limit	1 - 최대 메모리	Y	인스턴스에서 사용되는 PGA 메모리 집계에 대한 제한을 지정합니다.
processor_group_name	기본값이 없습니다.	N	지정된 운영 체제 프로세서 그룹 내에서 데이터베이스 인스턴스를 실행하도록 지시합니다.
spatial_vector_acceleration	TRUE, FALSE	N	공간 옵션의 일부인 공간 벡터 가속을 활성화 또는 비활성화합니다.
temp_undo_enabled	TRUE, FALSE(기본값)	Y	특정 세션 내의 트랜잭션에서 임시 실행 취소 로그를 사용할 수 있는지 여부를 결정합니다.
threaded_execution	TRUE, FALSE	N	멀티스레드 Oracle 모델을 활성화하지만 OS 인증을 방지합니다.
unified_audit_sga_queue_size	1MB - 30MB	Y	통합 감사에 대한 시스템 글로벌 영역 (SGA) 대기열 크기를 지정합니다.
use_dedicated_broker	TRUE, FALSE	N	전용 서버가 생성되는 방법을 결정합니다.

몇 개의 파라미터는 Amazon RDS 기반 Oracle 12c 버전 12.1.0.2에 대한 값 범위가 변경되었습니다. 이전 및 새 값 범위는 다음 표를 참조하십시오.

파라미터 이름	12c 버전 12.1.0.2 범위	11g 범위
audit_trail	os db [, extended] xml [, extended]	os db [, extended] xml [, extended] true false

파라미터 이름	12c 버전 12.1.0.2 범위	11g 범위
compatible	Oracle 11g에서 업그레이드된 DB 인스턴스의 경우, 업그레이드 중에 더 낮은 값(이를 테면 11.2.0)이 명시적으로 제공되는 경우가 아니면 Amazon RDS에서 12.0.0으로 자동 설정됩니다. 새 Oracle 12c 버전 12.1.0.2 DB 인스턴스의 경우 Amazon RDS에서 12.0.0부터 시작	Amazon RDS에서 11.2.0부터 시작
db_securefile	PERMITTED PREFERRED ALWAYS IGNORE FORCE	PERMITTED ALWAYS IGNORE FORCE
db_writer_processes	1-100	1-36
optimizer_features_era	8.0.0 ~ 12.1.0.2	8.0.0 ~ 11.2.0.4
parallel_degree_policy	MANUAL,LIMITED,AUTO,ADAPTIVE	MANUAL,LIMITED,AUTO
parallel_min_server	0 ~ parallel_max_servers	CPU_COUNT * PARALLEL_THREADS_PER_CPU * 2 ~ parallel_max_servers

한 파라미터는 Amazon RDS 기반 Oracle 12c에 대한 기본값이 변경되었습니다. 다음 표에는 새 기본값이나 있습니다.

파라미터 이름	Oracle 12c 기본값	Oracle 11g 기본값
job_queue_processes	50	1000

Amazon RDS의 파라미터는 파라미터 그룹을 사용하여 관리됩니다. 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오. 특정 Oracle 에디션 및 버전에 지원되는 파라미터를 보려면 AWS CLI `describe-engine-default-parameters` 명령을 실행합니다.

예를 들어 Oracle Enterprise Edition 12c, 버전 12.1.0.2에 지원되는 파라미터를 보려면 다음 명령을 실행합니다.

```
aws rds describe-engine-default-parameters --db-parameter-group-family oracle-ee-12.1
```

Oracle 12c 버전 12.1.0.2에 대한 Amazon RDS 시스템 권한

Oracle 12c 버전 12.1.0.2에 대한 시스템 계정에 다양한 새 시스템 권한이 부여되었습니다. 이러한 새로운 시스템 권한에는 다음이 포함됩니다.

- 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

Oracle 12c 버전 12.1.0.2에 대한 Amazon RDS 옵션

Oracle 11g와 Oracle 12c 버전 12.1.0.2 간에는 다양한 Oracle 옵션이 변경되었습니다. 대부분의 옵션은 두 버전 간에 동일하게 유지됩니다. Oracle 12c 버전 12.1.0.2의 변경 사항은 다음과 같습니다.

- Oracle Enterprise Manager Database Express 12c가 Oracle Enterprise Manager 11g Database Control로 대체되었습니다. 자세한 내용은 [Oracle Enterprise Manager Database Express \(p. 791\)](#) 단원을 참조하십시오.
- XMLDB 옵션은 Oracle 12c 버전 12.1.0.2에서 기본적으로 설치되므로, 더 이상 이 옵션을 설치할 필요가 없습니다.

Oracle 12c 버전 12.1.0.2용 Amazon RDS PL/SQL 패키지

Oracle 12c 버전 12.1.0.2에는 많은 새로운 PL/SQL 패키지가 기본적으로 포함되어 있습니다. Amazon RDS Oracle 12c 버전 12.1.0.2에 포함되는 패키지는 다음과 같습니다.

패키지 이름	설명
CTX_ANL	CTX_ANL 패키지는 AUTO_LEXER와 함께 사용되며 lexer에서 사용자 지정 사전을 추가 및 삭제하는 절차를 제공합니다.
DBMS_APP_CONT	DBMS_APP_CONT 패키지는 현재 사용할 수 없는 세션의 인플라이트 트랜잭션이 커밋되는지 여부와 해당 세션에 대한 마지막 호출이 완료되었는지 여부를 결정하는 인터페이스를 제공합니다.
DBMS_AUTO_REPORT	DBMS_AUTO_REPORT 패키지는 Automatic Workload Repository(AWR)로 캡처된 SQL 모니터링 및 실시간 Automatic Database Diagnostic Monitor(ADDM) 데이터를 볼 수 있는 인터페이스를 제공합니다.
DBMS_GOLDENGATE_AUTH	DBMS_GOLDENGATE_AUTH 패키지는 GoldenGate 관리자에게 권한을 부여하거나 권한을 해지하는 하위 프로그램을 제공합니다.
DBMS_HEAT_MAP	DBMS_HEAT_MAP 패키지는 블록, 범위, 세그먼트, 객체, 테이블스페이스 등 다양한 수준의 스토리지에서 열 지도를 표면화할 수 있는 인터페이스를 제공합니다.
DBMS_ILM	DBMS_ILM 패키지는 자동 데이터 최적화(ADO) 정책을 사용하여 정보 수명 주기 관리(ILM) 전략을 구현할 수 있는 인터페이스를 제공합니다.
DBMS_ILM_ADMIN	DBMS_ILM_ADMIN 패키지는 자동 데이터 최적화(ADO) 정책 실행을 사용자 지정할 수 있는 인터페이스를 제공합니다.

패키지 이름	설명
DBMS_PART	DBMS_PART 패키지는 분할된 객체에 대한 유지 관리 및 관리 작업을 위한 인터페이스를 제공합니다.
DBMS_PRIVILEGE_CAPTURE	DBMS_PRIVILEGE_CAPTURE 패키지는 데이터베이스 권한 분석에 대한 인터페이스를 제공합니다.
DBMS_QOPATCH	DBMS_QOPATCH 패키지는 설치된 데이터베이스 패치를 볼 수 있는 인터페이스를 제공합니다.
DBMS_REDACT	DBMS_REDACT 패키지는 권한이 없는 사용자 또는 애플리케이션에서 실행한 쿼리로부터 반환되는 데이터를 마스킹(redact)할 수 있는 Oracle Data Redaction에 대한 인터페이스를 제공합니다.
DBMS_SPD	DBMS_SPD 패키지는 SQL 계획 명령(SPD)을 관리할 수 있는 하위 프로그램을 제공합니다.
DBMS_SQL_TRANSLATOR	DBMS_SQL_TRANSLATOR 패키지는 SQL 트랜잭션 프로필 생성, 구성 및 사용을 위한 인터페이스를 제공합니다.
DBMS_SQL_MONITOR	DBMS_SQL_MONITOR 패키지는 실시간 SQL 모니터링 및 실시간 데이터베이스 운영 모니터링에 대한 정보를 제공합니다.
DBMS_SYNC_REFRESH	DBMS_SYNC_REFRESH 패키지는 구체화된 보기에 대한 동기식 새로 고침을 수행할 수 있는 인터페이스를 제공합니다.
DBMS_TSDP_MANAGE	DBMS_TSDP_MANAGE 패키지는 데이터베이스에서 중요한 열 및 열 유형을 가져와서 관리할 수 있는 인터페이스를 제공하며, 투명한 중요 데이터 보호(TSDP) 정책에 대해 DBMS_TSDP_PROTECT 패키지와 함께 사용됩니다. DBMS_TSDP_MANAGE는 Enterprise Edition에서만 사용할 수 있습니다.
DBMS_TSDP_PROTECT	DBMS_TSDP_PROTECT 패키지는 DBMS_TSDP_MANAGE 패키지와 함께 투명한 중요 데이터 보호(TSDP) 정책을 구성할 수 있는 인터페이스를 제공합니다. DBMS_TSDP_PROTECT는 Enterprise Edition에서만 사용할 수 있습니다.
DBMS_XDB_CONFIG	DBMS_XDB_CONFIG 패키지는 Oracle XML DB와 해당 리포지토리를 구성할 수 있는 인터페이스를 제공합니다.
DBMS_XDB_CONSTANTS	DBMS_XDB_CONSTANTS 패키지는 일반적으로 사용되는 상수에 대한 인터페이스를 제공합니다. 오타 방지를 위해 동적 문자열 대신 상수를 사용하는 것이 좋습니다.
DBMS_XDB_REPOS	DBMS_XDB_REPOS 패키지는 Oracle XML 데이터베이스 리포지토리에서 작동하기 위한 인터페이스를 제공합니다.
DBMS_XMLSCHEMA_ANNOTATE	DBMS_XMLSCHEMA_ANNOTATE 패키지는 주로 사전 등록 스키마 주석을 사용하여 구조화된 스토리지 모델을 관리 및 구성할 수 있는 인터페이스를 제공합니다.
DBMS_XMLSTORAGE_MANAGE	DBMS_XMLSTORAGE_MANAGE 패키지는 스키마 등록을 완료한 이후에 XML 스토리지를 관리 및 수정할 수 있는 인터페이스를 제공합니다.
DBMS_XSTREAM_ADMIN	DBMS_XSTREAM_ADMIN 패키지는 Oracle 데이터베이스와 다른 시스템 간에 데이터베이스 변경 내용을 스트리밍하기 위한 인터페이스를 제공합니다. XStream을 사용하면 애플리케이션에서 데이터베이스 변경 내용을 스트리밍 아웃 또는 스트리밍 인할 수 있습니다.

패키지 이름	설명
DBMS_XSTREAM_AUTH	DBMS_XSTREAM_AUTH 패키지는 XStream 관리자에게 권한을 부여하거나 권한을 해지하는 하위 프로그램을 제공합니다.
UTL_CALL_STACK	UTL_CALL_STACK 패키지는 현재 실행 중인 하위 프로그램에 대한 정보를 제공하는 인터페이스를 제공합니다.

지원되지 않는 Oracle 12c 버전 12.1.0.2 패키지

다음과 같은 일부 Oracle 11g PL/SQL 패키지는 Oracle 12c 버전 12.1.0.2에서 지원되지 않습니다. 이러한 패키지에는 다음이 포함됩니다.

- DBMS_AUTO_TASK_IMMEDIATE
- DBMS_CDC_PUBLISH
- DBMS_CDC_SUBSCRIBE
- DBMS_EXPFIL
- DBMS_OBFUSCATION_TOOLKIT
- DBMS_RLMGR
- SDO_NET_MEM

Oracle 데이터베이스 기능 지원

Oracle Database는 다양한 기능을 지원하며 대부분 Amazon RDS Oracle에서 지원됩니다. 일부 기능에는 제한된 지원 또는 제한된 권한이 있을 수 있습니다. 일부 기능은 Enterprise Edition에서만 사용할 수 있으며 일부 기능은 추가 라이선스가 필요합니다. 특정 Oracle 데이터베이스 버전의 Oracle Database 기능에 대한 자세한 내용은 사용 중인 버전의 Oracle 데이터베이스 라이선싱 정보 사용자 설명서를 참조하십시오.

Note

이러한 목록은 포괄적이지 않습니다.

Amazon RDS Oracle은 다음 Oracle 데이터베이스 기능을 지원합니다.

- 고급 앱축
- Application Express(APEX)

자세한 내용은 [Oracle Application Express \(p. 783\)](#) 단원을 참조하십시오.

- 자동 메모리 관리
- 자동 실행 취소 관리
- Automatic Workload Repository(AWR)

자세한 내용은 [Automatic Workload Repository\(AWR\) 작업 \(p. 849\)](#) 단원을 참조하십시오.

- 동일한 AWS 리전 내 최대 성능의 활성 데이터 보호

자세한 내용은 [Amazon RDS의 Oracle 읽기 전용 복제본 작업 \(p. 768\)](#) 단원을 참조하십시오.

- 데이터 개정
- 데이터베이스 인 메모리(버전 12.1 이상)
- 분산 쿼리 및 트랜잭션
- 에디션 기반 재정의

자세한 내용은 [DB 인스턴스의 기본 에디션 설정 \(p. 850\)](#) 단원을 참조하십시오.

- Enterprise Manager Database Control(11g) 및 EM Express(12c)

자세한 내용은 [Oracle Enterprise Manager \(p. 791\)](#) 단원을 참조하십시오.

- 세분화된 감사
- 플래시백 테이블, 플래시백 쿼리, 플래시백 트랜잭션 쿼리
- 가져오기/내보내기(레거시 및 데이터 펌프) 및 SQL*Loader

자세한 내용은 [Amazon RDS의 Oracle로 데이터 가져오기 \(p. 758\)](#) 단원을 참조하십시오.

- Java 가상 머신(JVM)

자세한 내용은 [Oracle Java 가상 머신 \(p. 798\)](#) 단원을 참조하십시오.

- 레이블 보안(버전 12.1 이상)

자세한 내용은 [Oracle 레이블 보안 \(p. 801\)](#) 단원을 참조하십시오.

- 로케이터

자세한 내용은 [Oracle Locator \(p. 803\)](#) 단원을 참조하십시오.

- 구체화된 보기
- 멀티미디어

자세한 내용은 [Oracle Multimedia \(p. 805\)](#) 단원을 참조하십시오.

- 네트워크 암호화

자세한 내용은 [Oracle 기본 네트워크 암호화 \(p. 807\)](#) 및 [Oracle 보안 소켓 Layer \(p. 810\)](#) 단원을 참조하십시오.

- 분할
- 공간 및 그래프

자세한 내용은 [Oracle Spatial \(p. 815\)](#) 단원을 참조하십시오.

- 스타 쿼리 최적화
- 스트림 및 고급 대기열
- 요약 관리 – 구체화된 보기 쿼리 다시 쓰기
- 텍스트(파일 및 URL 데이터 스토어 유형은 지원되지 않음)
- 토탈 리콜
- TDE(Transparent Data Encryption)

자세한 내용은 [Oracle Transparent Data Encryption \(p. 827\)](#) 단원을 참조하십시오.

- 통합 감사, 혼합 모드(버전 12.2 이상)

자세한 내용은 Oracle 설명서의 [혼합 모드 감사](#)를 참조하십시오.

- XML DB(XML DB Protocol Server 사용 안 함)

자세한 내용은 [Oracle XML DB \(p. 831\)](#) 단원을 참조하십시오.

- 가상 프라이빗 데이터베이스

Amazon RDS Oracle은 다음 Oracle 데이터베이스 기능을 지원하지 않습니다.

- Automatic Storage Management(ASM)
- 데이터베이스 볼트
- 플래시백 데이터베이스
- 멀티테넌트
- Oracle Enterprise Manager Cloud Control Management Repository

- 실제 애플리케이션 클러스터(Oracle RAC)
- 실제 애플리케이션 테스트
- 통합 감사, 순수 모드

Oracle 데이터베이스 파라미터 지원

Amazon RDS에서는 파라미터 그룹을 사용하여 파라미터를 관리합니다. 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오. 특정 Oracle 에디션 및 버전에 지원되는 파라미터를 보기 위해 AWS CLI `describe-engine-default-parameters` 명령을 실행할 수 있습니다.

예를 들어 Oracle Enterprise Edition 버전 12.2에 지원되는 파라미터를 보려면 다음 명령을 실행합니다.

```
aws rds describe-engine-default-parameters --db-parameter-group-family oracle-ee-12.2
```

Oracle 엔진 버전 관리

DB 엔진 버전 관리는 DB 인스턴스를 실행하는 데이터베이스 엔진 소프트웨어가 패치 및 업그레이드되는 시기와 방식을 제어할 수 있게 해주는 Amazon RDS의 기능입니다. 이 기능은 데이터베이스 엔진 패치 버전과의 호환성을 유지하여 프로덕션에 배포하기 전에 애플리케이션 관련 작업을 효과적으로 수행하고 사용자 고유 시간 및 시간대에 버전 업그레이드를 수행할 수 있도록 하는 유연성을 제공합니다.

Note

Amazon RDS는 Amazon RDS 관련 DB 엔진 버전을 사용하여 공식 Oracle 데이터베이스 패치를 정기적으로 수집합니다. Amazon RDS Oracle 관련 엔진 버전에 포함되는 Oracle 패치의 목록을 보려면 [Oracle 데이터베이스 엔진 출시 정보 \(p. 902\)](#) 단원을 참조하십시오.

현재 모든 Oracle 데이터베이스 업그레이드는 수동으로 실행합니다. Oracle DB 인스턴스 업그레이드에 대한 자세한 내용은 [Oracle DB 엔진 업그레이드 \(p. 752\)](#) 단원을 참조하십시오.

Oracle 11.2.0.2 운영 중단

Oracle 버전 11.2.0.2에 대한 Amazon RDS 지원은 2017년에 만료됩니다. Oracle은 이 버전에 대한 패치를 더 이상 제공하지 않습니다. 따라서 AWS 고객에게 최상의 환경을 제공하기 위해 이 버전을 만료할 예정입니다.

Oracle 버전 11.2.0.2를 실행하는 프로덕션 DB 인스턴스가 더 이상 없습니다. 11.2.0.2 DB 인스턴스의 스냅샷이 아직 남아 있을 수 있습니다.

Oracle 버전 11.2.0.2에 대한 Amazon RDS의 지원은 다음 일정에 따라 만료됩니다.

날짜	정보
2016년 8월 4일	Oracle 버전 11.2.0.2를 사용하는 DB 인스턴스를 더 이상 생성할 수 없습니다.
2019년 12월 18일	모든 11.2.0.2 스냅샷을 11.2.0.4로 업그레이드합니다. 이 날짜까지는 스냅샷을 직접 업그레이드할 수 있습니다. 자세한 내용은 Oracle DB 스냅샷 업그레이드 (p. 756) 단원을 참조하십시오.

Oracle 11.2.0.3 운영 중단

Oracle 버전 11.2.0.3에 대한 Amazon RDS 지원은 2017년에 만료됩니다. Oracle은 이 버전에 대한 패치를 더 이상 제공하지 않습니다. 따라서 AWS 고객에게 최상의 환경을 제공하기 위해 이 버전을 만료할 예정입니다.

Oracle 버전 11.2.0.3를 실행하는 프로덕션 DB 인스턴스가 더 이상 없습니다. 11.2.0.3 DB 인스턴스의 스냅샷이 아직 남아 있을 수 있습니다.

Oracle 버전 11.2.0.3에 대한 Amazon RDS의 지원은 다음 일정에 따라 만료됩니다.

날짜	정보
2016년 8월 4일	Oracle 버전 11.2.0.3을 사용하는 DB 인스턴스를 더 이상 생성할 수 없습니다.
2019년 12월 18일	모든 11.2.0.3 스냅샷을 11.2.0.4로 업그레이드합니다. 이 날짜까지는 스냅샷을 직접 업그레이드할 수 있습니다. 자세한 내용은 Oracle DB 스냅샷 업그레이드 (p. 756) 단원을 참조하십시오.

Oracle 12.1.0.1 운영 중단

Oracle 버전 12.1.0.1에 대한 Amazon RDS 지원은 2017년에 만료됩니다. Oracle은 이 버전에 대한 패치를 더 이상 제공하지 않습니다. 따라서 AWS 고객에게 최상의 환경을 제공하기 위해 이 버전을 만료할 예정입니다.

Oracle 버전 12.1.0.1을 실행하는 프로덕션 DB 인스턴스가 더 이상 없습니다. 12.1.0.1 DB 인스턴스의 스냅샷이 아직 남아 있을 수 있습니다.

Oracle 버전 12.1.0.1에 대한 Amazon RDS의 지원은 다음 일정에 따라 만료됩니다.

날짜	정보
2017년 2월 15일	Oracle 버전 12.1.0.1을 사용하는 DB 인스턴스를 더 이상 생성할 수 없습니다.
2019년 12월 18일	모든 12.1.0.1 스냅샷을 12.1.0.2로 업그레이드합니다. 이 날짜까지는 스냅샷을 직접 업그레이드할 수 있습니다. 자세한 내용은 Oracle DB 스냅샷 업그레이드 (p. 756) 단원을 참조하십시오.

Oracle DB 인스턴스에 방대한 페이지 사용

Oracle용 Amazon RDS는 Linux 커널 방대한 페이지를 지원해 데이터베이스 확장성을 높입니다. 방대한 페이지를 사용하면 페이지 표가 작아지고 메모리 관리에 사용되는 CPU 시간이 줄어 대용량 데이터베이스 인스턴스의 성능이 높아집니다. 자세한 내용은 Oracle 문서의 [Overview of HugePages](#) 단원을 참조하십시오.

방대한 페이지는 다음과 같은 Oracle 버전에 사용할 수 있습니다.

- 12.2.0.1, 모든 버전
- 12.1.0.2, 모든 버전

- 11.2.0.4, 모든 버전

`use_large_pages` 파라미터는 DB 인스턴스에서 방대한 페이지의 활성화 여부를 제어합니다. 이 파라미터는 `ONLY`, `FALSE` 또는 `{DBInstanceClassHugePagesDefault}`로 설정할 수 있습니다. Oracle용 기본 DB 파라미터 그룹에서는 `use_large_pages` 파라미터가 `{DBInstanceClassHugePagesDefault}`로 설정됩니다.

DB 인스턴스에서 방대한 페이지의 자동 활성화 여부를 제어하기 위해 파라미터 그룹에서 `DBInstanceClassHugePagesDefault` 수식 변수를 사용할 수 있습니다. 값은 다음과 같이 결정합니다.

- 아래 표에서 언급한 DB 인스턴스 클래스의 경우, `DBInstanceClassHugePagesDefault`는 항상 `FALSE`가 기본값이며 `use_large_pages`는 `FALSE`로 평가합니다. 이때는 DB 인스턴스 클래스의 메모리 크기가 14GiB 이상이라면 이러한 DB 인스턴스 클래스의 방대한 페이지를 수동으로 활성화할 수 있습니다.
- 아래 표에서 언급하지 않은 DB 인스턴스 클래스의 경우, DB 인스턴스 클래스의 메모리가 14GiB 미만이면 `DBInstanceClassHugePagesDefault`는 항상 `FALSE`로 평가하고 `use_large_pages`는 `FALSE`로 평가합니다.
- 아래 표에서 언급하지 않은 DB 인스턴스 클래스의 경우, 인스턴스 클래스의 메모리가 14GiB 이상 100GiB 미만이면 `DBInstanceClassHugePagesDefault`의 기본값이 `TRUE`이고 `use_large_pages`는 `ONLY`로 평가합니다. `use_large_pages`를 `FALSE`로 설정하면 방대한 페이지를 수동으로 비활성화할 수 있습니다.
- 아래 표에서 언급하지 않은 DB 인스턴스 클래스의 경우, 인스턴스 클래스의 메모리가 100GiB 이상이면 `DBInstanceClassHugePagesDefault`는 항상 `TRUE`로 평가하고 `use_large_pages`는 `ONLY`로 평가합니다.

다음 DB 인스턴스 클래스의 경우, 방대한 페이지가 기본적으로 활성화되지 않습니다.

DB 인스턴스 클래스 패밀리	방대한 페이지가 기본적으로 활성화되지 않는 DB 인스턴스 클래스
db.m5	db.m5.large
db.m4	db.m4.large, db.m4.xlarge, db.m4.2xlarge, db.m4.4xlarge, db.m4.10xlarge
db.m3	db.m3.medium, db.m3.large, db.m3.xlarge, db.m3.2xlarge
db.r3	db.r3.large, db.r3.xlarge, db.r3.2xlarge, db.r3.4xlarge, db.r3.8xlarge
db.t3	db.t3.micro, db.t3.small, db.t3.medium, db.t3.large
db.t2	db.t2.micro, db.t2.small, db.t2.medium, db.t2.large

DB 인스턴스 클래스에 대한 자세한 내용은 [사용 가능한 모든 DB 인스턴스 클래스의 하드웨어 사양 \(p. 79\)](#) 단원을 참조하십시오.

신규 또는 기존 DB 인스턴스에 방대한 페이지를 수동으로 활성화하려면 `use_large_pages` 파라미터를 `ONLY`로 설정합니다. Oracle Automatic Memory Management(AMM)에 방대한 페이지를 사용할 수 없습니다. 파라미터 `use_large_pages`를 `ONLY`로 설정하면 `memory_target` 및 `memory_max_target`도 0으로 설정해야 합니다. DB 인스턴스를 위한 DB 파라미터를 설정하는 자세한 방법은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

`sga_target`, `sga_max_size`, `pga_aggregate_target` 파라미터도 설정할 수 있습니다. 시스템 글로벌 영역(SGA)과 프로그램 글로벌 영역(PGA) 메모리 파라미터를 설정할 때 값을 모두 더합니다. 사용 가능한 인스턴스 메모리(`DBInstanceClassMemory`)에서 이 합산 값을 빼면 방대한 페이지 할당 후 가용 메모리를 알 수 있습니다. 최소 2GiB 또는 총 가용 인스턴스 메모리의 10퍼센트 중에서 적은 용량을 비워두어야 합니다.

파라미터를 구성한 후 DB 인스턴스를 재부팅해야 변경 사항이 적용됩니다. 자세한 내용은 [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#) 단원을 참조하십시오.

다음은 방대한 페이지를 수동으로 활성화할 수 있는 샘플 파라미터 구성입니다. 사용자의 필요에 맞게 값을 설정해야 합니다.

```
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
```

파라미터 그룹에서 파라미터 값이 다음과 같이 설정되어 있다고 가정합니다.

```
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}
```

위의 파라미터 그룹은 db.r4 DB 인스턴스 클래스의 메모리 크기가 100GiB 미만이고, db.r3 인스턴스의 메모리 크기가 100GiB 이상일 때 사용됩니다. 이렇게 파라미터를 설정하고 `use_large_pages`가 `{DBInstanceClassHugePagesDefault}`로 설정되면 db.r4 인스턴스에서 방대한 페이지가 활성화되지만 db.r3 인스턴스에서는 비활성화됩니다.

또 다른 예제에서는 파라미터 그룹의 파라미터 값이 다음과 같이 설정되어 있다고 가정합니다.

```
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
```

위의 파라미터 그룹은 db.r4 DB 인스턴스 클래스의 메모리 크기가 100GiB 미만이고, db.r3 인스턴스의 메모리 크기가 100GiB 이상일 때 사용됩니다. 이러한 파라미터 설정에서는 db.r4 인스턴스와 db.r3 인스턴스 모두에서 방대한 페이지가 비활성화됩니다.

Note

위의 파라미터 그룹이 메모리 크기가 100GiB 이상인 db.r4 DB 인스턴스 클래스에서 사용될 경우에는 `use_large_pages`의 `FALSE` 설정이 `ONLY` 설정으로 재정의됩니다. 이때는 재정의에 대한 알림 메시지가 고객에게 전송됩니다.

DB 인스턴스에 방대한 페이지가 활성화된 후 향상된 모니터링을 활성화하여 방대한 페이지 정보를 볼 수 있습니다. 자세한 내용은 [확장 모니터링 \(p. 238\)](#) 단원을 참조하십시오.

Oracle DB 인스턴스에 utl_http, utl_tcp 및 utl_smtp 사용

Amazon RDS는 Oracle을 실행하는 DB 인스턴스에서 아웃바운드 네트워크 액세스를 지원합니다. utl_http, utl_tcp 및 utl_smtp를 사용하여 DB 인스턴스에서 네트워크에 연결할 수 있습니다.

아웃바운드 네트워크 액세스 작업과 관련해 다음 사항에 유의하십시오.

- Oracle 11g를 실행하는 DB 인스턴스에서 utl_http를 사용하려면 XMLDB 옵션을 설치해야 합니다. 자세한 내용은 [Oracle XML DB \(p. 831\)](#) 단원을 참조하십시오.
- utl_http, utl_tcp 및 utl_smtp를 이용한 아웃바운드 네트워크 액세스는 VPC의 Oracle DB 인스턴스에 대해서만 지원됩니다. DB 인스턴스가 VPC에 있는지 확인하는 방법에 대해서는 [EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 \(p. 407\)](#) 단원을 참조하십시오. VPC에 있지 않은 DB 인스턴스를 VPC로 이동하는 방법에 대해서는 [VPC에 있지 않은 DB 인스턴스를 VPC로 이동 \(p. 420\)](#) 단원을 참조하십시오.
- SMTP를 UTL_MAIL 옵션으로 사용하려면 [Oracle UTL_MAIL \(p. 828\)](#) 단원을 참조하십시오.
- 원격 호스트의 DNS(Domain Name Server) 이름은 다음 중 어느 것이나 될 수 있습니다.
 - 공개적으로 확인할 수 있어야 함.
 - Amazon RDS DB 인스턴스의 엔드포인트.
- 사용자 지정 DNS 서버를 통해 확인 가능 자세한 내용은 [사용자 지정 DNS 서버 설정 \(p. 844\)](#) 단원을 참조하십시오.
- 동일한 VPC 또는 피어링된 VPC에 있는 Amazon EC2 인스턴스의 프라이빗 DNS 이름. 이 경우, 사용자 지정 DNS 서버를 통해 이름을 확인할 수 있어야 합니다. 또는 Amazon이 제공하는 DNS를 사용하기 위해 VPC 설정에서 enableDnsSupport 속성을 활성화하고 VPC 피어링 연결에 대한 DNS 확인 지원을 활성화할 수 있습니다. 자세한 내용은 [VPC에서 DNS 지원 및 VPC 피어링 연결 수정](#) 단원을 참조하십시오.

원격 SSL/TLS 리소스에 안전하게 연결하기 위해 사용자 지정 Oracle Wallet을 생성하고 업로드할 수 있습니다. Oracle 기능에 Amazon S3와 Amazon RDS의 통합을 사용하여 Oracle DB 인스턴스로 Amazon S3의 Wallet을 다운로드할 수 있습니다. Oracle에 대한 Amazon S3 통합에 대한 자세한 내용은 [Amazon S3통합 \(p. 773\)](#) 단원을 참조하십시오.

다음 예시는 utl_http를 통해 <https://status.aws.amazon.com/robots.txt>에 액세스하기 위한 Wallet을 생성합니다.

1. [Amazon Trust Services 리포지토리](#)에서 Amazon Root CA 1 인증서를 가져옵니다.
2. 새 Wallet을 생성하고 다음 인증서를 추가하십시오.

```
orapki wallet create -wallet . -auto_login_only
orapki wallet add -wallet . -trusted_cert -cert AmazonRootCA1.pem.pem -
auto_login_only orapki wallet display -wallet .
```

3. Amazon S3 버킷에 Wallet을 업로드합니다.
4. Oracle과 Amazon S3 통합을 위한 사전 조건을 만족시키고 S3_INTEGRATION 옵션을 Oracle DB 인스턴스에 추가하십시오. 옵션의 IAM 역할에 사용 중인 Amazon S3 버킷에 대한 액세스 권한이 있어야 합니다.

자세한 내용은 [Amazon S3통합 \(p. 773\)](#) 단원을 참조하십시오.

5. DB 인스턴스에 연결하고 데이터베이스에 Wallet을 보관할 디렉터리를 만듭니다. 다음은 SSL_WALLET이라는 이름의 디렉터리를 생성하는 예제입니다.

```
exec rdsadmin.rdsadmin_util.create_directory('SSL_WALLET');
```

6. Amazon S3 버킷에서 Oracle DB 인스턴스로 Wallet을 다운로드하십시오.

다음 예에서는 DB 인스턴스 디렉터리 SSL_WALLET으로 Wallet을 다운로드합니다.

```
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;
```

*bucket_name*을 현재 사용 중인 버킷의 이름으로 바꾸고 *wallet_name*을 Wallet의 이름으로 대체하십시오.

7. 다음 절차를 실행하여 utl_http 트랜잭션에 대해 이 Wallet을 설정하십시오.

```
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. SSL/TLS를 통해 위의 URL에 액세스하십시오.

```
SELECT utl_http.request('https://status.aws.amazon.com/robots.txt') AS ROBOTS_TXT FROM  
DUAL;  
  
ROBOTS_TXT  
-----  
User-agent: *  
Allow: /
```

Note

wallet에 필요한 특정 인증서는 서비스별로 다릅니다. AWS 서비스의 경우 일반적으로 [Amazon Trust Services 리포지토리](#)에서 인증서를 확인할 수 있습니다.

유사한 설정을 사용하여 SSL/TLS([Amazon Simple Email Service](#) 포함)를 통해 UTL_SMTP를 통해 이메일을 보낼 수 있습니다.

각 인스턴스에 Oracle SSL 옵션이 구성된 경우 SSL/TLS 엔드포인트를 통해 Oracle DB 인스턴스 사이에 데이터베이스 링크를 설정할 수 있습니다. 추가 구성이 필요하지 않습니다. 자세한 내용은 [Oracle 보안 소켓 Layer \(p. 810\)](#) 단원을 참조하십시오.

OEM, APEX, TDE 및 기타 옵션 사용

대부분의 Amazon RDS DB 엔진은 DB 인스턴스 관련 추가 기능을 선택할 수 있도록 옵션 그룹을 지원합니다. Oracle DB 인스턴스는 Oracle Enterprise Manager(OEM), Transparent Data Encryption(TDE), Application Express(APEX), Native Network Encryption 등 여러 옵션을 지원합니다. 지원되는 전체 Oracle 옵션 목록은 [Oracle DB 인스턴스 옵션 \(p. 772\)](#) 단원을 참조하십시오. 옵션 그룹 작업에 대한 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오.

확장 데이터 형식 사용

Amazon RDS Oracle 버전 12c는 확장 데이터 유형을 지원합니다. 확장 데이터 유형의 경우 VARCHAR2, NVARCHAR2 및 RAW 데이터 유형의 최대 크기는 32,767바이트입니다. 확장 데이터 유형을 사용하려면 MAX_STRING_SIZE 파라미터를 EXTENDED로 설정하십시오. 자세한 내용은 Oracle 설명서의 [확장 데이터 유형](#) 단원을 참조하십시오.

확장 데이터 유형을 사용하지 않을 경우 MAX_STRING_SIZE 파라미터를 STANDARD(기본값)로 유지하십시오. 이 파라미터가 STANDARD로 설정되어 있으면 VARCHAR2 및 NVARCHAR2의 크기 제한은 4,000바이트이며 RAW 데이터 유형은 2,000바이트입니다.

새 또는 기존 DB 인스턴스에 대해 확장 데이터 유형을 활성화할 수 있습니다. 새 DB 인스턴스의 경우 확장 데이터 형식을 활성화하면 일반적으로 DB 인스턴스 생성 시간이 길어집니다. 기존 DB 인스턴스의 경우 변환 과정 중 DB 인스턴스를 사용할 수 없습니다.

확장 데이터 유형이 활성화된 DB 인스턴스에 대한 고려 사항은 다음과 같습니다.

- DB 인스턴스에 대해 확장 데이터 형식을 활성화하면 데이터 유형에 표준 크기를 사용하도록 DB 인스턴스를 변경할 수 없습니다. 확장 데이터 유형을 사용하도록 DB 인스턴스를 변환한 후 MAX_STRING_SIZE 파라미터를 STANDARD로 다시 설정한 경우에는 incompatible-parameters 상태가 됩니다.
- 확장 데이터 유형을 사용하는 DB 인스턴스를 복원하면 MAX_STRING_SIZE 파라미터를 EXTENDED로 설정한 상태에서 파라미터 그룹을 지정해야 합니다. 복원 과정에서 MAX_STRING_SIZE를 STANDARD로 설정한 상태에서 기본값 파라미터 그룹 또는 기타 파라미터 그룹을 지정한 경우에는 incompatible-parameters 상태가 됩니다.
- t2.micro DB 인스턴스 클래스에서 실행하는 Oracle DB 인스턴스의 경우에는 확장 데이터 유형을 활성화하지 않는 것이 좋습니다.

MAX_STRING_SIZE 설정 때문에 DB 인스턴스 상태가 incompatible-parameters가 되면 MAX_STRING_SIZE 파라미터를 EXTENDED로 설정하고 DB 인스턴스를 재부팅해야만 DB 인스턴스를 사용 가능합니다.

새 DB 인스턴스에 대한 확장 데이터 유형 활성화

새 DB 인스턴스에 대해 확장 데이터 유형을 활성화하려면

1. 파라미터 그룹의 MAX_STRING_SIZE 파라미터를 EXTENDED로 설정하십시오.

파라미터를 설정하려면 새 파라미터 그룹을 생성하거나 기존 DB 파라미터 그룹을 수정하면 됩니다.

자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

2. 새 Amazon RDS Oracle DB 인스턴스를 생성하고 MAX_STRING_SIZE를 EXTENDED로 설정한 파라미터 그룹을 DB 인스턴스와 연결하십시오.

자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.

기존 DB 인스턴스에 대한 확장 데이터 유형 활성화

확장 데이터 유형을 활성화하도록 DB 인스턴스를 수정하면 데이터베이스의 데이터가 확장 크기를 사용하도록 변환됩니다. 변환 과정 중에는 DB 인스턴스를 사용할 수 없습니다. 데이터를 변환하는 데 걸리는 시간은 DB 인스턴스에서 사용하는 DB 인스턴스 클래스와 데이터베이스 크기에 따라 다릅니다.

Note

활성 데이터 유형을 활성화한 후에는 변환 과정 중 시간인 특정 시점으로 복원을 수행할 수 없습니다. 변환 직전 또는 변환 직후 시간으로는 복원할 수 있습니다.

기존 DB 인스턴스에 대해 확장 데이터 유형을 활성화하려면

- 데이터베이스의 스냅샷을 만드십시오.

데이터베이스에 잘못된 객체가 있는 경우 Amazon RDS는 이를 객체를 다시 컴파일하려고 합니다. Amazon RDS가 잘못된 객체를 다시 컴파일할 수 없는 경우 확장 데이터 유형으로의 변환은 실패할 수 있습니다. 변환에 문제가 있는 경우 스냅샷을 사용하면 데이터베이스를 복원할 수 있습니다. 변환하기 전에 항상 잘못된 객체가 있는지 점검하고 잘못된 객체를 수정하거나 제거하십시오. 프로덕션 데이터베이스의 경우 먼저 DB 인스턴스의 복사본에서 변환 프로세스를 테스트하는 것이 좋습니다.

자세한 내용은 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.

- 파라미터 그룹의 `MAX_STRING_SIZE` 파라미터를 `EXTENDED`로 설정하십시오.

파라미터를 설정하려면 새 파라미터 그룹을 생성하거나 기존 DB 파라미터 그룹을 수정하면 됩니다.

자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

- `MAX_STRING_SIZE`를 `EXTENDED`로 설정한 파라미터 그룹과 연결하도록 DB 인스턴스를 수정하십시오.

자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

- 파라미터 변경 사항을 적용하려면 DB 인스턴스를 재부팅하십시오.

자세한 내용은 [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#) 단원을 참조하십시오.

공개 동의어

자체 스키마에서 객체를 참조하는 공개 동의어를 생성할 수 있습니다. `SYS`, `SYSTEM` 및 `RDSADMIN`을 포함하여 Oracle에서 유지하는 스키마에 대한 공개 동의어를 생성하거나 수정하지 마십시오. 그러면 핵심 데이터베이스 구성 요소가 무효화되고 DB 인스턴스의 가용성에 영향을 미칠 수 있습니다.

Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성

Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. Oracle 데이터베이스를 실행하는 환경입니다.

Important

DB 인스턴스를 생성하거나 DB 인스턴스에 연결하려면 먼저 [Amazon RDS 설정 \(p. 5\)](#) 섹션의 작업을 완료해야 합니다.

사용자가 샘플 DB 인스턴스를 만들어 연결하는 절차를 실습하는 예제는 [Oracle DB 인스턴스를 만들고 Oracle DB 인스턴스의 데이터베이스에 연결 \(p. 33\)](#) 단원을 참조하십시오.

AWS Management 콘솔

Oracle DB 인스턴스를 실행하는 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. AWS Management 콘솔의 오른쪽 위 모서리에서 DB 인스턴스를 만들 AWS 리전을 선택합니다.
3. 탐색 창에서 데이터베이스를 선택합니다.

탐색 창이 닫혀 있는 경우 왼쪽 상단의 메뉴 아이콘을 선택하여 여십시오.

4. 데이터베이스 생성을 선택하여 엔진 선택 페이지를 여십시오.

사용 가능한 Oracle 에디션은 AWS 리전마다 다릅니다.

Select engine

Engine options

Amazon Aurora

Amazon
Aurora

MySQL



MariaDB



PostgreSQL



Oracle

ORACLE®

Microsoft SQL Server



Oracle

Edition

Oracle Enterprise Edition

Efficient, reliable, and secure database management system that delivers comprehensive high-end capabilities for mission-critical applications and demanding database workloads.

Oracle Standard Edition

Affordable and full-featured database management system supporting up to 32 vCPUs.

Oracle Standard Edition One

Affordable and full-featured database management system supporting up to 16 vCPUs.

Oracle Standard Edition Two

Affordable and full-featured database management system supporting up to 16 vCPUs. Oracle Database Standard Edition Two is a replacement for Standard Edition and Standard Edition One.

Only enable options eligible for RDS Free Usage Tier [info](#)

Cancel

Next

5. [Select engine] 창에서 사용할 Oracle DB 엔진에 해당하는 [Select] 버튼을 선택한 후 [Next]를 선택합니다.
6. 다음 단계에서는 생성하려는 DB 인스턴스를 프로덕션에 사용할지 여부를 묻는 메시지가 나타납니다. 사용할 계획이라면 [Production]을 선택합니다. 프로덕션을 선택하면 다음 단계에서 다음 설정이 모두 미리 선택됩니다.
 - 다중 AZ 배포 장애 조치 옵션
 - 프로비저닝된 IOPS 스토리지 옵션
 - 삭제 방지 활성화 옵션
7. 다음을 선택하여 계속 진행합니다. [Specify DB details] 페이지가 나타납니다.

DB 세부 정보 지정 페이지에서 DB 인스턴스 정보를 지정합니다. 각 설정에 대한 자세한 내용은 [Oracle DB 인스턴스 설정 \(p. 732\)](#) 단원을 참조하십시오.

Specify DB details

Instance specifications

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

DB engine [Info](#)
Oracle Database Enterprise Edition

License model [Info](#)

DB engine version [Info](#)

DB instance class [Info](#)

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

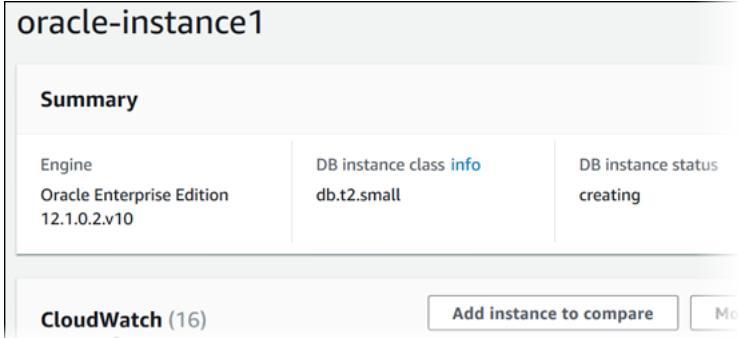
Storage type [Info](#)

8. 다음을 선택하여 계속 진행합니다. [Configure advanced settings] 페이지가 나타납니다.

Configure advanced settings(고급 설정 구성) 페이지에서 RDS가 DB 인스턴스를 시작하는 데 필요한 추가 정보를 입력합니다. 각 설정에 대한 자세한 내용은 [Oracle DB 인스턴스 설정 \(p. 732\)](#) 단원을 참조하십시오.

9. 데이터베이스 생성을 선택합니다.
10. 마지막 페이지에서 DB 인스턴스 세부 정보 보기를 선택합니다.

RDS 콘솔에 새 DB 인스턴스의 세부 정보가 표시됩니다. DB 인스턴스를 만들고 사용할 준비가 될 때까지 DB 인스턴스의 상태는 [creating]입니다. 상태가 available로 변경되면 DB 인스턴스에 연결할 수 있습니다. DB 인스턴스 클래스와 할당된 스토리지에 따라 새 인스턴스를 사용할 수 있을 때까지 몇 분 정도 걸릴 수 있습니다.



CLI

AWS CLI를 사용해 Oracle DB 인스턴스를 생성하려면 다음 파라미터와 함께 [create-db-instance](#) 명령을 호출합니다. 각 설정에 대한 자세한 내용은 [Oracle DB 인스턴스 설정 \(p. 732\)](#) 단원을 참조하십시오.

- `--db-instance-identifier`
- `--db-instance-class`
- `--db-security-groups`
- `--db-subnet-group`
- `--engine`
- `--master-user-name`
- `--master-user-password`
- `--allocated-storage`
- `--backup-retention-period`

Example

다음 명령은 예제 DB 인스턴스를 시작합니다.

Linux, OS X, Unix의 경우:

```
aws rds create-db-instance \
--engine oracle-se1 \
--db-instance-identifier mydbinstance \
--allocated-storage 20 \
--db-instance-class db.m1.small \
--db-security-groups mydbsecuritygroup \
--db-subnet-group mydbsubnetgroup \
--master-username masterawsuser \
--master-user-password masteruserpassword \
--backup-retention-period 3
```

Windows의 경우:

```
aws rds create-db-instance ^
--engine oracle-se1 ^
--db-instance-identifier mydbinstance ^
--allocated-storage 20 ^
--db-instance-class db.m1.small ^
--db-security-groups mydbsecuritygroup ^
--db-subnet-group mydbsubnetgroup ^
--master-username masterawsuser ^
--master-user-password masteruserpassword ^
```

```
--backup-retention-period 3
```

이 명령은 다음과 유사한 출력을 생성합니다.

```
DBINSTANCE mydbinstance db.m1.small oracle-se1 20 sa creating 3 **** n
11.2.0.4.v1
SECGROUP default active
PARAMGRP default.oracle-se1-11.2 in-sync
```

API

Amazon RDS API를 사용해 Oracle DB 인스턴스를 생성하려면 다음 파라미터와 함께 [CreateDBInstance](#) 작업을 호출합니다. 각 설정에 대한 자세한 내용은 [Oracle DB 인스턴스 설정 \(p. 732\)](#) 단원을 참조하십시오.

- AllocatedStorage
- BackupRetentionPeriod
- DBInstanceClass
- DBInstanceIdentifier
- DBSecurityGroups
- DBSubnetGroup
- Engine
- MasterUsername
- MasterUserPassword

Example

```
https://rds.amazonaws.com/
?Action=CreateDBInstance
&AllocatedStorage=250
&BackupRetentionPeriod=3
&DBInstanceClass=db.m1.large
&DBInstanceIdentifier=mydbinstance
&DBSecurityGroups.member.1=mysecuritygroup
&DBSubnetGroup=mydbsubnetgroup
&Engine=oracle-se1
&MasterUserPassword=masteruserpassword
&MasterUsername=masterawsuser
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-10-31
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20140305/us-west-1/rds/aws4_request
&X-Amz-Date=20140305T185838Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=b441901545441d3c7a48f63b5b1522c5b2b37c137500c93c45e209d4b3a064a3
```

Oracle DB 인스턴스 설정

다음 표에는 Oracle DB 인스턴스를 생성할 때 선택하는 설정에 대한 세부 정보가 나와 있습니다.

설정	설정 설명
할당된 스토리지	DB 인스턴스용으로 할당할 스토리지(단위: GB)입니다. 경우에 따라 DB 인스턴스에 대해 데이터베이스의 크기보다 많은 양의 스토리지를 할당하면 I/O 성능을 개선할 수 있습니다.

설정	설정 설명
	자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.
마이너 버전 자동 업그레이드	<p>기본 마이너 DB 엔진 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다. Amazon RDS는 유지 관리 기간에 마이너 버전 업그레이드를 수행합니다.</p> <p>Oracle Locator, Oracle Multimedia, Oracle Spatial 등의 일부 옵션을 사용하려면 마이너 버전 자동 업그레이드를 활성화해야 합니다. 이러한 옵션을 사용하는 DB 인스턴스의 업그레이드는 예약된 유지 관리 기간 중에 설치되며 업그레이드 중에 인스턴스가 중단됩니다. 이러한 옵션을 제거하도록 옵션 그룹을 수정하는 동시에 마이너 버전 자동 업그레이드를 비활성화할 수 없습니다.</p>
[Availability zone]	<p>DB 인스턴스의 가용 영역입니다. 특정 가용 영역을 지정해야 하는 경우가 아니면 기본값 No Preference를 그대로 사용합니다.</p> <p>자세한 내용은 리전 및 가용 영역 (p. 99) 단원을 참조하십시오.</p>
백업 보존 기간	<p>DB 인스턴스의 자동 백업을 보존할 기간(단위: 일)입니다. 중요한 인스턴스의 경우 이 값을 1 이상으로 설정해야 합니다.</p> <p>자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.</p>
백업 기간	<p>Amazon RDS가 자동으로 DB 인스턴스를 백업하는 기간입니다. 데이터베이스를 백업할 특정 시간을 지정하지 않으려면 기본값으로 No Preference를 사용합니다.</p> <p>자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.</p>
Character set name	<p>DB 인스턴스에 대한 문자 세트입니다. 기본값 AL32UTF8은 유니코드 5.0 UTF-8 범용 문자 세트입니다. DB 인스턴스를 만든 후에는 문자 집합을 변경할 수 없습니다.</p> <p>자세한 내용은 Amazon RDS에서 지원되는 Oracle 문자 집합 (p. 770) 단원을 참조하십시오.</p>
스냅샷으로 태그 복사	<p>스냅샷을 생성할 때 DB 인스턴스 태그를 DB 스냅샷에 복사하려면 이 옵션을 선택합니다.</p> <p>자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.</p>
데이터베이스 이름	<p>DB 인스턴스에 있는 데이터베이스의 이름입니다. 이름은 문자로 시작하고 최대 8자의 영숫자를 포함해야 합니다. 데이터베이스 이름으로 문자열 NULL 또는 기타 예약어를 지정할 수 없습니다. 이름을 제공하지 않으면 생성 중인 DB 인스턴스에 Amazon RDS가 데이터베이스를 생성하지 않습니다.</p>
데이터베이스 포트	DB 인스턴스에 액세스하는 데 사용할 포트입니다. Oracle 설치의 기본값은 포트 1521로 설정됩니다.
DB 엔진 버전	사용할 Oracle 버전입니다.

설정	설정 설명
DB 인스턴스 클래스	<p>사용할 DB 인스턴스 클래스.</p> <p>자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 및 Oracle을 위한 DB 인스턴스 클래스 지원 (p. 705) 단원을 참조하십시오.</p>
DB 인스턴스 식별자	<p>DB 인스턴스의 이름입니다. 이 이름은 계정 및 AWS 리전 내에서 고유해야 합니다. 예를 들어, <code>oracle-instance1</code>처럼 선택한 AWS 리전과 DB 엔진을 포함하는 등 알기 쉬운 이름을 지정할 수 있습니다.</p>
DB 파라미터 그룹	<p>DB 인스턴스의 파라미터 그룹입니다. 기본 파라미터 그룹을 사용하거나 사용자 지정 파라미터 그룹을 생성할 수 있습니다.</p> <p>자세한 내용은 DB 파라미터 그룹 작업 (p. 161) 및 Oracle sqlnet.ora 파라미터 수정 (p. 750) 단원을 참조하십시오.</p>
삭제 방지	<p>DB 인스턴스가 삭제되지 않도록 방지하려면, 삭제 방지를 활성화합니다. AWS Management 콘솔을 사용하여 프로덕션 DB 인스턴스를 생성할 경우 기본적으로 삭제 방지가 활성화됩니다. 자세한 내용은 DB 인스턴스 삭제 (p. 128) 단원을 참조하십시오.</p>
암호화	<p>이 DB 인스턴스에 대해 유튜 암호화를 활성화하기 위한 [Enable Encryption].</p> <p>자세한 내용은 Amazon RDS 리소스 암호화 (p. 385) 단원을 참조하십시오.</p>
확장 모니터링	<p>DB 인스턴스가 실행되는 운영 체제에 대한 실시간 측정치를 수집하려면 [Enable enhanced monitoring]을 선택합니다.</p> <p>자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.</p>
라이선스 모델	<p>사용할 라이선스 모델입니다. [license-included]를 선택하여 Oracle 일반 라이선스 계약을 사용합니다. 기존의 Oracle 라이선스를 사용하려면 [bring-your-own-license]를 선택합니다.</p> <p>자세한 내용은 Oracle 라이선싱 (p. 704) 단원을 참조하십시오.</p>
유지 관리 기간	<p>대기 중인 DB 인스턴스 설정 변경이 적용되기 위해 경과해야 하는 기간(30분)입니다. 이 시간이 중요하지 않은 경우 [No Preference]를 선택합니다.</p> <p>자세한 내용은 Amazon RDS 유지 관리 기간 (p. 116) 단원을 참조하십시오.</p>
Master username	<p>모든 데이터베이스 권한을 사용해 DB 인스턴스에 로그인하기 위해 마스터 사용자 이름으로 사용할 이름입니다. 이 사용자 계정은 DB 인스턴스에 로그인하는 데 사용되며 DBA 역할이 부여됩니다.</p> <p>자세한 내용은 Oracle 보안 (p. 708) 단원을 참조하십시오.</p>
Master password	<p>마스터 사용자 계정의 암호입니다. 암호는 8~30자의 인쇄 가능한 ASCII 문자를 포함해야 합니다(/, " 및 @ 제외).</p>

설정	설정 설명
다중 AZ 배포	<p>장애 조치를 위해 다른 가용 영역에 DB 인스턴스의 예비 복제본을 생성하려면 [Create replica in different zone]을 선택합니다. 이때 고가용성을 유지하려면 프로덕션 워크로드를 위한 다른 AZ를 권장합니다. 개발 및 테스팅을 위해 No를 선택할 수 있습니다.</p> <p>자세한 내용은 리전 및 가용 영역 (p. 99) 단원을 참조하십시오.</p>
옵션 그룹	<p>DB 인스턴스의 옵션 그룹입니다. 기본 옵션 그룹을 사용하거나 사용자 지정 옵션 그룹을 생성할 수 있습니다.</p> <p>자세한 내용은 옵션 그룹 작업 (p. 148) 단원을 참조하십시오.</p>
퍼블릭 액세스 가능성	<p>DB 인스턴스에 퍼블릭 IP 주소를 부여하려면 [Yes]를 선택합니다. 그러면 VPC 외부에서 DB 인스턴스에 액세스할 수 있습니다(DB 인스턴스도 VPC의 퍼블릭 서브넷에 있어야 함). VPC 내부에서만 DB 인스턴스에 액세스할 수 있게 하려면 [No]를 선택합니다.</p> <p>자세한 내용은 VPC에 있는 DB 인스턴스를 인터넷에서 숨기기 (p. 416) 단원을 참조하십시오.</p>
스토리지 유형	<p>DB 인스턴스의 스토리지 유형입니다.</p> <p>자세한 내용은 Amazon RDS 스토리지 유형 (p. 101) 단원을 참조하십시오.</p>
Subnet Group	<p>이 설정은 현재 사용 중인 플랫폼에 따라 다릅니다. AWS를 처음 사용하는 고객은 [default]를 선택합니다. 이 DB 서브넷 그룹은 계정에 대해 생성된 기본 DB 서브넷 그룹입니다. 이전 E2-Classic 플랫폼에서 DB 인스턴스를 만들어서 특정 VPC에 배포하려면 해당 VPC로 생성된 DB 서브넷 그룹을 선택합니다.</p>
Virtual Private Cloud(VPC)	<p>이 설정은 현재 사용 중인 플랫폼에 따라 다릅니다. AWS를 처음 사용하는 경우에는 기본 VPC를 선택합니다. 이전 E2-Classic 플랫폼에서 DB 인스턴스를 만드는 경우 [Not in VPC]를 선택합니다.</p> <p>자세한 내용은 Amazon Virtual Private Cloud(VPC) 및 Amazon RDS (p. 406) 단원을 참조하십시오.</p>
VPC 보안 그룹	<p>AWS를 처음 사용하는 경우에는 [Create new VPC security group]을 선택합니다. 처음 사용하는 경우가 아니라면 [Select existing VPC security groups]를 선택하고 앞서 생성한 보안 그룹을 선택합니다.</p> <p>RDS 콘솔에서 새 VPC 보안 그룹 생성을 선택하는 경우 브라우저에서 검색된 IP 주소에서 DB 인스턴스에 액세스하도록 허용하는 발신 규칙을 사용하여 새 보안 그룹이 생성됩니다.</p> <p>자세한 내용은 DB 보안 그룹 작업(EC2-Classic 플랫폼) (p. 394) 단원을 참조하십시오.</p>

관련 주제

- 자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 (p. 421)
- Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 (p. 736)
- Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 (p. 742)
- DB 인스턴스 삭제 (p. 128)

Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 에서 데이터베이스에 연결

Amazon RDS가 Oracle DB 인스턴스를 프로비저닝한 후에는 표준 SQL 클라이언트 애플리케이션을 사용해 DB 인스턴스에 연결할 수 있습니다. 이번 주제에서는 Oracle SQL Developer 또는 SQL*Plus를 사용해 Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에 연결합니다.

사용자가 샘플 DB 인스턴스를 만들어 연결하는 절차를 실습하는 예제는 [Oracle DB 인스턴스를 만들고 Oracle DB 인스턴스의 데이터베이스에 연결 \(p. 33\)](#) 단원을 참조하십시오.

DB 인스턴스의 엔드포인트 찾기

각 Amazon RDS DB 인스턴스에는 엔드포인트가 있으며, 각 엔드포인트에는 DB 인스턴스의 DNS 이름과 포트 번호가 있습니다. SQL 클라이언트 애플리케이션을 사용해 DB 인스턴스에 연결하려면 DB 인스턴스에 연결할 수 있는 DNS 이름과 포트 번호가 필요합니다.

Amazon RDS 콘솔 또는 AWS CLI를 사용하여 DB 인스턴스의 엔드포인트를 찾을 수 있습니다.

AWS Management 콘솔

콘솔을 사용하여 엔드포인트를 찾으려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 콘솔의 오른쪽 상단 모서리에서 DB 인스턴스의 AWS 리전을 선택합니다.
3. DB 인스턴스에 대한 DNS 이름과 포트 번호를 찾습니다.
 - a. DB 인스턴스 목록을 표시할 인스턴스를 선택합니다.
 - b. 인스턴스 세부 정보를 표시할 Oracle DB 인스턴스 이름을 선택합니다.
 - c. 연결 탭에서 엔드포인트를 복사합니다. 또한 포트 번호를 적어둡니다. DB 인스턴스에 연결하려면 엔드포인트와 포트 번호가 모두 필요합니다.

oracle-instance1

Summary

DB Name
oracle-instance1

Role
Instance

Connectivity Monitoring Logs & events Configuration

Connectivity

Endpoint & port

Endpoint
oracle-instance1. [REDACTED].rds.amazonaws.com

Port
1521

CLI

AWS CLI를 사용하여 Oracle DB 인스턴스의 엔드포인트를 찾으려면 [describe-db-instances](#) 명령을 호출하십시오.

Example AWS CLI를 사용하여 엔드포인트를 찾으려면

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

출력에서 Endpoint를 검색하여 DB 인스턴스의 DNS 이름과 포트 번호를 찾습니다. DNS 이름은 출력의 Address 라인에 포함되어 있습니다. 다음은 JSON 엔드포인트 출력의 예입니다.

```
"Endpoint": {  
    "HostedZoneId": "Z1PVIF0B656C1W",  
    "Port": 3306,  
    "Address": "myinstance.123456789012.us-west-2.rds.amazonaws.com"  
},
```

Note

출력에는 여러 DB 인스턴스 정보가 포함될 수 있습니다.

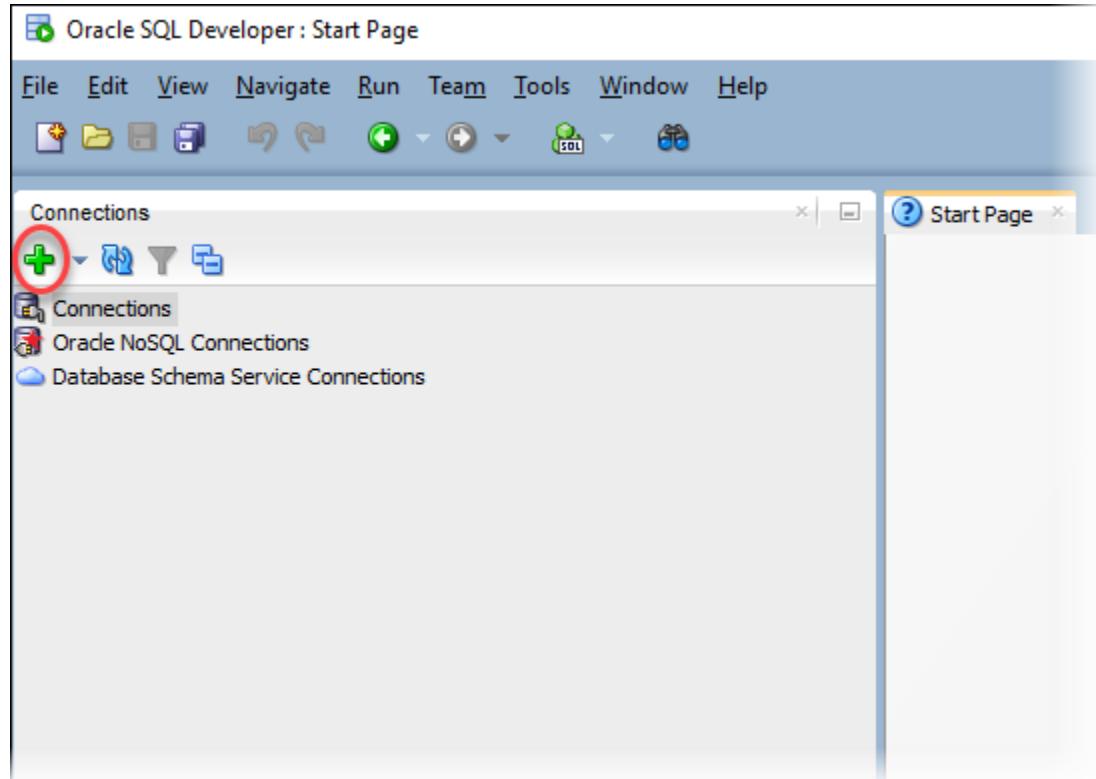
Oracle SQL Developer를 사용하여 DB 인스턴스에 연결

이 절차에서는 Oracle SQL Developer를 사용하여 DB 인스턴스에 연결합니다. 이 유ти리티의 독립 실행형 버전을 다운로드하려면 [Oracle SQL Developer 다운로드 페이지](#)를 참조하십시오.

DB 인스턴스에 연결하려면 인스턴스의 DNS 이름과 포트 번호가 필요합니다. DB 인스턴스의 DNS 이름 및 포트 번호를 찾는 자세한 내용은 [DB 인스턴스의 엔드포인트 찾기 \(p. 736\)](#)를 참조하십시오.

SQL Developer를 사용하여 DB 인스턴스에 연결하려면

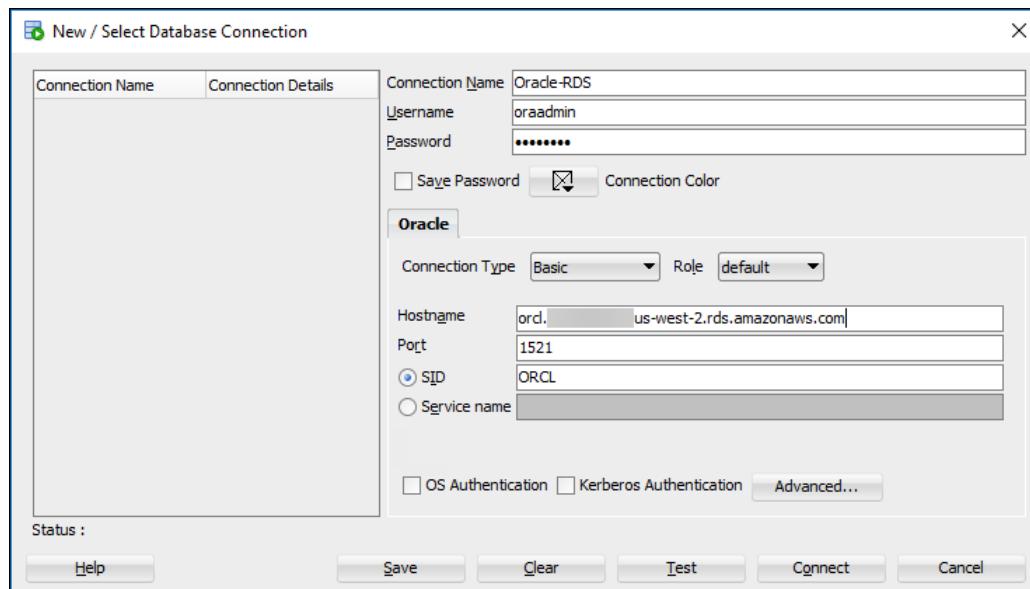
1. Oracle SQL Developer를 시작합니다.
2. [Connections] 탭에서 [add (+)] 아이콘을 선택합니다.



3. [New>Select Database Connection] 대화 상자에 다음과 같이 DB instance 정보를 입력합니다.

- 연결 이름에 연결을 설명하는 이름(예: Oracle-RDS)을 입력합니다.
- 사용자 이름에 DB 인스턴스의 데이터베이스 관리자 이름을 입력합니다.
- 암호에 데이터베이스 관리자 암호를 입력합니다.
- 호스트 이름에 DB 인스턴스의 DNS 이름을 입력하거나 붙여 넣습니다.
- 포트에 포트 번호를 입력합니다.
- SID에 Oracle 데이터베이스 SID를 입력합니다.

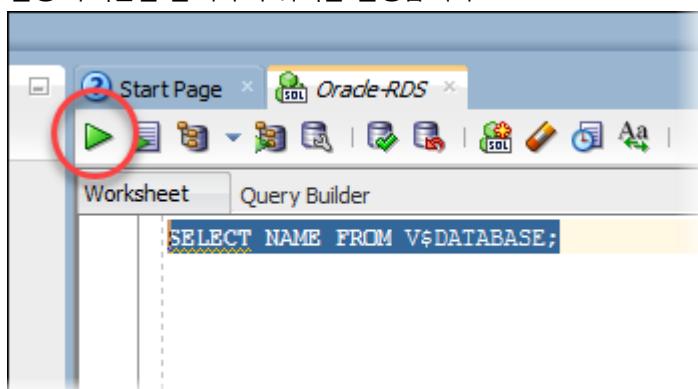
정보가 모두 입력된 대화 상자는 다음과 비슷한 모습이 되어야 합니다.



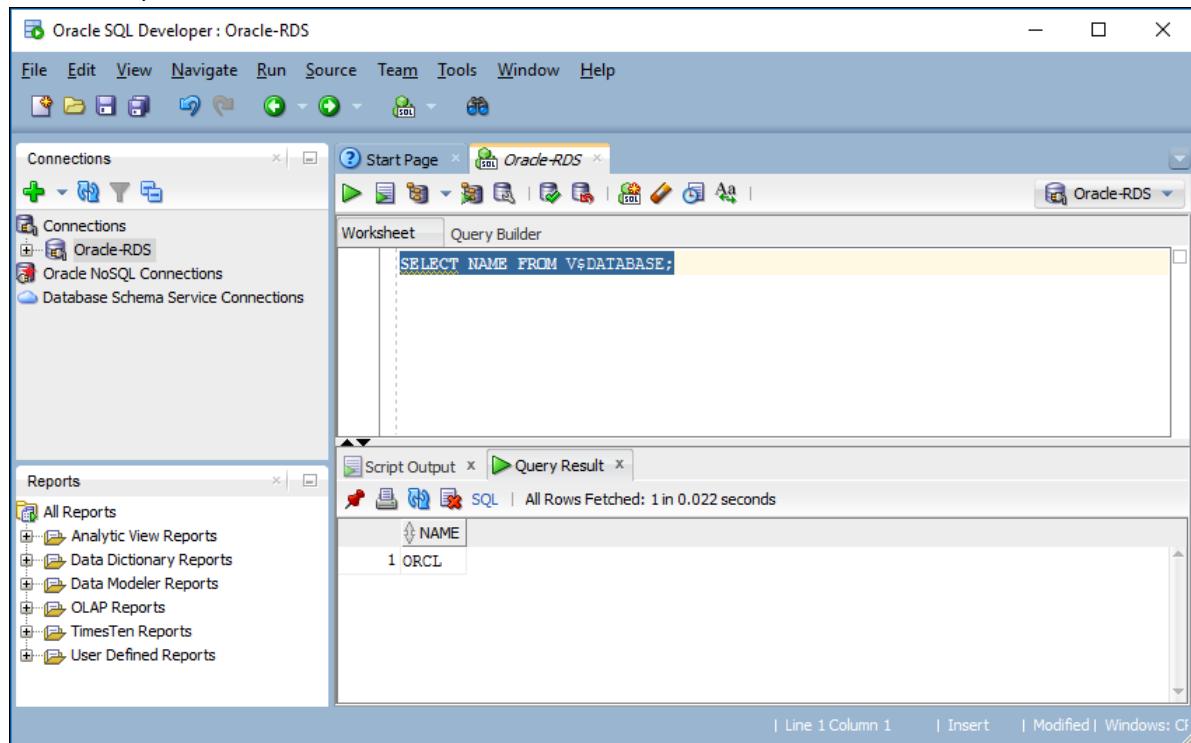
4. [Connect]를 클릭합니다.
5. 이제 자체 데이터베이스 생성을 시작하고 평소대로 DB 인스턴스와 데이터베이스에 대한 쿼리 실행을 시작할 수 있습니다. DB 인스턴스에 대한 테스트 쿼리를 실행하려면 다음 중 하나를 수행합니다.
- 해당 연결의 워크시트 탭에 다음 SQL 쿼리를 입력합니다.

```
SELECT NAME FROM V$DATABASE;
```

- 실행 아이콘을 클릭하여 쿼리를 실행합니다.



SQL Developer가 데이터베이스 이름을 반환합니다.



SQL*Plus를 사용하여 DB 인스턴스에 연결

SQL*Plus 같은 유ти리티를 사용하면 Oracle을 실행하는 Amazon RDS DB 인스턴스에 연결할 수 있습니다. SQL*Plus의 독립 실행형 버전을 다운로드하려면 [SQL*Plus 사용 설명서 및 참조](#)를 참조하십시오.

DB 인스턴스에 연결하려면 인스턴스의 DNS 이름과 포트 번호가 필요합니다. DB 인스턴스의 DNS 이름 및 포트 번호를 찾는 자세한 내용은 [DB 인스턴스의 엔드포인트 찾기 \(p. 736\)](#)을 참조하십시오.

Example SQL*Plus를 사용하여 Oracle DB 인스턴스에 연결하려면

다음 예제에서 DB 인스턴스 관리자의 사용자 이름을 대체합니다. 또한 DB 인스턴스를 DNS 이름으로 대체한 다음, 포트 번호와 Oracle SID를 포함시킵니다. SID 값은 DB 인스턴스의 이름이 아니라, DB 인스턴스를 생성할 때 지정한 DB 인스턴스의 데이터베이스 이름입니다.

Linux, OS X, Unix의 경우:

```
sqlplus 'user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))(CONNECT_DATA=(SID=database_name)))'
```

Windows의 경우:

```
sqlplus user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))(CONNECT_DATA=(SID=database_name)))
```

다음과 유사한 출력 화면이 표시되어야 합니다.

```
SQL*Plus: Release 12.1.0.2.0 Production on Mon Aug 21 09:42:20 2017
```

사용자 암호를 입력하면 SQL 프롬프트가 표시됩니다.

```
SQL>
```

Note

`sqlplus USER/PASSWORD@LONGER-THAN-63-CHARS-RDS-ENDPOINT-HERE:1521/DATABASE_IDENTIFIER`와 같은 짧은 형식의 연결 문자열(Easy Connect 또는 EZCONNECT)이 최대 문자 제한에 걸릴 수 있으며, 이런 문자열을 연결할 때 사용하면 안 됩니다.

보안 그룹 고려 사항

DB 인스턴스에 연결하려면 DB 인스턴스가 DB 인스턴스에 액세스할 때 사용하는 IP 주소와 네트워크 구성이 할당되어 있는 보안 그룹과 연동되어야 합니다. DB 인스턴스를 생성할 때 DB 인스턴스를 적당한 보안 그룹과 연동할 수 있습니다. 하지만 DB 인스턴스를 생성하면서 따로 설정할 필요가 없는 기본 보안 그룹을 할당한 경우에는 DB 인스턴스 방화벽이 연결을 차단합니다.

새로운 보안 그룹을 생성하여 액세스를 활성화해야 하는 경우 DB 인스턴스의 기반이 되는 Amazon EC2 플랫폼에 따라 생성할 보안 그룹 유형이 결정됩니다. 플랫폼을 확인하려면 [EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 \(p. 407\)](#) 단원을 참조하십시오. 일반적으로 DB 인스턴스가 EC2-Classic 플랫폼에 있으면 DB 보안 그룹을 만들고, DB 인스턴스가 VPC 플랫폼에 있으면 VPC 보안 그룹을 만듭니다. 새 보안 그룹 생성에 대한 자세한 내용은 [보안 그룹을 통한 액세스 제어 \(p. 390\)](#) 단원을 참조하십시오.

새 보안 그룹을 생성하였으면 보안 그룹과 연동되도록 DB 인스턴스 설정을 변경합니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

SSL을 사용하여 DB 인스턴스 연결을 암호화함으로써 보안을 강화할 수 있습니다. 자세한 내용은 [Oracle 보안 소켓 Layer \(p. 810\)](#) 단원을 참조하십시오.

전용 및 공유 서버 프로세스

서버 프로세스는 Oracle DB 인스턴스에 대한 사용자 연결을 처리합니다. 기본적으로 Oracle DB 인스턴스는 전용 서버 프로세스를 사용합니다. 전용 서버 프로세스에서는 서버 프로세스 하나로 사용자 프로세스 하나만 처리합니다. 원한다면 공유 서버 프로세스를 구성할 수 있습니다. 공유 서버 프로세스에서는 서버 프로세스 하나로 사용자 프로세스를 여러 개 처리할 수 있습니다.

다수의 사용자 세션에서 서버 메모리를 지나치게 많이 사용한다면 공유 서버 프로세스를 사용해 보십시오. 또한 세션이 빈번하게 연결 및 연결 해제되어 성능 문제가 발생하는 경우에도 공유 서버 프로세스의 사용을 고려해 볼 수 있습니다. 그러나 공유 서버 프로세스 사용에는 단점도 있습니다. 예를 들어, 공유 서버 프로세스는 CPU 리소스를 과도하게 사용할 수 있고 구성 및 관리하기가 훨씬 복잡합니다.

전용 및 공유 서버 프로세스에 대한 자세한 내용은 Oracle 설명서의 [About Dedicated and Shared Server Processes](#)를 참조하십시오. Amazon RDS Oracle DB 인스턴스에서 공유 서버 프로세스를 구성하는 방법에 대한 자세한 내용은 Knowledge Center에서 [공유 서버에서 Amazon RDS for Oracle Database가 작동하도록 구성하려면 어떻게 해야 합니까?](#)를 참조하십시오.

Oracle DB 인스턴스 연결 문제 해결

다음은 Oracle DB 인스턴스 연결을 시도할 때 발생할 수 있는 문제입니다.

문제	문제 해결 제안
DB 인스턴스에 연결할 수 없습니다.	새로 생성한 DB 인스턴스의 경우, DB 인스턴스를 사용할 준비가 될 때까지 DB 인스턴스의 상태는 [creating]입니다. 상태가 available로 변경되면 DB 인스턴스에 연결할 수 있습니다. DB 인스턴스 클래스와 스토리지의 양에 따라 새 DB 인스턴스를 사용할 수 있을 때까지 최대 20분이 걸릴 수 있습니다.
DB 인스턴스에 연결할 수 없습니다.	DB 인스턴스를 만들 때 지정한 포트를 통해 통신을 보내거나 받을 수 없으면 DB 인스턴스에 연결할 수 없습니다. 네트워크 관리자에게 문의해 DB 인스턴스에 대해 지정한 포트가 인바운드 및 아웃바운드 통신을 허용하는지 확인하십시오.
DB 인스턴스에 연결할 수 없습니다.	로컬 방화벽에서 적용되는 액세스 규칙과 DB 인스턴스의 보안 그룹에 있는 DB 인스턴스에 액세스하기 위한 권한을 부여한 IP 주소가 일치하지 않을 수 있습니다. 방화벽의 아웃바운드 또는 인바운드 규칙에 문제가 있을 가능성이 높습니다. 보안 그룹에 대한 자세한 내용은 보안 그룹을 통한 액세스 제어 (p. 390) 단원을 참조하십시오. 보안 그룹의 규칙 설정 절차에 대한 자세한 내용은 자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 (p. 421) 단원을 참조하십시오.
Connect failed because target host or object does not exist – Oracle, Error: ORA-12545(대상 호스트 또는 객체가 존재하지 않으므로 연결이 실패했습니다 - Oracle, 오류: ORA-12545)	서버 이름과 포트 번호를 정확하게 지정했는지 확인하십시오. 서버 이름에는 DNS 이름을 입력하거나 콘솔에서 볼여 넣습니다. DB 인스턴스의 DNS 이름 및 포트 번호를 찾는 자세한 내용은 DB 인스턴스의 앤드포인트 찾기 (p. 736) 를 참조하십시오.
Invalid username/password; logon denied – Oracle, Error: ORA-01017(잘못된 사용자 이름/암호이므로, 로그인이 거부되었습니다 - Oracle, 오류: ORA-01017)	DB 인스턴스에 연결할 수 있지만 연결이 거부되었습니다. 이 문제는 주로 사용자 이름이나 암호를 잘못 입력하면 발생합니다. 사용자 이름과 암호를 확인하고 다시 시도하십시오.

Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경

추가 스토리지를 더하거나 DB 인스턴스 클래스를 변경하는 것과 같은 작업을 완수하기 위해 DB 인스턴스의 설정을 변경할 수 있습니다. 이번 주제에서는 Amazon RDS Oracle DB 인스턴스의 수정 방법과 Oracle 인스턴스의 설정에 대해서 설명하겠습니다. 프로덕션 인스턴스를 수정하기 전에 테스트 인스턴스에서 변경 사항을 테스트하면 각 변경 사항이 미칠 영향을 완전히 이해하는 데 도움이 됩니다. 이러한 사례는 특히 데이터베이스 버전을 업그레이드할 때 중요합니다.

DB 인스턴스 설정을 수정한 후 변경 사항을 즉시 적용할 수도 있고, DB 인스턴스에 대한 다음 유지 관리 기간에 적용할 수도 있습니다. 일부 변경 사항의 경우 DB 인스턴스가 재시작될 수 있습니다.

아래에 서술한 대로 Oracle 인스턴스를 수정하는 일 외에도 이 주제의 마지막 [Oracle sqlnet.ora 파라미터 수정 \(p. 750\)](#)에 설명한 대로 Oracle DB 인스턴스의 sqlnet.ora 파라미터 설정도 변경할 수 있습니다.

AWS Management 콘솔

Oracle DB 인스턴스의 설정을 변경하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 변경하려는 DB 인스턴스를 선택합니다.
3. [Modify]를 선택합니다. [Modify DB Instance] 페이지가 나타납니다.
4. 원하는 설정을 모두 변경합니다. 각 설정에 대한 자세한 내용은 [Oracle DB 인스턴스 설정 \(p. 744\)](#) 단원을 참조하십시오.
5. 원하는 대로 모두 변경되었으면 [Continue]를 선택하고 수정 사항 요약을 확인합니다.
6. 변경 사항을 즉시 적용하려면 즉시 적용을 선택합니다. 일부의 경우 이 옵션을 선택하면 중단이 발생할 수 있습니다. 자세한 내용은 [즉시 적용 파라미터 사용 \(p. 112\)](#) 단원을 참조하십시오.
7. 확인 페이지에서 변경 내용을 검토합니다. 변경 내용이 정확할 경우 [Modify DB Instance]를 선택하여 변경 내용을 저장합니다.

그렇지 않으면 [Back]을 선택하여 변경 내용을 편집하거나 [Cancel]을 선택하여 변경 내용을 취소합니다.

CLI

AWS CLI를 사용해 Oracle DB 인스턴스를 수정하려면 `modify-db-instance` 명령을 호출합니다. DB 인스턴스 식별자와 수정하려는 설정의 파라미터를 지정합니다. 각 파라미터에 대한 자세한 내용은 [Oracle DB 인스턴스 설정 \(p. 744\)](#) 단원을 참조하십시오.

Example

다음 코드는 백업 보존 기간을 1주(7일)로 설정하여 `mydbinstance`를 수정합니다. 이 코드는 `--auto-minor-version-upgrade`를 사용하여 마이너 버전 자동 업그레이드를 활성화합니다. 마이너 버전 자동 업그레이드를 비활성화하려면 `--no-auto-minor-version-upgrade`를 사용합니다. 변경 사항은 `--no-apply-immediately`를 사용하여 다음 유지 관리 기간에 적용됩니다. 변경 사항을 바로 적용하려면 `--apply-immediately`를 사용합니다. 자세한 내용은 [즉시 적용 파라미터 사용 \(p. 112\)](#) 단원을 참조하십시오.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--backup-retention-period 7 \
--auto-minor-version-upgrade \
--no-apply-immediately
```

Windows의 경우:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--backup-retention-period 7 ^
--auto-minor-version-upgrade ^
--no-apply-immediately
```

API

Amazon RDS API를 사용해 Oracle DB 인스턴스를 수정하려면 `ModifyDBInstance` 작업을 호출합니다. DB 인스턴스 식별자와 수정하려는 설정의 파라미터를 지정합니다. 각 파라미터에 대한 자세한 내용은 [Oracle DB 인스턴스 설정 \(p. 744\)](#) 단원을 참조하십시오.

Example

다음은 백업 보존 기간을 1주일(7일)로 설정하고 마이너 버전 자동 업그레이드를 활성화하여 mydbinstance를 수정하는 코드입니다. 이 변경 사항은 다음 유지 관리 기간에 적용됩니다.

```
https://rds.amazonaws.com/
?Action=ModifyDBInstance
&ApplyImmediately=false
&AutoMinorVersionUpgrade=true
&BackupRetentionPeriod=7
&DBInstanceIdentifier=mydbinstance
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-10-31
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20131016/us-west-1/rds/aws4_request
&X-Amz-Date=20131016T233051Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=087a8eb41cb1ab0fc9ec1575f23e73757ffc6a1e42d7d2b30b9cc0be988cff97
```

Oracle DB 인스턴스 설정

다음 표에는 수정이 가능한 설정, 수정이 불가능한 설정, 변경 사항을 적용할 수 있는 시점, 변경 사항으로 인한 DB 인스턴스 가동 중지 여부에 대한 자세한 내용이 나와 있습니다.

설정	설정 설명	변경이 발생할 때	가동 중지 참고 사항
할당된 스토리지	<p>DB 인스턴스에 할당할 스토리지(단위: GB). 할당된 스토리지를 늘릴 수만 있고 줄일 수는 없습니다.</p> <p>DB 인스턴스 상태가 스토리지 최적화이거나 지난 6시간 동안 DB 인스턴스에 할당된 스토리지가 수정된 경우 할당된 스토리지를 수정할 수 없습니다.</p> <p>허용 가능한 최대 스토리지는 스토리지 유형에 따라 다릅니다. 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.</p>	[Apply immediately]가 true로 설정되는 즉시 변경됩니다. [Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.	가동 중지 없음. 변경 도중 성능이 저하될 수 있습니다.
마이너 버전 자동 업그레이드	기본 마이너 DB 엔진 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다. Amazon RDS는 유지 관리 기간에 마이너 버전 자동 업그레이드를 수행합니다.	-	-
백업 보존 기간	<p>자동 백업을 보존할 일수. 자동 백업을 비활성화하기 위해 백업 보존 기간을 0으로 설정합니다.</p> <p>자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되고 이 설정을 0이 아닌 값에서 다른 0이 아닌 값으로 변경하</p>	백업 보존 기간을 0에서 0이 아닌 값으로 또는 0이 아닌 값에서 0으로 변경할 경우 인스턴스가 종단됩니다.

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
		면 비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이때 적용되지 않을 경우, 다음 유지 관리 기간에 변경 사항이 적용됩니다.	
백업 기간	데이터베이스의 자동 백업이 실행되는 기간. 백업 기간은 국제 표준시(UTC)의 시작 시간과 시간 단위의 지속 기간으로 구성됩니다. 자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다.	-
인증 기관	사용할 인증서.	-	-
스냅샷으로 태그 복사	DB 인스턴스 태그가 있는 경우 이 옵션은 DB 스냅샷을 만들 때 태그를 복사합니다. 자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-
데이터베이스 포트	데이터베이스에 액세스하는 데 사용할 포트. 포트 값은 DB 인스턴스의 옵션 그룹에서 옵션에 대해 지정한 포트 값과 일치하지 않아야 합니다.	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	DB 인스턴스가 즉시 재부팅됩니다.
DB 엔진 버전	사용할 Oracle 데이터베이스 엔진의 버전. 프로덕션 DB 인스턴스를 업그레이드 하려면 먼저 테스트 인스턴스에서 업그레이드 프로세스를 테스트하여 업그레이드 시간을 확인하고 애플리케이션을 검증하는 것이 좋습니다. 마이크로 DB 인스턴스는 업그레이드를 추천하지 않습니다. CPU 리소스가 제한되어서 업그레이드를 마치는 데 몇 시간이 걸릴 수도 있기 때문입니다. 스토리지 용량이 작은(10~20GiB) 마이크로 DB 인스턴스를 업그레이드하기 위한 하나의 대안은 Data Pump를 사용해 데이터를 복사하는 것입니다. 하지만 이때도 프로덕션 인스턴스를 마이그레이션하기 전에 먼저 테스트를 권장합니다. 자세한 내용은 Oracle DB 엔진 업그레이드 (p. 752) 단원을 참조하십시오.	[Apply immediately]가 true로 설정되는 즉시 변경됩니다. [Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.	이 변경 도중 인스턴스가 중단됩니다.

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
DB 인스턴스 클래스	<p>사용할 DB 인스턴스 클래스. 자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 및 Oracle을 위한 DB 인스턴스 클래스 지원 (p. 705) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다.
DB 인스턴스 식별자	<p>DB 인스턴스 식별자 이 값은 소문자 문자열로 저장됩니다.</p> <p>DB 인스턴스 이름 바꾸기의 영향에 대한 자세한 내용은 DB 인스턴스 이름 바꾸기 (p. 121) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다. DB 인스턴스가 재부팅됩니다.
DB 파라미터 그룹	<p>DB 인스턴스와 연결할 파라미터 그룹. 자세한 내용은 DB 파라미터 그룹 작업 (p. 161) 및 Oracle sqlnet.ora 파라미터 수정 (p. 750) 단원을 참조하십시오.</p>	파라미터 그룹 변경 사항은 즉시 적용됩니다.	<p>이 변경 도중 인스턴스가 중단되지 않습니다. 파라미터 그룹을 변경하는 경우 일부 파라미터에 대한 변경 내용은 재부팅 없이 DB 인스턴스에 즉시 적용됩니다. 다른 파라미터에 대한 변경 내용은 DB 인스턴스를 재부팅한 후에만 적용됩니다.</p> <p>자세한 내용은 DB 클러스터에서 DB 인스턴스 (p. 123) 단원을 참조하십시오.</p>
삭제 방지	DB 인스턴스가 삭제되지 않도록 방지하려면, 삭제 방지를 활성화합니다. 자세한 내용은 DB 인스턴스 삭제 (p. 128) 단원을 참조하십시오.	-	-
확장 모니터링	<p>DB 인스턴스가 실행되는 운영 체제에 대한 실시간 측정치 수집을 활성화하려면 [Enable enhanced monitoring]을 선택합니다.</p> <p>자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.</p>	-	-
라이선스 모델	<p>Oracle 일반 라이선스 계약을 사용하려면 license-included를 선택하고, 기존 Oracle 라이선스를 사용하려면 bring-your-own-license를 선택합니다.</p> <p>자세한 내용은 Oracle 라이선싱 (p. 704) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다.

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
로그 내보내기	<p>Amazon CloudWatch Logs에 게시할 Oracle 데이터베이스 로그 파일의 유형을 선택합니다.</p> <p>자세한 내용은 Oracle 데이터베이스 로그 파일 (p. 326) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-
유지 관리 기간	<p>시스템 유지 관리를 실행하는 기간. 시스템 유지 관리는 업그레이드를 포함합니다(해당할 경우). 유지 관리 기간은 국제 표준시(UTC)의 시작 시간과 시간 단위의 지속 기간으로 구성됩니다.</p> <p>이 기간을 현재 시간으로 설정하려면 대기 중인 변경 사항이 모두 적용될 수 있도록 현재 시간과 기간 종료 시간 사이에 최소 30분 이상 필요합니다.</p> <p>자세한 내용은 Amazon RDS 유지 관리 기간 (p. 116) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	인스턴스가 중단될 수 있는 작업이 하나 이상 대기 중이고, 유지 관리 기간이 현재 시간을 포함하여 변경된 경우 대기 중인 작업들이 즉시 적용되고 인스턴스가 중단됩니다.
다중 AZ 배포	<p>다중 가용 영역에 DB 인스턴스를 배포하려면 [Yes], 그렇지 않으면 [No].</p> <p>자세한 내용은 리전 및 가용 영역 (p. 99) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	-
새 마스터 암호	마스터 사용자의 암호. 암호는 8~30자 의 영숫자 문자로 구성되어야 합니다.	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-
옵션 그룹	<p>DB 인스턴스와 연결할 옵션 그룹.</p> <p>자세한 내용은 옵션 그룹 작업 (p. 148) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	<p>기존 DB 인스턴스에 APEX 옵션을 추가하는 경우 DB 인스턴스를 자동으로 다시 시작하는 동안 인스턴스가 잠시 중단됩니다.</p> <p>OEM 옵션을 기존 DB 인스턴스에 추가하는 경우, 이 변경 사항 때문에 잠시 동안(1초 미만) 새 연결이 거부됩니다. 기존 연결은 중단되지 않습니다.</p>

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
퍼블릭 액세스 가능성	<p>DB 인스턴스에 퍼블릭 IP 주소를 부여하려면(즉 VPC 외부에서 액세스할 수 있음) [Yes]를 선택합니다. 공개적으로 액세스가 가능하려면 DB 인스턴스도 VPC의 퍼블릭 서브넷에 있어야 합니다. VPC 내부에서만 DB 인스턴스에 액세스할 수 있게 하려면 [No].</p> <p>자세한 내용은 VPC에 있는 DB 인스턴스를 인터넷에서 숨기기 (p. 416) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-
보안 그룹	<p>DB 인스턴스와 연결할 보안 그룹.</p> <p>자세한 내용은 DB 보안 그룹 작업 (EC2-Classic 플랫폼) (p. 394) 단원을 참조하십시오.</p>	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
스토리지 유형	<p>사용할 스토리지 유형. 자세한 내용은 Amazon RDS 스토리지 유형 (p. 101) 단원을 참조하십시오.</p>	<p>[Apply immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	<p>다음과 같이 변경하면 프로세스가 시작되는 동안 잠시 작동이 중단됩니다. 그 이후에 변경 사항이 적용되는 동안 데이터베이스를 정상적으로 사용할 수 있습니다.</p> <ul style="list-style-type: none"> • [General Purpose (SSD)] → [Magnetic]. • 범용(SSD)에서 프로비저닝된 IOPS(SSD)로. DB 인스턴스가 단일 AZ이고 사용자 지정 파라미터 그룹을 사용하는 경우에만 중단이 발생합니다. 다른 AZ DB 인스턴스가 중단되지는 않습니다. • [Magnetic] → [General Purpose (SSD)]. • [Magnetic] → [Provisioned IOPS (SSD)]. • [Provisioned IOPS (SSD)] → [Magnetic]. • 프로비저닝된 IOPS(SSD)에서 범용 (SSD)로. DB 인스턴스가 단일 AZ이고 사용자 지정 파라미터 그룹을 사용하는 경우에만 중단이 발생합니다. 다른 AZ DB 인스턴스가 중단되지는 않습니다.
Subnet Group	<p>DB 인스턴스에 대한 서브넷 그룹. 이 설정을 사용하여 DB 인스턴스를 다른 VPC로 이동할 수 있습니다. DB 인스턴스가 VPC 내에 있지 않을 경우 이 설정을 사용하여 DB 인스턴스를 VPC 내로 이동할 수 있습니다.</p> <p>자세한 내용은 VPC에 있지 않은 DB 인스턴스를 VPC로 이동 (p. 420) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	<p>이 변경 도중 인스턴스가 중단됩니다. DB 인스턴스가 재부팅됩니다.</p>

Oracle sqlnet.ora 파라미터 수정

sqlnet.ora 파일에는 Oracle 데이터베이스 서버와 클라이언트에서 Oracle Net 기능을 구성하는 파라미터가 포함되어 있습니다. sqlnet.ora 파일에서 이 파라미터를 사용하면 데이터베이스 안팎으로 연결하는 속성을 수정할 수 있습니다.

sqlnet.ora 파라미터의 설정 이유에 대한 자세한 정보는 Oracle 설명서의 [프로필 파라미터 구성](#)을 참조하십시오.

sqlnet.ora 파라미터 설정

Amazon RDS Oracle 파라미터 그룹에는 sqlnet.ora 파라미터의 하위 집합이 포함됩니다. 이것들은 다른 Oracle 파라미터와 같은 방법으로 설정해야 합니다. sqlnetora. 접두사는 어떤 파라미터가 sqlnet.ora 파라미터인지 식별합니다. 예를 들어 Amazon RDS의 Oracle 파라미터 그룹에서 default_sdu_size sqlnet.ora 파라미터는 sqlnetora.default_sdu_size입니다.

파라미터 그룹 관리 및 파라미터 값 설정에 대한 자세한 정보는 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

지원되는 sqlnet.ora 파라미터

Amazon RDS에서는 다음 sqlnet.ora 파라미터를 지원합니다. 동적 sqlnet.ora 파라미터의 변경 내용은 즉시 적용됩니다.

파라미터	유효한 값	정적/동적	설명
sqlnetora.default_sdu_size	Oracle 11g: 512 ~ 65535 Oracle 12c: – 512 ~ 2097152	동적	바이트로 표기하는 세션 데이터 단위(SDU). SDU는 버퍼에 두었다가 한 번에 네트워크로 전송하는 데이터의 양입니다.
sqlnetora.diagnosable_enabled	ON, OFF	동적	ADR(Automatic Diagnostic Repository) 추적을 활성화하거나 비활성화하는 값. ON은 사용하는 ADR 파일 추적을 지정합니다. OFF는 사용하는 비ADR 파일 추적을 지정합니다.
sqlnetora.recvbuf_size	268435456	동적	세션 작업 수신을 위한 버퍼 공간 제한. TCP/IP, SSL을 사용하는 TCP/IP, SDP 프로토콜이 지원합니다.
sqlnetora.sendbuf_size	268435456	동적	세션 작업 발송을 위한 버퍼 공간 제한. TCP/IP, SSL을 사용하는 TCP/IP, SDP 프로토콜이 지원합니다.
sqlnetora.sqlne1440_expire_time		동적	클라이언트-서버 연결이 활성인지 확인하기 위해 점검을 보내는 시간 간격(분).
sqlnetora.sqlne1440_inbound_connect_timeout	10~7200	동적	클라이언트가 데이터베이스 서버에 연결하고 필요한 인증 정보를 제공하는 시간(초).
sqlnetora.sqlne1440_outbound_connect_timeout	10~7200	동적	클라이언트가 Oracle Net 연결을 DB 인스턴스에 수립하는 시간(초).
sqlnetora.sqlne1440_recv_timeouts	10~7200	동적	데이터베이스 서버가 연결 수립 후 클라이언트 데이터를 기다리는 시간(초).

파라미터	유효한 값	정적/동적	설명
sqlnet.ora.send_timeo ^{드래그 및 드롭}	10~7200	동적	데이터베이스 서버가 연결 수립 후 클라이언트에 작업 전송을 완료하는 시간(초).
sqlnet.ora.tcp.connect_timeo ^{드래그 및 드롭}	10~7200	동적	클라이언트가 TCP 연결을 데이터베이스 서버에 수립하는 시간(초).
sqlnet.ora.trace_level_server ^{드래그 및 드롭}	16, OFF, USER, ADMIN, SUPPORT	동적	비ADR 추적의 경우, 지정된 레벨에서 서버 추적을 켜거나 고십시오.

지원되는 각 sqlnet.ora 파라미터의 기본값은 해당 릴리스에서 Oracle의 기본값입니다. Oracle 12c 기본값에 대한 자세한 정보는 12c Oracle 설명서의 [sqlnet.ora 파일의 파라미터](#)를 참조하십시오. Oracle 11g 기본값에 대한 자세한 정보는 11g Oracle 설명서의 [sqlnet.ora 파일의 파라미터](#)를 참조하십시오.

sqlnet.ora 파라미터 보기

AWS Management 콘솔, AWS CLI, SQL 클라이언트를 사용하여 sqlnet.ora 파라미터와 그 설정을 볼 수 있습니다.

콘솔을 사용하여 sqlnet.ora 파라미터 보기

파라미터 그룹에서 파라미터 보기에 대한 자세한 정보는 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

Oracle 파라미터 그룹에서 sqlnet.ora. 접두사는 어떤 파라미터가 sqlnet.ora 파라미터인지 식별합니다.

AWS CLI를 사용하여 sqlnet.ora 파라미터 보기

Oracle 파라미터 그룹에서 구성한 sqlnet.ora 파라미터를 보려면 AWS CLI [describe-db-parameters](#) 명령을 사용합니다.

Oracle DB 인스턴스의 sqlnet.ora 파라미터를 모두 보려면 AWS CLI [download-db-log-file-portion](#) 명령을 호출합니다. DB 인스턴스 식별자와 로그 파일 이름, 출력 유형을 지정합니다.

Example

다음 코드는 mydbinstance에 대한 모든 sqlnet.ora 파라미터를 나열합니다.

Linux, OS X, Unix의 경우:

```
aws rds download-db-log-file-portion \
--db-instance-identifier mydbinstance \
--log-file-name trace/sqlnet-parameters \
--output text
```

Windows의 경우:

```
aws rds download-db-log-file-portion ^
--db-instance-identifier mydbinstance ^
--log-file-name trace/sqlnet-parameters ^
--output text
```

SQL 클라이언트를 사용하여 sqlnet.ora 파라미터 보기

SQL 클라이언트에서 Oracle DB 인스턴스에 연결하면 다음 쿼리가 sqlnet.ora 파라미터를 나열합니다.

```
SELECT * FROM TABLE
(rdsadmin.rds_file_util.read_text_file(
    p_directory => 'BDUMP',
    p_filename   => 'sqlnet-parameters'));
```

SQL 클라이언트에서 Oracle DB 인스턴스에 연결하는 방법에 대한 자세한 정보는 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 \(p. 736\)](#)을 참조하십시오.

Oracle DB 엔진 업그레이드

Amazon RDS에서 새 버전의 Oracle을 지원하는 경우 DB 인스턴스를 새 버전으로 업그레이드할 수 있습니다. Amazon RDS는 Oracle DB 인스턴스에 대해 다음 업그레이드를 지원합니다.

- 메이저 버전 업그레이드 – 11g에서 12c로.

일반적으로 메이저 엔진 버전 업그레이드는 기존 애플리케이션과 호환되지 않는 변경 사항을 가져올 수 있습니다. 반대로 마이너 버전 업그레이드에는 기존 애플리케이션과 호환되는 변경 사항만 포함됩니다.

DB 인스턴스를 수동으로 수정하여 메이저 버전 업그레이드를 수행해야 합니다. DB 인스턴스에서 마이너 버전 자동 업그레이드를 사용하도록 설정하면 마이너 버전 업그레이드가 자동으로 수행됩니다. 그 밖의 경우에는 DB 인스턴스를 수동으로 수정하여 마이너 버전 업그레이드를 수행해야 합니다.

업그레이드를 수행하는 동안 중단이 발생합니다. 중단 시간은 엔진, 버전 및 DB 인스턴스의 크기에 따라 달립니다.

Amazon RDS에서 사용할 수 있는 Oracle 버전에 대한 자세한 내용은 [Oracle 데이터베이스 엔진 출시 정보 \(p. 902\)](#) 단원을 참조하십시오.

주제

- [업그레이드 개요 \(p. 752\)](#)
- [메이저 버전 업그레이드 \(p. 753\)](#)
- [Oracle 마이너 버전 업그레이드 \(p. 754\)](#)
- [Oracle SE2 업그레이드 경로 \(p. 754\)](#)
- [옵션 및 파라미터 그룹 고려 사항 \(p. 754\)](#)
- [업그레이드 테스트 \(p. 755\)](#)
- [Oracle DB 인스턴스 업그레이드 \(p. 756\)](#)

업그레이드 개요

Amazon RDS는 업그레이드 프로세스 중에 DB 스냅샷을 2개 캡처합니다. 첫 번째 DB 스냅샷은 업그레이드 변경 이전 DB 인스턴스의 스냅샷입니다. 업그레이드가 데이터베이스에 맞지 않는 경우에는 이 스냅샷을 복구하여 이전 버전의 DB 인스턴스를 생성할 수 있습니다. 두 번째 DB 스냅샷은 업그레이드 완료 이후에 캡처됩니다.

Note

DB 인스턴스에 대한 백업 보존 기간을 0보다 큰 수로 설정하면 Amazon RDS는 DB 스냅샷만 캡처합니다. 백업 보존 기간을 변경하려면 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

업그레이드가 완료되면 이전 버전의 데이터베이스 엔진으로 되돌릴 수 없습니다. 이때 이전 버전으로 되돌리려면 업그레이드 전에 캡처한 DB 스냅샷을 복구하여 새로운 DB 인스턴스를 생성해야 합니다.

DB 인스턴스를 다중 AZ 배포로 생성한 경우에는 기본 복제본과 예비 복제본이 모두 업그레이드됩니다. 운영 체제 업데이트가 필요하지 않은 경우, 기본 DB 인스턴스와 예비 DB 인스턴스가 모두 동시에 업그레이드되므로 업그레이드가 끝날 때까지 작동 중단을 겪게 됩니다.

DB 인스턴스가 다중 AZ 배포에 있고 운영 체제 업데이트가 필요한 경우, 데이터베이스 업그레이드를 요청하면 운영 체제 업데이트가 적용됩니다. 이 경우 운영 체제가 예비 DB 인스턴스에서 업데이트되고, 예비 DB 인스턴스가 업그레이드됩니다. 업그레이드가 완료되면 기본 DB 인스턴스가 예비 DB 인스턴스로 장애 조치되고, 운영 체제가 새 예비 DB 인스턴스(이전 기본 DB 인스턴스)에서 업데이트되며, 데이터베이스가 업그레이드됩니다.

메이저 버전 업그레이드

Amazon RDS는 다음과 같은 메이저 버전 업그레이드를 지원합니다.

- Oracle 버전 12.1.0.2~Oracle 버전 12.2.0.1을 실행하는 Oracle DB 인스턴스
- Oracle 버전 11.2.0.4~Oracle 버전 12.2.0.1을 실행하는 Oracle DB 인스턴스
- Oracle 버전 11.2.0.4~Oracle 버전 12.1.0.2.v5 이상을 실행하는 Oracle DB 인스턴스

메이저 버전 업그레이드를 수행하려면 DB 인스턴스를 수동으로 수정합니다. 메이저 버전 업그레이드는 자동으로 수행되지 않습니다.

경우에 따라 현재 Oracle DB 인스턴스가 업그레이드하려는 버전에서 지원되지 않는 DB 인스턴스 클래스에서 실행될 수 있습니다. 그런 경우 업그레이드하기 전에 DB 인스턴스를 지원되는 DB 인스턴스 클래스로 마이그레이션해야 합니다. 각 Amazon RDS Oracle 버전 및 에디션에서 지원되는 DB 인스턴스 클래스에 대한 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.

메이저 버전 업그레이드를 수행하기 전에 업그레이드하려는 DB 인스턴스에서 최적화 프로그램 통계를 수집하는 것이 좋습니다. 최적화 프로그램 통계를 수집하면 업그레이드 중에 DB 인스턴스 가동 중지 시간을 줄일 수 있습니다. 최적화 프로그램 통계를 수집하려면 다음 예제와 같이 마스터 사용자로 DB 인스턴스에 연결하고, `DBMS_STATS.GATHER_DICTIONARY_STATS` 프로시저를 실행합니다.

```
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS;
```

자세한 내용은 Oracle 문서의 [Gathering Optimizer Statistics to Decrease Oracle Database Downtime](#)을 참조하십시오.

Note

Oracle 버전 11.2.0.3 및 11.2.0.2 같은 더 이상 사용되지 않는 Oracle 버전에 대해서는 메이저 버전 업그레이드가 지원되지 않습니다.

메이저 버전 다운그레이드는 지원되지 않습니다.

11g에서 12c로의 메이저 버전 업그레이드는 동일 월이나 그 이후에 릴리스된 Oracle PSU(Patch Set Update)로 업그레이드해야 합니다.

예를 들면 Oracle 버전 11.2.0.4.v14에서 12.1.0.2.v11로의 메이저 버전 업그레이드가 지원됩니다.

그러나 Oracle 버전 11.2.0.4.v14에서 12.1.0.2.v9로의 메이저 버전 업그레이드는 지원되지 않습니다. Oracle 버전 11.2.0.4.v14는 2017년 10월에 릴리스되었고, Oracle 버전 12.1.0.2.v9는 2017년 7

월에 릴리스되었기 때문입니다. 각 Oracle PSU의 릴리스 날짜에 대한 자세한 내용은 [Oracle 데이터베이스 엔진 출시 정보 \(p. 902\)](#) 단원을 참조하십시오.

Oracle 마이너 버전 업그레이드

마이너 버전 업그레이드는 메이저 버전의 Oracle PSU를 적용합니다.

다음과 같은 마이너 버전 업그레이드는 지원되지 않습니다.

현재 버전	업그레이드 지원되지 않음
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

마이너 버전 다운그레이드는 지원되지 않습니다.

Oracle SE2 업그레이드 경로

다음 표에서는 Standard Edition Two(SE2)에 대한 지원 업그레이드 경로를 보여줍니다. 라이선스 포함 및 기존 보유 라이선스 사용(BYOL) 모델에 대한 자세한 내용은 [Oracle 라이선싱 \(p. 704\)](#) 단원을 참조하십시오.

기존 구성	지원되는 SE2 구성
12.2.0.1 SE2, BYOL	12.2.0.1 SE2, BYOL 또는 라이선스 포함
12.1.0.2 SE2, BYOL	12.2.0.1 SE2, BYOL 또는 라이선스 포함 12.1.0.2 SE2, BYOL 또는 라이선스 포함
11.2.0.4 SE1, BYOL 또는 라이선스 포함	12.2.0.1 SE2, BYOL 또는 라이선스 포함
11.2.0.4 SE, BYOL	12.1.0.2 SE2, BYOL 또는 라이선스 포함

기존 구성에서 지원되는 SE2 구성으로 업그레이드하려면 지원되는 업그레이드 경로를 사용하십시오. 자세한 내용은 [메이저 버전 업그레이드 \(p. 753\)](#) 단원을 참조하십시오.

옵션 및 파라미터 그룹 고려 사항

옵션 그룹 고려 사항

DB 인스턴스에서 사용자 지정 옵션 그룹을 사용할 경우 Amazon RDS에서 DB 인스턴스에 새 옵션 그룹을 할당할 수 없는 경우도 있습니다. 예를 들어, 새로운 메이저 버전으로 업그레이드할 경우 발생합니다. 이 경

우 새 옵션 그룹을 지정해야 합니다. 새 옵션 그룹을 생성하고 동일한 옵션을 기준 사용자 지정 옵션 그룹에 추가하는 것이 좋습니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 또는 [옵션 그룹 복사 \(p. 151\)](#) 단원을 참조하십시오.

DB 인스턴스가 APEX 옵션이 포함된 사용자 지정 옵션 그룹을 사용하면 경우에 따라 DB 인스턴스와 동시에 APEX의 버전을 업그레이드하여 DB 인스턴스의 업그레이드에 소요되는 시간을 단축할 수 있습니다. 자세한 내용은 [APEX 버전 업그레이드 \(p. 790\)](#) 단원을 참조하십시오.

파라미터 그룹 고려 사항

DB 인스턴스에서 사용자 지정 파라미터 그룹을 사용할 경우 Amazon RDS에서 DB 인스턴스에 새 파라미터 그룹을 할당할 수 없는 경우도 있습니다. 예를 들어, 새로운 메이저 버전으로 업그레이드할 경우 발생합니다. 이 경우 새 파라미터 그룹을 지정해야 합니다. 새 파라미터 그룹을 생성하고 기존 사용자 지정 파라미터 그룹에서와 같은 방법으로 파라미터를 구성하는 것이 좋습니다.

자세한 내용은 [DB 파라미터 그룹 생성 \(p. 163\)](#) 또는 [DB 파라미터 그룹 복사 \(p. 166\)](#) 단원을 참조하십시오.

업그레이드 테스트

DB 인스턴스에 대한 메이저 버전 업그레이드를 수행하기 전에 데이터베이스 및 해당 데이터베이스에 액세스하는 모든 애플리케이션이 새 버전과 호환되는지 여부를 철저하게 테스트해야 합니다. 다음 절차를 참조하는 것이 좋습니다.

메이저 버전 업그레이드를 테스트하려면

1. 다음과 같이 새 버전의 데이터베이스 엔진에 대한 Oracle 업그레이드 문서를 검토하여 데이터베이스나 애플리케이션에 영향을 끼칠 수도 있는 호환성 문제가 있는지 살펴봅니다. 자세한 내용은 Oracle 문서의 [Database Upgrade Guide](#) 단원을 참조하십시오.
2. DB 인스턴스에서 사용자 지정 옵션 그룹을 사용할 경우 업그레이드하려는 새 버전과 호환되는 새 옵션 그룹을 생성합니다. 자세한 내용은 [옵션 그룹 고려 사항 \(p. 754\)](#) 단원을 참조하십시오.
3. DB 인스턴스에서 사용자 지정 파라미터 그룹을 사용할 경우 업그레이드하려는 새 버전과 호환되는 새 파라미터 그룹을 생성합니다. 자세한 내용은 [파라미터 그룹 고려 사항 \(p. 755\)](#) 단원을 참조하십시오.
4. 업그레이드할 DB 인스턴스의 DB 스냅샷을 생성합니다. 자세한 내용은 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.
5. DB 스냅샷을 복구하여 새로운 테스트 DB 인스턴스를 생성합니다. 자세한 내용은 [DB 스냅샷에서 복원 \(p. 205\)](#) 단원을 참조하십시오.
6. 다음 방법 중 한 가지를 사용하여 이 새로운 테스트 DB 인스턴스를 변경하고 새로운 버전으로 업그레이드합니다.
 - [콘솔을 사용하여 DB 인스턴스의 엔진 버전 업그레이드 \(p. 119\)](#)
 - [AWS CLI를 사용하여 DB 인스턴스의 엔진 버전 업그레이드 \(p. 119\)](#)
 - [RDS API를 사용하여 DB 인스턴스의 엔진 버전 업그레이드 \(p. 120\)](#)
7. 테스트 수행:
 - 업그레이드한 DB 인스턴스와 관련하여 데이터베이스 및 애플리케이션과 새로운 버전의 호환성을 보장하는 데 필요하다면 최대한 많은 수의 품질 보증 테스트를 실행합니다.
 - 또한 1단계에서 발견된 호환성 문제의 영향을 평가하는 데 필요한 새로운 테스트도 모두 실행합니다.
 - 저장 프로시저와 함수, 트리거를 모두 테스트합니다.
 - 업그레이드한 DB 인스턴스에 대해 애플리케이션의 테스트 버전을 실행합니다. 새 버전에서 애플리케이션이 올바로 작동하는지 확인합니다.
 - 업그레이드한 인스턴스에서 사용할 스토리지를 평가하여 업그레이드 시 추가 스토리지의 필요 여부를 결정합니다. 프러덕션에서 새 버전을 지원하려면 더 큰 인스턴스 클래스를 선택해야 할 수도 있습니다. 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.

- 모든 테스트가 통과되면 프로덕션 환경의 DB 인스턴스에도 업그레이드를 실행합니다. 단, 모든 기능이 정상 작동하는 것을 확인할 때까지 쓰기 연산은 DB 인스턴스에 실행하지 않는 것이 좋습니다.

Oracle DB 인스턴스 업그레이드

Oracle DB 인스턴스의 수동 또는 자동 업그레이드에 대한 자세한 내용은 [DB 인스턴스 엔진 버전 업그레이드 \(p. 118\)](#) 단원을 참조하십시오.

Oracle DB 스냅샷 업그레이드

기존 수동 DB 스냅샷이 있는 경우 Oracle 데이터베이스 엔진의 최신 버전으로 스냅샷을 업그레이드하려고 할 수 있습니다.

Oracle이 버전에 대한 패치 제공을 종단해 Amazon RDS에서 해당 버전을 더 이상 사용할 수 없는 경우 더 이상 사용되지 않는 버전에 해당하는 스냅샷을 업그레이드할 수 있습니다. 자세한 내용은 [Oracle 엔진 버전 관리 \(p. 720\)](#) 단원을 참조하십시오.

다음과 같은 스냅샷 업그레이드는 현재 지원되지 않습니다.

현재 스냅샷 버전	지원되는 스냅샷 업그레이드
12.1.0.1	12.1.0.2.v8
11.2.0.3	11.2.0.4.v11
11.2.0.2	11.2.0.4.v12

Amazon RDS는 모든 AWS 리전에서 스냅샷 업그레이드를 지원합니다.

AWS Management 콘솔

Oracle DB 스냅샷을 업그레이드하려면

- AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
- 탐색 창에서 [Snapshots]를 선택한 다음 업그레이드하려는 DB 스냅샷을 선택합니다.
- 작업에서 스냅샷 수정을 선택합니다. [Modify DB Snapshot] 페이지가 나타납니다.
- DB 엔진 버전에서 스냅샷을 업그레이드할 버전을 선택합니다.
- (선택 사항) 옵션 그룹에서 업그레이드된 DB 스냅샷의 옵션 그룹을 선택합니다. DB 스냅샷을 업그레이드할 때 고려할 옵션 그룹은 DB 인스턴스를 업그레이드할 때와 동일합니다. 자세한 내용은 [옵션 그룹 고려 사항 \(p. 754\)](#) 단원을 참조하십시오.
- [Modify Snapshot]을 선택하여 변경 사항을 저장합니다.

또는 [Cancel]을 선택하여 변경 사항을 취소합니다.

CLI

AWS CLI를 사용하여 Oracle DB 스냅샷을 업그레이드하려면 다음 파라미터와 함께 `modify-db-snapshot` 명령을 호출합니다.

- **--db-snapshot-identifier** – DB 스냅샷의 이름입니다.
- **--engine-version** – 스냅샷을 업그레이드할 버전입니다.

또한 다음 파라미터를 포함해야 할 수 있습니다. DB 스냅샷을 업그레이드할 때 고려할 옵션 그룹은 DB 인스턴스를 업그레이드할 때와 동일합니다. 자세한 내용은 [옵션 그룹 고려 사항 \(p. 754\)](#) 단원을 참조하십시오.

- **--option-group-name** – 업그레이드된 DB 스냅샷에 대한 옵션 그룹입니다.

Example

다음 예제는 DB 스냅샷을 업그레이드합니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-snapshot \  
  --db-snapshot-identifier <mydbsnapshot> \  
  --engine-version <11.2.0.4.v12> \  
  --option-group-name <default:oracle-se1-11-2>
```

Windows의 경우:

```
aws rds modify-db-snapshot ^  
  --db-snapshot-identifier <mydbsnapshot> ^  
  --engine-version <11.2.0.4.v12> ^  
  --option-group-name <default:oracle-se1-11-2>
```

API

Amazon RDS API를 사용하여 Oracle DB 스냅샷을 업그레이드하려면 다음 파라미터와 함께 [ModifyDBSnapshot](#) 작업을 호출합니다.

- **DBSnapshotIdentifier** – DB 스냅샷의 이름입니다.
- **EngineVersion** – 스냅샷을 업그레이드할 버전입니다.

또한 다음 파라미터를 포함해야 할 수 있습니다. DB 스냅샷을 업그레이드할 때 고려할 옵션 그룹은 DB 인스턴스를 업그레이드할 때와 동일합니다. 자세한 내용은 [옵션 그룹 고려 사항 \(p. 754\)](#) 단원을 참조하십시오.

- **OptionGroupName** – 업그레이드된 DB 스냅샷에 대한 옵션 그룹입니다.

Example

다음 예제는 DB 스냅샷을 업그레이드합니다.

```
https://rds.amazonaws.com/  
?Action=ModifyDBSnapshot  
&DBSnapshotIdentifier=mydbsnapshot  
&EngineVersion=11.2.0.4.v12  
&OptionGroupName=default:oracle-se1-11-2  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4  
&Version=2014-10-31  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Credential=AKIADQKE4SARGYLE/20131016/us-west-1/rds/aws4_request  
&X-Amz-Date=20131016T233051Z
```

&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=087a8eb41cb1ab5f99e81575f23e73757ffc6a1e42d7d2b30b9cc0be988cff97

관련 주제

- Oracle 데이터베이스 엔진 출시 정보 (p. 902)
- Oracle DB 엔진 업그레이드 (p. 752)
- DB 인스턴스의 업데이트 적용 (p. 114)

Amazon RDS의 Oracle로 데이터 가져오기

데이터를 Amazon RDS DB 인스턴스로 가져오는 방법은 보유한 데이터의 양과 데이터베이스에 있는 데이터베이스 객체의 개수 및 다양성에 따라 달라집니다. 예를 들어 Oracle SQL Developer를 사용하여 간단한 20MB 데이터베이스를 가져올 수 있습니다. 또한 Oracle Data Pump를 사용하여 수백 메가바이트나 몇 테라바이트 크기의 데이터베이스나 복합 데이터베이스를 가져올 수 있습니다.

AWS Database Migration Service(AWS DMS)를 사용하여 데이터를 Amazon RDS DB 인스턴스로 가져올 수 있습니다. AWS DMS는 가동 중지 없이 데이터베이스를 마이그레이션할 수 있으며 대부분의 데이터베이스 엔진에서는 대상 데이터베이스로 전환할 준비가 될 때까지 지속적으로 복제를 계속할 수 있습니다. AWS DMS를 사용하여 동일하거나 다른 데이터베이스 엔진에서 Oracle로 마이그레이션할 수 있습니다. 다른 데이터베이스 엔진에서 마이그레이션하는 경우 AWS Schema Conversion Tool을 사용하여 AWS DMS를 통해 마이그레이션되지 않는 스키마 객체를 마이그레이션할 수 있습니다. AWS DMS에 대한 자세한 정보는 [AWS Database Migration Service란 무엇입니까?](#)를 참조하십시오.

이러한 마이그레이션 기술 중 하나를 사용하려면 먼저 데이터베이스 백업 모범 사례를 따르는 것이 좋습니다. 데이터를 가져온 후 스냅샷을 생성하여 Amazon RDS DB 인스턴스를 백업할 수 있습니다. 나중에 스냅샷에서 데이터베이스를 복구할 수 있습니다. 자세한 정보는 [Amazon RDS DB 인스턴스 백업 및 복원 \(p. 196\)](#) 단원을 참조하십시오.

Note

Amazon S3의 파일을 사용하여 Oracle로 데이터를 가져올 수도 있습니다. 예를 들어 Amazon S3의 Data Pump 파일을 DB 인스턴스 호스트로 다운로드할 수 있습니다. 자세한 내용은 [Amazon S3통합 \(p. 773\)](#) 단원을 참조하십시오.

Oracle SQL Developer를 사용한 가져오기

작은 데이터베이스의 경우 Oracle에서 무상으로 배포한 그래픽 Java 도구인 Oracle SQL Developer를 사용할 수 있습니다. 데스크톱 컴퓨터(Windows, Linux 또는 Mac) 또는 서버 중 하나에 이 도구를 설치할 수 있습니다. Oracle SQL Developer는 두 Oracle 데이터베이스 간의 데이터 마이그레이션이나 MySQL 등의 다른 데이터베이스에서 Oracle로 데이터 마이그레이션을 위한 옵션을 제공합니다. Oracle SQL Developer는 작은 데이터베이스를 마이그레이션하는 데 매우 적합합니다. 데이터 마이그레이션을 시작하기 전에 Oracle SQL Developer 제품을 읽는 것이 좋습니다.

SQL Developer를 설치한 후에는 SQL Developer를 사용하여 원본 및 대상 데이터베이스에 연결할 수 있습니다. Tools(도구) 메뉴의 Database Copy(데이터베이스 복사) 명령을 사용하여 데이터를 Amazon RDS 인스턴스에 복사합니다.

Oracle SQL Developer를 다운로드하려면 <http://www.oracle.com/technetwork/developer-tools/sql-developer>를 참조하십시오.

Oracle에서는 MySQL 및 SQL Server 등 다른 데이터베이스에서 마이그레이션하는 방법을 설명하는 문서도 제공합니다. 자세한 정보는 Oracle 설명서에서 <http://www.oracle.com/technetwork/database/migration>을 참조하십시오.

Oracle Data Pump를 사용한 가져오기

Oracle Data Pump는 Oracle 내보내기/가져오기 유ти리티를 장기적으로 대체합니다. Oracle Data Pump는 Oracle 설치에서 Amazon RDS DB 인스턴스로 대량의 데이터를 이동하는 기본적인 방법입니다. 다음과 같은 여러 시나리오에 대해 Oracle Data Pump를 사용할 수 있습니다.

- Oracle 데이터베이스(온프레미스 또는 Amazon EC2 인스턴스 중 하나)에서 Oracle DB 인스턴스용 Amazon RDS로 데이터를 가져옵니다.
- RDS Oracle DB 인스턴스에서 Oracle 데이터베이스(온프레미스 또는 Amazon EC2 인스턴스)로 데이터를 가져옵니다.
- RDS Oracle DB 인스턴스끼리 데이터를 가져옵니다(예: EC2-Classic에서 VPC로 데이터 마이그레이션).

Oracle Data Pump 유ти리티를 다운로드하려면 Oracle Technology Network 웹사이트의 [Oracle Database 소프트웨어 다운로드](#)를 참조하십시오.

Oracle 데이터베이스 버전 간에 마이그레이션할 때 호환성 고려 사항은 [Oracle 설명서](#)를 참조하십시오.

Oracle Data Pump를 사용하여 데이터를 가져올 때는 소스 데이터베이스의 데이터가 포함된 덤프 파일을 대상 데이터베이스로 전송해야 합니다. Amazon S3 버킷을 사용하거나 두 데이터베이스 간 데이터베이스 링크를 사용하여 덤프 파일을 전송할 수 있습니다.

다음은 Oracle Data Pump를 사용하여 Oracle용 Amazon RDS DB 인스턴스로 데이터를 가져오는 모범 사례입니다.

- 특정 스키마 및 객체를 가져오려면 schema 또는 table 모드로 가져오기를 수행하십시오.
- 가져오는 스키마를 애플리케이션에 필요한 스키마로 제한하십시오.
- full 모드로 가져오지 마십시오.

Oracle용 Amazon RDS는 sys 또는 sysdba 관리 사용자에 대한 액세스를 허용하지 않으므로 full 모드로 가져오거나 Oracle에서 관리하는 구성 요소에 대한 스키마를 가져오면 Oracle 데이터 사전을 손상시키고 데이터베이스의 안정성에 영향을 줄 수 있습니다.

- 많은 양의 데이터를 로드할 때 덤프 파일을 대상 Oracle용 Amazon RDS DB 인스턴스로 전송하고 인스턴스의 DB 스냅샷을 가져온 다음, 가져오기를 테스트하여 성공했는지 확인하십시오. 데이터베이스 구성 요소가 무효화된 경우 DB 인스턴스를 삭제하고 DB 스냅샷에서 다시 생성할 수 있습니다. 복원된 DB 인스턴스에는 DB 스냅샷을 가져왔을 때 DB 인스턴스에 준비된 모든 덤프 파일이 포함됩니다.
- Oracle Data Pump 내보내기 파라미터 TRANSPORT_TABLESPACES, TRANSPORTABLE 또는 TRANSPORT_FULL_CHECK을 사용하여 생성된 덤프 파일을 가져오지 마십시오. Oracle용 Amazon RDS DB 인스턴스는 이러한 덤프 파일 가져오기를 지원하지 않습니다.

주제

- [Oracle Data Pump와 Amazon S3 버킷으로 데이터 가져오기 \(p. 759\)](#)
- [Oracle Data Pump와 데이터베이스로 데이터 가져오기 \(p. 762\)](#)

Oracle Data Pump와 Amazon S3 버킷으로 데이터 가져오기

다음 가져오기 프로세스에서는 Oracle Data Pump와 Amazon S3 버킷을 사용합니다. 이 프로세스는 Oracle DBMS_DATAPUMP 패키지를 사용하여 소스 데이터베이스에서 데이터를 내보내고, 덤프 파일을 Amazon S3 버킷에 넣습니다. 그런 다음 덤프 파일을 Amazon S3 버킷에서 대상 Amazon RDS Oracle DB 인스턴스의 DATA_PUMP_DIR 디렉터리로 다운로드합니다. 마지막 단계에서는 DBMS_DATAPUMP 패키지를 사용하여 복사된 덤프 파일의 데이터를 Amazon RDS Oracle DB 인스턴스로 가져옵니다.

이 프로세스를 수행하려면 다음 요구 사항이 충족되어야 합니다.

- 파일 전송에 사용할 수 있는 Amazon S3 버킷이 있어야 하고 Amazon S3 버킷은 DB 인스턴스와 동일한 AWS 리전에 있어야 합니다. 지침을 보려면 Amazon Simple Storage Service 시작 안내서에서 [버킷 생성](#)을 참조하십시오.
- [Amazon RDS Oracle과 Amazon S3의 통합을 위한 사전 조건 \(p. 773\)](#)의 지침에 따라 Amazon RDS 통합을 위한 Amazon S3 버킷을 준비해야 합니다.
- 원본 인스턴스 및 대상 DB 인스턴스에 덤프 파일을 저장할 수 있는 충분한 스토리지 공간이 있는지 확인해야 합니다.

Note

이 프로세스는 덤프 파일을 모든 Oracle DB 인스턴스의 사전 구성된 디렉터리인 DATA_PUMP_DIR 디렉터리로 가져옵니다. 이 디렉터리는 데이터 파일과 동일한 스토리지 볼륨에 위치합니다. 덤프 파일을 가져올 때 기존 Oracle 데이터 파일은 더 많은 공간을 사용합니다. 따라서 DB 인스턴스가 공간의 추가 사용을 수용할 수 있는지 확인해야 합니다. 가져온 덤프 파일은 자동으로 삭제되거나 DATA_PUMP_DIR 디렉터리에서 제거됩니다. 가져온 덤프 파일을 제거하려면 Oracle 웹사이트에 있는 [UTL_FILE.FREMOVE](#)를 사용하십시오.

Oracle Data Pump 및 Amazon S3 버킷을 사용하는 가져오기 프로세스는 다음 단계로 이루어집니다.

주제

- [1단계: Amazon RDS 대상 인스턴스의 사용자에게 권한 부여 \(p. 760\)](#)
- [2단계: DBMS_DATAPUMP를 사용하여 덤프 파일 생성 \(p. 761\)](#)
- [3단계: 덤프 파일을 Amazon S3 버킷에 업로드 \(p. 761\)](#)
- [4단계: 내보낸 덤프 파일을 Amazon S3 버킷에서 대상 DB 인스턴스로 복사 \(p. 761\)](#)
- [5단계: DBMS_DATAPUMP를 사용하여 대상 DB 인스턴스의 데이터 파일 가져오기 \(p. 761\)](#)
- [6단계: 정리 \(p. 762\)](#)

1단계: Amazon RDS 대상 인스턴스의 사용자에게 권한 부여

RDS 대상 인스턴스에서 사용자에게 권한을 부여하려면 다음 단계를 수행하십시오.

- SQL Plus 또는 Oracle SQL Developer를 사용하여 데이터를 가져올 Amazon RDS 대상 Oracle DB 인스턴스에 연결합니다. Amazon RDS 마스터 사용자 권한으로 연결합니다. DB 인스턴스 연결에 대한 자세한 정보는 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 \(p. 736\)](#) 단원을 참조하십시오.
- 데이터를 가져오려면 먼저 테이블 스페이스를 생성해야 합니다. 자세한 내용은 [테이블스페이스 생성과 크기 조정 \(p. 846\)](#) 단원을 참조하십시오.
- 데이터를 가져올 사용자 계정이 존재하지 않으면 사용자 계정을 생성한 후 필요한 권한과 역할을 부여합니다. 데이터를 다수의 사용자 스키마로 가져오려는 경우에는 사용자 계정을 각각 생성한 후에 필요한 권한과 역할을 부여합니다.

예를 들어 다음은 새로운 사용자를 생성한 후 데이터를 해당 사용자의 스키마로 가져오기 위해 필요한 권한과 역할을 부여하는 명령입니다.

```
create user schema_1 identified by <password>;
grant create session, resource to schema_1;
alter user schema_1 quota 100M on users;
```

위 예제에서는 새로운 사용자에게 CREATE SESSION 권한과 RESOURCE 역할을 부여합니다. 하지만 가져오는 데이터베이스 객체에 따라 권한과 역할이 추가로 필요할 수도 있습니다.

Note

다음 단계에서 *schema_1*을 이 단계의 스키마 이름으로 대체합니다.

2단계: DBMS_DATAPUMP를 사용하여 덤프 파일 생성

SQL Plus 또는 Oracle SQL Developer를 사용하여 관리 사용자 권한으로 원본 Oracle 인스턴스에 연결합니다. 소스 데이터베이스가 Amazon RDS Oracle DB 인스턴스인 경우 Amazon RDS 마스터 사용자 권한으로 연결합니다. 그런 다음 Oracle Data Pump 유ти리티를 사용하여 덤프 파일을 생성합니다.

다음 스크립트는 DATA_PUMP_DIR 디렉터리에 sample.dmp라는 덤프 파일을 생성합니다.

```
DECLARE
hdnl NUMBER;
BEGIN
hdnl := DBMS_DATAPUMP.OPEN( operation => 'EXPORT', job_mode => 'SCHEMA', job_name=>null);
DBMS_DATAPUMP.ADD_FILE( handle => hdnl, filename => 'sample.dmp', directory =>
'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_dump_file);
DBMS_DATAPUMP.ADD_FILE( handle => hdnl, filename => 'exp.log', directory =>
'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_log_file);
DBMS_DATAPUMP.METADATA_FILTER(hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
DBMS_DATAPUMP.START_JOB(hdnl);
END;
/
```

Note

Data Pump 작업은 비동기로 시작됩니다. Data Pump 작업 모니터링에 대한 자세한 정보는 Oracle 설명서의 [Monitoring Job Status](#)를 참조하십시오.

3단계: 덤프 파일을 Amazon S3 버킷에 업로드

Amazon S3를 사용하여 덤프 파일을 Amazon S3 버킷에 업로드하십시오.

지침은 Amazon Simple Storage Service 시작 안내서의 [버킷에 객체 추가](#)를 참조하십시오.

4단계: 내보낸 덤프 파일을 Amazon S3 버킷에서 대상 DB 인스턴스로 복사

Amazon RDS 프로시저 rdsadmin.rdsadmin_s3_tasks.download_from_s3를 사용하여 덤프 파일을 Amazon S3 버킷에서 대상 DB 인스턴스로 복사하십시오. 다음 예제에서는 Amazon S3 버킷에서 *mys3bucket*이라는 이름의 모든 파일을 *DATA_PUMP_DIR* 디렉터리로 다운로드합니다.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
    p_bucket_name    => 'mys3bucket',
    p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

SELECT 문은 VARCHAR2 데이터 형식으로 작업 ID를 반환합니다.

자세한 내용은 [Amazon S3 버킷의 파일을 Oracle DB 인스턴스로 다운로드 \(p. 781\)](#) 단원을 참조하십시오.

5단계: DBMS_DATAPUMP를 사용하여 대상 DB 인스턴스의 데이터 파일 가져오기

DB 인스턴스에서 Oracle Data Pump를 사용하여 스키마를 가져옵니다. METADATA_REMAP 등 추가 옵션이 필요할 수 있습니다.

Amazon RDS 마스터 사용자 계정으로 DB 인스턴스에 연결하여 데이터를 가져옵니다.

```
DECLARE
hdnl NUMBER;
BEGIN
hdnl := DBMS_DATAPUMP.OPEN( operation => 'IMPORT', job_mode => 'SCHEMA', job_name=>null);
DBMS_DATAPUMP.ADD_FILE( handle => hdnl, filename => 'sample_copied.dmp', directory =>
'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_dump_file);
DBMS_DATAPUMP.METADATA_FILTER(hdnl, 'SCHEMA_EXPR', 'IN (''SCHEMA_1'')');
DBMS_DATAPUMP.START_JOB(hdnl);
END;
/
```

DB 인스턴스에서 해당 사용자의 테이블을 보고 데이터 가져오기를 확인할 수 있습니다. 예를 들어 다음 쿼리를 **SCHEMA_1**의 테이블 수를 반환합니다.

```
select count(*) from dba_tables where owner='SCHEMA_1';
```

6단계: 정리

데이터를 가져온 후에는 유지하지 않을 파일을 삭제할 수 있습니다. 다음 명령을 사용하여 DATA_PUMP_DIR에서 파일을 나열할 수 있습니다.

```
select * from table(RDSADMIN.RDS_FILE_UTIL.LISTDIR('DATA_PUMP_DIR')) order by mtime;
```

DATA_PUMP_DIR에서 더 이상 필요하지 않은 파일을 삭제하려면 다음 명령을 사용합니다.

```
exec utl_file.fremove('DATA_PUMP_DIR', '<file name>');
```

예를 들어, 다음 명령은 "sample_copied.dmp"라는 파일을 삭제합니다.

```
exec utl_file.fremove('DATA_PUMP_DIR', 'sample_copied.dmp');
```

Oracle Data Pump와 데이터베이스로 데이터 가져오기

다음 가져오기 프로세스에서는 Oracle Data Pump 및 Oracle DBMS_FILE_TRANSFER 패키지를 사용합니다. 이 프로세스에서는 소스 Oracle 인스턴스(온프레미스 또는 Amazon EC2 인스턴스이거나 Amazon RDS Oracle DB 인스턴스일 수 있음)에 연결합니다. 그런 다음 프로세스가 DBMS_DATAPUMP 패키지를 사용하여 데이터를 내보냅니다. 다음으로, DBMS_FILE_TRANSFER.PUT_FILE 메서드를 사용하여 Oracle 인스턴스의 덤프 파일을 데이터베이스 링크를 통해 연결된 대상 Amazon RDS Oracle DB 인스턴스의 DATA_PUMP_DIR 디렉터리로 복사합니다. 마지막 단계에서는 DBMS_DATAPUMP 패키지를 사용하여 복사된 덤프 파일의 데이터를 Amazon RDS Oracle DB 인스턴스로 가져옵니다.

이 프로세스를 수행하려면 다음 요구 사항이 충족되어야 합니다.

- DBMS_FILE_TRANSFER 및 DBMS_DATAPUMP 패키지에 대한 실행 권한이 있어야 합니다.
- 원본 DB 인스턴스의 DATA_PUMP_DIR 디렉터리에 대한 쓰기 권한이 있어야 합니다.
- 원본 인스턴스 및 대상 DB 인스턴스에 덤프 파일을 저장할 수 있는 충분한 스토리지 공간이 있는지 확인해야 합니다.

Note

이 프로세스는 덤프 파일을 모든 Oracle DB 인스턴스의 사전 구성된 디렉터리인 DATA_PUMP_DIR 디렉터리로 가져옵니다. 이 디렉터리는 데이터 파일과 동일한 스토리지 볼륨에 위치합니다. 덤프 파일을 가져올 때 기존 Oracle 데이터 파일은 더 많은 공간을 사용합니다. 따라서 DB 인스턴스가

공간의 추가 사용을 수용할 수 있는지 확인해야 합니다. 가져온 덤프 파일은 자동으로 삭제되거나 DATA_PUMP_DIR 디렉터리에서 제거됩니다. 가져온 덤프 파일을 제거하려면 Oracle 웹사이트에 있는 [UTL_FILE.FREMOVE](#)를 사용하십시오.

Oracle Data Pump 및 DBMS_FILE_TRANSFER 패키지를 사용하는 가져오기 프로세스는 다음 단계로 이루어집니다.

주제

- [1단계: Amazon RDS 대상 인스턴스의 사용자에게 권한 부여 \(p. 760\)](#)
- [2단계: 소스 데이터베이스에서 사용자에게 권한 부여 \(p. 763\)](#)
- [3단계: DBMS_DATAPUMP를 사용하여 덤프 파일 생성 \(p. 761\)](#)
- [4단계: 대상 DB 인스턴스의 데이터베이스 링크 생성 \(p. 761\)](#)
- [5단계: DBMS_FILE_TRANSFER를 사용하여 내보낸 덤프 파일을 대상 DB 인스턴스로 복사 \(p. 761\)](#)
- [6단계: DBMS_DATAPUMP를 사용하여 대상 DB 인스턴스로 데이터 파일 가져오기 \(p. 761\)](#)
- [7단계: 정리 \(p. 762\)](#)

1단계: Amazon RDS 대상 인스턴스의 사용자에게 권한 부여

RDS 대상 인스턴스에서 사용자에게 권한을 부여하려면 다음 단계를 수행하십시오.

1. SQL Plus 또는 Oracle SQL Developer를 사용하여 데이터를 가져올 Amazon RDS 대상 Oracle DB 인스턴스에 연결합니다. Amazon RDS 마스터 사용자 권한으로 연결합니다. DB 인스턴스 연결에 대한 자세한 정보는 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 \(p. 736\)](#) 단원을 참조하십시오.
2. 데이터를 가져오려면 먼저 테이블 스페이스를 생성해야 합니다. 자세한 내용은 [테이블스페이스 생성과 크기 조정 \(p. 846\)](#) 단원을 참조하십시오.
3. 데이터를 가져올 사용자 계정이 존재하지 않으면 사용자 계정을 생성한 후 필요한 권한과 역할을 부여합니다. 데이터를 다수의 사용자 스키마로 가져오려는 경우에는 사용자 계정을 각각 생성한 후에 필요한 권한과 역할을 부여합니다.

예를 들어 다음은 새로운 사용자를 생성한 후 데이터를 해당 사용자의 스키마로 가져오기 위해 필요한 권한과 역할을 부여하는 명령입니다.

```
create user schema_1 identified by <password>;
grant create session, resource to schema_1;
alter user schema_1 quota 100M on users;
```

위 예제에서는 새로운 사용자에게 CREATE SESSION 권한과 RESOURCE 역할을 부여합니다. 하지만 가져오는 데이터베이스 객체에 따라 권한과 역할이 추가로 필요할 수도 있습니다.

Note

다음 단계에서 `schema_1`을 이 단계의 스키마 이름으로 대체합니다.

2단계: 소스 데이터베이스에서 사용자에게 권한 부여

SQL Plus 또는 Oracle SQL Developer를 사용하여 가져올 데이터를 포함하는 Oracle 인스턴스에 연결합니다. 필요할 경우 사용자 계정을 생성하고 필요한 권한을 부여합니다.

Note

원본 데이터베이스가 Amazon RDS 인스턴스인 경우 이 단계를 건너뛸 수 있습니다. 이 경우 Amazon RDS 마스터 사용자 계정을 사용하여 데이터를 내보냅니다.

다음 명령은 새 사용자를 생성하고 필요한 권한을 부여합니다.

```
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;
```

3단계: DBMS_DATAPUMP를 사용하여 덤프 파일 생성

SQL Plus 또는 Oracle SQL Developer를 사용하여 관리 사용자 권한으로, 또는 2단계에서 생성한 사용자 권한으로 원본 Oracle 인스턴스에 연결합니다. 소스 데이터베이스가 Amazon RDS Oracle DB 인스턴스인 경우 Amazon RDS 마스터 사용자 권한으로 연결합니다. 그런 다음 Oracle Data Pump 유ти리티를 사용하여 덤프 파일을 생성합니다.

다음 스크립트는 DATA_PUMP_DIR 디렉터리에 sample.dmp라는 덤프 파일을 생성합니다.

```
DECLARE
hdnl NUMBER;
BEGIN
hdnl := DBMS_DATAPUMP.OPEN( operation => 'EXPORT', job_mode => 'SCHEMA', job_name=>null);
DBMS_DATAPUMP.ADD_FILE( handle => hdnl, filename => 'sample.dmp', directory =>
'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_dump_file);
DBMS_DATAPUMP.ADD_FILE( handle => hdnl, filename => 'exp.log', directory =>
'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_log_file);
DBMS_DATAPUMP.METADATA_FILTER(hdnl,'SCHEMA_EXPR','IN ('''SCHEMA_1''')');
DBMS_DATAPUMP.START_JOB(hdnl);
END;
/
```

Note

Data Pump 작업은 비동기로 시작됩니다. Data Pump 작업 모니터링에 대한 자세한 정보는 Oracle 설명서의 [Monitoring Job Status](#)를 참조하십시오.

4단계: 대상 DB 인스턴스의 데이터베이스 링크 생성

원본 인스턴스와 대상 DB 인스턴스 간의 데이터베이스 링크를 생성합니다. 데이터베이스 링크를 생성하고 내보내기 덤프 파일을 전송하려면 로컬 Oracle 인스턴스가 DB 인스턴스와 네트워크로 연결되어 있어야 합니다.

이번 단계에서도 이전 단계와 동일한 사용자 계정에 연결합니다.

동일한 VPC 또는 피어링된 VPC 내에서 두 DB 인스턴스 간에 데이터베이스 링크를 생성하려면 두 DB 인스턴스에 서로에게 이르는 유효한 경로가 있어야 합니다. 각 DB 인스턴스의 보안 그룹은 다른 DB 인스턴스로 (부터)의 수신 및 발신을 허용해야 합니다. 보안 그룹 인바운드 또는 아웃바운드 규칙은 동일한 VPC 또는 피어링된 VPC에서 보안 그룹을 참조할 수 있습니다. 자세한 내용은 [VPC의 DB 인스턴스에 사용하기 위한 데이터베이스 링크 조정 \(p. 850\)](#) 단원을 참조하십시오.

다음 명령은 대상 DB 인스턴스의 Amazon RDS 마스터 사용자에게 연결하는 to_rds라는 데이터베이스 링크를 생성합니다.

```
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>)))';
```

5단계: DBMS_FILE_TRANSFER를 사용하여 내보낸 덤프 파일을 대상 DB 인스턴스로 복사

DBMS_FILE_TRANSFER를 사용하여 원본 데이터베이스의 덤프 파일을 대상 DB 인스턴스로 복사합니다. 다음 스크립트는 원본 인스턴스에 있는 sample.dmp라는 덤프 파일을 to_rds(이전 단계에서 생성됨)라는 대상 데이터베이스 링크로 복사합니다.

```
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;
/
```

6단계: DBMS_DATAPUMP를 사용하여 대상 DB 인스턴스로 데이터 파일 가져오기

DB 인스턴스에서 Oracle Data Pump를 사용하여 스키마를 가져옵니다. METADATA_REMAP 등 추가 옵션이 필요할 수 있습니다.

Amazon RDS 마스터 사용자 계정으로 DB 인스턴스에 연결하여 데이터를 가져옵니다.

```
DECLARE
hdnl NUMBER;
BEGIN
hdnl := DBMS_DATAPUMP.OPEN( operation => 'IMPORT', job_mode => 'SCHEMA', job_name=>null);
DBMS_DATAPUMP.ADD_FILE( handle => hdnl, filename => 'sample_copied.dmp', directory =>
  'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_dump_file);
DBMS_DATAPUMP.METADATA_FILTER(hdnl, 'SCHEMA_EXPR', 'IN (''SCHEMA_1'')');
DBMS_DATAPUMP.START_JOB(hdnl);
END;
/
```

DB 인스턴스에서 해당 사용자의 테이블을 보고 데이터 가져오기를 확인할 수 있습니다. 예를 들어 다음 쿼리는 *schema_1*의 테이블 수를 반환합니다.

```
select count(*) from dba_tables where owner='SCHEMA_1';
```

7단계: 정리

데이터를 가져온 후에는 유지하지 않을 파일을 삭제할 수 있습니다. 다음 명령을 사용하여 DATA_PUMP_DIR에서 파일을 나열할 수 있습니다.

```
select * from table(RDSADMIN.RDS_FILE_UTIL.LISTDIR('DATA_PUMP_DIR')) order by mtime;
```

DATA_PUMP_DIR에서 더 이상 필요하지 않은 파일을 삭제하려면 다음 명령을 사용합니다.

```
exec utl_file.fremove('DATA_PUMP_DIR','<file name>');
```

예를 들어, 다음 명령은 "sample_copied.dmp"라는 파일을 삭제합니다.

```
exec utl_file.fremove('DATA_PUMP_DIR','sample_copied.dmp');
```

Oracle Export/Import 유틸리티

Oracle Export/Import 유틸리티는 데이터 크기가 작고 이진 부동 소수점 수 및 실수 등의 데이터 형식이 필요하지 않은 마이그레이션에 적합합니다. 가져오기 프로세스는 스키마 객체를 생성하므로 스키마 객체를 생성하기 위해 먼저 스크립트를 실행할 필요가 없습니다. 따라서 이 프로세스는 작은 테이블을 포함하는 데이터 베이스에 매우 적합합니다. 다음 예에서는 이러한 유틸리티를 사용하여 특정 테이블을 내보내고 가져오는 방법을 설명합니다.

Oracle 내보내기 및 가져오기 유틸리티를 다운로드 하려면 <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>로 이동하십시오.

아래 명령을 사용하여 원본 데이터베이스에서 테이블을 내보냅니다. 해당되는 경우 사용자 이름/암호를 대체합니다.

```
exp cust_dba@ORCL FILE=exp_file.dmp TABLES=(tab1,tab2,tab3) LOG=exp_file.log
```

내보내기 프로세스에서는 지정된 테이블에 대한 스키마 및 데이터를 모두 포함하는 이진 덤프 파일을 생성합니다. 이제 다음 명령을 사용하여 이 스키마와 데이터를 대상 데이터베이스로 가져올 수 있습니다.

```
imp cust_dba@targetdb FROMUSER=cust_schema TOUSER=cust_schema \
TABLES=(tab1,tab2,tab3) FILE=exp_file.dmp LOG=imp_file.log
```

사용자의 필요에 맞는 다양한 Export 및 Import 명령 버전이 있습니다. 자세한 정보는 Oracle 설명서를 참조하십시오.

Oracle SQL*Loader

Oracle SQL*Loader는 객체 수가 제한되어 있는 대규모 데이터베이스에 매우 적합합니다. 원본 데이터베이스에서 내보내고 대상 데이터베이스로 로드하는 프로세스는 스키마와 매우 밀접한 관계가 있으므로, 다음 예에서는 샘플 스키마 객체를 생성하여 원본에서 내보내고 대상 데이터베이스에 로드합니다.

Oracle SQL*Loader를 다운로드 하려면 <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>로 이동하십시오.

1. 아래 명령을 사용하여 샘플 원본 테이블을 생성합니다.

```
create table customer_0 tablespace users as select rnum id, o.* from
all_objects o, all_objects x where rnum <= 1000000;
```

2. 대상 Amazon RDS 인스턴스에서 데이터를 로드하는 데 사용되는 대상 테이블을 생성합니다.

```
create table customer_1 tablespace users as select 0 as id, owner,
object_name, created from all_objects where 1=2;
```

3. 원본 데이터베이스의 데이터를 일반 파일(구분 기호 사용)로 내보냅니다. 이 예에서는 이를 위해 SQL*Plus를 사용합니다. 실제 데이터의 경우 대개 데이터베이스의 모든 객체에 대한 내보내기를 수행하는 스크립트를 생성해야 할 것입니다.

```
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. 데이터를 설명하는 제어 파일을 생성해야 합니다. 앞에서 말했듯이, 데이터에 따라 이 단계를 수행하는 스크립트를 빌드해야 합니다.

```
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"
)
```

필요할 경우 이전 코드에서 생성된 파일을 Amazon EC2 인스턴스 등의 스테이징 영역으로 복사합니다.

5. 마지막으로 대상 데이터베이스에 대한 적절한 사용자 이름 및 암호와 함께 SQL*Loader를 사용하여 데이터를 가져옵니다.

```
sqlldr cust_dba@targetdb control=sqlldr_1.ctl BINDSIZE=10485760 READSIZE=10485760
ROWS=1000
```

Oracle 구체화 보기

Oracle 구체화 보기 복제를 사용하여 최신 데이터 세트를 효율적으로 마이그레이션할 수 있습니다. 복제를 통해 대상 테이블을 지속적으로 원본과 동일한 상태로 유지할 수 있으므로 필요할 경우 나중에 Amazon RDS로 실제 전환을 수행할 수 있습니다. 복제는 Amazon RDS 인스턴스에서 원본 데이터베이스로 연결되는 데이터베이스 링크를 사용하여 설정됩니다.

구체화 보기의 한 가지 요구 사항은 대상 데이터베이스에서 원본 데이터베이스로의 액세스를 허용해야 한다는 점입니다. 다음 예시에서는 SQLNet을 통해 Amazon RDS 대상 데이터베이스가 원본에 연결할 수 있도록 원본 데이터베이스에서 액세스 규칙이 활성화되었습니다.

1. 원본 및 Amazon RDS 대상 인스턴스 모두에서 동일한 암호로 인증할 수 있는 사용자 계정을 생성합니다.

```
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. 새로 생성된 dblink_user를 사용하여 Amazon RDS 대상 인스턴스에서 원본 인스턴스로 연결되는 데이터베이스 링크를 생성합니다.

```
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. 링크를 테스트합니다.

```
select * from v$instance@remote_site;
```

4. 원본 인스턴스에서 주 키와 구체화 보기 로그를 사용하여 샘플 테이블을 생성합니다.

```
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. 대상 Amazon RDS 인스턴스에서 구체화된 보기 생성합니다.

```
CREATE MATERIALIZED VIEW customer_0 BUILD IMMEDIATE REFRESH FAST AS
SELECT * FROM cust_dba.customer_0@remote_site;
```

6. 대상 Amazon RDS 인스턴스에서 구체화된 보기 새로 고칩니다.

```
exec DBMS_MV.REFRESH('CUSTOMER_0', 'f');
```

7. 구체화된 보기 끊고 PRESERVE TABLE 절을 포함하여 구체화된 보기 컨테이너 테이블과 그 내용을 보관합니다.

```
DROP MATERIALIZED VIEW customer_0 PRESERVE TABLE;
```

보관한 테이블에는 끊긴 구체화된 보기와 같은 이름이 있습니다.

Amazon RDS의 Oracle 읽기 전용 복제본 작업

일반적으로 읽기 전용 복제본을 사용하여 Amazon RDS DB 인스턴스 간 복제를 구성합니다. 읽기 전용 복제본에 대한 일반적인 정보는 [읽기 전용 복제본 작업 \(p. 135\)](#) 단원을 참조하십시오.

이 단원에서는 Oracle용 Amazon RDS에서의 읽기 전용 복제본 작업에 대한 특정 정보를 찾을 수 있습니다.

주제

- [Oracle용 읽기 전용 복제본 구성 \(p. 768\)](#)
- [Oracle을 사용한 읽기 전용 복제본 제한 \(p. 769\)](#)
- [Oracle 읽기 전용 복제본 문제 해결 \(p. 770\)](#)

Oracle용 읽기 전용 복제본 구성

Oracle 읽기 전용 복제본은 Oracle 데이터 보호를 사용하여 소스 DB 인스턴스의 데이터베이스 변경 사항을 읽기 전용 복제본으로 복제합니다. Oracle 데이터 보호에 대한 자세한 내용은 Oracle 설명서의 [Oracle 데이터 보호 개념 및 관리](#)를 참조하십시오.

임의의 DB 인스턴스를 원본 DB 인스턴스로 사용하려면 원본 DB 인스턴스의 자동 백업을 활성화해야 합니다. 이렇게 하려면 백업 보존 기간을 0 이외의 값으로 설정합니다. DB 인스턴트에서 강제 로깅 모드를 활성화하는 것이 좋습니다. 강제 로깅 모드를 활성화하려면 DB 인스턴스에 연결하고 다음 절차를 실행하십시오.

```
exec rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

이 절차에 대한 자세한 내용은 [강제 로깅 설정 \(p. 852\)](#) 단원을 참조하십시오.

로깅 구성을 변경하려는 경우 DB 인스턴스를 하나 이상의 읽기 전용 복제본의 소스로 만들기 전에 변경을 완료하는 것이 좋습니다. Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.add_logfile` 및 `rdsadmin.rdsadmin_util.drop_logfile`을 사용하여 DB 인스턴스의 로깅 구성을 수정할 수 있습니다. 자세한 내용은 [온라인 다시 실행 로그 추가 \(p. 853\)](#) 및 [온라인 다시 실행 로그 드롭 \(p. 854\)](#) 단원을 참조하십시오. 가장 좋은 방법은 읽기 전용 복제본을 만든 후에 로깅 구성을 수정하지 않는 것이 좋습니다. 수정으로 인해 온라인 재실행 로깅 구성이 대기 로깅 구성과 동기화되지 않을 수 있습니다.

Oracle 읽기 전용 복제본을 생성할 때는 마스터 DB 인스턴스를 중단할 필요가 없습니다. Amazon RDS가 원본 DB 인스턴스와 읽기 전용 복제본에 필요한 파라미터와 권한을 설정하기 때문에 서비스 중단은 발생하지 않습니다. 원본 DB 인스턴스를 캡처한 스냅샷이 읽기 전용 복제본이 됩니다. 읽기 전용 복제본을 삭제하더라도 중단은 발생하지 않습니다.

원본 DB 인스턴스 하나에서 최대 5개까지 읽기 전용 복제본을 생성할 수 있습니다. 읽기 전용 복제본을 만들기 전에 소스 DB 인스턴스와 읽기 전용 복제본의 `max_string_size` 파라미터 설정이 동일한지 확인하십시오. 동일한 파라미터 그룹과 연결하여 이 작업을 수행할 수 있습니다. 소스 및 읽기 전용 복제본에 서로 다른 파라미터 그룹이 있는 경우 `max_string_size`를 동일한 값으로 설정하여 이 작업을 수행할 수 있습니다.

소스 DB 인스턴스와 모든 읽기 전용 복제본의 Oracle DB 엔진 버전이 동일해야 합니다. Amazon RDS는 읽기 전용 복제본의 유지 관리 기간과 상관없이 소스 DB 인스턴스를 업그레이드 한 직후에 읽기 전용 복제본을 업그레이드합니다. DB 엔진 버전 업그레이드에 대한 자세한 내용은 [Oracle DB 엔진 업그레이드 \(p. 752\)](#) 단원을 참조하십시오.

읽기 전용 복제본이 소스에서 변경 사항을 받아 적용하려면 충분한 컴퓨팅 및 저장소 리소스가 있어야 합니다. 읽기 전용 복제본이 컴퓨팅, 네트워크 또는 스토리지 리소스 용량에 도달하면 읽기 전용 복제본은 소스에서 변경 사항을 수신하거나 적용하는 것을 중지합니다. 읽기 전용 복제본의 스토리지 및 CPU 리소스를 원본 및 다른 읽기 전용 복제본과 별도로 수정할 수 있습니다.

Oracle을 사용한 읽기 전용 복제본 제한

Oracle 읽기 전용 복제본의 제한은 다음과 같습니다.

- 활성 데이터 보호 라이선스가 있어야 합니다.
- Oracle 읽기 전용 복제본은 Oracle Enterprise Edition(EE) 엔진에서만 사용할 수 있습니다.
- Oracle 읽기 전용 복제본은 Oracle 버전 12.1.0.2.v10 이상 12.1 버전 및 모든 12.2 버전에서만 사용할 수 있습니다.
- Oracle 읽기 전용 복제본은 EC2-VPC 플랫폼의 DB 인스턴스에만 사용할 수 있습니다.
- Oracle 읽기 전용 복제본은 두 개 이상의 vCPU가 있는 DB 인스턴스 클래스에서 실행 중인 DB 인스턴스에 서만 사용할 수 있습니다.
- Oracle용 Amazon RDS는 개입을 통해 소스 DB 인스턴스와 읽기 전용 복제본 간의 긴 복제 지연을 완화하지 않습니다. 원본 DB 인스턴스와 해당 읽기 전용 복제본이 운영 로드에 맞게 컴퓨팅 및 스토리지 면에서 제대로 크기가 조정되었는지 확인합니다.
- Oracle 읽기 전용 복제본은 Amazon RDS는 소스 데이터베이스와 동일한 옵션 그룹에 속해야 합니다. 소스 옵션 그룹 또는 소스 옵션 그룹 멤버십에 대한 수정 사항은 읽기 전용 복제본으로 전파됩니다. 이러한 변경 사항은 읽기 전용 복제본의 유지 관리 기간과 상관없이 소스 DB 인스턴스에 적용된 직후 읽기 전용 복제본에 적용됩니다.

옵션 그룹에 대한 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오.

- 교차 리전 읽기 전용 복제본은 만들 수 없습니다.
- 현재 Oracle 읽기 전용 복제본은 EU(스톡홀름) 및 중국(닝샤) 리전에 지원되지 않습니다.

Oracle 읽기 전용 복제본 문제 해결

Amazon CloudWatch에서 Amazon RDS ReplicaLag 지표를 보고 복제 지연을 모니터링할 수 있습니다. 복제 지연 시간에 대한 자세한 내용은 [읽기 전용 복제본 모니터링 \(p. 146\)](#) 단원을 참조하십시오.

복제 지연이 너무 긴 경우, 지연에 대한 정보가 담긴 다음 보기의 쿼리할 수 있습니다.

- V\$ARCHIVED_LOG – 읽기 전용 복제본에 적용된 커밋을 표시합니다.
- V\$DATAGUARD_STATS – replicaLag 지표를 구성하는 구성 요소의 세부 분류를 표시합니다.
- V\$DATAGUARD_STATUS – Oracle의 내부 복제 프로세스의 로그 출력을 표시합니다.

Amazon RDS에서 지원되는 Oracle 문자 집합

다음 표에는 Amazon RDS에서 지원되는 Oracle 데이터베이스 문자 집합이 나와 있습니다. AWS CLI [create-db-instance](#) 명령의 --character-set-name 파라미터 또는 Amazon RDS API [CreateDBInstance](#) 작업의 CharacterSetName 파라미터와 함께 이 표의 값을 사용할 수 있습니다.

클라이언트 환경에서 NLS_LANG 환경 파라미터를 설정하는 것이 가장 간단하게 Oracle에 대한 로캘 동작을 지정하는 방법입니다. 이 파라미터는 클라이언트 애플리케이션과 데이터베이스 서버에서 사용되는 언어와 지역을 설정합니다. 또한, 이 파라미터는 클라이언트 애플리케이션에서 입력되거나 표시되는 데이터에 대한 문자 집합에 해당하는 클라이언트의 문자 집합을 표시합니다. Amazon RDS를 사용하여 DB 인스턴스를 만들 때 문자 집합을 설정할 수 있습니다. NLS_LANG 및 문자 집합에 대한 자세한 정보는 Oracle 문서의 [What is a Character set or Code?](#)를 참조하십시오.

값	설명
AL32UTF8	유니코드 5.0 UTF-8 범용 문자 집합(기본값)
AR8ISO8859P6	ISO 8859-6 라틴어/아랍어
AR8MSWIN1256	Microsoft Windows 코드 페이지 1256 8비트 라틴어/아랍어
BLT8ISO8859P13	ISO 8859-13 발트어
BLT8MSWIN1257	Microsoft Windows 코드 페이지 1257 8비트 발트어
CL8ISO8859P5	ISO 8859-5 라틴어/키릴 자모
CL8MSWIN1251	Microsoft Windows 코드 페이지 1251 8비트 라틴어/키릴 자모
EE8ISO8859P2	ISO 8859-2 동유럽어
EL8ISO8859P7	ISO 8859-7 라틴어/그리스어
EE8MSWIN1250	Microsoft Windows 코드 페이지 1250 8비트 동유럽어
EL8MSWIN1253	Microsoft Windows 코드 페이지 1253 8비트 라틴어/그리스어

값	설명
IW8ISO8859P8	ISO 8859-8 라틴어/하브리어
IW8MSWIN1255	Microsoft Windows 코드 페이지 1255 8비트 라틴어/하브리어
JA16EUC	EUC 24비트 일본어
JA16EUCTILDE	유니코드와의 사이에서 물결표 매핑을 제외하면 JA16EUC와 동일
JA16SJIS	Shift-JIS 16비트 일본어
JA16SJISTILDE	유니코드와의 사이에서 물결표 매핑을 제외하면 JA16SJIS와 동일
KO16MSWIN949	Microsoft Windows 코드 페이지 949 한국어
NE8ISO8859P10	ISO 8859-10 북유럽어
NEE8ISO8859P4	ISO 8859-4 북유럽 및 북동 유럽어
TH8TISASCII	태국 산업 표준 620-2533-ASCII 8비트
TR8MSWIN1254	Microsoft Windows 코드 페이지 1254 8비트 터키어
US7ASCII	ASCII 7비트 영어
UTF8	유니코드 3.0 UTF-8 범용 문자 집합, CESU-8 준수
VN8MSWIN1258	Microsoft Windows 코드 페이지 1258 8비트 베트남어
WE8ISO8859P1	서유럽어 8비트 ISO 8859 Part 1
WE8ISO8859P15	ISO 8859-15 서유럽어
WE8ISO8859P9	ISO 8859-9 서유럽어 및 터키어
WE8MSWIN1252	Microsoft Windows 코드 페이지 1252 8비트 서유럽어
ZHS16GBK	GBK 16비트 중국어 간체
ZHT16HKSCS	Microsoft Windows 코드 페이지 950, 홍콩 보조 문자 집합 HKSCS-2001 포함. 문자 집합 번환은 유니코드 3.0을 기반으로 합니다.
ZHT16MSWIN950	Microsoft Windows 코드 페이지 950 중국어 번체
ZHT32EUC	EUC 32비트 중국어 번체

Amazon RDS의 Oracle DB 인스턴스에 대한 인스턴스 수준에서 다음 National Language Support(NLS) 초기화 파라미터를 설정할 수도 있습니다.

- NLS_DATE_FORMAT
- NLS_LENGTH_SEMANTICS
- NLS_NCHAR_CONV_EXCP
- NLS_TIME_FORMAT

- NLS_TIME_TZ_FORMAT
- NLS_TIMESTAMP_FORMAT
- NLS_TIMESTAMP_TZ_FORMAT

인스턴스 파라미터 수정에 대한 자세한 정보는 [DB 파라미터 그룹 작업 \(p. 161\)](#)을 참조하십시오.

SQL 클라이언트에서 다른 NLS 초기화 파라미터를 설정할 수 있습니다. 예를 들어 다음 명령문은 Oracle DB 인스턴스에 연결된 SQL 클라이언트에서 NLS_LANGUAGE 초기화 파라미터를 GERMAN으로 설정합니다.

```
ALTER SESSION SET NLS_LANGUAGE=GERMAN;
```

SQL 클라이언트를 사용하여 Oracle DB 인스턴스에 연결하는 방법에 대한 자세한 정보는 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 \(p. 736\)](#)을 참조하십시오.

Oracle DB 인스턴스 옵션

다음에는 Oracle DB 엔진을 실행하는 Amazon RDS 인스턴스에 사용 가능한 옵션 또는 추가 기능에 대한 설명이 나와 있습니다. 이러한 옵션을 활성화 하려면 먼저 옵션 그룹에 추가한 다음 옵션 그룹을 DB 인스턴스에 연결해야 합니다. 자세한 정보는 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오.

일부 옵션은 DB 인스턴스에서 실행하려면 추가 메모리가 필요합니다. 예를 들어 Oracle Enterprise Manager Database Control은 약 300MB의 RAM을 사용합니다. 작은 DB 인스턴스에 대해 이 옵션을 활성화할 경우 메모리 제약으로 인해 성능 문제가 발생할 수 있습니다. 데이터베이스에서 RAM에 대한 필요성이 줄어들도록 Oracle 파라미터를 조정할 수 있습니다. 또는 더 큰 DB 인스턴스로 확장할 수 있습니다.

Amazon RDS는 Oracle DB 인스턴스에 대해 다음 옵션을 지원합니다.

옵션	옵션 ID
Amazon S3통합 (p. 773)	S3_INTEGRATION
Oracle Application Express (p. 783)	APEX
	APEX-DEV
Oracle Enterprise Manager (p. 791)	OEM
	OEM_AGENT
Oracle Java 가상 머신 (p. 798)	JVM
Oracle 레이블 보안 (p. 801)	OLS
Oracle Locator (p. 803)	LOCATOR
Oracle Multimedia (p. 805)	MULTIMEDIA
Oracle 기본 네트워크 암호화 (p. 807)	NATIVE_NETWORK_ENCRYPTION
Oracle 보안 소켓 Layer (p. 810)	SSL
Oracle Spatial (p. 815)	SPATIAL
Oracle SQLT (p. 817)	SQLT

옵션	옵션 ID
Oracle Statspack (p. 822)	STATSPACK
Oracle 시간대 (p. 824)	Timezone
Oracle Transparent Data Encryption (p. 827)	TDE
Oracle UTL_MAIL (p. 828)	UTL_MAIL
Oracle XML DB (p. 831)	XMLDB

Amazon S3통합

Oracle용 Amazon RDS DB 인스턴스와 Amazon S3 버킷 사이에서 파일을 전송할 수 있습니다. Data Pump 와 같은 Oracle 기능과의 Amazon S3 통합을 사용할 수 있습니다. 예를 들어 Amazon S3의 Data Pump 파일을 DB 인스턴스 호스트로 다운로드할 수 있습니다.

Note

DB 인스턴스와 Amazon S3 버킷은 같은 AWS 리전에 있어야 합니다.

주제

- [Amazon RDS Oracle과 Amazon S3의 통합을 위한 사전 조건 \(p. 773\)](#)
- [Amazon S3 통합 옵션 추가 \(p. 778\)](#)
- [Oracle용 Amazon RDS와 Amazon S3 버킷 사이의 파일 전송 \(p. 779\)](#)
- [Amazon S3 통합 옵션 제거 \(p. 783\)](#)

Amazon RDS Oracle과 Amazon S3의 통합을 위한 사전 조건

Oracle용 Amazon RDS와 Amazon S3를 통합하는 작업을 하려면 Amazon RDS DB 인스턴스에 Amazon S3 버킷에 대한 액세스가 있어야 합니다. 이를 위해 AWS Identity and Access Management(IAM) 정책과 IAM 역할을 만들습니다.

Note

DB 인스턴스에 역할을 추가하려면 DB 인스턴스의 상태가 available이어야 합니다.

콘솔

Amazon S3 버킷에 Amazon RDS 액세스를 허용하는 IAM 정책을 생성하려면

1. [IAM 관리 콘솔](#)을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. [Create policy]를 선택합니다.
4. [Visual editor] 탭에서 [Choose a service]를 선택한 다음 [S3]을 선택합니다.
5. 작업에서 Expand all(모두 확장)을 선택한 다음 IAM 정책에 필요한 버킷 권한 및 객체 권한을 선택합니다.

필요한 액세스 유형에 따라 정책에 적절한 조치를 포함합니다.

- `GetObject` – Amazon S3 버킷에서 Amazon RDS로 파일을 전송하는 데 필요합니다.
- `ListBucket` – Amazon S3 버킷에서 Amazon RDS로 파일을 전송하는 데 필요합니다.
- `PutObject` – Amazon RDS에서 Amazon S3 버킷으로 파일을 전송하는 데 필요합니다.

객체 권한은 Amazon S3의 객체 작업에 대한 권한으로, 버킷 자체가 아닌 버킷의 객체에 대해 부여해야 합니다. Amazon S3의 객체 작업의 권한에 대한 자세한 내용은 [객체 작업에 대한 권한](#)을 참조하십시오.

6. 리소스를 선택하고 버킷에 대해 Add ARN(ARN 추가)을 선택합니다.
7. [Add ARN(s)] 대화 상자에서 리소스에 대한 세부 정보를 제공하고 [Add]를 선택합니다.

액세스를 허용할 Amazon S3 버킷을 지정하십시오. 예를 들어 Amazon RDS가 example-bucket라는 Amazon S3 버킷에 액세스할 수 있게 허용하려면 Amazon 리소스 이름(ARN) 값을 arn:aws:s3:::example-bucket으로 설정하십시오.

8. [object] 리소스가 나열되면 객체에 대해 [Add ARN]을 선택합니다.
9. [Add ARN(s)] 대화 상자에서 리소스에 대한 세부 정보를 제공합니다.

Amazon S3 버킷의 경우 액세스를 허용할 Amazon S3 버킷을 지정하십시오. 객체의 경우 [Any]를 선택하여 버킷의 모든 객체에 권한을 허용할 수 있습니다.

Note

Amazon Resource Name(ARN)(Amazon 리소스 이름(ARN))을 더 구체적인 ARN 값으로 설정하여 Amazon RDS에서 Amazon S3 버킷의 특정 파일 또는 폴더에만 액세스하도록 허용할 수 있습니다. Amazon S3에 대한 액세스 정책을 정의하는 방법에 대한 자세한 내용은 [Amazon S3 리소스에 대한 액세스 권한 관리](#) 단원을 참조하십시오.

10. (선택 사항) Add additional permissions(권한 추가)를 선택하여 다른 Amazon S3 버킷을 정책에 추가하고 버킷에 대해 이전 단계를 반복합니다.

Note

이 단계를 반복하여 Amazon RDS에서 액세스할 각 Amazon S3 버킷의 정책에 해당 버킷 권한 설명문을 추가할 수 있습니다. 또는 Amazon S3의 모든 버킷 및 객체에 대한 액세스 권한을 부여할 수도 있습니다.

11. [Review policy]를 선택합니다.
12. 이름에서 IAM 정책의 이름을 입력합니다(예: rds-s3-integration-policy). IAM 역할을 만들어 DB 인스턴스와 연결할 때 이 이름을 사용합니다. Description 값(선택 사항)을 추가할 수도 있습니다.
13. [Create policy]를 선택합니다.

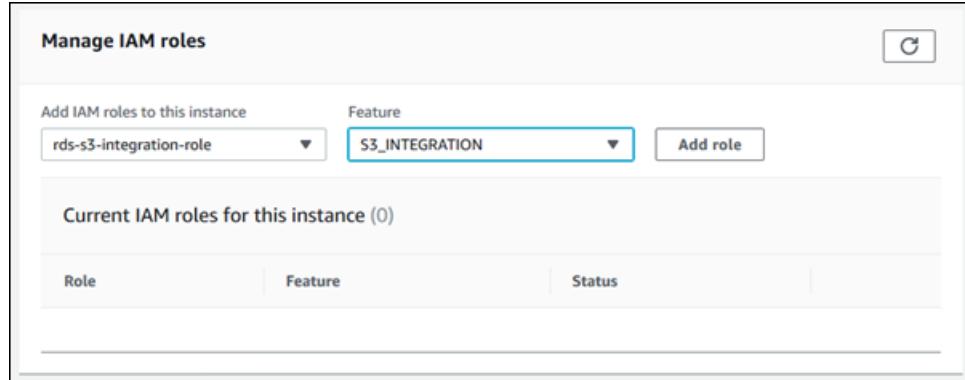
Amazon S3 버킷에 Amazon RDS 액세스를 허용하는 IAM 역할을 생성하려면

1. 탐색 창에서 역할을 선택합니다.
2. Create role을 선택합니다.
3. AWS 서비스에서 RDS를 선택합니다.
4. Select your use case(사용 사례 선택)를 선택하려면 RDS- Add Role to Database(데이터베이스에 역할 추가)를 선택하십시오.
5. 다음: 권한을 선택합니다.
6. Attach permissions policies(권한 정책 연결) 아래 검색에서 생성한 IAM 정책의 이름을 입력하고, 목록에 표시된 정책을 선택하십시오.
7. Next: Tags(다음: 태그)를 선택한 후 Next: Review(다음: 검토)를 선택합니다.
8. 역할 이름을 IAM 역할의 이름으로 설정합니다(예: rds-s3-integration-role). Role description(역 할 설명) 값(선택 사항)을 추가할 수도 있습니다.
9. [Create Role]을 선택합니다.

IAM 역할을 DB 인스턴스와 연결하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.

2. 세부 정보를 표시하고자 하는 Oracle DB 인스턴스 이름을 선택합니다.
3. Connectivity & security(연결성 및 보안) 탭에 있는 Manage IAM roles(IAM 역할 관리) 섹션의 이 인스턴스에 IAM 역할 추가에서 추가할 역할을 선택합니다.
4. 기능에서 S3_INTEGRATION을 선택하십시오.



5. [Add role]을 선택합니다.

AWS CLI

Amazon S3 버킷에 대한 액세스 권한을 Amazon RDS에 부여하려면

1. Amazon S3 버킷에 대한 액세스 권한을 Amazon RDS에 부여하는 AWS Identity and Access Management(IAM) 정책을 생성합니다.

필요한 액세스 유형에 따라 정책에 적절한 조치를 포함합니다.

- GetObject – Amazon S3 버킷에서 Amazon RDS로 파일을 전송하는 데 필요합니다.
- ListBucket – Amazon S3 버킷에서 Amazon RDS로 파일을 전송하는 데 필요합니다.
- PutObject – Amazon RDS에서 Amazon S3 버킷으로 파일을 전송하는 데 필요합니다.

다음 AWS CLI 명령은 이 옵션으로 **rds-s3-integration-policy**라는 이름의 IAM 정책을 만듭니다. **your-s3-bucket-arn**이라는 버킷에 대한 액세스 권한을 부여합니다.

Example

Linux, OS X, Unix의 경우:

```
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"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::your-s3-bucket-arn",
                "arn:aws:s3:::your-s3-bucket-arn/*"
            ]
    ]
}'
```

```
        }
    ]
}'
```

Windows의 경우:

```
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"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::your-s3-bucket-arn",
        "arn:aws:s3:::your-s3-bucket-arn/*"
      ]
    }
  ]
}'
```

2. 정책을 만든 후에 정책의 Amazon 리소스 이름(ARN)을 기록하십시오. 이후 단계에 이 ARN이 필요합니다.
3. Amazon RDS가 Amazon S3 버킷에 액세스하기 위해 사용자 대신 가정할 수 있는 IAM 역할을 만듭니다.

다음 AWS CLI 명령은 이 목적으로 **rds-s3-integration-role**을 만듭니다.

Example

Linux, OS X, Unix의 경우:

```
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"
    }
  ]
}'
```

Windows의 경우:

```
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"
        }
    ]
}'
```

자세한 내용은 IAM사용 설명서의 역할을 만들어 [IAM 사용자에게 권한 위임](#) 단원을 참조하십시오.

4. 역할이 생성되면 역할의 ARN을 기록하십시오. 이후 단계에 이 ARN이 필요합니다.
5. 생성한 정책을 생성한 역할에 연결하십시오.

다음 AWS CLI 명령은 정책을 *rds-s3-integration-role*이라는 역할에 연결합니다.

Example

Linux, OS X, Unix의 경우:

```
aws iam attach-role-policy \
--policy-arn your-policy-arn \
--role-name rds-s3-integration-role
```

Windows의 경우:

```
aws iam attach-role-policy ^
--policy-arn your-policy-arn ^
--role-name rds-s3-integration-role
```

*your-policy-arn*을 이전 단계에서 기록한 정책 ARN으로 바꾸십시오.

6. Oracle DB 인스턴스에 역할을 추가하십시오.

다음 AWS CLI 명령은 *mydbinstance*라는 Oracle DB 인스턴스에 역할을 추가합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds add-role-to-db-instance \
--db-instance-identifier mydbinstance \
--feature-name S3_INTEGRATION \
--role-arn your-role-arn
```

Windows의 경우:

```
aws rds add-role-to-db-instance ^
--db-instance-identifier mydbinstance ^
--feature-name S3_INTEGRATION ^
--role-arn your-role-arn
```

*your-role-arn*을 이전 단계에서 기록한 역할 ARN으로 바꿉니다. --feature-name 옵션에 대해 S3_INTEGRATION을 지정해야 합니다.

Amazon S3 통합 옵션 추가

Oracle용 Amazon RDS를 Amazon S3와의 통합에 사용하려면, Amazon RDS Oracle DB 인스턴스가 S3_INTEGRATION 옵션을 포함하는 옵션 그룹과 연결되어 있어야 합니다.

콘솔

Amazon S3 통합을 위한 옵션 그룹을 구성하려면

- 새 옵션 그룹을 만들거나 S3_INTEGRATION 옵션을 추가할 수 있는 기존 옵션 그룹을 식별하십시오.
옵션 그룹의 생성에 대한 자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.
- [S3_INTEGRATION] 옵션을 옵션 그룹에 추가합니다.
옵션 그룹에 옵션을 추가하는 방법에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#) 단원을 참조하십시오.
- 새 Oracle DB 인스턴스를 생성하고 옵션 그룹을 연결하거나, Oracle DB 인스턴스를 수정하여 옵션 그룹을 연결합니다.
Oracle DB 인스턴스 생성에 대한 정보는 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
Oracle DB 인스턴스 수정에 대한 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

AWS CLI

Amazon S3 통합을 위한 옵션 그룹을 구성하려면

- 새 옵션 그룹을 만들거나 S3_INTEGRATION 옵션을 추가할 수 있는 기존 옵션 그룹을 식별하십시오.
옵션 그룹의 생성에 대한 자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.
- [S3_INTEGRATION] 옵션을 옵션 그룹에 추가합니다.
예를 들어 다음 AWS CLI 명령은 S3_INTEGRATION 옵션을 *myoptiongroup*이라는 옵션 그룹에 추가합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds add-option-to-option-group \
--option-group-name myoptiongroup \
--options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

Windows의 경우:

```
aws rds add-option-to-option-group ^
--option-group-name myoptiongroup ^
--options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

3. 새 Oracle DB 인스턴스를 생성하고 옵션 그룹을 연결하거나, Oracle DB 인스턴스를 수정하여 옵션 그룹을 연결합니다.

Oracle DB 인스턴스 생성에 대한 정보는 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.

Oracle DB 인스턴스 수정에 대한 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

Oracle용 Amazon RDS와 Amazon S3 버킷 사이의 파일 전송

Amazon RDS 프로시저를 사용하여 Oracle DB 인스턴스의 파일을 Amazon S3 버킷으로 업로드할 수 있습니다. 또한 Amazon RDS 프로시저를 사용하여 Amazon S3 버킷에서 Oracle DB 인스턴스로 파일을 다운로드 할 수도 있습니다.

주제

- [Oracle DB 인스턴스에서 Amazon S3 버킷으로 파일 업로드 \(p. 779\)](#)
- [Amazon S3 버킷의 파일을 Oracle DB 인스턴스로 다운로드 \(p. 781\)](#)
- [파일 전송 상태 모니터링 \(p. 782\)](#)

Oracle DB 인스턴스에서 Amazon S3 버킷으로 파일 업로드

Oracle DB 인스턴스에서 Amazon S3 버킷으로 파일을 업로드하려면 Amazon RDS 프로시저 `rdsadmin.rdsadmin_s3_tasks.upload_to_s3`를 사용하십시오. 예를 들어 Oracle Recovery Manager(RMAN) 백업 파일을 업로드할 수 있습니다. RMAN 백업 수행에 대한 자세한 내용은 [Oracle DB 인스턴스에 대한 공통 DBA RMAN\(Recovery Manager\) 작업 \(p. 858\)](#) 단원을 참조하십시오.

`rdsadmin.rdsadmin_s3_tasks.upload_to_s3` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	필수 여부	설명
<code>p_bucket_name</code>	VARCHAR2	-	필수	파일을 업로드할 Amazon S3 버킷의 이름입니다.
<code>p_directory_name</code>	VARCHAR2	-	필수	파일을 업로드할 Oracle 디렉터리 객체의 이름입니다. 이 디렉터리는 사용자가 생성한 디렉터리 객체 또는 Data Pump 디렉터리(예: <code>DATA_PUMP_DIR</code>)일 수 있습니다. Note 지정된 디렉터리의 파일만 업로드할 수 있습니다. 지정된 디렉터리의 하위 디렉터리에서는 파일

파라미터 이름	데이터 형식	기본값	필수 여부	설명
				을 업로드할 수 없습니다.
p_s3_prefix	VARCHAR2	-	필수	<p>파일이 업로드되는 Amazon S3 파일 이름 접두사입니다. 빈 접두사는 모든 파일을 지정된 Amazon S3 버킷의 최상위 레벨에 업로드하며, 접두사를 파일 이름에 추가하지 않습니다.</p> <p>예를 들어 접두사가 folder_1/oradb이면 파일이 folder_1에 업로드됩니다. 이 경우 oradb 접두사가 각 파일에 추가됩니다.</p>
p_prefix	VARCHAR2	-	필수	업로드되기 위해 파일 이름과 일치해야 하는 파일 이름 접두사입니다. 빈 접두사는 지정된 디렉터리의 모든 파일을 업로드합니다.

다음 예제에서는 **DATA_PUMP_DIR** 디렉터리에 있는 모든 파일을 **mys3bucket**이라는 이름의 Amazon S3 버킷에 업로드합니다.

```
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;
```

다음 예제에서는 **DATA_PUMP_DIR** 디렉터리에 있는 **db** 접두사의 모든 파일을 **mys3bucket**이라는 이름의 Amazon S3 버킷에 업로드합니다.

```
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;
```

다음 예제에서는 **DATA_PUMP_DIR** 디렉터리에 있는 모든 파일을 **mys3bucket**이라는 이름의 Amazon S3 버킷에 업로드합니다. 파일은 **dbfiles** 폴더에 업로드됩니다.

```
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;
```

다음 예제에서는 **DATA_PUMP_DIR** 디렉터리에 있는 모든 파일을 **mys3bucket**이라는 이름의 Amazon S3 버킷에 업로드합니다. 파일은 dbfiles 폴더에 업로드되고 ora는 각 파일 이름의 시작 부분에 추가됩니다.

```
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;
```

각 예에서, SELECT 문은 VARCHAR2 데이터 형식으로 작업 ID를 반환합니다.

Amazon S3 버킷의 파일을 Oracle DB 인스턴스로 다운로드

Oracle DB 인스턴스로 Amazon S3 버킷의 파일을 다운로드하려면 Amazon RDS 프로시저 `rdsadmin.rdsadmin_s3_tasks.download_from_s3`를 사용하십시오.
`rdsadmin.rdsadmin_s3_tasks.download_from_s3` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	필수 여부	설명
p_bucket_name	VARCHAR2	-	필수	파일을 다운로드할 수 있는 Amazon S3 버킷의 이름입니다.
p_directory_name	VARCHAR2	-	필수	파일을 다운로드할 Oracle 디렉터리 객체의 이름입니다. 이 디렉터리는 사용자가 생성한 디렉터리 객체 또는 Data Pump 디렉터리(예: DATA_PUMP_DIR)일 수 있습니다.
p_s3_prefix	VARCHAR2	"	선택 사항	<p>다운로드되기 위해 파일 이름과 일치해야 하는 파일 이름 접두사입니다. 빈 접두사는 지정된 Amazon S3 버킷의 모든 파일을 다운로드합니다.</p> <p>이 프로시저는 접두사와 일치하는 첫 레벨 폴더에서만 Amazon S3 개 객체를 다운로드합니다. 지정된 접두사와 일치하는 중첩된 디렉터리 구조는 다운로드되지 않습니다. 예를 들어 Amazon S3 버킷에 <code>folder_1/folder_2/folder_3</code> 폴더 구조가 있다고 가정합니다. '<code>folder_1/folder_2</code>' 접두사도 지정한다고 가정하십시오. 이 경우 <code>folder_2</code>의 파일만 다</p>

파라미터 이름	데이터 형식	기본값	필수 여부	설명
				운로드되고 <code>folder_1</code> 또는 <code>folder_3</code> 의 파일은 다운로드되지 않습니다.

다음 예제에서는 `mys3bucket`이라는 이름의 Amazon S3 버킷의 모든 파일을 `DATA_PUMP_DIR` 디렉터리로 다운로드합니다.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
    p_bucket_name    => 'mys3bucket',
    p_directory_name => 'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

다음 예제에서는 `mys3bucket`이라는 Amazon S3 버킷에서 접두사가 `db`인 모든 파일을 `DATA_PUMP_DIR` 디렉터리로 다운로드합니다.

```
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;
```

각 예에서, SELECT 문은 VARCHAR2 데이터 형식으로 작업 ID를 반환합니다.

다음 예제에서는 `mys3bucket`이라는 Amazon S3 버킷의 `myfolder/` 폴더에 있는 모든 파일을 `DATA_PUMP_DIR` 디렉터리로 다운로드합니다. 접두사 파라미터 설정을 사용하여 Amazon S3 폴더를 지정 하십시오.

```
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;
```

각 예에서, SELECT 문은 VARCHAR2 데이터 형식으로 작업 ID를 반환합니다.

Note

`UTL_FILE.FREMOVE` Oracle 프로시저를 사용하여 디렉터리에서 파일을 제거할 수 있습니다. 자세한 내용은 Oracle 설명서의 [FREMOVE 프로시저](#)를 참조하십시오.

파일 전송 상태 모니터링

파일 전송 작업은 Amazon RDS 이벤트가 시작되고 완료될 때 이벤트를 게시합니다. 이벤트 보기에 대한 자세한 내용은 [Amazon RDS 이벤트 보기 \(p. 305\)](#) 단원을 참조하십시오.

진행 중인 작업의 상태를 `bdump` 파일에서 볼 수 있습니다. 이 `bdump` 파일은 `/rdsdbdata/log/trace` 디렉터리에 위치합니다. 각 `bdump` 파일 이름은 다음 형식으로 되어 있습니다.

```
dbtask-task-id.log
```

task-id를 모니터링하고자 하는 작업의 ID로 바꾸십시오.

rdsadmin.rds_file_util.read_text_file 저장 프로시저를 사용하여 bdump 파일의 내용을 볼 수 있습니다. 예를 들어, 다음 쿼리는 **dbtask-1546988886389-2444.log** bdump 파일의 내용을 반환합니다.

```
SELECT text FROM
  table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1546988886389-2444.log'));
```

Amazon S3 통합 옵션 제거

DB 인스턴스에서 Amazon S3 통합 옵션을 제거할 수 있습니다.

DB 인스턴스에서 Amazon S3 통합 옵션을 제거하려면 다음 중 하나를 수행합니다.

- 여러 DB 인스턴스에서 Amazon S3 통합 옵션을 제거하려면 DB 인스턴스가 속한 옵션 그룹에서 해당 S3_INTEGRATION 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- 단일 DB 인스턴스에서 Amazon S3 통합 옵션을 제거하려면 DB 인스턴스를 수정하고 S3_INTEGRATION 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

Oracle Application Express

Amazon RDS는 APEX 및 APEX-DEV 옵션 사용을 통해 Oracle Application Express(APEX)를 지원합니다. Oracle APEX는 웹 기반 애플리케이션을 위한 런타임 환경 또는 전체 개발 환경으로 배포할 수 있습니다. 개발자들은 Oracle APEX를 사용하여 웹 브라우저 내에서 전체 애플리케이션을 빌드할 수 있습니다. 자세한 내용은 Oracle 문서의 [Oracle Application Express](#)를 참조하십시오.

Oracle APEX는 다음과 같은 두 가지 주요 구성 요소로 구성됩니다.

- 리포지토리 - APEX 애플리케이션 및 구성 요소에 대한 메타데이터를 저장합니다. 리포지토리는 Amazon RDS DB 인스턴스에 설치되는 테이블, 인덱스 및 기타 객체로 구성됩니다.
- listener - Oracle APEX 클라이언트와의 HTTP 통신을 관리합니다. 리스너는 웹 브라우저에서 수신되는 유입 연결을 허용하고 처리를 위해 해당 연결을 Amazon RDS DB 인스턴스에 전달한 다음, 리포지토리의 결과를 브라우저로 다시 보냅니다. Oracle 12c에서 APEX Listener의 이름이 Oracle Rest Data Services(ORDS)로 변경되었습니다.

Amazon RDS APEX 옵션을 DB 인스턴스에 추가하면 Amazon RDS에서 Oracle APEX 리포지토리만 설치합니다. Amazon EC2 인스턴스, 회사의 온프레미스 서버 또는 사용자의 데스크톱 컴퓨터와 같은 별도의 호스트에 Oracle APEX Listener를 설치해야 합니다.

APEX 옵션은 DB 인스턴스에 대한 DB 인스턴스 클래스의 스토리지를 사용합니다.

다음은 Amazon RDS에서 Oracle 12c 버전 12.2용 Oracle APEX에 대한 지원 버전과 대략적인 스토리지 요구 사항입니다.

- Oracle APEX 버전 18.2.v1 – 146MiB
- Oracle APEX 버전 18.1.v1 – 145MiB

- Oracle APEX 버전 5.1.4.v1 – 220MiB

다음은 Amazon RDS에서 Oracle 12c 버전 12.1용 Oracle APEX에 대한 지원 버전과 대략적인 스토리지 요구 사항입니다.

- Oracle APEX 버전 18.2.v1 – 146MiB
- Oracle APEX 버전 18.1.v1 – 145MiB
- Oracle APEX 버전 5.1.4.v1 – 220MiB
- Oracle APEX 버전 5.1.2.v1 – 150MiB
- Oracle APEX 버전 5.0.4.v1 – 140MiB
- Oracle APEX 버전 4.2.6.v1 – 160MiB

다음은 Amazon RDS에서 Oracle 11g용 Oracle APEX에 대한 지원 버전과 대략적인 스토리지 요구 사항입니다.

- Oracle APEX 버전 18.2.v1 – 146MiB
- Oracle APEX 버전 18.1.v1 – 145MiB
- Oracle APEX 버전 5.1.4.v1 – 220MiB
- Oracle APEX 버전 5.1.2.v1 – 150MiB
- Oracle APEX 버전 5.0.4.v1 – 140MiB
- Oracle APEX 버전 4.2.6.v1 – 160MiB
- Oracle APEX 버전 4.1.1.v1 – 130MiB

Note

DB 인스턴스에 사용되는 DB 인스턴스 클래스에 하나의 vCPU만 있는 경우 Oracle 11g용 Oracle APEX 5가 지원되지 않습니다. DB 인스턴스 클래스에 대한 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#)을(를) 참조하십시오.

Oracle APEX 및 APEX Listener 사전 요구사항

다음은 Oracle APEX 및 APEX Listener 사용을 위한 사전 요구사항입니다.

- DB 인스턴스에서 관리 작업을 수행하려면 SQL*Plus가 필요합니다.
- 또한 Oracle APEX Listener 역할을 하는 호스트 컴퓨터에 다음과 같은 소프트웨어를 설치해야 합니다.
 - Java Runtime Environment(JRE).
 - Oracle Net Services - Amazon RDS 인스턴스에 연결할 수 있도록 Oracle APEX Listener를 활성화합니다.

Amazon RDS APEX 옵션 추가

Amazon RDS APEX 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 해당 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동 시킵니다.

Amazon RDS APEX 옵션을 추가하는 경우 DB 인스턴스를 자동으로 다시 시작하는 동안 잠시 작동이 중단됩니다.

DB 인스턴스에 APEX 옵션을 추가하려면

1. 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않으면 다음 설정을 사용하여 사용자 지정 DB 옵션을 생성합니다.
 - a. [Engine]에서 사용할 Oracle 버전을 선택합니다. 모든 버전에서 APEX 옵션이 지원됩니다.
 - b. 메이저 엔진 버전에서 DB 인스턴스의 버전을 선택합니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.
2. 해당 옵션을 옵션 그룹에 추가합니다. Oracle APEX 런타임 환경만 배포하려면 APEX 옵션만 추가합니다. 전체 개발 환경을 배포하려면 APEX 및 APEX-DEV 옵션을 모두 추가합니다.
 - Oracle 12c의 경우, APEX 및 APEX-DEV 옵션을 추가합니다.
 - Oracle 11g는 먼저 [XMLDB] 옵션을 사전 요구사항으로 추가한 다음 [APEX] 및 [APEX-DEV] 옵션을 추가합니다.

[Version]에서 사용하고자 하는 APEX 버전을 선택합니다. 버전을 선택하지 않으면 11g의 기본값은 버전 4.1.1.v1이고, 12c의 기본값은 버전 4.2.6.v1입니다.

Important

하나 이상의 DB 인스턴스에 이미 연결되어 있는 기존 옵션 그룹에 APEX 옵션을 추가하면 모든 DB 인스턴스가 자동으로 다시 시작되는 동안 인스턴스가 잠시 중단됩니다.

옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#)을(를) 참조하십시오.

3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:
 - 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
 - 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 기존 DB 인스턴스에 APEX 옵션을 추가하는 경우 DB 인스턴스를 자동으로 다시 시작하는 동안 인스턴스가 잠시 중단됩니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

퍼블릭 사용자 계정 잠금 해제

Amazon RDS APEX 옵션을 설치한 후 APEX 퍼블릭 사용자 계정 암호를 변경한 다음 계정을 잠금 해제해야 합니다. 이 작업은 Oracle SQL*Plus 명령줄 유ти리티를 사용하여 수행할 수 있습니다. DB 인스턴스에 마스터 사용자로 연결하고 다음 명령을 실행합니다. `new_password`를 원하는 암호로 바꿉니다.

```
alter user APEX_PUBLIC_USER identified by new_password;
alter user APEX_PUBLIC_USER account unlock;
```

Oracle APEX용 RESTful 서비스 구성

APEX에 RESTful 서비스를 구성하려면(APEX 4.1.1.V1에는 필요하지 않음) SQL*Plus를 사용하여 마스터 사용자로서 DB 인스턴스에 연결한 후 `rdsadmin.rdsadmin_run_apex_rest_config` 저장 프로시저를 실행하십시오. 저장 프로시저를 실행할 때 다음 사용자를 위한 암호를 제공합니다.

- APEX_LISTENER
- APEX_REST_PUBLIC_USER

저장 프로시저는 이들 사용자를 위해 새 데이터베이스 계정을 만드는 `apex_rest_config.sql` 스크립트를 실행합니다.

Note

Oracle APEX 버전 4.1.1.v1을 위한 구성은 필요하지 않습니다. 이 Oracle APEX 버전에 한해 저장 프로시저를 실행할 필요가 없습니다.

다음 명령으로 저장 프로시저를 실행합니다.

```
exec rdsadmin.rdsadmin_run_apex_rest_config('apex_listener_password',
'apex_rest_public_user_password');
```

APEX Listener 설치 및 구성

이제 Oracle APEX에서 사용하기 위해 Listener를 설치하고 구성할 수 있습니다. 이를 위해 다음 제품 중 하나를 사용할 수 있습니다.

- APEX 버전 5.0 이상은 Oracle Rest Data Services(ORDS) 사용
- APEX 버전 4.1.1은 Oracle APEX Listener 버전 1.1.4 사용
- Oracle HTTP 서버 및 `mod_plsql`

Note

Amazon RDS는 PL/SQL 게이트웨이가 포함된 Oracle XML DB HTTP 서버를 지원하지 않으므로, 이 게이트웨이를 APEX Listener로 사용할 수 없습니다. 일반적으로 Oracle은 인터넷 기반 애플리케이션으로 포함된 PL/SQL 게이트웨이 사용을 권장합니다.

Amazon EC2 인스턴스, 회사의 온프레미스 서버 또는 사용자의 데스크톱 컴퓨터와 같은 별도의 호스트에 APEX Listener를 설치해야 합니다.

다음 절차에서는 APEX Listener를 설치하고 구성하는 방법을 보여 줍니다. 여기에서는 호스트의 이름이 `myapexhost.example.com`이고 이 호스트에서 Linux를 실행 중이라고 가정합니다.

APEX Listener를 설치하고 구성하려면

1. `myapexhost.example.com`에 root로 로그인합니다.
2. APEX Listener 설치를 소유할 권한 없는 OS 사용자를 생성합니다. 다음 명령은 `apexuser`라는 새 사용자를 생성합니다.

```
useradd -d /home/apexuser apexuser
```

다음 명령은 새로운 사용자에게 암호를 할당합니다.

```
passwd apexuser;
```

3. `myapexhost.example.com`에 `apexuser`로 로그인하고, Oracle의 APEX 설치 파일을 `/home/apexuser` 디렉터리에 다운로드합니다.
 - <http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html>
 - [Oracle Application Express Prior Release Archives](#)
4. `/home/apexuser` 디렉터리에서 파일의 압축을 풉니다.

```
unzip apex_<version>.zip
```

파일의 압축을 풀면 /home/apexuser 디렉터리에 apex 디렉터리가 있습니다.

5. myapexhost.example.com에 apexuser로 여전히 로그인되어 있는 경우, Oracle의 APEX 리스터 파일을 /home/apexuser 디렉터리에 다운로드합니다.
 - <http://www.oracle.com/technetwork/developer-tools/apex-listener/downloads/index.html>
6. 새 디렉터리를 생성하고 APEX Listener 파일을 업니다.

Listener 유형	지침
ORDS	<p>다음 코드를 실행합니다.</p> <pre>mkdir /home/apexuser/ORDS cd /home/apexuser/ORDS unzip/ords.<version>.zip</pre>
APEX Listener	<p>다음 코드를 실행합니다.</p> <pre>mkdir /home/apexuser/apexlistener cd /home/apexuser/apexlistener unzip/apex_listener.<version>.zip</pre>

7. 이전 단계의 디렉터리에서 Listener 프로그램을 실행합니다.

Listener 유형	지침
ORDS	<p>다음 코드를 실행합니다.</p> <pre>java -jar ords.war setup</pre> <p>프로그램에서 다음 정보를 묻는 메시지를 표시합니다. 기본값은 팔호 안에 표시되어 있습니다.</p> <ul style="list-style-type: none"> • 구성 데이터를 저장할 위치 /home/apexuser/ORDS를 입력합니다. • 데이터베이스 서버의 이름[localhost] 기본값을 선택하거나 알맞은 값을 입력합니다. • 데이터베이스 수신 대기 포트[1521] 기본값을 선택하거나 알맞은 값을 입력합니다. • 데이터베이스 서비스 이름 또는 데이터베이스 SID[1] 데이터베이스 서비스 이름을 지정하려면 1을, 데이터베이스 SID를 지정하려면 2를 선택합니다. • 데이터베이스 SID[xe] 기본값을 선택하거나 알맞은 값을 입력합니다. • Oracle REST Data Services 스키마를 확인/설치 또는 이 단계 건너뛰기 [1] 2을 선택합니다. • PL/SQL 게이트웨이 사용 또는 이 단계 건너뛰기 [1]

Listener 유형	지침
	<p>기본값을 선택합니다.</p> <ul style="list-style-type: none">• PL/SQL 게이트웨이 데이터베이스 사용자 이름 [APEX_PUBLIC_USER] <p>기본값을 선택합니다.</p> <ul style="list-style-type: none">• APEX_PUBLIC_USER의 데이터베이스 암호 <p>암호를 입력합니다.</p> <ul style="list-style-type: none">• Application Express RESTful Services 데이터베이스 사용자 (APEX_LISTENER, APEX_REST_PUBLIC_USER)를 위한 암호 지정 또는 이 단계 건너뛰기 [1] <p>APEX 4.1.1.V1의 경우 2를 선택하고 다른 모든 APEX 버전의 경우 1을 선택합니다.</p> <ul style="list-style-type: none">• [APEX 4.1.1.v1에는 필요하지 않음] APEX_LISTENER의 데이터베이스 암호 (필요한 경우) 암호를 입력합니다.• [APEX 4.1.1.v1에는 필요하지 않음] APEX_REST_PUBLIC_USER의 데이터베이스 암호 (필요한 경우) 암호를 입력합니다.

Listener 유형	지침
APEX Listener	<p>다음 코드를 실행합니다.</p> <pre>java -Dapex.home=./apex -Dapex.images=/home/apexuser/apex/images -Dapex.erase -jar ./apex.war</pre> <p>프로그램에서 다음 정보를 묻는 메시지를 표시합니다.</p> <ul style="list-style-type: none"> • APEX Listener 관리자 이름. 기본값은 adminlistener입니다. • APEX Listener 관리자의 암호 • APEX Listener Manger의 사용자 이름입니다. 기본값은 managerlistener입니다. • APEX Listener 관리자의 암호 <p>다음과 같이 구성을 완료하는 데 필요한 URL이 인쇄됩니다:</p> <pre>INFO: Please complete configuration at: http://localhost:8080/apex/listenerConfigure Database is not yet configured</pre> <p>APEX Listener를 실행 중인 상태로 들플니다. Oracle Application Express를 사용하면 Listener가 계속 실행되고 있어야 합니다. 이 구성 절차를 마치면 Listener를 백그라운드에서 실행할 수 있습니다.</p> <p>웹 브라우저에서 APEX Listener 프로그램에 제공된 URL로 이동합니다. Oracle Application Express Listener 관리 창이 나타납니다. 다음 정보를 입력합니다.</p> <ul style="list-style-type: none"> • Username(사용자 이름) – APEX_PUBLIC_USER • Password(암호) – APEX_PUBLIC_USER에 대한 암호입니다. 앞에서 APEX 리포지토리를 구성할 때 지정한 암호입니다. 자세한 내용은 퍼블릭 사용자 계정 잠금 해제 (p. 785) 단원을 참조하십시오. • Connection Type(연결 유형) – 기본 • Hostname(호스트 이름) – Amazon RDS DB 인스턴스의 엔드포인트입니다(예: mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com). • Port(포트) – 1521 • SID – Amazon RDS DB 인스턴스에 있는 데이터베이스의 이름입니다(예: mydb). <p>[Apply]를 선택합니다. APEX 관리 창이 나타납니다.</p>

8. APEX admin 사용자용 암호를 설정해야 합니다. 이를 위해 SQL*Plus를 사용하여 DB 인스턴스에 마스터 사용자로 연결하고 다음 명령을 실행합니다:

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
grant APEX_ADMINISTRATOR_ROLE to master;
@/home/apexuser/apex/apxchpwd.sql
```

master를 마스터 사용자 이름으로 바꿉니다. apxchpwd.sql 스크립트에서 메시지가 표시되면 새 admin 암호를 입력합니다.

9. ORDS에서 APEX Listener를 시작합니다. 다음 코드를 실행합니다.

```
java -jar ords.war
```

APEX Listener를 처음으로 시작할 때 APEX 고정 리소스의 위치를 제공하라는 메시지가 나타납니다. 이 이미지 폴더는 APEX 설치 디렉터리의 /apex/images 디렉터리에 위치합니다.

10. 브라우저에서 APEX 관리 창으로 돌아가서 [Administration]을 선택합니다. 그런 다음 [Application Express Internal Administration]을 선택합니다. AWS 자격 증명을 요구하는 메시지가 표시되면 다음 정보를 입력합니다.
 - User name(사용자 이름) – admin
 - Password(암호) – apxchpwd.sql 스크립트를 사용하여 설정한 암호

[Login]을 선택한 다음 admin 사용자용 새 암호를 설정합니다.

이제 APEX Listener를 사용할 준비가 끝났습니다.

APEX 버전 업그레이드

Important

APEX를 업그레이드하기 전에 DB 인스턴스를 백업하십시오. 자세한 내용은 [DB 스냅샷 생성 \(p. 203\)](#) 및 [업그레이드 테스트 \(p. 755\)](#) 단원을 참조하십시오.

DB 인스턴스와 함께 APEX를 업그레이드하려면 다음을 실행합니다.

- 업그레이드된 DB 인스턴스 버전의 새 옵션 그룹을 생성합니다.
- 업그레이드된 APEX 및 APEX-DEV 버전을 새 옵션 그룹에 추가합니다. DB 인스턴스가 사용하는 다른 옵션도 포함시켜야 합니다. 자세한 내용은 [옵션 그룹 고려 사항 \(p. 754\)](#) 단원을 참조하십시오.
- DB 인스턴스를 업그레이드할 때 업그레이드된 DB 인스턴스의 새 옵션 그룹을 지정합니다.

APEX 버전을 업그레이드한 후에도 이전 버전의 APEX 스키마가 데이터베이스에 남아 있을 수 있습니다. 더 이상 필요 없는 경우에는 업그레이드 후에 데이터베이스에서 기존 APEX 스키마를 제거할 수 있습니다.

APEX 버전을 업그레이드하고 이전 APEX 버전에서 RESTful 서비스가 구성되지 않은 경우 RESTful 서비스를 구성하는 것이 좋습니다. 자세한 내용은 [Oracle APEX용 RESTful 서비스 구성 \(p. 785\)](#) 단원을 참조하십시오.

DB 인스턴스의 주요 버전 업그레이드를 실행하고 대상 데이터베이스 버전과 호환되지 않는 APEX 버전을 사용할 경우 DB 인스턴스를 업그레이드하기 전에 APEX 버전을 업그레이드할 수 있습니다. APEX를 먼저 업그레이드하면 DB 인스턴스를 업그레이드하는 시간을 줄일 수 있습니다.

Note

APEX 업그레이드 이후 업그레이드된 버전과 사용할 리스너를 설치 및 구성합니다. 자침은 [APEX Listener 설치 및 구성 \(p. 786\)](#)을 참조하십시오.

APEX 옵션 제거

DB 인스턴스에서 Amazon RDS APEX 옵션을 제거할 수 있습니다. DB 인스턴스에서 APEX 옵션을 제거하려면 다음 중 하나를 수행합니다.

- 여러 DB 인스턴스에서 APEX 옵션을 제거하려면 DB 인스턴스가 속한 옵션 그룹에서 해당 APEX 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 여러 DB 인스턴스에 연결된 옵션 그룹에서 APEX 옵션을 제거하면 모든 DB 인스턴스가 다시 시작되는 동안 인스턴스가 잠시 중단됩니다.

자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.

- 단일 DB 인스턴스에서 APEX 옵션을 제거하려면 DB 인스턴스를 수정하고 APEX 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. APEX 옵션을 제거하는 경우 DB 인스턴스를 자동으로 다시 시작하는 동안 인스턴스가 잠시 중단됩니다.

자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

DB 인스턴스에서 APEX 옵션을 제거하면 데이터베이스에서 APEX 스키마가 제거됩니다.

Oracle Enterprise Manager

Amazon RDS는 Oracle Enterprise Manager(OEM)를 지원합니다. OEM은 엔터프라이즈 정보 기술의 통합 관리를 위한 Oracle 제품 라인입니다.

Amazon RDS는 다음 옵션을 통해 OEM을 지원합니다.

옵션	옵션 ID	지원
OEM 데이터베이스 (p. 791)	OEM	OEM Database Express 12c OEM 11g Database Control
OEM Management Agent (p. 793)	OEM_AGENT	OEM Cloud Control for 13c OEM Cloud Control for 12c

Note

OEM 데이터베이스 또는 OEM Management Agent를 사용할 수 있지만 둘 다 사용할 수는 없습니다.

Oracle Enterprise Manager Database Express

Amazon RDS는 OEM 옵션 사용을 통해 Oracle Enterprise Manager(OEM) Database Express를 지원합니다. Amazon RDS는 OEM 데이터베이스의 다음 버전을 지원합니다.

- Oracle Enterprise Manager Database Express 12c
- Oracle Enterprise Manager 11g Database Control

OEM Database Express 및 Database Control은 Oracle 데이터베이스 관리용 웹 기반 인터페이스가 있는 서로 비슷한 도구입니다. 이러한 도구에 대한 자세한 내용은 Oracle 설명서의 [Accessing Enterprise Manager Database Express 12c](#) 및 [Accessing Enterprise Manager 11g Database Control](#)을 참조하십시오.

다음은 OEM Database를 사용할 때 적용되는 몇 가지 제한 사항입니다.

- 2., db.t2.small, db.t.micro, db.m1.small DB 인스턴스 클래스에 대해서는 OEM Database가 지원되지 않습니다.

DB 인스턴스 클래스에 대한 자세한 내용은 [Oracle을 위한 DB 인스턴스 클래스 지원 \(p. 705\)](#) 섹션을 참조하십시오.

- OEM 11g Database Control은 아메리카/아르헨티나/부에노스아이레스, 아메리카/마타모로스, 아메리카/몬테레이, 아메리카/토론토, 아시아/아수기바트, 아시아/다카, 아시아/카트만두, 아시아/콜카타, 아시아/ول란바토르, 대서양/케이프베르데, 오스트레일리아/유클라, 태평양/키리티마티 등의 시간대와 호환되지 않습니다.

시간대에 대한 자세한 내용은 [Oracle 시간대 \(p. 824\)](#) 단원을 참조하십시오.

OEM Database 옵션 설정

Amazon RDS는 OEM 옵션에 대해 다음 설정을 지원합니다.

옵션 설정	유효한 값	설명
포트	정수 값	OEM 데이터베이스에 대해 수신 대기하는 DB 인스턴스의 포트입니다. OEM 데이터베이스 Express 12c의 기본값은 5500입니다. OEM 11g Database Control의 기본값은 1158입니다.
보안 그룹	—	Port(포트)에 액세스할 수 있는 보안 그룹입니다.

OEM 데이터베이스 옵션 추가

OEM 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동시킵니다.

OEM 옵션을 추가한 후 DB 인스턴스를 재시작할 필요가 없습니다. 옵션 그룹이 활성화되자마자 OEM 데이터베이스가 활성화됩니다.

DB 인스턴스에 OEM 옵션을 추가하려면

1. 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않으면 다음 설정을 사용하여 사용자 지정 DB 옵션을 생성합니다.
 - a. [Engine]에서 DB 인스턴스의 Oracle 에디션을 선택합니다.
 - b. 메이저 엔진 버전에서 DB 인스턴스의 버전을 선택합니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.

2. [OEM] 옵션을 옵션 그룹에 추가하고 옵션 설정을 구성합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#)을(를) 참조하십시오. 각 설정에 대한 자세한 내용은 [OEM Database 옵션 설정 \(p. 792\)](#)을(를) 참조하십시오.
3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:
 - 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
 - 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

OEM 데이터베이스 사용

OEM 옵션을 활성화한 다음, 웹 브라우저에서 OEM 데이터베이스 도구 사용을 시작하면 됩니다.

웹 브라우저에서 OEM Database Control 또는 OEM Database Express에 액세스할 수 있습니다. 예를 들어 Amazon RDS DB 인스턴스에 대한 엔드포인트가 `mydb.f9rbfa893tft.us-`

east-1.rds.amazonaws.com이고 OEM 포트가 1158인 경우 OEM Database Control에 액세스하기 위한 URL은 다음과 같습니다.

`https://mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com:1158/em`

웹 브라우저에서 도구에 액세스하면 사용자 이름과 암호를 묻는 로그인 창이 나타납니다. DB 인스턴스에 대한 마스터 사용자 이름과 마스터 암호를 입력합니다. Oracle 데이터베이스를 관리할 준비가 완료되었습니다.

OEM Database 설정 수정

OEM Database를 활성화한 후 옵션의 보안 그룹 설정을 수정할 수 있습니다.

옵션 그룹을 DB 인스턴스와 연동한 이후에는 OEM 포트 번호를 수정할 수 없습니다. DB 인스턴스의 OEM 포트 번호를 변경하려면 다음과 같이 합니다.

1. 새 옵션 그룹을 생성합니다.
2. 새 포트 번호가 포함된 OEM 옵션에 새 옵션 그룹을 추가합니다.
3. DB 인스턴스에서 기존 옵션 그룹을 제거합니다.
4. 새 옵션 그룹을 DB 인스턴스에 추가합니다.

옵션 설정을 변경하는 방법에 대한 자세한 내용은 [옵션 설정 변경 \(p. 156\)](#)을(를) 참조하십시오. 각 설정에 대한 자세한 내용은 [OEM Database 옵션 설정 \(p. 792\)](#) 단원을 참조하십시오.

OEM 데이터베이스 옵션 제거

DB 인스턴스에서 OEM 옵션을 제거할 수 있습니다. OEM 옵션을 제거한 후 DB 인스턴스를 재시작할 필요가 없습니다.

DB 인스턴스에서 OEM 옵션을 제거하려면 다음 중 하나를 수행합니다.

- OEM Agent가 속한 옵션 그룹에서 OEM 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- DB 인스턴스를 수정하고, OEM 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 이 변경은 단일 DB 인스턴스에 영향을 미칩니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

관련 주제

- [옵션 그룹 작업 \(p. 148\)](#)
- [Oracle DB 인스턴스 옵션 \(p. 772\)](#)

Oracle Management Agent for Enterprise Manager Cloud Control

Amazon RDS는 OEM_AGENT 옵션 사용을 통해 Oracle Enterprise Manager(OEM) Management Agent를 지원합니다. Amazon RDS는 OEM의 다음 버전에 대해 Management Agent를 지원합니다.

- Oracle Enterprise Manager Cloud Control for 13c
- Oracle Enterprise Manager Cloud Control for 12c

Management Agent는 호스트에서 실행되는 대상을 모니터링하고 해당 정보를 미들티어 Oracle Management Service(OMS)에 전달하는 소프트웨어 구성 요소입니다. 자세한 내용은 Oracle 설명서에서

[Overview of Oracle Enterprise Manager Cloud Control 12c](#) 및 [Overview of Oracle Enterprise Manager Cloud Control 13c](#) 단원을 참조하십시오.

다음은 Management Agent를 사용할 때 적용되는 몇 가지 제한 사항입니다.

- 작업 실행 및 데이터베이스 패치 적용과 같이 호스트 자격 증명이 필요한 관리 작업은 지원되지 않습니다.
- 호스트 측정치 및 프로세스 목록에는 실제 시스템 상태가 반영되지 않을 수 있습니다.
- 자동 검색은 지원되지 않습니다. 데이터베이스 대상을 수동으로 추가해야 합니다.
- OMS 모듈 사용성은 데이터베이스 에디션에 따라 다릅니다. 예를 들어, 데이터베이스 성능 진단 및 튜닝 모듈은 Oracle Database Enterprise Edition에만 사용할 수 있습니다.
- Management Agent는 추가 메모리 및 컴퓨팅 리소스를 사용합니다. OEM_AGENT 옵션을 활성화한 후 성능 문제가 발생할 경우 더 큰 DB 인스턴스로 조정하는 것이 좋습니다. 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 및 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.
- 경고 로그의 운영 체제 액세스는 Amazon RDS 호스트에서 OEM_AGENT를 실행하는 사용자에게 부여되지 않으므로 OEM에서 DB Alert Log 및 DB Alert Log Error Status에 대한 지표를 수집할 수 없습니다.

Management Agent의 사전 요구 사항

다음은 Management Agent 사용을 위한 필수 선행 조건입니다.

- Oracle 버전 12.2.0.1, 12.1.0.2 또는 11.2.0.4를 실행하는 Amazon RDS DB 인스턴스.
- OEM 13c2일 때 최소 3.3GiB의 스토리지 공간.
- OEM 13c1일 때 최소 3GiB의 스토리지 공간.
- OEM 12c일 때 최소 2GiB의 스토리지 공간.
- Amazon RDS DB 인스턴스에 연결하도록 구성된 Oracle Management Service(OMS).
- Oracle 패치 25163555가 적용된 OMS 13c2일 때는 OEM Agent 13.2.0.0.v2 이상을 사용합니다.

OMSPatcher를 사용하여 패치를 적용합니다.

- 패치가 적용되지 않은 OMS 13c2일 때는 OEM Agent 13.2.0.0.v1을 사용합니다.
- 대부분의 경우 OMS에서 DB 인스턴스로 연결을 허용하도록 VPC를 구성해야 합니다. Amazon Virtual Private Cloud(Amazon VPC)에 친숙하지 않은 경우 계속하기 전에 [자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 \(p. 421\)](#)의 단계를 완료하는 것이 좋습니다.

OMS 호스트와 Amazon RDS DB 인스턴스가 통신하려면 추가 구성이 필요합니다. 다음 작업도 수행해야 합니다.

- OMS가 방화벽 뒤에 있는 경우 Management Agent에서 OMS로 연결하려면 DB 인스턴스의 IP 주소를 OMS에 추가해야 합니다.
- OMS에 공개적으로 확인할 수 있는 호스트 이름이 있는 경우 OMS에서 Management Agent로 연결하려면 OMS 주소를 보안 그룹에 추가해야 합니다. 보안 그룹에는 DB 인스턴스 포트 및 Management Agent 포트에 대한 액세스를 허용하는 인바운드 규칙이 있어야 합니다. 보안을 생성하고 인바운드 규칙을 추가하는 방법의 예는 [자습서: Amazon RDS DB 인스턴스에 사용할 Amazon VPC 생성 \(p. 421\)](#) 단원을 참조하십시오.
- OMS에 공개적으로 확인할 수 있는 호스트 이름이 없는 경우 OMS에서 Management Agent로 연결하려면 다음 중 하나를 사용합니다.
 - OMS가 프라이빗 VPC의 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 호스팅된 경우 VPC 피어링을 설정하여 OMS에서 Management Agent로 연결할 수 있습니다. 자세한 내용은 [VPC에 있는 DB 인스턴스에 다른 VPC에 있는 EC2 인스턴스가 액세스 \(p. 410\)](#) 단원을 참조하십시오.
 - OMS가 온프레미스에 호스팅된 경우 VPN 연결을 설정하여 OMS에서 Management Agent로 액세스를 허용할 수 있습니다. 자세한 내용은 [클라이언트 애플리케이션이 인터넷을 통해 VPC에 있는 DB 인스턴스에 액세스 \(p. 411\)](#) 또는 [VPN 연결을 참조하십시오.](#)

Management Agent 옵션 설정

Amazon RDS는 Management Agent 옵션에 대해 다음 설정을 지원합니다. OEM_AGENT 옵션을 추가할 때 모든 설정은 필수입니다.

Note

모든 설정은 필수입니다.

옵션 설정	유효한 값	설명
버전 (AGENT_VERSION)	13.2.0.0.v2 13.2.0.0.v1 13.1.0.0.v1 12.1.0.5.v1 12.1.0.4.v1	Management Agent 소프트웨어의 버전입니다. AWS CLI 옵션 이름은 OptionVersion입니다. Note AWS GovCloud (US-West) 리전에서는 버전 13.2.0.0.v1 및 13.2.0.0.v2만 사용할 수 있습니다.
포트 (AGENT_PORT)	정수 값	OMS 호스트에 대해 수신 대기하는 DB 인스턴스의 포트입니다. 기본값은 3872입니다. OMS 호스트는 이 포트에 액세스할 수 있는 보안 그룹에 속해야 합니다. AWS CLI 옵션 이름은 Port입니다.
보안 그룹	기존 보안 그룹	Port(포트)에 액세스할 수 있는 보안 그룹입니다. OMS 호스트는 이 보안 그룹에 속해야 합니다. AWS CLI 옵션 이름은 VpcSecurityGroupMemberships 또는 DBSecurityGroupMemberships입니다.
OMS_HOST	문자열 값, 예: <i>my.example.oms</i>	OMS의 공개적으로 액세스할 수 있는 호스트 이름 또는 IP 주소입니다. AWS CLI 옵션 이름은 OEM_HOST입니다.
OMS_PORT	정수 값	Management Agent에 대해 수신 대기하는 OMS 호스트의 HTTPS 업로드 포트입니다. HTTPS 업로드 포트를 결정하려면 OMS 호스트를 연결하고 다음 명령을 실행합니다(SYSMAN 암호 필요).

옵션 설정	유효한 값	설명
		<pre>emctl status oms -details</pre> <p>AWS CLI 옵션 이름은 OMS_PORT입니다.</p>
AGENT_REGISTRATION_PASSWORD	GRD 파일 값	<p>Management Agent가 OMS에 자신을 인증하기 위해 사용하는 암호입니다. OEM_AGENT 옵션을 활성화하기 전에 OMS에서 지속적인 암호를 생성하는 것이 좋습니다. 지속적인 암호가 있으면 단일 Management Agent 옵션 그룹을 여러 Amazon RDS 데이터베이스 간에 공유할 수 있습니다.</p> <p>AWS CLI 옵션 이름은 AGENT_REGISTRATION_PASSWORD입니다.</p>

Management Agent 옵션 추가

Management Agent 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동시킵니다.

Management Agent 옵션을 추가한 후 DB 인스턴스를 재시작할 필요가 없습니다. 옵션 그룹이 활성화되자마자 OEM Agent가 활성화됩니다.

AWS Management 콘솔

Management Agent 옵션을 DB 인스턴스에 추가하려면

1. 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않으면 다음 설정을 사용하여 사용자 지정 DB 옵션을 생성합니다.
 - a. [Engine]에서 DB 인스턴스의 Oracle 에디션을 선택합니다.
 - b. 메이저 엔진 버전에서 DB 인스턴스의 버전을 선택합니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.

2. [OEM_AGENT] 옵션을 옵션 그룹에 추가하고 옵션 설정을 구성합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#)을(를) 참조하십시오. 각 설정에 대한 자세한 내용은 [Management Agent 옵션 설정 \(p. 795\)](#)을(를) 참조하십시오.
3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:
 - 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
 - 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

AWS CLI

다음 예에서는 AWS CLI `add-option-to-option-group` 명령을 사용하여 OEM_AGENT 옵션을 myoptiongroup이라는 옵션 그룹에 추가합니다.

Linux, OS X, Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

Management Agent 옵션 사용

Management Agent 옵션을 활성화한 후에는 다음 절차를 사용하여 사용을 시작합니다.

Management Agent 옵션을 사용하려면

- DB 인스턴스의 대상 데이터베이스에서 다음 코드를 실행하고 마스터 사용자 계정을 사용하여 DBSNMP 계정 자격 증명을 잠금 해제하고 재설정합니다.

```
ALTER USER dbsnmp IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

- 대상을 OMS 콘솔에 수동으로 추가합니다.
 - OMS 콘솔에서 [Setup], [Add Target], [Add Targets Manually]를 선택합니다.
 - [Add Targets Declaratively by Specifying Target Monitoring Properties]를 선택합니다.
 - [Target Type]에서 [Database Instance]를 선택합니다.
 - Monitoring Agent(모니터링 에이전트)에서 Amazon RDS DB 인스턴스 식별자와 동일한 식별자가 있는 에이전트를 선택합니다.
 - [Add Manually]를 선택합니다.
 - Amazon RDS DB 인스턴스의 엔드포인트를 입력하거나 호스트 이름 목록에서 이를 선택합니다. 지정된 호스트 이름이 Amazon RDS DB 인스턴스의 엔드포인트와 일치하는지 확인합니다.

사용자의 Amazon RDS DB 인스턴스에 대한 엔드포인트를 찾는 방법은 [DB 인스턴스의 엔드포인트 찾기 \(p. 736\)](#) 단원을 참조하십시오.

- 다음 데이터베이스 속성을 지정합니다.
 - [Target name]에 이름을 입력합니다.
 - [Database system name]에 이름을 입력합니다.
 - [Monitor username]에 dbsnmp를 입력합니다.
 - [Monitor password]에 1단계의 암호를 입력합니다.

- [Role]에 [normal]을 입력합니다.
 - [Oracle home path]에 /oracle을 입력합니다.
 - [Listener Machine name]에는 에이전트 식별자가 이미 나타나 있습니다.
 - [Port]에 데이터베이스 포트를 입력합니다. RDS 기본 포트는 1521입니다.
 - [Database name]에 데이터베이스 이름을 입력합니다.
- h. [Test Connection]을 선택합니다.
- i. [Next]를 선택합니다. 모니터링되는 리소스 목록에 대상 데이터베이스가 나타납니다.

Management Agent 옵션 설정 수정

Management Agent를 활성화한 후 옵션 설정을 수정할 수 있습니다. 옵션 설정을 변경하는 방법에 대한 자세한 내용은 [옵션 설정 변경 \(p. 156\)](#) 단원을 참조하십시오. 각 설정에 대한 자세한 내용은 [Management Agent 옵션 설정 \(p. 795\)](#) 단원을 참조하십시오.

Management Agent 옵션 제거

DB 인스턴스에서 OEM Agent를 제거할 수 있습니다. OEM Agent를 제거한 후 DB 인스턴스를 재시작할 필요가 없습니다.

DB 인스턴스에서 OEM Agent를 제거하려면 다음 중 하나를 수행합니다.

- OEM Agent가 속한 옵션 그룹에서 OEM Agent 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- DB 인스턴스를 수정하고, OEM Agent 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 이 변경은 단일 DB 인스턴스에 영향을 미칩니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정 할 수 있습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

관련 주제

- [옵션 그룹 작업 \(p. 148\)](#)
- [Oracle DB 인스턴스 옵션 \(p. 772\)](#)

Oracle Java 가상 머신

Amazon RDS는 JVM 옵션을 사용함으로써 Oracle Java 가상 머신(JVM)을 지원합니다. Oracle Java는 Oracle 데이터베이스에서 Oracle Java 기능을 지원하는 SQL 스키마 및 함수를 제공합니다. 자세한 내용은 Oracle 설명서의 [Oracle 데이터베이스에 Java 도입](#) 단원을 참조하십시오.

Oracle JVM을 다음의 Oracle 데이터베이스 버전에 사용할 수 있습니다.

- Oracle 12c, 12.2.0.1, 모든 버전
- Oracle 12c, 12.1.0.2.v13 또는 이후 버전
- Oracle 11g, 11.2.0.4.v17 또는 이후 버전

Amazon RDS의 Java 구현에는 권한 집합이 제한되어 있습니다. 마스터 사용자에게는 RDS_JAVA_ADMIN 역할이 부여되며, JAVA_ADMIN 역할에 의해 부여된 권한의 하위 집합이 부여됩니다. RDS_JAVA_ADMIN 역할에 부여된 권한을 나열하려면 DB 인스턴스에서 다음 쿼리를 실행하십시오.

```
SELECT * FROM dba_java_policy
  WHERE grantee IN ('RDS_JAVA_ADMIN', 'PUBLIC')
    AND enabled = 'ENABLED'
  ORDER BY type_name, name, grantee;
```

Oracle JVM 사전 조건

다음은 Oracle Java 사용 시 사전 조건입니다.

- DB 인스턴스는 VPC(가상 사설 클라우드) 내에 있어야 합니다. 자세한 내용은 [EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 \(p. 407\)](#)을(를) 참조하십시오.
- DB 인스턴스 클래스는 충분히 커야 합니다. Oracle Java는 db.m1.small, db.t2.micro 또는 db.t2.small DB 인스턴스 클래스에 대해 지원되지 않습니다. 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#)을(를) 참조하십시오.
- DB 인스턴스는 마이너 버전 자동 업그레이드 옵션이 활성화되어야 합니다. 이 옵션은 DB 인스턴스를 활성화하여 사용 가능할 때 마이너 DB 엔진 버전을 자동으로 업그레이드 받을 수 있습니다. Amazon RDS는 이 옵션을 사용하여 DB 인스턴스를 최신 Oracle Patch Set Update(PSU)로 업데이트합니다. 특히, CVSS(공통 취약성 평가 시스템) 점수가 9.0 이상인 보안 취약성 또는 발표된 기타 보안 취약성이 있을 경우 그렇게 합니다. 자세한 내용은 [Oracle DB 인스턴스 설정 \(p. 744\)](#)를 참조하십시오.
- DB 인스턴스가 메이저 버전 11.2에서 실행되는 경우 XMLDB 옵션을 설치해야 합니다. 자세한 내용은 [Oracle XML DB \(p. 831\)](#) 단원을 참조하십시오.

Oracle JVM의 모범 사례

다음은 Oracle Java 사용에 관한 모범 사례입니다.

- 보안을 극대화하기 위해 Secure Sockets Layer(SSL)와 함께 JVM 옵션을 사용합니다. 자세한 내용은 [Oracle 보안 소켓 Layer \(p. 810\)](#) 단원을 참조하십시오.
- DB 인스턴스를 구성하여 네트워크 액세스를 제한하십시오. 자세한 내용은 [VPC에서 DB 인스턴스에 액세스하는 시나리오 \(p. 408\)](#) 및 [VPC에서 Amazon RDS DB 인스턴스를 사용한 작업 \(p. 415\)](#) 단원을 참조하십시오.

Oracle JVM 옵션 추가

JVM 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동시킵니다.

JVM 옵션을 추가하는 동안 잠시 작동이 중단됩니다. 옵션을 추가한 후 DB 인스턴스를 재시작할 필요가 없습니다. 옵션 그룹이 활성화되는 즉시 Oracle java를 사용 가능합니다.

DB 인스턴스에 JVM 옵션을 추가하려면

1. 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않으면 다음 설정을 사용하여 사용자 지정 DB 옵션을 생성합니다.
 - 엔진에는 DB 인스턴스에서 사용하는 DB 엔진을 선택하십시오(oracle-ee, oracle-se, oracle-se1 또는 oracle-se2).

- 메이저 엔진 버전에서 DB 인스턴스의 버전을 선택합니다.
- 자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.
2. [JVM] 옵션을 옵션 그룹에 추가합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#)을(를) 참조하십시오.
 3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:
 - 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용하십시오. 자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#)을(를) 참조하십시오.
 - 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용하십시오. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#)을(를) 참조하십시오.
 4. 필수 권한을 사용자에게 부여하십시오.

Amazon RDS 마스터 사용자는 기본값으로 JVM 옵션을 사용하는 권한을 가져야 합니다. 다른 사용자가 이러한 사용 권한을 필요로 하는 경우 SQL 클라이언트에서 DB 인스턴스에 마스터 사용자로 연결하고 사용자에게 사용 권한을 부여하십시오.

다음 예제에서는 test_proc 사용자에게 JVM 옵션 사용 권한을 부여합니다.

```
create user test_proc identified by password;
CALL dbms_java.grant_permission('TEST_PROC',
  'oracle.aurora.security.JServerPermission', 'LoadClassInPackage.*', ''');
```

사용자에게 권한이 부여되면 다음 쿼리는 출력을 반환해야 합니다.

```
select * from dba_java_policy where grantee='TEST_PROC';
```

Note

Oracle 사용자 이름은 대소문자를 구분하며 일반적으로 모두 대문자입니다.

Oracle JVM 옵션 제거

DB 인스턴스에서 JVM 옵션을 제거할 수 있습니다. 옵션을 제거하는 동안 잠시 작동이 중단됩니다. JVM 옵션을 제거한 후 DB 인스턴스를 재시작할 필요가 없습니다.

Warning

JVM 옵션을 삭제했을 때 DB 인스턴스가 옵션의 일부로 활성화되어 있는 데이터 형식을 사용하고 있다면 데이터가 손실될 수 있습니다. 따라서 처리 전에 데이터를 백업해야 합니다. 자세한 내용은 [Amazon RDS DB 인스턴스 백업 및 복원 \(p. 196\)](#) 단원을 참조하십시오.

DB 인스턴스에서 JVM 옵션을 제거하려면 다음 중 하나를 수행합니다.

- 소속 옵션 그룹에서 JVM 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- DB 인스턴스를 수정하고 JVM 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 이 변경은 단일 DB 인스턴스에 영향을 미칩니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

Oracle 레이블 보안

Amazon RDS에서는 OLS 옵션을 사용하여 Oracle Enterprise Edition, 버전 12c에 대한 Oracle 레이블 보안을 지원합니다.

대부분의 데이터베이스 보안에서는 객체 수준에서 액세스를 제어합니다. Oracle 레이블 보안에서는 개별 테이블 행에 대한 액세스를 세부적으로 제어합니다. 예를 들어 레이블 보안을 사용하면 정책 기반 관리 모델을 통해 규제 준수를 이행할 수 있습니다. 레이블 보안 정책을 사용하여 중요 데이터에 대한 액세스를 제어하고 적절한 권한을 가진 사용자로 액세스를 제한할 수 있습니다. 자세한 내용은 Oracle 설명서의 [Introduction to Oracle Label Security](#)를 참조하십시오.

Important

Amazon RDS의 Oracle 12c 버전 12.2의 경우 Oracle 레이블 보안이 영구적이고 지속적인 옵션입니다. Oracle 버전 12.2 DB 인스턴스에서 Oracle 레이블 보안을 제거할 수 없습니다.

Oracle 레이블 보안에 대한 필수 선행 조건

다음은 Oracle 레이블 보안 사용을 위한 필수 선행 조건입니다.

- DB 인스턴스에서 기본 보유 라이선스 사용 모델을 사용해야 합니다. 자세한 내용은 [Oracle 라이선싱 \(p. 704\)](#) 단원을 참조하십시오.
- Oracle Enterprise Edition에 유효한 라이선스(소프트웨어 업데이트 라이선스 및 지원 포함)가 있어야 합니다.
- Oracle 라이선스에 레이블 보안 옵션이 포함되어 있어야 합니다.

Oracle 레이블 보안 옵션 추가

Oracle 레이블 보안 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동시킵니다.

레이블 보안 옵션을 추가하면 옵션 그룹이 활성화되고 레이블 보안이 활성화됩니다.

DB 인스턴스에 레이블 보안 옵션을 추가하려면

1. 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않으면 다음 설정을 사용하여 사용자 지정 DB 옵션을 생성합니다.
 - a. [Engine]에서 [oracle-ee]를 선택합니다.
 - b. 메이저 엔진 버전에서 DB 인스턴스의 버전을 선택합니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.

2. [OLS] 옵션을 옵션 그룹에 추가합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#) 단원을 참조하십시오.

Important

하나 이상의 DB 인스턴스에 이미 연결되어 있는 기존 옵션 그룹에 레이블 보안을 추가하면 모든 DB 인스턴스가 다시 시작됩니다.

3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:

- 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 정보는 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
- 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 기존 DB 인스턴스에 레이블 보안 옵션을 추가하는 경우 DB 인스턴스를 자동으로 다시 시작하는 동안 인스턴스가 잠시 중단됩니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

Oracle 레이블 보안 사용

Oracle 레이블 보안을 사용하려면 테이블의 특정 행에 대한 액세스를 제어하는 정책을 생성합니다. 자세한 내용은 Oracle 설명서의 [Creating an Oracle Label Security Policy](#)를 참조하십시오.

레이블 보안을 사용할 경우 모든 작업을 LBAC_DBA 역할로 수행합니다. LBAC_DBA 역할은 DB 인스턴스에 대한 마스터 사용자에게 부여됩니다. 다른 사용자에게 레이블 보안 정책을 관리할 수 있는 LBAC_DBA 역할을 부여할 수 있습니다.

Oracle 12.2용 Amazon RDS DB 인스턴스의 경우, Oracle 레이블 보안에 액세스해야 하는 새로운 사용자에게 OLS_ENFORCEMENT 패키지에 대한 액세스 권한을 부여해야 합니다. OLS_ENFORCEMENT 패키지에 대한 액세스 권한을 부여하려면 마스터 사용자로 DB 인스턴스에 연결하고 다음 SQL 문을 실행합니다.

```
GRANT ALL ON LBACSYS.OLS_ENFORCEMENT TO username;
```

Oracle Enterprise Manager(OEM) Cloud Control을 통해 레이블 보안을 구성할 수 있습니다. Amazon RDS는 Management Agent 옵션을 통해 OEM Cloud Control을 지원합니다. 자세한 내용은 [Oracle Management Agent for Enterprise Manager Cloud Control \(p. 793\)](#) 단원을 참조하십시오.

Oracle 레이블 보안 옵션 제거

DB 인스턴스에서 Oracle 레이블 보안을 제거할 수 있습니다.

DB 인스턴스에서 레이블 보안을 제거하려면 다음 중 하나를 수행합니다.

- 여러 DB 인스턴스에서 레이블 보안을 제거하려면 레이블 보안이 속한 옵션 그룹에서 해당 레이블 보안 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 여러 DB 인스턴스에 연결된 옵션 그룹에서 레이블 보안을 제거하면 모든 DB 인스턴스가 다시 시작됩니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- 단일 DB 인스턴스에서 레이블 보안을 제거하려면 DB 인스턴스를 수정하고 레이블 보안 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. 레이블 보안 옵션을 제거하는 경우 DB 인스턴스를 자동으로 다시 시작하는 동안 인스턴스가 잠시 중단됩니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

문제 해결

다음은 Oracle 레이블 보안을 사용할 때 생길 수 있는 문제입니다.

문제	문제 해결 제안
정책을 생성하려고 하면 다음과 비슷한 오류 메시지가 표시됩니다. <code>insufficient authorization for the SYSDBA package.</code>	사용자의 이름이 16자 또는 24자인 경우 레이블 보안 명령을 실행할 수 없는 Oracle 레이블 보안 기능의 알려진 문제입니다. 새 사용자의 이름 문자 수를 변경하고 새

문제	문제 해결 제안
	사용자에게 LBAC_DBAs를 부여한 다음 새 사용자로 로그인하여 OLS 명령을 실행할 수 있습니다. 자세한 내용은 Oracle 지원 부서에 문의하십시오.

관련 주제

- 옵션 그룹 작업 (p. 148)
- Oracle DB 인스턴스 옵션 (p. 772)

Oracle Locator

Amazon RDS는 LOCATOR 옵션 사용을 통해 Oracle Locator를 지원합니다. Oracle Locator에는 인터넷 및 무선 서비스 기반 애플리케이션과 파트너 기반 GIS 솔루션을 지원할 때 필요한 기능이 있습니다. Oracle Locator는 Oracle Spatial의 제한된 서브셋입니다. 자세한 내용은 Oracle 문서의 [Oracle Locator](#)를 참조하십시오.

Important

Oracle Locator를 사용하면 CVSS(공통 취약성 평가 시스템) 점수가 9 이상인 보안 취약성 또는 기타 발표된 보안 취약성이 있는 경우 Amazon RDS에서 DB 인스턴스를 최신 Oracle PSU로 자동 업데이트합니다.

Amazon RDS는 Oracle의 다음 에디션 및 버전에 대한 Oracle Locator를 지원합니다.

- Oracle Standard Edition(SE2) 또는 Enterprise Edition, 버전 12.2.0.1, 모든 버전
- Oracle 스탠다드 에디션(SE2) 또는 엔터프라이즈 에디션, 버전 12.1.0.2.v13 또는 이후 버전
- Oracle 스탠다드 에디션(SE, SE1) 또는 엔터프라이즈 에디션, 버전 11.2.0.4.v17 또는 이후 버전

Oracle Locator의 사전 요구 사항

Oracle Locator 사용을 위한 사전 요구 사항은 다음과 같습니다.

- DB 인스턴스는 VPC(가상 사설 클라우드) 내에 있어야 합니다. 자세한 내용은 [EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 \(p. 407\)](#)을(를) 참조하십시오.
- DB 인스턴스 클래스는 충분해야 합니다. Oracle Locator는 db.m1.smal, db.t2.micro 또는 db.t2.small DB 인스턴스 클래스에 대해 지원되지 않습니다. 자세한 내용은 [Oracle을 위한 DB 인스턴스 클래스 지원 \(p. 705\)](#) 단원을 참조하십시오.
- DB 인스턴스에서 자동 마이너 버전 업그레이드를 활성화해야 합니다. CVSS 점수가 9 이상인 보안 취약성 또는 기타 발표된 보안 취약성이 있는 경우 Amazon RDS는 DB 인스턴스를 최신 Oracle PSU로 업데이트 합니다. 자세한 내용은 [Oracle DB 인스턴스 설정 \(p. 744\)](#) 단원을 참조하십시오.
- DB 인스턴스가 메이저 버전 11.2에서 실행되는 경우 XMLDB 옵션을 설치해야 합니다. 자세한 내용은 [Oracle XML DB \(p. 831\)](#) 단원을 참조하십시오.

Oracle Locator의 모범 사례

다음은 Oracle Locator 사용에 관한 모범 사례입니다.

- 보안을 극대화하기 위해 Secure Sockets Layer(SSL)와 함께 LOCATOR 옵션을 사용합니다. 자세한 내용은 [Oracle 보안 소켓 Layer \(p. 810\)](#) 단원을 참조하십시오.

- DB 인스턴스에 대한 액세스를 제한하도록 DB 인스턴스를 구성합니다. 자세한 내용은 [VPC에서 DB 인스턴스에 액세스하는 시나리오 \(p. 408\)](#) 및 [VPC에서 Amazon RDS DB 인스턴스를 사용한 작업 \(p. 415\)](#) 단원을 참조하십시오.

Oracle Locator 옵션 추가

LOCATOR 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동시킵니다.

LOCATOR 옵션을 추가하는 동안 잠시 작동이 중단됩니다. 옵션을 추가한 후 DB 인스턴스를 재시작할 필요가 없습니다. 옵션 그룹이 활성화되는 즉시 Oracle Locator를 사용할 수 있습니다.

LOCATOR 옵션을 DB 인스턴스에 추가하려면

1. 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않으면 다음 설정을 사용하여 사용자 지정 DB 옵션을 생성합니다.
 - a. [Engine]에서 DB 인스턴스의 Oracle 에디션을 선택합니다.
 - b. 메이저 엔진 버전에서 DB 인스턴스의 버전을 선택합니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.

2. [LOCATOR] 옵션을 옵션 그룹에 추가합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#)을(를) 참조하십시오.
3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:
 - 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
 - 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

Oracle Locator 사용

Oracle Locator 옵션을 활성화한 후에 사용할 수 있습니다. Oracle Locator 기능만 사용해야 합니다. Oracle Spatial 기능을 사용하려면 Oracle Spatial 라이선스가 있어야 합니다.

Oracle Locator에 지원되는 기능 목록은 Oracle 설명서에서 [Locator에 포함되는 기능](#)을 참조하십시오.

Oracle Locator에 지원되지 않는 기능 목록은 Oracle 설명서에서 [Locator에 포함되지 않는 기능](#)을 참조하십시오.

Oracle Locator 옵션 제거

DB 인스턴스에서 LOCATOR 옵션을 제거할 수 있습니다. 옵션을 제거하는 동안 잠시 작동이 중단됩니다. LOCATOR 옵션을 제거한 후 DB 인스턴스를 재시작할 필요가 없습니다.

Warning

LOCATOR 옵션을 삭제했을 때 DB 인스턴스가 옵션의 일부로 활성화되어 있는 데이터 형식을 사용하고 있다면 데이터가 손실될 수 있습니다. 따라서 처리 전에 데이터를 백업해야 합니다. 자세한 내용은 [Amazon RDS DB 인스턴스 백업 및 복원 \(p. 196\)](#) 단원을 참조하십시오.

DB 인스턴스에서 LOCATOR 옵션을 제거하려면 다음 중 하나를 수행합니다.

- 소속 옵션 그룹에서 LOCATOR 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- DB 인스턴스를 수정하고 LOCATOR 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 이 변경은 단일 DB 인스턴스에 영향을 미칩니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

관련 주제

- Oracle Spatial (p. 815)
- Oracle DB 인스턴스 옵션 (p. 772)
- 옵션 그룹 작업 (p. 148)

Oracle Multimedia

Amazon RDS는 MULTIMEDIA 옵션 사용을 통해 Oracle Multimedia를 지원합니다. Oracle Multimedia를 사용하여 이미지, 오디오, 동영상 및 기타 이종 미디어 데이터를 저장하고 관리하며 검색할 수 있습니다. 자세한 내용은 Oracle 문서의 [Oracle Multimedia](#)를 참조하십시오.

Important

Oracle Multimedia를 사용하면 CVSS(공통 취약성 평가 시스템) 점수가 9 이상인 보안 취약성 또는 기타 발표된 보안 취약성이 있는 경우 Amazon RDS에서 DB 인스턴스를 최신 Oracle PSU로 자동 업데이트합니다.

Amazon RDS는 Oracle의 다음 에디션 및 버전에 대한 Oracle Multimedia를 지원합니다.

- Oracle Enterprise Edition, 버전 12.2.0.1, 모든 버전
- Oracle 엔터프라이즈 에디션, 버전 12.1.0.2.v13 또는 이후 버전
- Oracle 엔터프라이즈 에디션, 버전 11.2.0.4.v17 또는 이후 버전

Oracle Multimedia의 사전 요구 사항

Oracle Multimedia 사용을 위한 사전 요구 사항은 다음과 같습니다.

- DB 인스턴스는 VPC(가상 사설 클라우드) 내에 있어야 합니다. 자세한 내용은 [EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 \(p. 407\)](#)을(를) 참조하십시오.
- DB 인스턴스 클래스는 충분해야 합니다. Oracle Multimedia는 db.m1.small, db.t2.micro 또는 db.t2.small DB 인스턴스 클래스에 대해 지원되지 않습니다. 자세한 내용은 [Oracle을 위한 DB 인스턴스 클래스 지원 \(p. 705\)](#) 단원을 참조하십시오.
- DB 인스턴스에서 자동 마이너 버전 업그레이드를 활성화해야 합니다. CVSS 점수가 9 이상인 보안 취약성 또는 기타 발표된 보안 취약성이 있는 경우 Amazon RDS는 DB 인스턴스를 최신 Oracle PSU로 업데이트 합니다. 자세한 내용은 [Oracle DB 인스턴스 설정 \(p. 744\)](#) 단원을 참조하십시오.
- DB 인스턴스가 메이저 버전 11.2에서 실행되는 경우 XMLDB 옵션을 설치해야 합니다. 자세한 내용은 [Oracle XML DB \(p. 831\)](#) 단원을 참조하십시오.

Oracle Multimedia의 모범 사례

Oracle Multimedia 사용을 위한 모범 사례는 다음과 같습니다.

- 보안을 극대화하기 위해 Secure Sockets Layer(SSL)와 함께 MULTIMEDIA 옵션을 사용합니다. 자세한 내용은 [Oracle 보안 소켓 Layer \(p. 810\)](#) 단원을 참조하십시오.
- DB 인스턴스에 대한 액세스를 제한하도록 DB 인스턴스를 구성합니다. 자세한 내용은 [VPC에서 DB 인스턴스에 액세스하는 시나리오 \(p. 408\)](#) 및 [VPC에서 Amazon RDS DB 인스턴스를 사용한 작업 \(p. 415\)](#) 단원을 참조하십시오.

Oracle Multimedia 옵션 추가

MULTIMEDIA 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

- 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
- 옵션을 옵션 그룹에 추가합니다.
- 옵션 그룹을 DB 인스턴스에 연동시킵니다.

MULTIMEDIA 옵션을 추가하는 동안 잠시 작동이 중단됩니다. 옵션을 추가한 후 DB 인스턴스를 재시작할 필요가 없습니다. 옵션 그룹이 활성화되는 즉시 Oracle Multimedia를 사용할 수 있습니다.

MULTIMEDIA 옵션을 DB 인스턴스에 추가하려면

- 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않으면 다음 설정을 사용하여 사용자 지정 DB 옵션을 생성합니다.
 - [Engine]에서 [oracle-ee]를 선택합니다.
 - 메이저 엔진 버전에서 DB 인스턴스의 버전을 선택합니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.

- [MULTIMEDIA] 옵션을 옵션 그룹에 추가합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#)을(를) 참조하십시오.
- 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:
 - 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
 - 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

Oracle Multimedia 옵션 제거

DB 인스턴스에서 MULTIMEDIA 옵션을 제거할 수 있습니다. 옵션을 제거하는 동안 잠시 작동이 중단됩니다. MULTIMEDIA 옵션을 제거한 후 DB 인스턴스를 재시작할 필요가 없습니다.

Warning

MULTIMEDIA 옵션을 삭제했을 때 DB 인스턴스가 옵션의 일부로 활성화되어 있는 데이터 형식을 사용하고 있다면 데이터가 손실될 수 있습니다. 따라서 처리 전에 데이터를 백업해야 합니다. 자세한 내용은 [Amazon RDS DB 인스턴스 백업 및 복원 \(p. 196\)](#) 단원을 참조하십시오.

DB 인스턴스에서 MULTIMEDIA 옵션을 제거하려면 다음 중 하나를 수행합니다.

- 소속 옵션 그룹에서 MULTIMEDIA 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- DB 인스턴스를 수정하고 MULTIMEDIA 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 이 변경은 단일 DB 인스턴스에 영향을 미칩니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정

할 수 있습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

관련 주제

- [옵션 그룹 작업 \(p. 148\)](#)
- [Oracle DB 인스턴스 옵션 \(p. 772\)](#)

Oracle 기본 네트워크 암호화

Amazon RDS는 Oracle 기본 네트워크 암호화(NNE)를 지원합니다. 기본 네트워크 암호화를 사용하면 DB 인스턴스에서 양방향으로 이동하는 데이터를 암호화할 수 있습니다. Amazon RDS는 Oracle의 모든 에디션에 NNE를 지원합니다.

Oracle 기본 네트워크 암호화에 대한 자세한 설명은 본 문서의 범위에서 벗어나지만 배포할 솔루션을 결정하려면 각 알고리즘과 키의 장단점은 잘 알고 있어야 합니다. Oracle 기본 네트워크 암호화를 통해 제공되는 알고리즘 및 키에 대한 자세한 내용은 Oracle 설명서의 [Configuring Network Data Encryption](#) 단원을 참조하십시오. AWS 보안에 대한 자세한 내용은 [AWS 보안 센터](#)를 참조하십시오.

Note

Native Network Encryption 또는 Secure Sockets Layer를 사용할 수 있지만 둘 다 사용할 수는 없습니다. 자세한 내용은 [Oracle 보안 소켓 Layer \(p. 810\)](#) 단원을 참조하십시오.

NNE 옵션 설정

Amazon RDS는 NNE 옵션에 대해 다음 설정을 지원합니다.

옵션 설정	유효한 값	기본값	설명
SQLNET.ENCRYPTION_SERVER	OPTION, Rejected, Requested, Required	Rejected	클라이언트 또는 클라이언트 역할을 하는 서버가 DB 인스턴스에 연결된 경우의 암호화 동작입니다. Requested는 DB 인스턴스에서 클라이언트의 송신 트래픽을 암호화 할 필요가 없음을 나타냅니다.
SQLNET.CRYPTO_CHECKSUM_SERVER	OPTION, CHECKSUM, SERVER	None	클라이언트 또는 클라이언트 역할을 하는 서버가 DB 인스턴스에 연결된 경우의 데이터 무결성 동작입니다. Requested는 DB 인스턴스에서 클라이언트에 체크섬을 수행하도록 요구할 필요가 없음을 나타냅니다.
SQLNET.ENCRYPTION_TYPE_SERVER	AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	DB 인스턴스에서 사용하는 암호화 알고리즘 목록입니다. DB 인스턴스에서는 알고리즘이 성공하거나 목록의 끝에 도달할 때까지 각 알고리즘을 순서대로 사용하여 클라이언트 입력을 해독합니다. Amazon RDS는 다음과 같은 Oracle의 기본 목록을 사용합니다. 순서를 변경

옵션 설정	유효한 값	기본값	설명
	RC4_40, DES40		<p>하거나 DB 인스턴스가 수락할 알고리즘을 제한할 수 있습니다.</p> <ol style="list-style-type: none"> 1. RC4_256: RSA RC4(256비트 키 크기) 2. AES256: AES(256비트 키 크기) 3. AES192: AES(192비트 키 크기) 4. 3DES168112: 3키 Triple-DES(비트 유효 키 크기) 5. RC4_128: RSA RC4(128비트 키 크기) 6. AES128: AES(128비트 키 크기) 7. 3DES1122: Triple-DES(80비트 유효 키 크기) 8. RC4_56: RSA RC4(56비트 키 크기) 9. DES: Standard DES(56비트 키 크기) 10. RC4_40: RSA RC4(40비트 키 크기) 11. DES40: DES40(40비트 키 크기)
SQLNET.CRYPTOGRAPHIC_CHECKSUM_PASSWORDSERVER	SHA056, SHA384, SHA512, SHA1, MD5	MD5	체크섬 알고리즘입니다.

NNE 옵션 추가

NNE 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동 시킵니다.

NNE 옵션을 추가하면 옵션 그룹이 활성화되는 순간, NNE가 활성화됩니다.

DB 인스턴스에 NNE 옵션을 추가하려면

1. [Engine]에서 사용할 Oracle 버전을 선택합니다. NNE는 모든 에디션에서 지원됩니다.
2. 메이저 엔진 버전에서 DB 인스턴스의 버전을 선택합니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.

3. [NNE] 옵션을 옵션 그룹에 추가합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#) 단원을 참조하십시오.

Note

NNE 옵션을 추가한 후 DB 인스턴스를 재시작할 필요가 없습니다. 옵션 그룹이 활성화되자마자 NNE가 활성화됩니다.

4. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:

- 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 정보는 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
- 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. NNE 옵션을 추가한 후 DB 인스턴스를 재시작할 필요가 없습니다. 옵션 그룹이 활성화되자마자 NNE가 활성화됩니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

NNE 사용

Oracle 기본 네트워크 암호화를 사용하여 클라이언트 쪽에서 네트워크 암호화를 지정할 수도 있습니다. 클라이언트(DB 인스턴스에 연결하는 데 사용된 컴퓨터)에서 sqlnet.ora 파일을 사용하여 SQLNET.CRYPTO_CHECKSUM_CLIENT, SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT, SQLNET.ENCRYPTION_CLIENT 및 SQLNET.ENCRYPTION_TYPES_CLIENT 클라이언트 설정을 지정할 수 있습니다. 자세한 내용은 Oracle 설명서의 [Configuring Network Data Encryption and Integrity for Oracle Servers and Clients](#) 단원을 참조하십시오.

DB 인스턴스에서 애플리케이션의 연결 요청을 거부하는 경우도 있습니다(예: 클라이언트와 서버의 암호화 알고리즘이 일치하지 않는 경우).

Oracle 기본 네트워크 암호화를 테스트하려면 클라이언트의 sqlnet.ora 파일에 다음 줄을 추가합니다.

```
DIAG_ADR_ENABLED=off
TRACE_DIRECTORY_CLIENT=/tmp
TRACE_FILE_CLIENT=nettrace
TRACE_LEVEL_CLIENT=16
```

이 줄은 연결이 시도될 때 클라이언트에서 /tmp/nettrace*라는 추적 파일을 생성합니다. 추적 파일에는 연결 관련 정보가 포함되어 있습니다. Oracle 기본 네트워크 암호화를 사용할 때 연결 관련 문제에 대한 자세한 내용은 Oracle 설명서의 [About Negotiating Encryption and Integrity](#)를 참조하십시오.

NNE 설정 수정

NNE를 활성화한 후 옵션 설정을 수정할 수 있습니다. 옵션 설정을 변경하는 방법에 대한 자세한 내용은 [옵션 설정 변경 \(p. 156\)](#) 단원을 참조하십시오. 각 설정에 대한 자세한 내용은 [NNE 옵션 설정 \(p. 807\)](#) 단원을 참조하십시오.

NNE 옵션 제거

DB 인스턴스에서 NNE를 제거할 수 있습니다.

DB 인스턴스에서 NNE를 제거하려면 다음 중 하나를 수행합니다.

- 여러 DB 인스턴스에서 NNE를 제거하려면 인스턴스가 속한 옵션 그룹에서 해당 NNE 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. NNE 옵션을 제거한 후 DB 인스턴스를 재시작할 필요가 없습니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- 단일 DB 인스턴스에서 NNE를 제거하려면 DB 인스턴스를 수정하고 NNE 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. NNE 옵션을 제거한 후 DB 인스턴스를 재시작할 필요가 없습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

관련 주제

- [옵션 그룹 작업 \(p. 148\)](#)
- [Oracle DB 인스턴스 옵션 \(p. 772\)](#)

Oracle 보안 소켓 Layer

Oracle DB 인스턴스와 연결된 옵션 그룹에 Oracle SSL 옵션을 추가하여 Oracle DB 인스턴스에 대해 Secure Sockets Layer(SSL) 암호화를 활성화할 수 있습니다. SSL을 사용하여 통신에 사용할 포트를 지정합니다. 다음 단원과 같이 SQL*Plus를 구성해야 합니다.

Oracle DB 인스턴스와 연결된 옵션 그룹에 Oracle SSL 옵션을 추가하여 DB 인스턴스에 대해 SSL 암호화를 활성화할 수 있습니다. Amazon RDS는 Oracle의 요구 사항에 따라 두 번째 포트를 SSL 연결에 사용합니다. 이 접근 방식에서는 DB 인스턴스와 SQL*Plus 간에 클리어 텍스트 통신과 SSL로 암호화된 통신이 동시에 발생할 수 있습니다. 예를 들어 이 포트를 클리어 텍스트 통신에 사용하여 VPC 내의 다른 리소스와 통신하면서 동일한 포트를 SSL로 암호화된 통신에 사용하여 VPC 외부의 리소스와 통신할 수 있습니다.

Note

Secure Sockets Layer 또는 Native Network Encryption를 사용할 수 있지만 둘 다 사용할 수는 없습니다. 자세한 내용은 [Oracle 기본 네트워크 암호화 \(p. 807\)](#) 단원을 참조하십시오.

SSL 암호화는 다음과 같은 Oracle 데이터베이스 버전에 사용할 수 있습니다.

- 12.2.0.1: 모든 버전, 모든 에디션(Standard Edition Two 포함)
- 12.1.0.2: 모든 버전, 모든 에디션(Standard Edition Two 포함)
- 11.2.0.4: 모든 버전, Enterprise Edition
- 11.2.0.4: 버전 6 이상, Standard Edition, Standard Edition One, Enterprise Edition

Note

동일한 인스턴스에서 SSL과 Oracle NNE(기본 네트워크 암호화)를 모두 사용할 수 없습니다. SSL 암호화를 사용할 경우 다른 모든 연결 암호화를 비활성화해야 합니다.

Oracle SSL 옵션에 대한 TLS 버전

Oracle용 Amazon RDS는 TLS(전송 계층 보안) 버전 1.0 및 1.2를 지원합니다. Oracle SSL 옵션을 사용하려면 `SQLNET.SSL_VERSION` 옵션 설정을 사용해야 합니다. 다음은 이 옵션 설정으로 허용되는 값입니다.

- "1.0" – 클라이언트는 TLS 1.0만 사용하여 DB 인스턴스에 연결할 수 있습니다.
- "1.2" – 클라이언트는 TLS 1.2만 사용하여 DB 인스턴스에 연결할 수 있습니다.
- "1.2 or 1.0" – 클라이언트는 TLS 1.2 또는 1.0을 사용하여 DB 인스턴스에 연결할 수 있습니다.

Oracle SSL 옵션을 사용하려면 `SQLNET.SSL_VERSION` 옵션 설정도 필요합니다.

- 기존 Oracle SSL 옵션에서 `SQLNET.SSL_VERSION`은 "1.0"으로 자동 설정됩니다. 필요할 경우 설정을 변경할 수 있습니다.
- 새 Oracle SSL 옵션을 추가할 때 유효한 값에 `SQLNET.SSL_VERSION`을 명시적으로 설정해야 합니다.

다음 표는 다른 Oracle 엔진 버전과 에디션에서 지원되는 TLS 옵션 설정입니다.

Oracle 엔진 버전	<code>SQLNET.SSL_VERSION</code> =	<code>SQLNET.SSL_VERSION</code> =	<code>SQLNET.SSL_VERSION</code> ="1.2 또는 1.0"
12.2.0.1(모든 버전)	지원	지원	지원
12.1.0.2(모든 버전)	지원	지원	지원
11.2.0.4(Oracle EE)	지원	11.2.0.4.v8 이상 지원	11.2.0.4.v8 이상 지원

Oracle 엔진 버전	SQLNET.SSL_VERSION=	SQLNET.SSL_VERSION=	SQLNET.SSL_VERSION="1.2 또는 1.0"
11.2.0.4(Oracle SE1)	지원	지원되지 않음	지원되지 않음
11.2.0.4(Oracle SE)	지원	지원되지 않음	지원되지 않음

Oracle DB 인스턴스에 SSL을 사용하도록 SQL*Plus 구성

Oracle SSL 옵션을 사용하는 Oracle DB 인스턴스에 연결하기 전에 SQL*Plus를 구성해야 합니다.

Note

해당 클라이언트에서 DB 인스턴스에 액세스하는 것을 허용하려면 보안 그룹을 올바르게 구성해야 합니다. 자세한 정보는 [보안 그룹을 통한 액세스 제어 \(p. 390\)](#) 단원을 참조하십시오. 또한 이러한 지침은 Oracle 품을 직접 사용하는 SQL*Plus 및 기타 클라이언트에 대한 것입니다. JDBC 연결에 대한 자세한 정보는 [JDBC를 통한 SSL 연결 설정 \(p. 813\)](#) 단원을 참조하십시오.

SSL을 사용하여 Oracle DB 인스턴스에 연결하도록 SQL*Plus를 구성하려면

1. ORACLE_HOME 환경 변수를 Oracle 품 디렉터리의 위치로 설정합니다.

Oracle 품 디렉터리 경로는 설치에 따라 달라집니다. 다음은 ORACLE_HOME 환경 변수를 설정하는 예제입니다.

```
prompt>export ORACLE_HOME=/home/user/app/user/product/12.1.0/dbhome_1
```

Oracle 환경 변수 설정에 대한 자세한 정보는 Oracle 설명서의 [SQL*Plus Environment Variables](#) 및 운영 체제에 해당하는 Oracle 설치 안내서를 참조하십시오.

2. \$ORACLE_HOME/lib를 LD_LIBRARY_PATH 환경 변수에 추가합니다.

다음은 LD_LIBRARY_PATH 환경 변수를 설정하는 예제입니다.

```
prompt>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

3. \$ORACLE_HOME/ssl_wallet에 Oracle Wallet을 위한 디렉터리를 만듭니다.

다음은 Oracle Wallet 디렉터리를 생성하는 예제입니다.

```
prompt>mkdir $ORACLE_HOME/ssl_wallet
```

4. 모든 리전에서 작동하는 루트 인증서를 다운로드하고 파일을 ssl_wallet 디렉터리에 저장하십시오.

루트 인증서 다운로드에 대한 자세한 내용은 [SSL을 사용하여 DB 인스턴스에 대한 연결 암호화 \(p. 388\)](#) 단원을 참조하십시오.

5. \$ORACLE_HOME/network/admin 디렉터리에서 tnsnames.ora 파일을 수정하거나 만들고 다음 항목을 포함합니다.

```
<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. 동일한 디렉터리에서 sqlnet.ora 파일을 수정하거나 만들고 다음 파라미터를 포함합니다.

Note

TLS 보안 연결을 통해 개체와 통신하기 위해 Oracle이 인증에 필요한 인증서가 있는 wallet을 요구합니다. 7단계에서와 같이 Oracle의 ORAPKI 유ти리티를 사용하여 Oracle Wallet을 만들고 유지 관리할 수 있습니다. 자세한 내용은 Oracle 설명서에서 [ORAPKI를 이용한 Oracle Wallet 설정](#)을 참조하십시오.

```
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

DB 인스턴스가 지원하는 경우 SSL_VERSION을 더 높은 값으로 설정할 수 있습니다.

7. 다음 명령을 실행하여 Oracle Wallet을 만듭니다.

```
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-2015-root.pem -auto_login_only
```

파일 이름을 다운로드한 파일 이름으로 바꿉니다.

SSL을 사용하여 Oracle DB 인스턴스에 연결

앞에서 설명한 대로 SSL을 사용하도록 SQL*Plus를 구성한 후에는 SSL 옵션을 통해 Oracle DB 인스턴스에 연결할 수 있습니다. 선택적으로, tnsnames.ora 및 sqlnet.ora 파일이 들어있는 디렉터리를 가리키는 TNS_ADMIN 값을 먼저 내보낼 수 있습니다. 그러면 SQL *Plus가 이러한 파일을 일관되게 찾을 수 있습니다. 다음 예에서는 TNS_ADMIN 값을 내보냅니다.

```
export TNS_ADMIN = ${ORACLE_HOME}/network/admin
```

DB 인스턴스에 연결합니다. 예를 들어 SQL*Plus와 tnsnames.ora 파일의 <net_service_name>을 사용하여 연결할 수 있습니다.

```
sqlplus <mydbuser>@<net_service_name>
```

다음 명령을 사용하면 tnsnames.ora 파일 사용 없이 SQL*Plus를 사용하여 DB 인스턴스에 연결할 수도 있습니다.

```
sqlplus '<mydbuser>@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCPS)(HOST = <endpoint>) (PORT = <ssl port number>))(CONNECT_DATA = (SID = <database name>)))'
```

SSL을 사용하지 않고 Oracle DB 인스턴스에 연결할 수도 있습니다. 예를 들어 다음 명령을 실행하면 SSL 암호화 없이 클리어 텍스트 포트를 통해 DB 인스턴스에 연결됩니다.

```
sqlplus '<mydbuser>@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = <endpoint>) (PORT = <port number>))(CONNECT_DATA = (SID = <database name>)))'
```

TCP(Transmission Control Protocol) 포트 액세스를 닫으려는 경우 IP 주소 수신이 없는 보안 그룹을 만들어 인스턴스에 추가합니다. 이렇게 하면 TCP 포트를 통한 연결이 닫히지만 SSL 옵션 보안 그룹에 의해 허용되는 범위 내에 있는 IP 주소를 사용하여 지정되는 SSL 포트를 통해 계속 연결할 수 있습니다.

JDBC를 통한 SSL 연결 설정

JDBC를 통한 SSL 연결을 사용하려면 키 스토어를 만들고 Amazon RDS 루트 CA 인증서를 신뢰하고 아래에 지정된 코드 조각을 사용해야 합니다.

키 스토어를 JKS 형식으로 만들려면 다음 명령을 사용합니다. 키 스토어 만들기에 대한 자세한 정보는 [Oracle 설명서](#)를 참조하십시오.

```
keytool -keystore clientkeystore -genkey -alias client
```

그 다음에는 다음 단계에 따라 Amazon RDS 루트 CA 인증서를 신뢰합니다.

Amazon RDS 루트 CA 인증서를 신뢰하려면

- 모든 리전에서 작동하는 루트 인증서를 다운로드하고 파일을 ssl_wallet 디렉터리에 저장하십시오.

루트 인증서 다운로드에 대한 자세한 내용은 [SSL을 사용하여 DB 인스턴스에 대한 연결 암호화 \(p. 388\)](#) 단원을 참조하십시오.

- 다음 명령을 사용하여 인증서를 .der 형식으로 변환합니다.

```
openssl x509 -outform der -in rds-ca-2015-root.pem -out rds-ca-2015-root.der
```

파일 이름을 다운로드한 파일 이름으로 바꿉니다.

- 다음 명령을 사용하여 인증서를 키 스토어로 가져옵니다.

```
keytool -import -alias rds-root -keystore clientkeystore -file rds-ca-2015-root.der
```

- 키 스토어가 성공적으로 만들어졌는지 확인하십시오.

```
keytool -list -v -keystore clientkeystore.jks
```

메시지가 표시되면 키 스토어 암호를 입력하십시오.

다음은 JDBC를 사용하여 SSL 연결을 설정하는 방법을 보여 주는 코드 예입니다.

```
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: https://s3.amazonaws.com/rds-
downloads/rds-ca-2015-root.pem
    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
    }
}
```

SSL 연결과 DN 일치 적용

Oracle 파라미터 SSL_SERVER_DN_MATCH를 사용하여 데이터베이스 서버에 대한 DN(고유 이름)을 서비스 이름과 일치하도록 적용할 수 있습니다. 일치 확인을 적용한 경우 SSL에서 인증서가 서버에서 가져온 것인지를 확인합니다. 일치 확인을 적용하지 않은 경우 SSL에서 확인을 수행하지만 일치 여부에 상관없이 연결을 허용합니다. 일치를 적용하지 않은 경우 서버 ID가 위조될 수 있습니다.

DN 일치를 적용하려면 DN 일치 속성을 추가하고 아래 지정된 연결 문자열을 사용합니다.

다음과 같이 속성을 클라이언트 연결에 추가하여 DN 일치를 적용합니다.

```
properties.put("oracle.net.ssl_server_dn_match", "TRUE");
```

SSL을 사용할 때 DN 일치를 적용하려면 다음 연결 문자열을 사용합니다.

```
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 RDS는 SPATIAL 옵션 사용을 통해 Oracle Spatial을 지원합니다. Oracle Spatial에는 Oracle 데이터베이스에서 공간 데이터의 저장, 검색, 업데이트 및 쿼리를 신속하게 실행할 수 있는 SQL 스키마 및 기능이 있습니다. 자세한 내용은 Oracle 설명서에서 [Spatial Concepts](#)를 참조하십시오.

Important

Oracle Spatial을 사용하면 CVSS(공통 취약성 평가 시스템) 점수가 9 이상인 보안 취약성 또는 기타 발표된 보안 취약성이 있는 경우 Amazon RDS에서 DB 인스턴스를 최신 Oracle PSU로 자동 업데이트합니다.

Amazon RDS는 Oracle의 다음 에디션 및 버전에 대해 Oracle Spatial을 지원합니다.

- Oracle Enterprise Edition, 버전 12.2.0.1, 모든 버전
- Oracle 엔터프라이즈 에디션, 버전 12.1.0.2.v13 또는 이후 버전
- Oracle 엔터프라이즈 에디션, 버전 11.2.0.4.v17 또는 이후 버전

Oracle Spatial의 사전 요구 사항

Oracle Spatial 사용을 위한 사전 요구 사항은 다음과 같습니다.

- DB 인스턴스는 VPC(가상 사설 클라우드) 내에 있어야 합니다. 자세한 내용은 [EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 \(p. 407\)](#)을(를) 참조하십시오.
- DB 인스턴스 클래스는 충분해야 합니다. Oracle Spatial은 db.m1.small, db.t2.micro 또는 db.t2.small DB 인스턴스 클래스에 대해 지원되지 않습니다. 자세한 내용은 [Oracle을 위한 DB 인스턴스 클래스 지원 \(p. 705\)](#) 단원을 참조하십시오.
- DB 인스턴스에서 자동 마이너 버전 업그레이드를 활성화해야 합니다. CVSS 점수가 9 이상인 보안 취약성 또는 기타 발표된 보안 취약성이 있는 경우 Amazon RDS는 DB 인스턴스를 최신 Oracle PSU로 업데이트 합니다. 자세한 내용은 [Oracle DB 인스턴스 설정 \(p. 744\)](#) 단원을 참조하십시오.
- DB 인스턴스가 메이저 버전 11.2에서 실행되는 경우 XMLDB 옵션을 설치해야 합니다. 자세한 내용은 [Oracle XML DB \(p. 831\)](#) 단원을 참조하십시오.
- Oracle의 Oracle Spatial 라이선스 자세한 내용은 Oracle 설명서의 [Oracle Spatial and Graph](#)을 참조하십시오.

Oracle Spatial의 모범 사례

다음은 Oracle Spatial 사용에 관한 모범 사례입니다.

- 보안을 극대화하기 위해 Secure Sockets Layer(SSL)와 함께 SPATIAL 옵션을 사용합니다. 자세한 내용은 [Oracle 보안 소켓 Layer \(p. 810\)](#) 단원을 참조하십시오.
- DB 인스턴스에 대한 액세스를 제한하도록 DB 인스턴스를 구성합니다. 자세한 내용은 [VPC에서 DB 인스턴스에 액세스하는 시나리오 \(p. 408\)](#) 및 [VPC에서 Amazon RDS DB 인스턴스를 사용한 작업 \(p. 415\)](#) 단원을 참조하십시오.

Oracle Spatial 옵션 추가

SPATIAL 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동시킵니다.

SPATIAL 옵션을 추가하는 동안 잠시 작동이 중단됩니다. 옵션을 추가한 후 DB 인스턴스를 재시작할 필요가 없습니다. 옵션 그룹이 활성화되는 즉시 Oracle Spatial을 사용할 수 있습니다.

SPATIAL 옵션을 DB 인스턴스에 추가하려면

1. 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않으면 다음 설정을 사용하여 사용자 지정 DB 옵션을 생성합니다.
 - a. [Engine]에서 [oracle-ee]를 선택합니다.
 - b. 메이저 엔진 버전에서 DB 인스턴스의 버전을 선택합니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.

2. [SPATIAL] 옵션을 옵션 그룹에 추가합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#)을(를) 참조하십시오.
3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:
 - 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
 - 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

Oracle Spatial 옵션 제거

DB 인스턴스에서 SPATIAL 옵션을 제거할 수 있습니다. 옵션을 제거하는 동안 잠시 작동이 중단됩니다. SPATIAL 옵션을 제거한 후 DB 인스턴스를 재시작할 필요가 없습니다.

Warning

SPATIAL 옵션을 삭제했을 때 DB 인스턴스가 옵션의 일부로 활성화되어 있는 데이터 형식을 사용하고 있다면 데이터가 손실될 수 있습니다. 따라서 처리 전에 데이터를 백업해야 합니다. 자세한 내용은 [Amazon RDS DB 인스턴스 백업 및 복원 \(p. 196\)](#) 단원을 참조하십시오.

DB 인스턴스에서 SPATIAL 옵션을 제거하려면 다음 중 하나를 수행합니다.

- 소속 옵션 그룹에서 SPATIAL 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- DB 인스턴스를 수정하고 SPATIAL 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 이 변경은 단일 DB 인스턴스에 영향을 미칩니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

관련 주제

- [Oracle Locator \(p. 803\)](#)

- Oracle DB 인스턴스 옵션 (p. 772)
- 옵션 그룹 작업 (p. 148)

Oracle SQLT

Amazon RDS는 SQLT 옵션 사용을 통해 Oracle SQLTXPLAIN(SQLT)을 지원합니다.

Oracle EXPLAIN PLAN 문은 SQL 문의 실행 계획을 확인할 수 있습니다. 이 문은 Oracle 옵티マイ저가 특정 실행 계획(예: 중첩 루프 조인)을 선택했는지 확인할 수 있습니다. 또한 옵티マイ저의 결정(예: 해시 조인이 아니라 중첩 루프 조인을 선택한 이유)을 이해하도록 돕습니다. 따라서 EXPLAIN PLAN을 통해 문의 성능을 파악할 수 있습니다.

SQLT는 보고서를 생성하는 Oracle 유ти리티입니다. 이 보고서에는 객체 통계, 객체 메타데이터, 옵티マイ저 관련 초기화 파라미터와 데이터베이스 관리자가 최적의 성능을 위해 SQL 문을 튜닝하는 데 사용할 수 있는 기타 정보가 포함됩니다. SQLT는 모든 셕션에 대한 하이퍼링크가 포함된 HTML 보고서를 생성합니다.

Automatic Workload Repository 또는 Statspack 보고서와 달리 SQLT는 개별 SQL 문에서 작동합니다. SQLT는 성능 데이터를 수집, 저장 및 표시하는 SQLT, PL/SQL 및 SQL*Plus 파일의 모음입니다.

Oracle용 Amazon RDS에서는 현재 다음과 같은 SQLT 버전을 지원합니다.

- 12.1.160429
- 12.2.180331
- 12.2.180725

이를 사용하기 위해 SQLT 및 액세스 지침을 다운로드하려면:

- My Oracle 지원 계정에 로그인하고 다음 문서를 여십시오.
- SQLT를 다운로드하려면: [문서 215187.1](#)
- SQLT 사용 지침: [문서 1614107.1](#)
- SQLT에 대한 FAQ: [문서 1454160.1](#)
- SQLT 출력 읽기에 대한 자세한 내용: [문서 1456176.1](#)
- 주요 보고서: [문서 1922234.1](#)를 해석하려면

SQLT는 다음과 같은 Oracle 데이터베이스 버전에 사용할 수 있습니다.

- Oracle 12c, 12.2.0.1
- Oracle 12c, 12.1.0.2
- Oracle 11g, 11.2.0.4

Amazon RDS는 다음 SQLT 메서드를 지원하지 않습니다.

- XPORE
- XHUME

SQLT의 사전 조건

다음은 SQLT 사용을 위한 사전 조건입니다.

- SQLT에 필요한 사용자 및 역할을 제거해야 합니다(있는 경우).

SQLT 옵션이 DB 인스턴스에 다음 사용자 및 역할을 생성합니다.

- SQLTXPLAIN user
- SQLTXADMIN user
- SQLT_USER_ROLE 역할

DB 인스턴스에 이러한 사용자 또는 역할이 있는 경우 SQL 클라이언트를 사용해 DB 인스턴스에 로그인하여 다음 문을 사용하여 삭제하십시오.

```
DROP USER SQLTXPLAIN CASCADE;
DROP USER SQLTXADMIN CASCADE;
DROP ROLE SQLT_USER_ROLE CASCADE;
```

- SQLT에 필요한 테이블 공간을 제거해야 합니다(있는 경우).

SQL 옵션이 DB 인스턴스에 다음 테이블 공간을 생성합니다.

- RDS_SQLT_TS
- RDS_TEMP_SQLT_TS

DB 인스턴스에 이러한 테이블 공간이 있는 경우 SQL 클라이언트를 사용해 DB 인스턴스에 로그인하여 삭제하십시오.

SQLT 옵션 설정

SQLT에서는 Oracle Tuning Pack 및 Oracle Diagnostics Pack에서 제공하는 사용이 허가된 기능을 사용할 수 있습니다. Oracle Tuning Pack에는 SQL Tuning Advisor가 들어 있고, Oracle Diagnostics Pack에는 Automatic Workload Repository가 들어 있습니다. SQLT 설정에 따라 SQLT에서 이러한 기능에 대한 액세스가 활성화 또는 비활성화됩니다.

Amazon RDS는 SQLT 옵션에 대해 다음 설정을 지원합니다.

옵션 설정	유효한 값	기본값	설명
LICENSE_PACK	T, D, N	T	<p>SQLT를 사용하여 액세스하려고 하는 Oracle Management Pack입니다. 다음 값 중 하나를 입력합니다.</p> <ul style="list-style-type: none">• T는 Oracle Tuning Pack 및 Oracle Diagnostics Pack에 대한 라이선스가 있고 SQLT에서 SQL Tuning Advisor 및 Automatic Workload Repository에 액세스하려고 함을 나타냅니다.• D는 Oracle Diagnostics Pack에 대한 라이선스가 있고 SQLT에서 Automatic Workload Repository에 액세스하려고 함을 나타냅니다.• N은 Oracle Tuning Pack 및 Oracle Diagnostics Pack에 대한 라이선스가 없거나 둘 중 하나 또는 둘 다에 대한 라이선스가 있지만 SQLT에서 이러한 팩에 액세스하려고 하지 않음을 나타냅니다. <p>Note</p> <p>Amazon RDS는 Oracle Management Pack에 대한 라이선스를 제공하지 않습니다. DB에 포함되어 있는 않은 팩을 사용하려고 하는 경우에는 DB 인스턴스에서 SQLT를 사용할 수 있습니다. 그러나 SQLT에서 해당 팩에 액세스할 수 없고 SQLT</p>

옵션 설정	유효한 값	기본값	설명
			보고서에 해당 팩에 대한 데이터가 포함되지 않습니다. 예를 들어, <code>T</code> 를 지정한 경우 DB 인스턴스에는 Oracle Tuning Pack이 포함되지 않고, SQLT가 DB 인스턴스에서 작동하지만 SQLT 보고서에는 Oracle Tuning Pack과 관련된 데이터가 들어 있지 않습니다.
VERSION	2016-04-29, 2016-04-29, 설치하려는 SQLT 버전 2018-03-31, v1, 2018-07-25, v1		

SQLT 옵션 추가

SQLT 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. SQLT 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동시킵니다.

SQLT 옵션을 추가하면 옵션 그룹이 활성화되는 순간, SQLT가 활성화됩니다.

DB 인스턴스에 SQLT 옵션을 추가하려면

1. 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않으면 다음 설정을 사용하여 사용자 지정 DB 옵션을 생성합니다.
 - a. [Engine]에서 사용할 Oracle 버전을 선택합니다. 모든 버전에서 SQLT 옵션이 지원됩니다.
 - b. 메이저 엔진 버전에서 DB 인스턴스의 버전을 선택합니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.

2. [SQLT] 옵션을 옵션 그룹에 추가합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#) 단원을 참조하십시오.
3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:

- 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 정보는 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
- 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

4. (선택 사항) SQLT 옵션을 사용하여 각 DB 인스턴스에서 SQLT 설치를 확인합니다.

- a. SQL 클라이언트를 사용하여 마스터 관리자로 DB 인스턴스에 연결합니다.

SQL 클라이언트를 사용하여 Oracle DB 인스턴스에 연결하는 방법에 대한 자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 \(p. 736\)](#) 단원을 참조하십시오.

- b. 다음 쿼리를 실행합니다.

```
SELECT sqltxplain.sqlt$a.get_param('tool_version') sqlt_version FROM DUAL;
```

이 쿼리는 Amazon RDS에 있는 SQLT 옵션의 현재 버전을 반환합니다. 12.1.160429는 Amazon RDS에서 사용 가능한 SQLT 버전의 예입니다.

5. SQLT 옵션에서 생성한 사용자의 암호를 변경합니다.

- a. SQL 클라이언트를 사용하여 마스터 관리자로 DB 인스턴스에 연결합니다.
- b. 다음 SQL 문을 실행하여 SQLTXADMIN 사용자의 암호를 변경합니다.

```
ALTER USER SQLTXADMIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

- c. 다음 SQL 문을 실행하여 SQLTXPLAIN 사용자의 암호를 변경합니다.

```
ALTER USER SQLTXPLAIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

Note

SQLT를 업그레이드하려면 이전 버전의 SQLT를 제거한 다음 새 버전을 설치해야 합니다. 따라서 SQLT를 업그레이드하면 모든 SQLT 메타데이터가 손실될 수 있습니다. 데이터베이스의 메이저 버전 업그레이드 역시 SQLT를 제거한 후 다시 설치합니다. 메이저 버전 업그레이드의 예로는 Oracle 11g에서 Oracle 12c로의 업그레이드가 있습니다.

SQLT 사용

SQLT는 Oracle SQL*Plus 유ти리티와 함께 작동합니다.

SQLT를 사용하려면

1. My Oracle Support 사이트의 [문서 215187.1](#)에서 SQLT .zip 파일을 다운로드합니다.

Note

My Oracle Support 사이트에서 SQLT 12.1.160429를 다운로드할 수 없습니다. Oracle에는 이 구 버전이 사용되지 않습니다.

2. SQLT .zip 파일의 압축을 풉니다.
3. 명령 프롬프트에서 파일 시스템의 sqlt/run 디렉터리로 변경합니다.
4. 명령 프롬프트에서 SQL*Plus를 열고 DB 인스턴스에 마스터 사용자로 연결합니다.

SQL*Plus를 사용한 DB 인스턴스 연결에 대한 자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 \(p. 736\)](#) 단원을 참조하십시오.

5. SQL 문의 SQL ID를 가져옵니다.

```
SELECT SQL_ID FROM V$SQL WHERE SQL_TEXT='sql_statement';
```

다음과 유사하게 출력됩니다.

```
SQL_ID
-----
chvsmttqjzjkn
```

6. SQLT를 사용하여 SQL 문을 분석합니다.

```
START sqltxtract.sql sql_id sqltxplain_user_password
```

예를 들어 SQL ID chvsmttqjzjkn의 경우 다음과 같이 입력합니다.

```
START sqltxtract.sql chvsmttqjzjkn sqltxplain_user_password
```

SQLT가 SQL 명령이 실행된 디렉터리에 HTML 보고서 및 관련 리소스를 .zip 파일로 생성합니다.

7. (선택 사항) 애플리케이션 사용자가 SQLT를 사용하여 SQL 문을 진단하도록 하려면 다음 문을 사용하여 각 애플리케이션 사용자에게 SQLT_USER_ROLE을 부여하십시오.

```
GRANT ROLE SQLT_USER_ROLE TO application_user_name;
```

Note

Oracle에서는 sys 사용자 또는 DBA 역할을 가진 사용자가 SQLT를 실행하는 것을 권장하지 않습니다. 애플리케이션 사용자에게 SQLT_USER_ROLE을 부여해 애플리케이션 사용자의 계정으로 SQLT 진단을 실행하는 것이 가장 좋습니다.

SQLT 옵션 업그레이드

Oracle용 Amazon RDS를 사용하면 SQLT 옵션을 기존 버전에서 상위 버전으로 업그레이드할 수 있습니다. SQLT 옵션을 업그레이드하려면 SQLT 새 버전에 대한 [SQLT 사용 \(p. 820\)](#)의 1단계-3단계를 완료하십시오. 또한, 이 섹션의 7단계에서 SQLT 이전 버전에 대한 권한을 부여한 경우 새 SQLT 버전에 대한 이 권한을 다시 부여하십시오.

SQLT 옵션을 업그레이드하면 이전 SQLT 버전의 메타데이터가 손실됩니다. 이전 SQLT 버전의 스키마 및 관련 객체는 삭제되고 최신 버전의 SQLT가 설치됩니다. 최신 SQLT 버전의 변경 사항에 대한 자세한 내용은 My Oracle Support 사이트의 [문서 1614201.1](#)을 참조하십시오.

Note

버전 다운그레이드는 지원되지 않습니다.

SQLT 설정 수정

SQLT를 활성화한 후에는 이 옵션에 대한 LICENSE_PACK 및 VERSION 설정을 수정할 수 있습니다.

옵션 설정을 변경하는 방법에 대한 자세한 내용은 [옵션 설정 변경 \(p. 156\)](#)을(를) 참조하십시오. 각 설정에 대한 자세한 내용은 [SQLT 옵션 설정 \(p. 818\)](#) 단원을 참조하십시오.

SQLT 옵션 제거

DB 인스턴스에서 SQLT를 제거할 수 있습니다.

DB 인스턴스에서 SQLT를 제거하려면 다음 중 하나를 수행합니다.

- 여러 DB 인스턴스에서 SQLT를 제거하려면 DB 인스턴스가 속한 옵션 그룹에서 해당 SQLT 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- 단일 DB 인스턴스에서 SQLT를 제거하려면 DB 인스턴스를 수정하고 SQLT 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

관련 주제

- [옵션 그룹 작업 \(p. 148\)](#)
- [Oracle DB 인스턴스 옵션 \(p. 772\)](#)

Oracle Statspack

Oracle 옵션(STATSPACK)은 Oracle Statspack 성능 통계 기능을 설치하고 활성화합니다. Oracle Statspack은 성능 데이터를 수집, 저장 및 표시하는 SQL, PL/SQL 및 SQL*Plus 스크립트의 모음입니다. Oracle Statspack 사용에 대한 자세한 정보는 Oracle 설명서의 [Oracle Statspack](#) 단원을 참조하십시오.

Note

Oracle Statspack은 Oracle에서 더 이상 지원되지 않으며 어드밴스 Automatic Workload Repository(AWR)로 대체되었습니다. AWR은 Diagnostics Pack을 구입한 Oracle Enterprise Edition 고객만 사용할 수 있습니다. Oracle Statspack은 Amazon RDS의 모든 Oracle DB 엔진에서 사용할 수 있습니다.

다음 단계는 Amazon RDS에서 Oracle Statspack을 사용하는 방법을 보여줍니다.

1. PERFSTAT 계정이 이미 생성된 기존 DB 인스턴스가 있고 Oracle Statspack을 해당 인스턴스와 함께 사용하려면 DB 인스턴스에 연결된 옵션 그룹에 Statspack 옵션을 추가하기 전에 PERFSTAT 계정을 삭제해야 합니다. PERFSTAT 계정이 이미 생성된 DB 인스턴스와 연결된 옵션 그룹에 Statspack 옵션을 추가하려고 하면 오류가 발생하고 RDS 이벤트 RDS-Event-0058이 생성됩니다.

이미 Statspack을 설치했으며 PERFSTAT 계정이 Statspack에 연결된 경우에는 이 단계를 건너뛰고 PERFSTAT 사용자를 삭제하지 마십시오.

다음 명령을 실행하여 PERFSTAT 계정을 삭제할 수 있습니다.

```
DROP USER perfstat CASCADE;
```

2. Statspack 옵션을 옵션 그룹에 추가한 다음 해당 옵션 그룹을 DB 인스턴스와 연결합니다. Amazon RDS는 DB 인스턴스에 Statspack 스크립트를 설치한 다음, PERFSTAT 사용자 계정을 설정합니다. 이 계정은 Statspack 스크립트를 실행하는데 사용할 계정입니다. Statspack을 설치한 경우 이 단계를 건너뜁니다.
3. Amazon RDS에서 Statspack을 DB 인스턴스에 설치한 후, 마스터 사용자 이름과 마스터 암호를 사용하여 DB 인스턴스에 로그인해야 합니다. 그런 다음 Statspack을 설치할 때 Amazon RDS에서 임의로 생성된 값으로 PERFSTAT 암호를 재설정해야 합니다. PERFSTAT 암호를 재설정한 이후에 PERFSTAT 사용자 계정을 사용하여 로그인하고 Statspack 스크립트를 실행할 수 있습니다.

다음 명령을 사용하여 암호를 재설정합니다.

```
ALTER USER perfstat IDENTIFIED BY <new_password> ACCOUNT UNLOCK;
```

4. PERFSTAT 계정을 사용하여 로그온한 후 Statspack 스냅샷을 수동으로 생성하거나 지정된 시간이 경과한 이후에 Statspack 스냅샷을 만드는 작업을 생성할 수 있습니다. 예를 들어, 다음 작업은 Statspack 스냅샷을 1시간마다 생성합니다.

```
variable jn number;
execute dbms_job.submit(:jn, 'statspack.snap;', sysdate, 'trunc(SYSDATE+1/24, ''HH24''));
commit;
```

5. 최소 두 개의 Statspack 스냅샷을 생성한 후 다음 쿼리를 사용하여 스냅샷을 확인할 수 있습니다.

```
select snap_id, snap_time from stats$snapshot order by 1;
```

6. Statspack 보고서를 생성하려면 분석할 두 스냅샷을 선택하고 다음과 같이 Amazon RDS 명령을 실행합니다.

```
exec RDSADMIN.RDS_RUN_SPREPORT(<begin snap>,<end snap>);
```

예를 들어, 다음 Amazon RDS 명령은 Statspack 스냅샷 1과 2 사이의 간격을 근거로 보고서를 생성합니다.

```
exec RDSADMIN.RDS_RUN_SPREPORT(1,2);
```

생성되는 Statspack 보고서의 파일 이름에는 사용된 두 Statspack 스냅샷의 번호가 포함되어 있습니다. 예를 들어, Statspack 스냅샷 1과 2를 사용하여 생성된 보고서 파일의 이름은 ORCL_sreport_1_2.lst입니다. RDS 콘솔에 있는 DB 인스턴스 세부 정보의 [Log] 섹션에서 보고서를 선택하고 [Download]를 클릭하여 Statspack 보고서를 다운로드하거나, [Oracle 추적 파일을 사용한 작업 \(p. 327\)](#)에 설명된 추적 파일 절차를 사용할 수 있습니다.

Logs (342)		
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_sreport_1_2.lst	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_2018011817315366624143795.aud	Thu Jan 18 09:51:37 GMT-800 2018	3.5 kB

보고서를 생성할 때 오류가 발생하는 경우 오류 파일은 동일한 명명 규칙에 따라 생성되지만 확장자가 .err입니다. 예를 들어, Statspack 스냅샷 1과 7을 사용하여 보고서를 생성하는 동안 오류가 발생한 경우 보고서 파일의 이름은 ORCL_sreport_1_7.err입니다. RDS 콘솔의 [Log] 섹션에서 보고서를 선택하고 [Download]를 클릭하여 오류 보고서를 다운로드하거나 [Oracle 추적 파일을 사용한 작업 \(p. 327\)](#)에 설명된 추적 파일 절차를 사용할 수 있습니다.

Oracle Statspack은 보고서를 실행하기 전에 몇 가지 기본 검사를 수행하므로 명령 프롬프트에 오류 메시지가 표시될 수도 있습니다. 예를 들어, 잘못된 범위를 기준으로 보고서를 생성하려고 하면(예: 시작 Statspack 스냅샷 값이 종료 Statspack 스냅샷 값보다 큰 경우) 명령 프롬프트에 오류 메시지가 표시되고 오류 파일은 생성되지 않습니다.

```
exec RDSADMIN.RDS_RUN_SPREPORT(2,1);
*
ERROR at line 1:
ORA-20000: Invalid snapshot IDs. Find valid ones in perfstat.stats$snapshot.
```

Statspack 스냅샷 중 하나에 대해 잘못된 번호를 사용하는 경우에도 명령 프롬프트에 오류 메시지가 나타납니다. 예를 들어, 20개의 Statspack 스냅샷이 있지만 Statspack 스냅샷 1과 50을 사용하여 보고서를 실행하도록 요청하면 명령 프롬프트에 오류가 표시됩니다.

```
exec RDSADMIN.RDS_RUN_SPREPORT(1,50);
*
ERROR at line 1:
```

ORA-20000: Could not find both snapshot IDs

스냅샷 수준을 조정하여 캡처되는 데이터의 양을 조정하는 방법을 비롯하여 Oracle Statspack을 사용하는 방법에 대한 자세한 내용을 보려면 Oracle [Statspack 설명서 페이지](#)로 이동하십시오.

Oracle Statspack 파일을 제거하려면 다음 명령을 사용합니다.

```
execute statspack.purge(<begin snap>, <end snap>);
```

Oracle 시간대

시간대 옵션을 사용하면 Oracle DB 인스턴스에서 사용하는 시스템 시간대를 변경할 수 있습니다. 예를 들면 온프레미스 환경 또는 기존 애플리케이션과 시간을 호환하기 위해 DB 인스턴스의 시간대를 변경할 수 있습니다. 시간대 옵션은 호스트 레벨에서 시간대를 변경합니다. 시간대를 변경하면 SYSDATE 및 SYSTIMESTAMP를 비롯한 모든 날짜 열과 값이 영향을 받습니다.

시간대 옵션은 `rdsadmin_util.alter_db_time_zone` 명령과 같습니다. `alter_db_time_zone` 명령은 특정 데이터 유형의 시간대만 변경합니다. 시간대 옵션은 모든 날짜 열과 값의 시간대를 변경합니다. `alter_db_time_zone`에 대한 자세한 내용은 [데이터베이스 시간대 설정 \(p. 848\)](#) 섹션을 참조하십시오.

Note

이 옵션을 사용하는 DB 스냅샷은 공유할 수 없습니다. DB 스냅샷 공유에 대한 자세한 내용은 [DB 스냅샷 공유 \(p. 215\)](#) 단원을 참조하십시오.

시간대의 필수 선행 조건

시간대 옵션은 영구적이고 지속적인 옵션입니다. 추가한 후에는 옵션 그룹에서 이 옵션을 제거할 수 없습니다. 추가한 후에는 DB 인스턴스에서 옵션 그룹을 제거할 수 없습니다. 옵션의 시간대 설정을 다른 시간대로 수정할 수 없습니다.

시간대 옵션을 DB 인스턴스에 추가하기 전에 DB 인스턴스의 DB 스냅샷을 만들어 두는 것이 좋습니다. 시간대 옵션이 잘못 설정된 경우 스냅샷을 이용해 DB 인스턴스를 복구할 수 있습니다. 자세한 정보는 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.

프로덕션 DB 인스턴스에 추가하기 전에 테스트 DB 인스턴스에서 시간대 옵션 설정을 테스트하는 것이 좋습니다. 시간대 옵션을 추가하면 시스템 날짜를 이용해 날짜나 시간을 추가하는 테이블에 문제가 발생할 수 있습니다. 데이터와 애플리케이션을 분석해 시간대 변경의 영향을 파악해야 합니다.

시간대 옵션 설정

Amazon RDS는 시간대 옵션에 대해 다음 설정을 지원합니다.

옵션 설정	유효한 값	설명
TIME_ZONE	사용 가능한 시간대 중 하나입니다. 전체 목록은 사용 가능한 표준 시간대 (p. 826) 단원을 참조하십시오.	DB 인스턴스에 대한 새 시간대를 선택합니다.

시간대 옵션 추가

시간대 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동 시킵니다.

시간대 옵션을 추가하는 경우 DB 인스턴스를 자동으로 다시 시작하는 동안 인스턴스가 잠시 중단됩니다.

AWS Management 콘솔

DB 인스턴스에 시간대 옵션을 추가하려면

1. 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않으면 다음 설정을 사용하여 사용자 지정 DB 옵션을 생성합니다.
 - a. [Engine]에서 DB 인스턴스의 Oracle 에디션을 선택합니다.
 - b. 메이저 엔진 버전에서 DB 인스턴스의 버전을 선택합니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.

2. [Timezone] 옵션을 옵션 그룹에 추가하고 옵션 설정을 구성합니다.

Important

하나 이상의 DB 인스턴스에 이미 연결되어 있는 기존 옵션 그룹에 시간대 옵션을 추가하면 모든 DB 인스턴스가 자동으로 다시 시작되는 동안 인스턴스가 잠시 중단됩니다.

옵션 추가에 대한 자세한 정보는 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#) 단원을 참조하십시오. 각 설정에 대한 자세한 정보는 [시간대 옵션 설정 \(p. 824\)](#) 단원을 참조하십시오.

3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:

- 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
- 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 기존 DB 인스턴스에 시간대 옵션을 추가하는 경우 DB 인스턴스를 자동으로 다시 시작하는 동안 인스턴스가 잠시 중단됩니다. 자세한 정보는 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

CLI

다음 예에서는 AWS CLI `add-option-to-option-group` 명령을 사용하여 Timezone 옵션 및 `TIME_ZONE` 옵션 설정을 `myoptiongroup`이라는 옵션 그룹에 추가합니다. 표준 시간대는 Africa/Cairo로 설정되어 있습니다.

Linux, OS X, Unix의 경우:

```
aws rds add-option-to-option-group \
--option-group-name "myoptiongroup" \
--options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/Cairo}]" \
--apply-immediately
```

Windows의 경우:

```
aws rds add-option-to-option-group ^
--option-group-name "myoptiongroup" ^
--options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/Cairo}]" ^
--apply-immediately
```

시간대 설정 수정

시간대 옵션은 영구적이고 지속적인 옵션입니다. 추가한 후에는 옵션 그룹에서 이 옵션을 제거할 수 없습니다. 추가한 후에는 DB 인스턴스에서 옵션 그룹을 제거할 수 없습니다. 옵션의 시간대 설정을 다른 시간대로 수정할 수 없습니다. 시간대가 잘못 설정된 경우, 시간대 옵션을 추가하기 전에 DB 인스턴스의 스냅샷을 복구합니다.

시간대 옵션 제거

시간대 옵션은 영구적이고 지속적인 옵션입니다. 추가한 후에는 옵션 그룹에서 이 옵션을 제거할 수 없습니다. 추가한 후에는 DB 인스턴스에서 옵션 그룹을 제거할 수 없습니다. 시간대 옵션을 제거하려면, 시간대 옵션을 추가하기 전에 DB 인스턴스의 스냅샷을 복구합니다.

사용 가능한 표준 시간대

다음은 시간대 옵션에 사용할 수 있는 값입니다.

영역	시간대
아프리카	Africa/Cairo, Africa/Casablanca, Africa/Harare, Africa/Lagos, Africa/Luanda, Africa/Monrovia, Africa/Nairobi, Africa/Tripoli, Africa/Windhoek
아메리카	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
아시아	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
대서양	Atlantic/Azores, Atlantic/Cape_Verde
호주	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Eucla, Australia/Hobart, Australia/Lord_Howe, Australia/Perth, Australia/Sydney
브라질	Brazil/DeNoronha, Brazil/East
캐나다	Canada/Newfoundland, Canada/Saskatchewan
기타	Etc/GMT-3
유럽	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
태평양	Pacific/Apia, Pacific/Auckland, Pacific/Chatham, Pacific/Fiji, Pacific/Guam, Pacific/Honolulu, Pacific/Kiritimati, Pacific/Marquesas, Pacific/Samoa, Pacific/Tongatapu, Pacific/Wake
US	US/Alaska, US/Central, US/East-Indiana, US/Eastern, US/Pacific

영역	시간대
UTC	UTC

관련 주제

- 옵션 그룹 작업 (p. 148)
- Oracle DB 인스턴스 옵션 (p. 772)

Oracle Transparent Data Encryption

Amazon RDS는 Oracle Enterprise Edition에서 지원되는 Oracle Advanced Security 옵션의 한 가지 기능인 Oracle Transparent Data Encryption(TDE)을 지원합니다. 이 기능은 스토리지에 데이터를 쓰기 전에 자동으로 데이터를 암호화한 뒤에 데이터를 스토리지에서 읽을 때 다시 자동으로 해독합니다.

Oracle Transparent Data Encryption은 민감한 데이터를 암호화해야 하는 시나리오에 사용됩니다. 데이터 파일과 백업 파일이 제3자의 수중에 들어가거나 보안 관련 규정 준수 문제를 해결해야 하는 경우를 예로 들 수 있습니다.

TDE 옵션은 옵션 그룹에서 제거할 수 없는 영구 옵션입니다. 인스턴스가 Oracle TDE 옵션을 포함하는 옵션 그룹에 연결되어 있는 경우 DB 인스턴스에서 TDE를 비활성화할 수 없습니다. TDE 옵션을 사용하는 DB 인스턴스의 옵션 그룹을 변경할 수 있지만 DB 인스턴스와 연결된 옵션 그룹에는 TDE 옵션이 포함되어야 합니다. 다른 옵션을 추가하거나 제거하여 TDE 옵션이 포함된 옵션 그룹을 수정할 수도 있습니다.

Oracle Transparent Data Encryption에 대한 자세한 정보는 본 문서의 범위에서 벗어납니다. Oracle Transparent Data Encryption 사용에 대한 자세한 정보는 [Securing Stored Data Using Transparent Data Encryption](#) 단원을 참조하십시오. Oracle Advanced Security에 대한 자세한 정보는 Oracle 설명서의 [Oracle Advanced Security](#) 단원을 참조하십시오. AWS 보안에 대한 자세한 정보는 [AWS 보안 센터](#)를 참조하십시오.

Note

이 옵션을 사용하는 DB 스냅샷은 공유할 수 없습니다. DB 스냅샷 공유에 대한 자세한 내용은 [DB 스냅샷 공유 \(p. 215\)](#) 단원을 참조하십시오.

TDE 암호화 모델

Oracle Transparent Data Encryption은 TDE 테이블스페이스 암호화 및 TDE 열 암호화의 두 가지 모드를 지원합니다. TDE 테이블스페이스 암호화는 전체 애플리케이션 테이블을 암호화하는 데 사용됩니다. TDE 열 암호화는 중요 데이터를 포함하는 개별 데이터 요소를 암호화하는 데 사용됩니다. TDE 테이블스페이스 암호화와 열 암호화를 모두 사용하는 하이브리드 암호화 솔루션을 적용할 수도 있습니다.

Note

Amazon RDS가 DB 인스턴스의 Oracle Wallet 및 TDE 마스터 키를 관리합니다. [ALTER SYSTEM set encryption key] 명령을 사용하여 암호화 키를 설정하지 않아도 됩니다.

TDE 모범 사례에 대한 자세한 정보는 [Oracle Advanced Security Transparent Data Encryption Best Practices](#)를 참조하십시오.

옵션을 활성화한 경우 다음 명령을 사용하여 Oracle Wallet의 상태를 확인할 수 있습니다.

```
SELECT * FROM v$encryption_wallet;
```

암호화된 테이블스페이스를 생성하려면 다음 명령을 사용합니다.

```
CREATE TABLESPACE encrypt_ts ENCRYPTION DEFAULT STORAGE (ENCRYPT);
```

암호화 알고리즘을 지정하려면 다음 명령을 사용하십시오.

```
CREATE TABLESPACE encrypt_ts ENCRYPTION USING 'AES256' DEFAULT STORAGE (ENCRYPT);
```

테이블스페이스를 암호화하는 이전 명령은 Amazon RDS에서 Oracle을 설치하지 않은 경우에 사용하는 명령과 동일하며, 열을 암호화하는 ALTER TABLE 구문 역시 Amazon RDS에서 Oracle을 설치하지 않은 경우에 사용하는 명령과 동일합니다.

DB 인스턴스가 TDE 옵션이 있는 옵션 그룹과 연결되어 있는지 먼저 확인해야 합니다. DB 인스턴스와 연동되어 있는 옵션 그룹은 RDS 콘솔, [describe-db-instance](#) AWS CLI 명령, 또는 API 작업 [DescribeDBInstances](#)를 사용하여 확인할 수 있습니다.

또한 몇 가지 보안 기준을 준수하기 위해 Amazon RDS는 주기적으로 마스터 키를 자동으로 교체하고 있습니다.

TDE 옵션 추가

Amazon RDS에서 Oracle Transparent Data Encryption(TDE)을 사용하는 프로세스는 다음과 같습니다.

1. DB 인스턴스가 TDE 옵션이 추가된 옵션 그룹과 연동되어 있지 않으면 먼저 옵션 그룹을 생성한 후 TDE 옵션을 추가하거나 연동되어 있는 옵션 그룹의 설정을 변경하여 TDE 옵션을 추가합니다. 옵션 그룹의 생성 및 변경에 대한 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오. 옵션 그룹에 옵션을 추가하는 방법에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#) 단원을 참조하십시오.
2. DB 인스턴스를 TDE 옵션이 있는 옵션 그룹과 연결합니다. DB 인스턴스와 옵션 그룹의 연동에 대한 자세한 정보는 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

TDE 옵션 제거

더 이상 TDE 옵션을 DB 인스턴스와 함께 사용하지 않으려면 DB 인스턴스에서 모든 데이터를 해독하고 TDE가 활성화된 옵션 그룹과 연결되지 않은 새 DB 인스턴스에 데이터를 복사한 다음 원본 인스턴스를 삭제해야 합니다. 원하는 경우 새 인스턴스의 이름을 이전 DB 인스턴스와 동일하게 변경할 수 있습니다.

Data Pump에서 TDE 사용

Oracle Data Pump를 사용하여 암호화된 덤프 파일을 가져오거나 내보낼 수 있습니다. Amazon RDS는 Oracle Data Pump에 대한 암호 암호화 모드(ENCRYPTION_MODE=PASSWORD)를 지원합니다. Amazon RDS는 Oracle Data Pump에 대한 투명한 암호화 모드(ENCRYPTION_MODE=TRANSPARENT)를 지원하지 않습니다. Amazon RDS에서 Oracle Data Pump를 사용하는 방법에 대한 자세한 정보는 [Oracle Data Pump를 사용한 가져오기 \(p. 759\)](#) 단원을 참조하십시오.

관련 주제

- [옵션 그룹 작업 \(p. 148\)](#)
- [Oracle DB 인스턴스 옵션 \(p. 772\)](#)

Oracle UTL_MAIL

Amazon RDS는 UTL_MAIL 옵션 및 SMTP 서버 사용을 통해 Oracle UTL_MAIL을 지원합니다. UTL_MAIL 패키지를 사용하여 데이터베이스에서 직접 이메일을 전송할 수 있습니다. Amazon RDS는 Oracle의 다음 버전에 대해 UTL_MAIL을 지원합니다.

- Oracle 버전 12.2.0.1, 모든 버전
- Oracle 버전 12.1.0.2.v5 이상
- Oracle 버전 11.2.0.4.v9 이상

다음은 UTL_MAIL을 사용할 때 적용되는 몇 가지 제한 사항입니다.

- UTL_MAIL에서는 TLS(전송 계층 보안)를 지원하지 않으므로 이메일이 암호화되지 않습니다.

사용자 지정 Oracle wallet을 생성하고 업로드하여 원격 SSL/TLS 리소스에 안전하게 연결하려면 [Oracle DB 인스턴스에 utl_http, utl_tcp 및 utl_smtp 사용 \(p. 724\)](#)의 지침을 따르십시오.

wallet에 필요한 특정 인증서는 서비스별로 다릅니다. AWS 서비스의 경우 일반적으로 [Amazon Trust Services 리포지토리](#)에서 이 정보를 확인할 수 있습니다.

- UTL_MAIL은 SMTP 서버를 통한 인증을 지원하지 않습니다.
- 단일 첨부만 이메일로 보낼 수 있습니다.
- 32K를 초과하는 첨부를 보낼 수 없습니다.
- ASCII 및 EBCDIC(Extended Binary Coded Decimal Interchange Code) 문자 인코딩만 사용할 수 있습니다.
- SMTP 포트(25)는 탄력적 네트워크 인터페이스 소유자의 정책에 따라 조절됩니다.

UTL_MAIL을 활성화하는 경우 DB 인스턴스의 마스터 사용자에게만 실행 권한이 부여됩니다. 필요한 경우, 마스터 사용자는 다른 사용자에게 UTL_MAIL을 사용할 수 있는 실행 권한을 부여할 수 있습니다.

Important

UTL_MAIL 절차 사용을 추적하려면 Oracle의 기본 제공 감사 기능을 사용하는 것이 좋습니다.

Oracle UTL_MAIL 필수 선행 조건

다음은 Oracle UTL_MAIL 사용을 위한 필수 선행 조건입니다.

- 하나 이상의 SMTP 서버와 해당 IP 주소 또는 퍼블릭 또는 프라이빗 DNS(Domain Name Server) 이름. 사용자 지정 DNS 서버를 통해 확인되는 프라이빗 DNS 이름에 대한 자세한 내용은 [사용자 지정 DNS 서버 설정 \(p. 844\)](#) 단원을 참조하십시오.
- Oracle 12c 이전 버전의 경우 DB 인스턴스에서 XML DB 옵션을 사용해야 합니다. 자세한 내용은 [Oracle XML DB \(p. 831\)](#) 단원을 참조하십시오.

Oracle UTL_MAIL 옵션 추가

Oracle UTL_MAIL 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동시킵니다.

UTL_MAIL 옵션을 추가하면 옵션 그룹이 활성화되고 UTL_MAIL이 활성화됩니다.

DB 인스턴스에 UTL_MAIL 옵션을 추가하려면

1. 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않으면 다음 설정을 사용하여 사용자 지정 DB 옵션을 생성합니다.
 - a. [Engine]에서 사용할 Oracle 버전을 선택합니다.
 - b. 메이저 엔진 버전에서 DB 인스턴스의 버전을 선택합니다.

자세한 내용은 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.

2. [UTL_MAIL] 옵션을 옵션 그룹에 추가합니다. 옵션 추가에 대한 자세한 내용은 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#) 단원을 참조하십시오.
3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:
 - 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
 - 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

Oracle UTL_MAIL 사용

UTL_MAIL 옵션을 활성화한 후 사용하기 전에 SMTP 서버를 구성해야 합니다.

SMTP_OUT_SERVER 파라미터를 유효한 IP 주소 또는 퍼블릭 DNS 이름으로 설정하여 SMTP 서버를 구성합니다. SMTP_OUT_SERVER 파라미터에 대해 여러 서버 주소를 쉼표로 구분된 목록으로 지정할 수 있습니다. 첫 번째 서버를 사용할 수 없는 경우 UTL_MAIL에서는 다음 서버를 순서대로 시도합니다.

[DB 파라미터 그룹](#)을 사용하여 DB 인스턴스에 대한 기본 SMTP_OUT_SERVER를 설정할 수 있습니다. DB 인스턴스에서 데이터베이스에 대해 다음 코드를 실행하여 세션에 대한 SMTP_OUT_SERVER 파라미터를 설정할 수 있습니다.

```
ALTER SESSION SET smtp_out_server = mailserver.domain.com:25;
```

UTL_MAIL 옵션을 활성화하고 SMTP_OUT_SERVER를 구성한 후 SEND 절차를 사용하여 메일을 보낼 수 있습니다. 자세한 내용은 Oracle 설명서의 [UTL_MAIL](#)을 참조하십시오.

Oracle UTL_MAIL 옵션 제거

DB 인스턴스에서 Oracle UTL_MAIL을 제거할 수 있습니다.

DB 인스턴스에서 UTL_MAIL을 제거하려면 다음 중 하나를 수행합니다.

- 여러 DB 인스턴스에서 UTL_MAIL을 제거하려면 UTL_MAIL이 속한 옵션 그룹에서 해당 UTL_MAIL 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 내용은 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- 단일 DB 인스턴스에서 UTL_MAIL을 제거하려면 DB 인스턴스를 수정하고 UTL_MAIL 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

문제 해결

다음은 Amazon RDS에서 UTL_MAIL을 사용할 때 발생할 수 있는 문제입니다.

- Throttling. SMTP 포트(25)는 탄력적 네트워크 인터페이스 소유자의 정책에 따라 조절됩니다. UTL_MAIL을 사용하여 이메일을 발송할 수 있지만 오류 ORA-29278: SMTP transient error: 421 Service not available이 표시될 경우 포트가 조절되는 것일 수 있습니다. 이메일 발송 시 조절을 경험할 경우 백오프 알고리즘을 구현하는 것이 좋습니다. 백오프 알고리즘에 대한 자세한 내용은 [AWS의 오류 재시도 횟수 및 지수 백오프 섹션](#) 및 [How to handle a "Throttling – Maximum sending rate exceeded" error](#) 블로그를 참조하십시오.

이러한 조절은 제거를 요청할 수 있습니다. 자세한 내용은 [EC2 인스턴스에서 포트 25의 제한을 제거할 수 있습니까?](#) 단원을 참조하십시오.

관련 주제

- [옵션 그룹 작업 \(p. 148\)](#)
- [Oracle DB 인스턴스 옵션 \(p. 772\)](#)

Oracle XML DB

Oracle XML DB는 DB 인스턴스에 기본 XML 지원을 추가합니다. XML DB를 사용하면 관계형 데이터베이스 뿐 아니라 정형 또는 비정형 XML도 저장 및 검색할 수 있습니다.

XML DB는 Oracle 버전 12c 이상에 사전 설치되어 있습니다. Amazon RDS는 XMLDB 옵션을 사용하여 버전 11g용 Oracle XML DB를 지원합니다. DB 인스턴스에 XMLDB 옵션을 적용하면 Oracle XML DB 리포지토리에 대한 전체 액세스 권한이 부여되고, 사후 설치 작업이 필요하지 않습니다.

Note

Amazon RDS XMLDB 옵션은 Oracle XML DB Protocol Server를 지원하지 않습니다.

Oracle XML DB 옵션 추가

Oracle XML DB 옵션을 DB 인스턴스에 추가하는 일반적인 프로세스는 다음과 같습니다.

1. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 복사 또는 수정합니다.
2. 옵션을 옵션 그룹에 추가합니다.
3. 옵션 그룹을 DB 인스턴스에 연동시킵니다.

XML DB 옵션을 추가하면 옵션 그룹이 활성화되는 순간, XML DB가 활성화됩니다.

DB 인스턴스에 XML DB 옵션을 추가하려면

1. 사용할 옵션 그룹을 결정합니다. 새 옵션 그룹을 생성하거나 기존 옵션 그룹을 사용합니다. 기존 옵션 그룹을 사용하려면 다음 단계로 건너뛰십시오. 그렇지 않으면 다음 설정을 사용하여 사용자 지정 DB 옵션을 생성합니다.
 - a. [Engine]에서 사용할 Oracle 버전을 선택합니다.
 - b. [Major engine version]에서 [11.2]를 선택합니다.

자세한 정보는 [옵션 그룹 생성 \(p. 150\)](#) 단원을 참조하십시오.

2. [XMLDB] 옵션을 옵션 그룹에 추가합니다. 옵션 추가에 대한 자세한 정보는 [옵션을 옵션 그룹에 추가하기 \(p. 152\)](#) 단원을 참조하십시오.
3. 옵션 그룹을 새 DB 인스턴스 또는 기존 DB 인스턴스에 적용합니다:
 - 새 DB 인스턴스의 경우, 인스턴스를 시작할 때 옵션 그룹을 적용합니다. 자세한 내용은 [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 \(p. 727\)](#) 단원을 참조하십시오.
 - 기존 DB 인스턴스의 경우, 해당 인스턴스를 수정하고 새 옵션 그룹을 연결하여 옵션 그룹을 적용합니다. 자세한 정보는 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

Oracle XML DB 옵션 제거

버전 11g를 실행하는 DB 인스턴스에서 XML DB 옵션을 제거할 수 있습니다.

버전 11g를 실행하는 DB 인스턴스에서 XML DB 옵션을 제거하려면 다음 중 하나를 수행합니다.

- 여러 DB 인스턴스에서 XMLDB 옵션을 제거하려면 DB 인스턴스가 속한 옵션 그룹에서 해당 XMLDB 옵션을 제거합니다. 이 변경은 해당 옵션 그룹을 사용하는 모든 DB 인스턴스에 영향을 미칩니다. 자세한 정보는 [옵션을 옵션 그룹에서 제거하기 \(p. 158\)](#) 단원을 참조하십시오.
- 단일 DB 인스턴스에서 XMLDB 옵션을 제거하려면 DB 인스턴스를 수정하고 XMLDB 옵션이 포함되지 않은 다른 옵션 그룹을 지정합니다. 기본(빈) 옵션 그룹을 지정하거나 다른 사용자 지정 옵션 그룹을 지정할 수 있습니다. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

Oracle DB 인스턴스에 대한 일반 DBA 작업

다음은 Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에 대한 몇 가지 공통 DBA 작업을 Amazon RDS에 따라 구현하는 방법에 관해 설명한 것입니다. 관리형 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 shell 액세스를 제공하지 않으며, 고급 권한이 필요한 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다.

다음은 Oracle을 실행 중인 DB 인스턴스에 대한 공통 DBA 작업입니다.

- [시스템 작업 \(p. 836\)](#)

세션 분리 (p. 836)	Amazon RDS 메서드: <code>disconnect</code> Oracle 메서드: <code>alter system disconnect session</code>
세션 종료 (p. 837)	Amazon RDS 메서드: <code>kill</code> Oracle 메서드: <code>alter system kill session</code>
제한 세션 활성화 및 비활성화 (p. 837)	Amazon RDS 메서드: <code>restricted_session</code> Oracle 메서드: <code>alter system enable restricted session</code>
공유 풀 플러시 (p. 838)	Amazon RDS 메서드: <code>flush_shared_pool</code> Oracle 메서드: <code>alter system flush shared_pool</code>
버퍼 캐시 플러시 (p. 838)	Amazon RDS 메서드: <code>flush_buffer_cache</code> Oracle 메서드: <code>alter system flush buffer_cache</code>
SYS 객체에 대한 SELECT 또는 EXECUTE 권한 부여 (p. 838)	Amazon RDS 메서드: <code>grant_sys_object</code> Oracle 메서드: <code>grant</code>

SYS 객체에 대한 SELECT 또는 EXECUTE 권한 취소 (p. 840)	Amazon RDS 메서드: <code>revoke_sys_object</code> Oracle 메서드: <code>REVOKE</code>
마스터가 아닌 사용자에게 권한 부여 (p. 840)	Amazon RDS 메서드: <code>grant</code> Oracle 메서드: <code>GRANT</code>
DBMS_SCHEDULER 작업 변경 (p. 841)	Amazon RDS 메서드: <code>dbms_scheduler.set_attribute</code> Oracle 메서드: <code>DBMS_SCHEDULER.SET_ATTRIBUTE</code>
사용자 지정 암호 확인 함수 생성 (p. 841)	Amazon RDS 메서드: <code>create_verify_function</code> Amazon RDS 메서드: <code>create_passthrough_verify_fcn</code>
사용자 지정 DNS 서버 설정 (p. 844)	—

- 데이터베이스 작업 (p. 845)

데이터베이스의 전역 이름 변경 (p. 846)	Amazon RDS 메서드: <code>rename_global_name</code> Oracle 메서드: <code>ALTER DATABASE RENAME</code>
테이블스페이스 생성과 크기 조정 (p. 846)	Amazon RDS 메서드: <code>CREATE TABLESPACE</code> Oracle 메서드: <code>ALTER DATABASE</code>
기본 테이블스페이스 설정 (p. 847)	Amazon RDS 메서드: <code>ALTER_DEFAULT_TABLESPACE</code> Oracle 메서드: <code>ALTER DATABASE DEFAULT TABLESPACE</code>
기본 임시 테이블스페이스 설정 (p. 847)	Amazon RDS 메서드: <code>ALTER_DEFAULT_TEMP_TABLESPACE</code> Oracle 메서드: <code>ALTER DATABASE DEFAULT TEMPORARY TABLESPACE</code>
데이터베이스 체크포인트 (p. 847)	Amazon RDS 메서드: <code>checkpoint</code> Oracle 메서드: <code>ALTER SYSTEM CHECKPOINT</code>
분산 복구 설정 (p. 847)	Amazon RDS 메서드: <code>enable_distr_recovery</code> Oracle 메서드: <code>ALTER SYSTEM ENABLE DISTRIBUTED RECOVERY</code>
데이터베이스 시간대 설정 (p. 848)	Amazon RDS 메서드: <code>alter_db_time_zone</code> Oracle 메서드: <code>ALTER DATABASE SET TIME_ZONE</code>
Oracle 외부 테이블 작업 (p. 848)	—

Automatic Workload Repository(AWR) 작업 (p. 849)	—
VPC의 DB 인스턴스에 사용하기 위한 데이터베이스 링크 조정 (p. 850)	—
DB 인스턴스의 기본 에디션 설정 (p. 850)	Amazon RDS 메서드: <code>alter_default_edition</code> Oracle 메서드: <code>alter database default edition</code>
SYS.AUD\$ 테이블에 대한 감사 활성화 (p. 850)	Amazon RDS 메서드: <code>audit_all_sys_aud_table</code> Oracle 메서드: <code>audit</code>
SYS.AUD\$ 테이블에 대한 감사 비활성화 (p. 851)	Amazon RDS 메서드: <code>noaudit_all_sys_aud_table</code> Oracle 메서드: <code>noaudit</code>

- [RMAN 작업 \(p. 858\)](#)

DB 인스턴스 파일 확인 (p. 862)	Amazon RDS 메서드: <code>rdsadmin_rman_util.procedure</code> Oracle 메서드: <code>RMAN VALIDATE</code>
블록 변경 추적 활성화 및 비활성화 (p. 865)	Amazon RDS 메서드: <code>rdsadmin_rman_util.procedure</code> Oracle 메서드: <code>ALTER DATABASE</code>
보관된 재실행 로그 대조 확인 (p. 865)	Amazon RDS 메서드: <code>rdsadmin_rman_util.crosscheck_archi</code> Oracle 메서드: <code>RMAN BACKUP</code>
보관된 재실행 로그 백업 (p. 866)	Amazon RDS 메서드: <code>rdsadmin_rman_util.procedure</code> Oracle 메서드: <code>RMAN BACKUP</code>
전체 데이터베이스 백업 수행 (p. 871)	Amazon RDS 메서드: <code>rdsadmin_rman_util.backup_database</code> Oracle 메서드: <code>RMAN BACKUP</code>
증분 데이터베이스 백업 수행 (p. 871)	Amazon RDS 메서드: <code>rdsadmin_rman_util.backup_database</code> Oracle 메서드: <code>RMAN BACKUP</code>
테이블스페이스 백업 수행 (p. 872)	Amazon RDS 메서드: <code>rdsadmin_rman_util.backup_database</code> Oracle 메서드: <code>RMAN BACKUP</code>

- [로그 작업 \(p. 852\)](#)

강제 로깅 설정 (p. 852)	Amazon RDS 메서드: <code>force_logging</code> Oracle 메서드: <code>alter database force logging</code>
보충 로깅 설정 (p. 852)	Amazon RDS 메서드: <code>alter_supplemental_logging</code> Oracle 메서드: <code>alter database add supplemental log</code>
온라인 로그 파일 전환 (p. 853)	Amazon RDS 메서드: <code>switch_logfile</code> Oracle 메서드: <code>alter system switch logfile</code>
온라인 다시 실행 로그 추가 (p. 853)	Amazon RDS 메서드: <code>add_logfile</code>
온라인 다시 실행 로그 드롭 (p. 854)	Amazon RDS 메서드: <code>drop_logfile</code>
온라인 다시 실행 로그 크기 조절 (p. 854)	—
보관된 다시 실행 로그 보존 (p. 856)	Amazon RDS 메서드: <code>set_configuration</code>
트랜잭션 로그 액세스 (p. 857)	Amazon RDS 메서드: <code>create_archivelog_dir</code> Amazon RDS 메서드: <code>create_onlinelog_dir</code>

- [기타 작업 \(p. 873\)](#)

메인 데이터 스토리지 공간에 새로운 디렉터리 생성 (p. 874)	Amazon RDS 메서드: <code>create_directory</code> Oracle 메서드: <code>create directory</code>
DB 인스턴스 디렉터리의 파일 목록 표시 (p. 874)	Amazon RDS 메서드: <code>listdir</code> Oracle 메서드: —
DB 인스턴스 디렉터리의 파일 목록 읽기 (p. 875)	Amazon RDS 메서드: <code>read_text_file</code> Oracle 메서드: —

Oracle DB 인스턴스에 대한 공통 DBA 시스템 작업

이 단원에서는 Oracle을 실행하는 Amazon RDS DB 인스턴스에서 시스템과 관련된 공통 DBA 작업을 수행하는 방법에 대해 살펴보겠습니다. 관리형 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 shell 액세스를 제공하지 않으며, 고급 권한이 필요한 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다.

주제

- [세션 분리 \(p. 836\)](#)
- [세션 종료 \(p. 837\)](#)
- [제한 세션 활성화 및 비활성화 \(p. 837\)](#)
- [공유 풀 플러시 \(p. 838\)](#)
- [버퍼 캐시 플러시 \(p. 838\)](#)
- [SYS 객체에 대한 SELECT 또는 EXECUTE 권한 부여 \(p. 838\)](#)
- [SYS 객체에 대한 SELECT 또는 EXECUTE 권한 취소 \(p. 840\)](#)
- [마스터가 아닌 사용자에게 권한 부여 \(p. 840\)](#)
- [DBMS_SCHEDULER 작업 변경 \(p. 841\)](#)
- [사용자 지정 암호 확인 함수 생성 \(p. 841\)](#)
- [사용자 지정 DNS 서버 설정 \(p. 844\)](#)

세션 분리

전용 서버 프로세스를 종료하여 현재 세션을 분리하는 Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.disconnect`를 사용할 수 있습니다. `disconnect` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	필수 여부	설명
<code>sid</code>	숫자	—	Required	세션 식별자입니다.
<code>serial</code>	숫자	—	Required	세션 일련번호입니다.
<code>method</code>	varchar	'IMMEDIATE'	선택	유효한 값은 ' <code>IMMEDIATE</code> ' 또는 ' <code>POST_TRANSACTION</code> '입니다.

다음 예제에서는 세션을 분리합니다.

```
begin
    rdsadmin.rdsadmin_util.disconnect(
        sid    => sid,
        serial => serial_number);
end;
/
```

세션 식별자와 세션 일련번호를 얻으려면, `V$SESSION` 뷰를 쿼리하십시오. 다음 예제에서는 사용자 `AWSUSER`의 모든 세션을 확보합니다.

```
select SID, SERIAL#, STATUS from V$SESSION where USERNAME = 'AWSUSER';
```

이 메서드를 사용하려면 데이터베이스가 열려 있어야 합니다. 세션 분리에 대한 자세한 내용은 Oracle 문서에서 [ALTER SYSTEM](#) 단원을 참조하십시오.

세션 종료

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.kill`을 사용하여 세션을 종료할 수 있습니다. `kill` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
<code>sid</code>	숫자	—	Required	세션 식별자입니다.
<code>serial</code>	숫자	—	Required	세션 일련번호입니다.
<code>method</code>	varchar	null	선택	유효한 값은 ' <code>IMMEDIATE</code> ' 또는 ' <code>PROCESS</code> '입니다.

다음 예제는 세션을 종료합니다.

```
begin
    rdsadmin.rdsadmin_util.kill(
        sid    => sid,
        serial => serial_number);
end;
/
```

세션 식별자와 세션 일련번호를 얻으려면, `V$SESSION` 뷰를 쿼리하십시오. 다음 예제에서는 사용자 `AWSUSER`의 모든 세션을 확보합니다.

```
select SID, SERIAL#, STATUS from V$SESSION where USERNAME = 'AWSUSER';
```

`IMMEDIATE` 또는 `PROCESS`를 `method` 파라미터 값으로 지정할 수 있습니다. `PROCESS`를 `method` 값으로 지정하면 세션과 연결된 프로세스를 종료할 수 있습니다. 이 방법은 `IMMEDIATE`를 `method` 값으로 사용하여 세션을 종료하지 못했을 경우에만 사용해야 합니다.

제한 세션 활성화 및 비활성화

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.restricted_session`을 사용하여 제한 세션을 활성화 또는 비활성화할 수 있습니다. `restricted_session` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
<code>p_enable</code>	부울	true	선택	<code>true</code> 는 제한 세션을 활성화하고, <code>false</code> 는 제한 세션을 비활성화하도록 설정됩니다.

다음은 제한 세션을 활성화 및 비활성화하는 방법을 나타낸 예제입니다.

```
/* 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
```

공유 풀 플러시

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.flush_shared_pool`을 사용하여 공유 풀을 플러시할 수 있습니다. `flush_shared_pool` 프로시저에는 파라미터가 없습니다.

다음 예제는 공유 풀을 플러시합니다.

```
exec rdsadmin.rdsadmin_util.flush_shared_pool;
```

버퍼 캐시 플러시

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.flush_buffer_cache`를 사용하여 버퍼 캐시를 플러시할 수 있습니다. `flush_buffer_cache` 프로시저에는 파라미터가 없습니다.

다음 예제는 버퍼 캐시를 플러시합니다.

```
exec rdsadmin.rdsadmin_util.flush_buffer_cache;
```

SYS 객체에 대한 SELECT 또는 EXECUTE 권한 부여

일반적으로 여러 객체가 포함되어 있을 수 있는 역할을 사용하여 권한을 이전합니다. Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.grant_sys_object`를 사용하여 단일 객체에 권한을 부여할 수 있습니다. 이 프로시저는 마스터 계정이 이미 가지고 있는 권한만 역할을 통해, 혹은 직접 부여합니다.

`grant_sys_object` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
p_obj_name	varchar2	—	Required	권한을 부여할 객체의 이름입니다. 디렉터리, 기

파라미터 이름	데이터 형식	기본값	Required	설명
				능, 패키지, 프로시저, 시퀀스, 테이블, 뷰가 객체가 될 수 있습니다. 객체 이름은 DBA_OBJECTS에 표시된 대로 정확하게 입력해야 합니다. 대부분의 시스템 객체는 대문자로 정의되므로 먼저 대문자로 시도하는 것이 좋습니다.
p_grantee	varchar2	—	Required	권한을 부여할 객체의 이름입니다. 스키마나 역할이 객체가 될 수 있습니다.
p_privilege	varchar2	null	Required	—
p_grant_option	boolean	false	선택	true는 부여 옵션과 함께 사용하도록 설정됩니다. p_grant_option 파라미터는 Oracle 버전 11.2.0.4.v8 이상 및 12.1.0.2.v4 이상에서 지원됩니다.

다음 예에서는 v_\$SESSION이라는 객체에 대한 선택 권한을 USER1이라는 사용자에게 부여합니다.

```
begin
    rdsadmin.rdsadmin_util.grant_sys_object(
        p_obj_name    => 'V_$SESSION',
        p_grantee     => 'USER1',
        p_privilege   => 'SELECT');
end;
/
```

다음 예에서는 v_\$SESSION이라는 객체에 대한 선택 권한을 USER1이라는 사용자에게 부여 옵션과 함께 부여합니다.

```
begin
    rdsadmin.rdsadmin_util.grant_sys_object(
        p_obj_name    => 'V_$SESSION',
        p_grantee     => 'USER1',
        p_privilege   => 'SELECT',
        p_grant_option => true);
end;
/
```

객체에 권한을 부여할 수 있으려면 해당 권한을 부여 옵션을 통해 계정에 직접 부여하거나 with admin option을 사용해 부여된 역할을 통해 계정에 부여해야 합니다. 대부분 이미 SELECT_CATALOG_ROLE 역할에 부여된 DBA 뷰에 SELECT 권한을 부여하는 경우가 많습니다. 해당 역할이 아직 with admin option을 통해 사용자에게 직접 부여되지 않았다면 권한을 양도할 수 없습니다. DBA 권한이 있다면 역할을 직접 다른 사용자에게 부여할 수 있습니다.

다음은 SELECT_CATALOG_ROLE과 EXECUTE_CATALOG_ROLE을 USER1에게 부여하는 예제입니다. with admin option을 사용했기 때문에, USER1은 이제 SELECT_CATALOG_ROLE에 부여된 SYS 객체 액세스 권한을 부여할 수 있습니다.

```
grant SELECT_CATALOG_ROLE to USER1 with admin option;
```

```
grant EXECUTE_CATALOG_ROLE to USER1 with admin option;
```

PUBLIC에 이미 부여된 객체는 다시 부여할 필요가 없습니다. grant_sys_object 프로시저를 사용하여 액세스 권한을 다시 부여한다면, 프로시저는 성공을 호출합니다.

SYS 객체에 대한 SELECT 또는 EXECUTE 권한 취소

Amazon RDS 프로시저 rdsadmin.rdsadmin_util.revoke_sys_object를 사용하여 단일 객체에 권한을 취소할 수 있습니다. 이 프로시저는 마스터 계정이 이미 가지고 있는 권한만 역할을 통해, 혹은 직접 취소할 수 있습니다.

revoke_sys_object 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
p_obj_name	varchar2	—	Required	권한을 취소할 객체의 이름입니다. 디렉터리, 기능, 패키지, 프로시저, 시퀀스, 테이블, 뷰가 객체가 될 수 있습니다. 객체 이름은 DBA_OBJECTS에 표시된 대로 정확하게 입력해야 합니다. 대부분의 시스템 객체는 대문자로 정의되므로 먼저 대문자로 시도해보는 것이 좋습니다.
p_revoker	varchar2	—	Required	권한을 취소할 객체의 이름입니다. 스키마나 역할이 객체가 될 수 있습니다.
p_privilege	varchar2	null	Required	—

다음 예에서는 v_\$SESSION이라는 객체에 대한 선택 권한을 user1이라는 사용자에게서 취소합니다.

```
begin
    rdsadmin.rdsadmin_util.revoke_sys_object(
        p_obj_name  => 'V_$SESSION',
        p_revoker   => 'USER1',
        p_privilege => 'SELECT');
end;
/
```

마스터가 아닌 사용자에게 권한 부여

SELECT_CATALOG_ROLE 역할을 사용하여 SYS 스키마에 있는 여러 객체에 선택 권한을 부여할 수 있습니다. SELECT_CATALOG_ROLE 역할은 사용자에게 데이터 딕셔너리 뷰에 대한 SELECT 권한을 부여합니다. 다음은 역할 SELECT_CATALOG_ROLE을 사용자 user1에 부여하는 예제입니다.

```
grant SELECT_CATALOG_ROLE to user1;
```

EXECUTE_CATALOG_ROLE 역할을 사용하여 SYS 스키마에 있는 여러 객체에 실행 권한을 부여할 수 있습니다. EXECUTE_CATALOG_ROLE 역할은 사용자에게 데이터 딕셔너리의 패키지와 프로시저에 대한 EXECUTE 권한을 부여합니다. 다음 예에서는 EXECUTE_CATALOG_ROLE이라는 역할을 user1이라는 사용자에게 부여합니다.

```
grant EXECUTE_CATALOG_ROLE to user1;
```

다음 예에서는 SELECT_CATALOG_ROLE 및 EXECUTE_CATALOG_ROLE이라는 역할에서 허용하는 권한을 얻습니다.

```
select *
  from ROLE_TAB_PRIVS
 where ROLE in ('SELECT_CATALOG_ROLE', 'EXECUTE_CATALOG_ROLE')
order by ROLE, TABLE_NAME asc;
```

다음 예에서는 sh.sales라는 데이터베이스에서 마스터 사용자가 아닌 user1이라는 사용자를 생성한 후 CREATE SESSION 권한과 SELECT 권한을 부여합니다.

```
create user user1 identified by password;
grant CREATE SESSION to user1;
grant SELECT on sh.sales TO user1;
```

DBMS_SCHEDULER 작업 변경

Oracle 프로시저 dbms_scheduler.set_attribute를 사용하여 DBMS_SCHEDULER 업무를 수정할 수 있습니다. 자세한 내용은 Oracle 문서의 [DBMS_SCHEDULER와 SET_ATTRIBUTE Procedure](#) 단원을 참조 하십시오.

Amazon RDS DB 인스턴스를 작업할 때는, 스키마 이름 SYS를 객체 이름 앞에 붙이십시오. 다음은 월요일 창 객체에 자원 계획 속성을 설정하는 예제입니다.

```
begin
  dbms_scheduler.set_attribute(
    name      => 'SYS.MONDAY_WINDOW',
    attribute => 'RESOURCE_PLAN',
    value     => 'resource_plan_1');
end;
/
```

사용자 지정 암호 확인 함수 생성

사용자 지정 암호 확인 함수는 두 가지 방법으로 생성할 수 있습니다. 표준 확인 로직을 사용하고 SYS 스키마에 함수를 저장하려면, `create_verify_function` 프로시저를 사용합니다. 사용자 지정 확인 로직을 사용하거나 SYS 스키마에 함수를 저장하지 않으려면, `create_passthrough_verify_fcn` 프로시저를 사용합니다.

`create_verify_function` 프로시저

`create_verify_function` 프로시저는 Oracle 버전 11.2.0.4.v9 이상과 12.1.0.2.v5 이상에서 지원됩니다.

Amazon RDS 프로시저 `rdsadmin.rdsadmin_password_verify.create_verify_function`을 사용하여 사용자 지정 암호 확인 함수를 만들 수 있습니다. `create_verify_function` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
p_verify_function_name	varchar2	—	Required	사용자 정의 함수 이름입니다. 이 함수는 SYS 스키마에서 자동으로 만들어집니다. 이 함수를 사용자 프로필에 할당합니다.

파라미터 이름	데이터 형식	기본값	Required	설명
p_min_length	숫자	8	선택	필요한 최소 문자 수입니다.
p_max_length	숫자	256	선택	허용되는 최대 문자 수입니다.
p_min_letters	숫자	1	선택	필요한 최소 글자 수입니다.
p_min_uppercase	숫자	0	선택	필요한 최소 대문자 수입니다.
p_min_lowercase	숫자	0	선택	필요한 최소 소문자 수입니다.
p_min_digits	숫자	1	선택	필요한 최소 자릿수 숫자입니다.
p_min_special	숫자	0	선택	필요한 최소 특수문자 수입니다.
p_min_different_chars	숫자	3	선택	이전 암호와 새 암호 간에 필요한 서로 다른 문자의 최소 개수입니다.
p_disallow_username	부울	true	선택	true로 설정되어 암호에 사용자 이름을 사용할 수 없습니다.
p_disallow_reverse	부울	true	선택	true로 설정되어 암호에 사용자 이름 반전을 사용할 수 없습니다.
p_disallow_db_name	부울	true	선택	true로 설정되어 암호에 데이터베이스나 서버 이름을 사용할 수 없습니다.
p_disallow_simple_string	부울	true	선택	true로 설정되어 단순 문자열을 암호로 사용할 수 없습니다.
p_disallow_whitespace	boolean	false	선택	true로 설정되어 암호에 공백 문자를 사용할 수 없습니다.
p_disallow_at_sign	boolean	false	선택	true로 설정되어 암호에 @ 문자를 사용할 수 없습니다.

암호 확인 기능을 여러 개 만들 수 있습니다.

사용자 지정 기능의 이름에 대한 제한 사항이 있습니다. 사용자 지정 함수는 기존 시스템 객체와 이름이 같을 수 없고, 30자를 넘어서는 안 됩니다. 또한 PASSWORD, VERIFY, COMPLEXITY, ENFORCE, STRENGTH 중 한 가지 문자열을 포함해야 합니다.

다음 예시에서는 CUSTOM_PASSWORD_FUNCTION이라는 기능을 생성합니다. 이 함수에는 최소한 문자 12개, 대문자 2개, 자릿수 1개, 특수 문자 1개가 필요하며 암호에는 @ 문자를 사용할 수 없습니다.

```
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;  
/
```

확인 함수의 텍스트를 보려면 DBA_SOURCE를 쿼리하십시오. 다음은 CUSTOM_PASSWORD_FUNCTION이라는 이름의 사용자 지정 암호 기능의 텍스트를 얻는 예제입니다.

```
col text format a150  
  
select TEXT  
  from DBA_SOURCE  
 where OWNER = 'SYS' and NAME = 'CUSTOM_PASSWORD_FUNCTION'  
order by LINE;
```

확인 함수를 사용자 프로파일과 연결하려면 alter profile을 사용하십시오. 다음은 확인 함수를 DEFAULT 사용자 프로필과 연결하는 예제입니다.

```
alter profile DEFAULT limit PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

어떤 사용자 프로파일이 어떤 확인 기능과 연결되어 있는지 보려면 DBA_PROFILES를 쿼리하십시오. 다음은 CUSTOM_PASSWORD_FUNCTION이라는 이름의 사용자 지정 확인 기능과 연결된 프로필을 얻는 예제입니다.

```
select *  
  from DBA_PROFILES  
 where RESOURCE = 'PASSWORD' and LIMIT = 'CUSTOM_PASSWORD_FUNCTION';  
  
PROFILE          RESOURCE_NAME          RESOURCE  LIMIT  
-----           -----  
-----  
DEFAULT          PASSWORD_VERIFY_FUNCTION    PASSWORD  
CUSTOM_PASSWORD_FUNCTION
```

다음은 모든 프로필과 프로필에 연결된 암호 확인 기능을 얻는 예제입니다.

```
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
```

create_passthrough_verify_fcn 프로시저

create_passthrough_verify_fcn 프로시저는 Oracle 버전 11.2.0.4.v11 이상과 12.1.0.2.v7 이상에서 지원됩니다.

Amazon RDS 프로시저

rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn을 사용하여 사용자 지

정 암호 확인 함수를 만들 수 있습니다. `create_passthrough_verify_fcn` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
<code>p_verify_function_name</code>	varchar2	—	Required	사용자 정의 확인 함수의 이름입니다. SYS 스키마에서 자동으로 생성되는 래퍼 함수로서 확인 로직을 포함하지 않습니다. 이 함수를 사용자 프로필에 할당합니다.
<code>p_target_owner</code>	varchar2	—	Required	사용자 지정 확인 함수의 스키마 소유자입니다.
<code>p_target_function_name</code>	varchar2	—	Required	확인 로직을 포함하는 기존 사용자 지정 함수의 이름입니다. 사용자 정의 함수는 부울 값을 반환합니다. 사용자의 함수는 암호가 유효할 경우 <code>true</code> 를 반환하고 암호가 유효하지 않을 경우 <code>false</code> 를 반환해야 합니다.

다음은 `PASSWORD_LOGIC_EXTRA_STRONG`이라는 함수 로직을 사용하는 암호 확인 함수를 생성하는 예제입니다.

```
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;
/
```

확인 기능을 사용자 프로파일과 연결하려면 `alter profile`를 사용하십시오. 다음은 확인 함수를 `DEFAULT` 사용자 프로파일과 연결하는 예제입니다.

```
alter profile DEFAULT limit PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

사용자 지정 DNS 서버 설정

Amazon RDS는 Oracle을 실행하는 DB 인스턴스에서 아웃바운드 네트워크 액세스를 지원합니다. 사전 요구 사항을 포함한 아웃바운드 네트워크 액세스에 대한 자세한 내용은 [Oracle DB 인스턴스에 utl_http, utl_tcp 및 utl_smtp 사용 \(p. 724\)](#) 단원을 참조하십시오.

Amazon RDS Oracle에서는 고객이 소유한 사용자 지정 DNS 서버에서의 DNS(Domain Name Service) 확인이 가능합니다. 사용자 지정 DNS 서버를 통해 Amazon RDS DB 인스턴스에서 전체 주소 도메인 이름만을 확인할 수 있습니다.

사용자 지정 DNS 이름 서버를 설정한 후 변경 사항이 DB 인스턴스에 전파되는 데 최대 30분이 걸립니다. 변경 사항이 DB 인스턴스에 전파된 후 DNS 조회를 필요로 하는 모든 아웃바운드 네트워크 트래픽은 포트 53을 통해 DNS 서버를 쿼리합니다.

Oracle Amazon RDS DB 인스턴스의 사용자 지정 DNS 서버를 설정하려면 다음과 같이 하십시오.

- VPC에 연결된 DHCP 옵션 세트에서 DNS 이름 서버의 IP 주소에 대해 `domain-name-servers` 옵션을 설정합니다. 자세한 내용은 [DHCP 옵션 세트](#) 단원을 참조하십시오.

Note

`domain-name-servers` 옵션은 최대 4개의 값을 받아들이지만 Amazon RDS DB 인스턴스는 첫 번째 값만을 사용합니다.

- DNS 서버가 DNS 이름, Amazon EC2 프라이빗 DNS 이름, 고객별 DNS 이름을 비롯한 모든 조회 쿼리를 확인할 수 있는지 확인합니다. 아웃바운드 네트워크 트래픽에 DNS 서버가 처리할 수 없는 DNS 조회가 포함된 경우, DNS 서버에 적절한 업스트림 DNS 공급자가 구성되어 있어야 합니다.
- 512바이트 이하의 UDP(User Datagram Protocol) 응답을 생성하도록 DNS 서버를 구성하십시오.
- 1,024바이트 이하의 TCP(Transmission Control Protocol) 응답을 생성하도록 DNS 서버를 구성하십시오.
- 포트 53을 통한 Amazon RDS DB 인스턴스로부터의 인바운드 트래픽을 허용하도록 DNS 서버를 구성하십시오. DNS 서버가 Amazon VPC에 있는 경우, VPC에는 포트 53에서 UDP 및 TCP 트래픽을 허용하는 인바운드 규칙이 포함된 보안 그룹이 있어야 합니다. DNS 서버가 Amazon VPC에 없는 경우, 포트 53에서 UDP 및 TCP 트래픽을 허용하는 적절한 방화벽 화이트리스트가 있어야 합니다.

자세한 내용은 [VPC의 보안 그룹 및 규칙 추가 및 제거](#) 단원을 참조하십시오.

- 포트 53을 통한 아웃바운드 트래픽을 허용하도록 Amazon RDS DB 인스턴스의 VPC를 구성하십시오. VPC에는 포트 53에서 UDP 및 TCP 트래픽을 허용하는 아웃바운드 규칙이 포함된 보안 그룹이 있어야 합니다.

자세한 내용은 [VPC의 보안 그룹 및 규칙 추가 및 제거](#) 단원을 참조하십시오.

- Amazon RDS DB 인스턴스와 DNS 서버 간 라우팅 경로가 DNS 트래픽을 허용하도록 올바로 구성되어야 합니다.
- Amazon RDS DB 인스턴스와 DNS 서버가 같은 VPC에 있지 않은 경우, 그 사이에 피어링 연결을 구축해야 합니다. 자세한 내용은 [VPC 피어링이란?](#)을 참조하십시오.

Oracle DB 인스턴스에 대한 공통 DBA 데이터베이스 작업

이 단원에서는 Oracle을 실행하는 Amazon RDS DB 인스턴스에서 데이터베이스와 관련된 공통 DBA 작업을 수행하는 방법에 대해 살펴보겠습니다. 관리형 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 shell 액세스를 제공하지 않으며, 고급 권한이 필요한 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다.

주제

- [데이터베이스의 전역 이름 변경](#) (p. 846)
- [테이블스페이스 생성과 크기 조정](#) (p. 846)
- [기본 테이블스페이스 설정](#) (p. 847)
- [기본 임시 테이블스페이스 설정](#) (p. 847)
- [데이터베이스 체크포인트](#) (p. 847)
- [분산 복구 설정](#) (p. 847)
- [데이터베이스 시간대 설정](#) (p. 848)
- [Oracle 외부 테이블 작업](#) (p. 848)
- [Automatic Workload Repository\(AWR\) 작업](#) (p. 849)
- [VPC의 DB 인스턴스에 사용하기 위한 데이터베이스 링크 조정](#) (p. 850)
- [DB 인스턴스의 기본 에디션 설정](#) (p. 850)
- [SYS.AUD\\$ 테이블에 대한 감사 활성화](#) (p. 850)

- SYS.AUD\$ 테이블에 대한 감사 비활성화 (p. 851)

데이터베이스의 전역 이름 변경

Amazon RDS 프로시저 rdsadmin.rdsadmin_util.rename_global_name을 사용하여 데이터베이스의 전역 이름을 변경할 수 있습니다. rename_global_name 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
p_new_global_name	varchar2	—	Required	데이터베이스의 새로운 전역 이름입니다.

이름 변경이 적용될 수 있도록 데이터베이스가 열려 있어야 합니다. 데이터베이스의 전역 이름 변경에 대한 자세한 내용은 Oracle 문서의 [ALTER DATABASE](#)를 참조하십시오.

다음은 데이터베이스의 전역 이름을 new_global_name으로 변경하는 예제입니다.

```
exec rdsadmin.rdsadmin_util.rename_global_name(p_new_global_name => 'new_global_name');
```

테이블스페이스 생성과 크기 조정

Amazon RDS는 데이터 파일, 로그 파일 및 제어 파일에 대해 Oracle Managed Files(OMF)만 지원합니다. 데이터 파일과 로그 파일을 생성할 때는 물리적인 파일 이름을 지정할 수 없습니다.

기본적으로, 테이블스페이스는 자동 확장 가능 상태로 생성되며 최대 크기는 지정되지 않습니다. 이러한 기본 설정 때문에, 테이블스페이스가 할당된 모든 스토리지를 차지할 때까지 확장되기도 합니다. 되도록 영구 및 임시 테이블스페이스에 적절한 최대 크기를 지정하고, 공간 사용량을 자세히 살펴보십시오.

다음은 시작 크기가 1GB이고 최대 크기가 10GB인 users2라는 이름의 테이블스페이스를 생성하는 예제입니다.

```
create tablespace users2 datafile size 1G autoextend on maxsize 10G;
```

다음은 temp01이라는 이름의 임시 테이블스페이스를 생성하는 예제입니다.

```
create temporary tablespace temp01;
```

Oracle용 Amazon RDS로는 스몰파일 테이블스페이스의 크기를 변경할 수 없으므로 스몰파일 테이블스페이스를 사용하지 않는 것이 좋습니다. 그러나 스몰파일 테이블스페이스에 데이터파일을 추가할 수 있습니다.

ALTER TABLESPACE를 사용하여 빅파일 테이블스페이스 크기를 조정할 수 있습니다. 크기는 킬로바이트(K), 메가바이트(M), 기가바이트(G)나 테라바이트(T)로 설정할 수 있습니다.

다음 예에서는 users2라는 빅파일 테이블스페이스의 크기를 200MB로 조정합니다.

```
alter tablespace users2 resize 200M;
```

다음 예에서는 users2라는 스몰파일 테이블스페이스에 데이터파일을 추가합니다.

```
alter tablespace users2 add datafile size 100000M autoextend on next 250m maxsize UNLIMITED;
```

기본 테이블스페이스 설정

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.alter_default_tablespace`를 사용하여 기본 테이블스페이스를 설정할 수 있습니다. `alter_default_tablespace` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
<code>tablespace_name</code>	<code>varchar</code>	—	Required	기본 테이블스페이스의 이름입니다.

다음은 기본 테이블스페이스를 `users2`로 설정하는 예제입니다.

```
exec rdsadmin.rdsadmin_util.alter_default_tablespace(tablespace_name => 'users2');
```

기본 임시 테이블스페이스 설정

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.alter_default_temp_tablespace`를 사용하여 기본 임시 테이블스페이스를 설정할 수 있습니다. `alter_default_temp_tablespace` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
<code>tablespace_name</code>	<code>varchar</code>	—	Required	기본 임시 테이블스페이스의 이름입니다.

다음 예에서는 기본 임시 테이블스페이스를 `temp01`으로 설정합니다.

```
exec rdsadmin.rdsadmin_util.alter_default_temp_tablespace(tablespace_name => 'temp01');
```

데이터베이스 체크포인트

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.checkpoint`를 사용하여 데이터베이스에 체크포인트를 만들 수 있습니다. `checkpoint` 프로시저에는 파라미터가 없습니다.

다음 예에서는 데이터베이스에 체크포인트를 만듭니다.

```
exec rdsadmin.rdsadmin_util.checkpoint;
```

분산 복구 설정

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.enable_distr_recovery`와 `disable_distr_recovery`를 사용하여 분산 복구를 설정할 수 있습니다. 프로시저에는 파라미터가 없습니다.

다음 예에서는 분산 복구를 활성화합니다.

```
exec rdsadmin.rdsadmin_util.enable_distr_recovery;
```

다음 예에서는 분산 복구를 비활성화합니다.

```
exec rdsadmin.rdsadmin_util.disable_distr_recovery;
```

데이터베이스 시간대 설정

Amazon RDS Oracle 데이터베이스의 표준 시간대는 두 가지 방법으로 설정할 수 있습니다.

- **Timezone** 옵션을 사용해도 됩니다.

Timezone 옵션은 호스트의 시간대를 변경하여 SYSDATE를 포함한 모든 데이터 열과 값에 영향을 끼칩니다. Timezone 옵션에 대한 자세한 내용은 [Oracle 시간대 \(p. 824\)](#) 단원을 참조하십시오.

- Amazon RDS 프로시저 rdsadmin.rdsadmin_util.alter_db_time_zone을 사용해도 됩니다.

alter_db_time_zone 절차는 특정 데이터 유형의 시간대만 변경하며 SYSDATE는 변경하지 않습니다. 시간대 설정에 대한 자세한 제한 사항은 [Oracle 문서](#)를 참조하십시오.

alter_db_time_zone 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
p_new_tz	varchar2	—	Required	지정된 리전이나 협정 세계 시(UTC)의 절대 오프셋으로 지정된 새로운 표준 시간대입니다. 사용할 수 있는 오프셋은 -12:00~+14:00입니다.

다음 예에서는 시간대를 UTC+3시간으로 변경합니다.

```
exec rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => '+3:00');
```

다음 예에서는 시간대를 아프리카/알제 리전의 시간대로 변경합니다.

```
exec rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => 'Africa/Algiers');
```

alter_db_time_zone을 사용하여 시간대를 변경한 후에는 DB 인스턴스를 재부팅해야만 변경 사항이 적용됩니다. 자세한 내용은 [DB 클러스터에서 DB 인스턴스 \(p. 123\)](#) 단원을 참조하십시오.

Oracle 외부 테이블 작업

Oracle 외부 테이블이란 데이터가 데이터베이스에 저장되어 있지 않은 테이블을 말합니다. 오히려 데이터베이스가 액세스할 수 있는 외부 파일에 데이터가 저장되어 있습니다. 외부 테이블을 사용하면 데이터베이스에 테이블을 로드하지 않고도 데이터에 액세스할 수 있습니다. 외부 테이블에 대한 자세한 내용은 Oracle 설명서에서 [Managing External Tables](#)를 참조하십시오.

Amazon RDS에서는 외부 테이블 파일을 디렉터리 객체에 저장할 수 있습니다. 디렉터리 객체는 생성할 수도 있지만, DATA_PUMP_DIR 디렉터리처럼 Oracle 데이터베이스에 사전 정의되어 있는 객체를 사용할 수도 있습니다. 디렉터리 객체 생성에 대한 자세한 내용은 [메인 데이터 스토리지 공간에 새로운 디렉터리 생성 \(p. 874\)](#) 단원을 참조하십시오. ALL_DIRECTORIES 뷰에 대한 쿼리를 실행하여 Amazon RDS Oracle DB 인스턴스의 디렉터리 객체 목록을 조회할 수 있습니다.

Note

디렉터리 객체는 인스턴스에서 사용하는 메인 데이터 스토리지 공간(Amazon EBS 볼륨)을 가리킵니다. 데이터 파일, 재실행 로그, 감사, 트레이스 및 기타 파일 등에 사용되는 공간은 할당된 공간에서 차감됩니다.

외부 데이터 파일은 [DBMS_FILE_TRANSFER](#) 패키지 또는 [UTL_FILE](#) 패키지를 사용하여 Oracle 데이터베이스에서 다른 Oracle 데이터베이스로 이동시킬 수 있습니다. 그러면 원본 데이터베이스의 디렉터리에서 대상 데이터베이스의 특정 디렉터리로 외부 데이터 파일이 이동합니다. DBMS_FILE_TRANSFER 사용에 대한 자세한 내용은 [Oracle Data Pump를 사용한 가져오기 \(p. 759\)](#) 단원을 참조하십시오.

외부 데이터 파일을 이동시켰으면 이제 이 파일을 이용해 외부 테이블을 생성할 수 있습니다. 다음 예에서는 USER_DIR1 디렉터리의 emp_xt_file1.txt 파일을 사용하는 외부 테이블을 생성합니다.

```
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;
```

예를 들어 Amazon RDS Oracle DB 인스턴스에 저장된 데이터를 외부 데이터 파일로 이동시킨다고 가정하겠습니다. 이때는 외부 테이블을 생성한 후 데이터베이스 테이블에서 데이터를 선택하여 외부 데이터 파일을 채울 수 있습니다. 다음은 데이터베이스의 orders 테이블에 대해 쿼리를 실행하여 orders_xt 외부 테이블을 생성하는 SQL 문입니다.

```
CREATE TABLE orders_xt
ORGANIZATION EXTERNAL
(
    TYPE ORACLE_DATAPUMP
    DEFAULT DIRECTORY DATA_PUMP_DIR
    LOCATION ('orders_xt.dmp')
)
AS SELECT * FROM orders;
```

위 예제에서는 DATA_PUMP_DIR 디렉터리의 orders_xt.dmp 파일이 데이터로 채워집니다.

Automatic Workload Repository(AWR) 작업

Oracle Database Enterprise Edition을 사용하고 진단 및 튜닝 팩 사용을 허가한 경우 Automatic Workload Repository(AWR)를 사용할 수 있습니다. AWR을 활성화하려면 CONTROL_MANAGEMENT_PACK_ACCESS 파라미터를 변경하십시오.

AWR 보고서는 일반적으로 데이터베이스 호스트 서버에 설치된 awrrpt.sql과 같은 보고서 생성 스크립트를 사용해 생성합니다. 호스트에 직접 액세스할 순 없지만 다른 Oracle Database 설치 파일에서 스크립트의 사본을 얻을 수는 있습니다. 또는 DBMS_WORKLOAD_REPOSITORY 패키지를 사용해 보고서를 생성할 수 있습니다.

VPC의 DB 인스턴스에 사용하기 위한 데이터베이스 링크 조정

동일한 VPC 또는 피어링된 VPC 내에서 Amazon RDS DB 인스턴스 간에 Oracle 데이터베이스 링크를 사용하려면 두 DB 인스턴스에 서로에게 이르는 유효한 경로가 있어야 합니다. VPC 라우팅 테이블과 네트워크 ACL(액세스 제어 목록)을 사용하여 DB 인스턴스 간 유효 경로를 확인합니다.

각 DB 인스턴스의 보안 그룹은 다른 DB 인스턴스로(부터)의 수신 및 발신을 허용해야 합니다. 인바운드 및 아웃바운드 규칙은 동일한 VPC 또는 피어링된 VPC에서 보안 그룹을 참조할 수 있습니다. 자세한 내용은 [피어링된 VPC 보안 그룹을 참조하도록 보안 그룹 업데이트 단원](#)을 참조하십시오.

VPC에서 DHCP 옵션 세트를 이용해 사용자 지정 DNS 서버를 구성했다면, 사용자 지정 DNS 서버가 데이터베이스 링크 타깃의 이름을 확인할 수 있어야 합니다. 자세한 내용은 [사용자 지정 DNS 서버 설정 \(p. 844\)](#) 단원을 참조하십시오.

Oracle Data Pump로 데이터베이스 링크를 사용하는 방법에 대한 자세한 내용은 [Oracle Data Pump를 사용한 가져오기 \(p. 759\)](#) 단원을 참조하십시오.

DB 인스턴스의 기본 에디션 설정

데이터베이스 객체는 에디션이라고 하는 프라이빗 환경에서 재정의할 수 있습니다. 이러한 에디션 기반 재정의를 통해 서비스 중단을 최소화하면서 애플리케이션의 데이터베이스 객체를 업그레이드할 수 있습니다.

Amazon RDS Oracle DB 인스턴스의 기본 에디션은 Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.alter_default_edition`을 사용하여 설정할 수 있습니다.

다음은 Amazon RDS Oracle DB 인스턴스의 기본 에디션을 `RELEASE_V1`으로 설정하는 예제입니다.

```
exec rdsadmin.rdsadmin_util.alter_default_edition('RELEASE_V1');
```

다음은 Amazon RDS Oracle DB 인스턴스의 기본 에디션을 다시 Oracle 기본값으로 설정하는 예제입니다.

```
exec rdsadmin.rdsadmin_util.alter_default_edition('ORA$BASE');
```

Oracle 에디션 기반 재정의에 대한 자세한 내용은 Oracle 설명서에서 [About Editions and Edition-Based Redefinition](#)을 참조하십시오.

SYS.AUD\$ 테이블에 대한 감사 활성화

Amazon RDS 프로시저인 `rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table`을 사용해 데이터베이스 감사 주적 테이블인 `SYS.AUD$`에서 감사를 활성화할 수 있습니다. 지원되는 유일한 감사 속성은 `ALL`입니다. 개별 문 또는 연산을 감사하거나 감사하지 않을 수 있습니다.

감사 활성화는 다음 버전을 실행하는 Oracle DB 인스턴스에 대해 지원됩니다.

- 11.2.0.4.v18 이상의 11.2 버전
- 12.1.0.2.v14 이상의 12.1 버전
- 12.2 이상 버전 전체

`audit_all_sys_aud_table` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
p_by_access	부울	true	선택	true로 설정하여 BY ACCESS를 감사합니다. false로 설정하여 BY SESSION을 감사합니다.

다음 쿼리는 데이터베이스의 SYS.AUD\$에 대한 현재 감사 구성을 반환합니다.

```
select * from dba_obj_audit_opts where owner='SYS' and object_name='AUD$';
```

다음 명령은 SYS.AUD\$ BY ACCESS의 ALL에 대한 감사를 활성화합니다.

```
exec rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table;  
exec rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => true);
```

다음 명령은 SYS.AUD\$ BY SESSION의 ALL에 대한 감사를 활성화합니다.

```
exec rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => false);
```

자세한 내용은 Oracle 설명서의 [AUDIT\(기존 감사\)](#)를 참조하십시오.

SYS.AUD\$ 테이블에 대한 감사 비활성화

Amazon RDS 프로시저인 `rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table`을 사용해 데이터베이스 감사 추적 테이블인 SYS.AUD\$에서 감사를 비활성화할 수 있습니다. 이 프로시저에는 파라미터가 없습니다.

다음 쿼리는 데이터베이스의 SYS.AUD\$에 대한 현재 감사 구성을 반환합니다.

```
select * from dba_obj_audit_opts where owner='SYS' and object_name='AUD$';
```

다음 명령은 SYS.AUD\$의 ALL에 대한 감사를 비활성화합니다.

```
exec rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table;
```

자세한 내용은 Oracle 설명서의 [NOAUDIT\(기존 감사\)](#)를 참조하십시오.

Oracle DB 인스턴스에 대한 공통 DBA 로그 작업

이 단원에서는 Oracle을 실행하는 Amazon RDS DB 인스턴스에서 로깅과 관련된 공통 DBA 작업을 수행하는 방법에 대해 살펴보겠습니다. 관리형 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 shell 액세스를 제공하지 않으며, 고급 권한이 필요한 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다.

자세한 내용은 [Oracle 데이터베이스 로그 파일 \(p. 326\)](#) 단원을 참조하십시오.

주제

- [강제 로깅 설정 \(p. 852\)](#)
- [보충 로깅 설정 \(p. 852\)](#)
- [온라인 로그 파일 전환 \(p. 853\)](#)
- [온라인 다시 실행 로그 추가 \(p. 853\)](#)
- [온라인 다시 실행 로그 드롭 \(p. 854\)](#)
- [온라인 다시 실행 로그 크기 조절 \(p. 854\)](#)
- [보관된 다시 실행 로그 보존 \(p. 856\)](#)
- [트랜잭션 로그 액세스 \(p. 857\)](#)

강제 로깅 설정

강제 로깅 모드에서는 Oracle이 임시 테이블스페이스와 임시 세그먼트의 변경 사항을 제외하고 데이터베이스의 모든 변경 사항을 기록합니다(NOLOGGING 절은 무시됩니다). 자세한 내용은 Oracle 문서의 [Specifying FORCE LOGGING Mode](#) 단원을 참조하십시오.

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.force_logging`을 사용하여 강제 로깅을 설정할 수 있습니다. `force_logging` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
<code>p_enable</code>	부울	<code>true</code>	선택	<code>true</code> 로 설정하면 데이터베이스를 강제 로깅 모드로 설정하고, <code>false</code> 로 설정하면 데이터베이스를 강제 로깅 모드에서 해제합니다.

다음은 데이터베이스를 강제 로깅 모드로 설정하는 예제입니다.

```
exec rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

보충 로깅 설정

보충 로깅은 LogMiner와 LogMiner 기술을 사용하는 제품이 충분한 정보를 바탕으로 행 체인을 비롯해 클러스터 테이블 같은 스토리지 정렬까지 지원하게 합니다. 자세한 내용은 Oracle 문서의 [Supplemental Logging](#)을 참조하십시오.

Oracle 데이터베이스는 기본적으로 보충 로깅이 활성화되어 있지 않습니다. Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.alter_supplemental_logging`을 사용하여 보충 로깅을 활성화 또는 비활성화할 수 있습니다. Amazon RDS가 Oracle DB 인스턴스를 위해 보관된 다시 실행 로그 보존을 관리하는 방법에 대한 자세한 내용은 [보관된 다시 실행 로그 보존 \(p. 856\)](#) 단원을 참조하십시오.

`alter_supplemental_logging` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
p_action	varchar2	—	Required	'ADD'는 보충 로깅을 추가하며, 'DROP'은 보충 로깅을 드롭합니다.
p_type	varchar2	null	선택	보충 로깅 유형입니다. 유효한 값은 'ALL', 'FOREIGN KEY', 'PRIMARY KEY' 또는 'UNIQUE'입니다.

다음 예에서는 보충 로깅을 활성화합니다.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD');
end;
/
```

다음 예에서는 고정 길이 최대 크기 열 전체에 대해 보충 로깅을 활성화합니다.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD',
        p_type    => 'ALL');
end;
/
```

다음 예에서는 기본 키 열에 대한 보충 로깅을 활성화합니다.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD',
        p_type   => 'PRIMARY KEY');
end;
/
```

온라인 로그 파일 전환

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.switch_logfile`을 사용하여 로그 파일을 바꿀 수 있습니다. `switch_logfile` 프로시저에는 파라미터가 없습니다.

다음은 로그 파일을 바꾸는 예제입니다.

```
exec rdsadmin.rdsadmin_util.switch_logfile;
```

온라인 다시 실행 로그 추가

Oracle을 실행하는 Amazon RDS DB 인스턴스는 각각 128MB인 온라인 다시 실행 로그 4개로 시작합니다. Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.add_logfile`을 이용해 다른 다시 실행 로그를 추가할 수 있습니다.

Oracle 버전에 상관없이, `add_logfile` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
bytes	positive	null	선택	로그 파일 크기(바이트)입니다.

`add_logfile` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
p_size	varchar2	—	Required	로그 파일 크기입니다. 크기는 킬로바이트(K), 메가바이트(M)나 기가바이트(G)로 설정할 수 있습니다.

다음 명령을 실행하면 100MB 로그 파일이 추가됩니다.

```
exec rdsadmin.rdsadmin_util.add_logfile(p_size => '100M');
```

온라인 다시 실행 로그 드롭

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.drop_logfile`을 이용해 다시 실행 로그를 드롭할 수 있습니다. `drop_logfile` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
grp	positive	—	Required	로그의 그룹 번호입니다.

다음 예에서는 그룹 번호가 3인 로그를 드롭합니다.

```
exec rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
```

상태가 미사용이거나 비활성인 로그만 드롭할 수 있습니다. 다음 예에서는 로그의 상태를 가져옵니다.

```
select GROUP#, STATUS from V$LOG;
GROUP#      STATUS
----- -----
1           CURRENT
2           INACTIVE
3           INACTIVE
4           UNUSED
```

온라인 다시 실행 로그 크기 조절

Oracle을 실행하는 Amazon RDS DB 인스턴스는 각각 128MB인 온라인 다시 실행 로그 4개로 시작합니다. 다음은 Amazon RDS 프로시저를 사용하여 로그를 각각 128MB에서 512MB로 조정하는 예제입니다.

```
/* 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

/* Issue a checkpoint to clear log 2. */

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

```

보관된 다시 실행 로그 보존

보관된 다시 실행 로그는 Oracle LogMiner(DBMS_LOGMNR) 같은 제품에서 사용할 수 있도록 DB 인스턴스에 로컬 보존할 수 있습니다. 다시 실행 로그를 보존하면 LogMiner를 사용하여 로그를 분석할 수 있습니다. 자세한 내용은 Oracle 문서의 [Using LogMiner to Analyze Redo Log Files](#) 단원을 참조하십시오.

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.set_configuration`을 사용하여 보관된 다시 실행 로그를 보존할 수 있습니다. `set_configuration` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
<code>name</code>	varchar	—	Required	업데이트할 구성의 이름입니다.
<code>value</code>	varchar	—	Required	구성 값입니다.

다음 예에서는 24시간 동안 다시 실행 로그를 보존합니다.

```
begin
```

```
rdsadmin.rdsadmin_util.set_configuration(  
    name => 'archivelog retention hours',  
    value => '24');  
end;  
/  
commit;
```

Note

변경 사항을 적용하려면 커밋해야 합니다.

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.show_configuration`을 사용하여 DB 인스턴스 재실행 로그의 보관 기간을 볼 수 있습니다.

다음 예에서는 로그 보관 시간을 보여줍니다.

```
set serveroutput on  
exec rdsadmin.rdsadmin_util.show_configuration;
```

`archivelog retention hours`의 현재 설정이 출력됩니다. 다음 출력에서는 보관된 재실행 로그가 48시간 동안 보관되었음을 보여줍니다.

```
NAME:archivelog retention hours  
VALUE:48  
DESCRIPTION:ArchiveLog expiration specifies the duration in hours before archive/redo log  
files are automatically deleted.
```

보관된 다시 실행 로그는 DB 인스턴스에 보존되므로 보존된 로그를 수용하기에 충분한 스토리지가 DB 인스턴스에 할당되어 있는지 확인해야 합니다. 지난 X 시간 동안 DB 인스턴스가 사용한 공간을 측정하려면 X 자리에 사용 시간을 대입하여 다음 쿼리를 실행합니다.

```
select sum(BLOCKS * BLOCK_SIZE) bytes  
  from V$ARCHIVED_LOG  
 where FIRST_TIME >= SYSDATE-(X/24) and DEST_ID=1;
```

보관된 다시 실행 로그는 DB 인스턴스의 백업 보존 기간이 0보다 큰 경우에만 생성됩니다. 백업 보존 기간은 기본적으로 0보다 크므로, 명시적으로 0으로 설정하지 않는 한 보관된 다시 실행 로그는 DB 인스턴스에 생성됩니다. DB 인스턴스의 백업 보존 기간을 수정하려면 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

보관된 다시 실행 로그를 DB 인스턴스에서 제거한 후에 DB 인스턴스에 다시 다운로드할 수 있습니다. Amazon RDS는 보관된 다시 실행 로그를 DB 인스턴스 외부에 보존하여 일정 시점에 DB 인스턴스를 복구하도록 지원합니다. Amazon RDS는 DB 인스턴스에 구성된 백업 보존 기간에 따라 DB 인스턴스 외부에 보관된 다시 실행 로그를 보존합니다. DB 인스턴스의 백업 보존 기간을 수정하려면 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

Note

어떤 경우에는 Linux 기반 JDBC를 사용해 보관된 재실행 로그를 다운로드하면 지역 시간이 길어지고 연결이 재설정되는 일을 겪을 수 있습니다. 그러한 경우 Java 클라이언트의 기본 난수 생성기 설정이 문제의 원인일 수 있습니다. 비차단형 난수 생성기를 사용하도록 JDBC 드라이버를 설정하는 것이 좋습니다.

트랜잭션 로그 액세스

트랜잭션 로그 액세스는 Oracle 버전 11.2.0.4.v11 이상과 12.1.0.2.v7 이상에서 지원됩니다.

GoldenGate, Attunity, Informatica 등과 같은 외부 도구로 마이닝하려면 온라인 및 아카이브된 다시 실행 로그 파일에 액세스해야 합니다. 온라인 및 아카이브된 다시 실행 로그 파일에 액세스하려면, 먼저 물리적 파일 경로에 읽기 전용으로 액세스할 수 있는 디렉터리 객체를 생성해야 합니다.

다음은 온라인 및 아카이브된 다시 실행 로그 파일에 읽기 전용으로 액세스할 수 있는 디렉터리를 생성하는 코드입니다.

Important

이 코드는 DROP ANY DIRECTORY 권한도 취소합니다.

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

온라인 및 아카이브된 다시 실행 로그 파일용 디렉터리 객체를 생성한 후 PL/SQL을 사용하여 이 파일을 읽을 수 있습니다. 디렉터리 객체의 파일을 읽는 방법에 대한 자세한 내용은 [DB 인스턴스 디렉터리의 파일 목록 표시 \(p. 874\)](#) 및 [DB 인스턴스 디렉터리의 파일 목록 읽기 \(p. 875\)](#) 단원을 참조하십시오.

다음은 온라인 및 보관된 재실행 로그 파일용 디렉터리를 삭제하는 코드입니다.

```
exec rdsadmin.rdsadmin_master_util.drop_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.drop_onlinelog_dir;
```

다음은 DROP ANY DIRECTORY 권한을 부여하고 취소하는 코드입니다.

```
exec rdsadmin.rdsadmin_master_util.revoke_drop_any_directory;
exec rdsadmin.rdsadmin_master_util.grant_drop_any_directory;
```

Oracle DB 인스턴스에 대한 공통 DBA RMAN(Recovery Manager) 작업

아래 단원에서는 Oracle을 실행하는 Amazon RDS DB 인스턴스에서 Oracle RMAN(Recovery Manager) DBA 작업을 수행하는 방식을 확인하실 수 있습니다. 관리형 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 shell 액세스를 제공하지 않습니다. 또한 고급 권한이 필요한 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다.

Amazon RDS 패키지인 `rdsadmin.rdsadmin_rman_util`을 사용해 Oracle 데이터베이스용 Amazon RDS의 RMAN 백업을 디스크에 수행할 수 있습니다. `rdsadmin.rdsadmin_rman_util` 패키지에서는 전체 및 충분 데이터베이스 파일 백업, 테이블스페이스 백업 및 아카이브 로그 백업을 지원합니다.

RMAN 백업은 Amazon RDS DB 인스턴스 호스트의 스토리지 스페이스를 소비합니다. 백업을 수행할 때 Oracle 디렉터리 객체를 프로시저 호출 시 파라미터로 지정하십시오. 백업 파일은 지정된 디렉터리에 있습니다. `DATA_PUMP_DIR`과 같은 기본 디렉터리를 사용하거나 새 디렉터리를 생성할 수 있습니다. 자세한 내용은 [메인 데이터 스토리지 공간에 새로운 디렉터리 생성 \(p. 874\)](#) 단원을 참조하십시오.

RMAN 백업이 종료된 후에는 Oracle DB 인스턴스 호스트에 대해 Amazon RDS에서 백업 파일을 복사할 수 있습니다. 이 작업은 RDS가 아닌 호스트 또는 백업 장기 보관에 대해 복원 목적으로 수행할 수 있습니다. 예를 들어 백업 파일을 Amazon S3 버킷에 복사할 수 있습니다. 자세한 내용은 [Amazon S3통합 \(p. 773\)](#) 사용 단원을 참조하십시오.

RMAN 백업을 위한 백업 파일은 수동으로 제거하기 전에는 Amazon RDS DB 인스턴스에 남아 있습니다. `UTL_FILE.FREMOVE` Oracle 프로시저를 사용하여 디렉터리에서 파일을 제거할 수 있습니다. 자세한 내용은 Oracle 설명서의 [FREMOVE 프로시저](#)를 참조하십시오.

보관된 재실행 로그를 백업하거나 보관된 재실행 로그가 포함된 전체 또는 충분 백업을 수행할 때는 재실행 로그 보존을 0이 아닌 값으로 설정해야 합니다. 자세한 내용은 [보관된 다시 실행 로그 보존 \(p. 856\)](#) 단원을 참조하십시오.

Note

또 하나의 Oracle DB 인스턴스용 Amazon RDS에 대한 백업 및 복원을 위해서는 Amazon RDS 백업 및 복원 기능을 계속 사용할 수 있습니다. 자세한 내용은 [Amazon RDS DB 인스턴스 백업 및 복원 \(p. 196\)](#) 단원을 참조하십시오.

현재 RMAN 복원은 Oracle DB용 Amazon RDS 인스턴스에는 지원되지 않습니다.

주제

- [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#)
- [DB 인스턴스 파일 확인 \(p. 862\)](#)
- [블록 변경 추적 활성화 및 비활성화 \(p. 865\)](#)
- [보관된 재실행 로그 대조 확인 \(p. 865\)](#)
- [보관된 재실행 로그 백업 \(p. 866\)](#)
- [전체 데이터베이스 백업 수행 \(p. 871\)](#)
- [증분 데이터베이스 백업 수행 \(p. 871\)](#)
- [테이블스페이스 백업 수행 \(p. 872\)](#)

RMAN 프로시저용 공통 파라미터

Amazon RDS 패키지인 `rdsadmin.rdsadmin_rman_util`에서 프로시저를 사용해 RMAN으로 작업을 수행할 수 있습니다. 이 패키지에서 몇 가지 파라미터는 프로시저에 공통됩니다. 이 패키지에는 다음과 같은 공통 파라미터가 있습니다.

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
<code>p_owner</code>	<code>varchar2</code>	<code>p_directory_name</code> 에 지정된 디렉터리의 유효한 소유자입니다.		Required	백업 파일을 담을 디렉터리의 소유자입니다.
<code>p_directory_name</code>	<code>varchar2</code>	유효한 데이터베이스 디렉터리 이름입니다.	-	Required	백업 파일을 담을 디렉터리의 이름입니다.
<code>p_label</code>	<code>varchar2</code>	a-z, A-Z, 0-9, '_', '-', '.'	-	선택	백업 파일 이름에 포함된 고유 문자열입니다. Note 제한은 30자입니다.
<code>p_compress</code>	부울	TRUE, FALSE	FALSE	선택	TRUE를 지정하여 기본 백업 압축을 활성화합니다. FALSE를 지정하여 기본 백업 압축을 비활성화합니다.

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
p_include_archive_logs	부울	TRUE, FALSE	FALSE	선택	<p>TRUE를 지정하여 보관된 재실행 로그를 백업에 포함합니다.</p> <p>FALSE를 지정하여 보관된 재실행 로그를 백업에서 제외합니다.</p> <p>보관된 재실행 로그를 백업에 포함하는 경우 <code>rdsadmin.rdsadmin_util.set_co</code> 프로시저를 사용해 보존 기간을 1시간 이상으로 설정하십시오. 또한 백업을 실행하기 전에 <code>rdsadmin.rdsadmin_rman_util.c</code> 프로시저를 즉시 호출하십시오. 이렇게 하지 않으면 Amazon RDS 관리 프로시저에서 삭제한 보관된 재실행 로그가 누락되어 백업이 실패할 수 있습니다.</p>
p_include_controlfile	부울	TRUE, FALSE	FALSE	선택	<p>TRUE를 지정하여 백업에 제어 파일을 포함합니다.</p> <p>FALSE를 지정하여 백업에서 제어 파일을 제외합니다.</p>
p_optimize	부울	TRUE, FALSE	TRUE	선택	<p>보관된 재실행 로그가 포함된 경우 백업 크기를 줄이기 위해 TRUE를 지정하여 백업 최적화를 활성화합니다.</p> <p>FALSE를 지정하여 백업 최적화를 비활성화합니다.</p>

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
p_parallel	숫자	Oracle Database Enterprise Edition(EE)에서 1과 254 사이의 유효한 정수입니다. 다른 Oracle Database 에디션일 때는 1입니다.	1	선택	채널 수입니다.
p_rman_to_dbms_output	부울	TRUE, FALSE	FALSE	선택	TRUE일 때는 RMAN 출력이 DBMS_OUTPUT 키지를 비롯해 BDUMP 디렉터리의 파일로 전송됩니다. SQL*Plus를 사용할 경우에는 SET SERVEROUTPUT ON을 실행하여 출력을 확인합니다. FALSE일 때는 RMAN 출력이 BDUMP 디렉터리의 파일로만 전송됩니다.
p_section_size_mb	숫자	유효한 정수입니다.	NULL	선택	섹션 크기(MB)입니다. 각 파일을 지정된 섹션 크기로 나누어서 함께 확인합니다. NULL일 때는 파라미터를 무시합니다.

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
p_validation_type	varchar2	'PHYSICAL','PHYSICAL+LOGICAL'	'선택'		<p>손상 탐지 수준입니다.</p> <p>물리적 손상 여부를 알아보고 싶다면 'PHYSICAL'을 지정합니다. 물리적 손상을 예로 들면 헤더와 푸터가 서로 일치하는 않는 블록이 있습니다.</p> <p>물리적 손상 외에 논리적 블일치 여부도 알아보고 싶다면 'PHYSICAL+LOGICAL'을 지정합니다. 논리적 손상의 예로는 잘못된 블록을 들 수 있습니다.</p>

DB 인스턴스 파일 확인

데이터 파일, 테이블스페이스, 제어 파일, 서버 파라미터 파일(SPFILE) 같은 Oracle DB용 Amazon RDS 인스턴스 파일은 Amazon RDS 패키지 `rdsadmin.rdsadmin_rman_util`을 사용하여 확인할 수 있습니다.

RMAN 확인에 대한 자세한 내용은 Oracle 설명서에서 [Validating Database Files and Backups](#) 및 [VALIDATE](#)를 참조하십시오.

주제

- [DB 인스턴스 확인 \(p. 862\)](#)
- [테이블 스페이스 확인 \(p. 863\)](#)
- [제어 파일 확인 \(p. 864\)](#)
- [SPFILE 확인 \(p. 864\)](#)
- [데이터 파일 확인 \(p. 864\)](#)

DB 인스턴스 확인

Amazon RDS Oracle DB 인스턴스에서 사용되는 관련 파일은 모두 Amazon RDS 프로시저 `rdsadmin.rdsadmin_rman_util.validate_database`를 사용하여 확인할 수 있습니다.

이 프로시저에서는 RMAN 작업을 위해 다음과 같은 공통 파라미터를 사용합니다.

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

다음 예에서는 파라미터의 기본값을 사용하여 DB 인스턴스를 확인합니다.

```
exec rdsadmin.rdsadmin_rman_util.validate_database;
```

다음 예에서는 지정된 값을 파라미터에 사용하여 DB 인스턴스를 확인합니다.

```
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;
/
```

p_rman_to_dbms_output 파라미터가 FALSE로 설정되면 RMAN 출력이 BDUMP 디렉터리의 파일로 작성됩니다.

BDUMP 디렉터리의 파일을 보려면 다음과 같이 SELECT 문을 실행합니다.

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime;
```

BDUMP 디렉터리의 파일 내용을 보려면 다음과 같이 SELECT 문을 실행합니다.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','rds-rman-validate-nnn.txt'));
```

파일 이름을 보려고 하는 파일 이름으로 변경합니다.

테이블 스페이스 확인

테이블 스페이스와 연결된 파일은 Amazon RDS 프로시저
rdsadmin.rdsadmin_rman_util.validate_tablespace를 사용하여 확인할 수 있습니다.

이 프로시저에서는 RMAN 작업을 위해 다음과 같은 공통 파라미터를 사용합니다.

- p_validation_type
- p_parallel
- p_section_size_mb
- p_rman_to_dbms_output

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

이 프로시저에서는 다음과 같은 추가 파라미터도 사용합니다.

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
p_tablespace_name	varchar2	유효한 테이블 스페이스 이름	—	Required	테이블스페이스의 이름입니다.

제어 파일 확인

Amazon RDS Oracle DB 인스턴스에서 사용되는 제어 파일은 Amazon RDS 프로시저 `rdsadmin.rdsadmin_rman_util.validate_current_controlfile`을 사용하여 확인할 수 있습니다.

이 프로시저에서는 RMAN 작업을 위해 다음과 같은 공통 파라미터를 사용합니다.

- `p_validation_type`
- `p_rman_to_dbms_output`

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

SPFILE 확인

Amazon RDS Oracle DB 인스턴스에서 사용되는 서버 파라미터 파일(SPFILE)은 Amazon RDS 프로시저 `rdsadmin.rdsadmin_rman_util.validate_spfile`을 사용하여 확인할 수 있습니다.

이 프로시저에서는 RMAN 작업을 위해 다음과 같은 공통 파라미터를 사용합니다.

- `p_validation_type`
- `p_rman_to_dbms_output`

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

데이터 파일 확인

데이터 파일은 Amazon RDS 프로시저 `rdsadmin.rdsadmin_rman_util.validate_datafile`을 사용하여 확인할 수 있습니다.

이 프로시저에서는 RMAN 작업을 위해 다음과 같은 공통 파라미터를 사용합니다.

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

이 프로시저에서는 다음과 같은 추가 파라미터도 사용합니다.

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
<code>p_datafile</code>	<code>varchar2</code>	유효한 데이터파일 ID 번호 또는 유효한 데이터파일 이름(전체 경로 포함)	—	Required	데이터파일 ID 번호(<code>v\$datafile.file#</code>) 또는 경로가 포함된 전체 데이터파일 이름(<code>v\$datafile.name</code>)입니다.
<code>p_from_block</code>	숫자	유효한 정수입니다.	<code>NULL</code>	선택	데이터 파일 내에서 확인이 시작되는 블록 번호입니다.

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
					니다. 이 값이 NULL인 경우 1이 사용됩니다.
p_to_block	숫자	유효한 정수입니다.	NULL	선택	데이터 파일 내에서 확인이 끝나는 블록 번호입니다. 이 값이 NULL일 때는 데이터 파일에서 최대 블록이 사용됩니다.

블록 변경 추적 활성화 및 비활성화

Amazon RDS 프로시저인 `rdsadmin.rdsadmin_rman_util.enable_block_change_tracking`을 사용해 DB 인스턴스에 대한 블록 변경 추적을 활성화할 수 있습니다.

Amazon RDS 프로시저인 `rdsadmin.rdsadmin_rman_util.disable_block_change_tracking`을 사용해 DB 인스턴스에 대한 블록 변경 추적을 비활성화할 수 있습니다.

이 프로시저에는 파라미터가 없습니다. 블록 변경 추적을 활성화하면 충분 백업의 성능을 향상 시킬 수 있습니다.

이 프로시저는 다음 Oracle DB용 Amazon RDS 엔진 버전에서 지원됩니다.

- 11.2.0.4.v19 이상의 11.2 버전
- 12.1.0.2.v15 이상의 12.1 버전
- 12.2.0.1.ru-2019-01.rur-2019-01.r1 이상의 12.2 버전

DB 인스턴스에 대해 블록 변경 추적이 활성화되어 있는지 확인하려면 다음 쿼리를 실행하십시오.

```
SELECT status, filename FROM V$BLOCK_CHANGE_TRACKING;
```

다음 예에서는 DB 인스턴스에 대한 블록 변경 추적을 활성화합니다.

```
exec rdsadmin.rdsadmin_rman_util.enable_block_change_tracking;
```

다음 예에서는 DB 인스턴스에 대한 블록 변경 추적을 비활성화합니다.

```
exec rdsadmin.rdsadmin_rman_util.disable_block_change_tracking;
```

보관된 재실행 로그 대조 확인

보관된 재실행 로그를 Amazon RDS 프로시저 `rdsadmin.rdsadmin_rman_util.crosscheck_archivelog`를 사용해 대조 확인할 수 있습니다.

이 프로시저를 사용해 제어 파일에 등록된 보관된 재실행 로그를 대조 확인하고 선택 사항으로 만료된 로그를 삭제할 수 있습니다. RMAN 백업을 수행할 때마다 제어 파일에 레코드 파일이 생성됩니다. 시간이 지남에

따라 이 레코드로 인해 제어 파일의 크기가 증가합니다. 오래되고 만료된 레코드는 주기적으로 비워주는 것
이 좋습니다.

Note

표준 Amazon RDS 백업은 RMAN을 사용하지 않으므로 제어 파일에 레코드를 생성하지 않습니다.

이 프로시저에서는 RMAN 작업을 위해 공통 파라미터인 `p_rman_to_dbms_output`을 사용합니다.

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

이 프로시저에서는 다음과 같은 추가 파라미터도 사용합니다.

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
<code>p_delete_expired</code>	부울	TRUE, FALSE	TRUE	선택	TRUE인 경우 제어 파일 에서 만료된 보관된 재실행 로그를 삭제합니다. FALSE인 경우 제어 파일 에 만료된 보관된 재실행 로그를 보관합니다.

이 프로시저는 다음 Oracle DB용 Amazon RDS 엔진 버전에서 지원됩니다.

- 11.2.0.4.v19 이상의 11.2 버전
- 12.1.0.2.v15 이상의 12.1 버전
- 12.2.0.1.ru-2019-01.rur-2019-01.r1 이상의 12.2 버전

다음 예에서는 제어 파일에서 만료된 보관된 재실행 로그를 삭제합니다.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
        p_delete_expired      => 4,
        p_rman_to_dbms_output => FALSE);
END;
/
```

보관된 재실행 로그 백업

Amazon RDS 패키지인 `rdsadmin.rdsadmin_rman_util`을 사용해 Amazon RDS Oracle DB 인스턴스에
대해 보관된 재실행 로그를 백업할 수 있습니다.

보관된 재실행 로그 백업 프로시저는 다음과 같은 Oracle DB용 Amazon RDS 엔진 버전에서 지원됩니다.

- 11.2.0.4.v19 이상의 11.2 버전
- 12.1.0.2.v15 이상의 12.1 버전
- 12.2.0.1.ru-2019-01.rur-2019-01.r1 이상의 12.2 버전

주제

- 보관된 재실행 로그 전체 백업 (p. 867)
- 날짜 범위에서 보관된 재실행 로그 백업 (p. 867)
- SCN 범위에서 보관된 재실행 로그 백업 (p. 868)
- 시퀀스 번호 범위에서 보관된 재실행 로그 백업 (p. 870)

보관된 재실행 로그 전체 백업

Amazon RDS 프로시저인 `rdsadmin.rdsadmin_rman_util.backup_archivelog_all`을 사용해 Amazon RDS Oracle DB 인스턴스에 대해 보관된 재실행 로그를 모두 백업할 수 있습니다.

이 프로시저에서는 RMAN 작업을 위해 다음과 같은 공통 파라미터를 사용합니다.

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

다음 예에서는 DB 인스턴스에 대해 모든 보관된 재실행 로그를 백업합니다.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_all(
        p_owner          => 'MYDBADMIN',
        p_directory_name => 'MYDIRECTORY',
        p_parallel       => 4,
        p_rman_to_dbms_output => FALSE);
END;
/
```

날짜 범위에서 보관된 재실행 로그 백업

Amazon RDS 프로시저인 `rdsadmin.rdsadmin_rman_util.backup_archivelog_date`를 사용해 날짜 범위를 지정하여 Amazon RDS Oracle DB 인스턴스에 대해 특정 보관된 재실행 로그를 백업할 수 있습니다. 날짜 범위에서는 백업할 보관된 재실행 로그를 지정합니다.

이 프로시저에서는 RMAN 작업을 위해 다음과 같은 공통 파라미터를 사용합니다.

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

이 프로시저에서는 다음과 같은 추가 파라미터도 사용합니다.

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
p_from_date	날짜	디스크에 있는 보관된 재실행 로그의 start_date 및 next_date 사이의 날짜입니다. 이 값은 p_to_date에 대해 지정된 값과 같거나 이보다 작아야 합니다.	—	Required	보관된 로그 백업의 시작 날짜입니다.
p_to_date	날짜	디스크에 있는 보관된 재실행 로그의 start_date 및 next_date 사이의 날짜입니다. 이 값은 p_from_date에 대해 지정된 값과 같거나 이보다 커야 합니다.	—	Required	보관된 로그 백업의 종료 날짜입니다.

다음 예에서는 DB 인스턴스의 날짜 범위에서 보관된 재실행 로그를 백업합니다.

```

BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_date(
        p_owner          => 'MYDBADMIN',
        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;
/

```

SCN 범위에서 보관된 재실행 로그 백업

Amazon RDS 프로시저인 `rdsadmin.rdsadmin_rman_util.backup_archivelog_scn`을 사용해 시스템 변경 번호(SCN) 범위를 지정하여 Amazon RDS Oracle DB 인스턴스에 대해 특정 보관된 재실행 로그를 백업할 수 있습니다. SCN 범위에서는 백업할 보관된 재실행 로그를 지정합니다.

이 프로시저에서는 RMAN 작업을 위해 다음과 같은 공통 파라미터를 사용합니다.

- p_owner
- p_directory_name
- p_label
- p_parallel
- p_compress
- p_rman_to_dbms_output

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

이 프로시저에서는 다음과 같은 추가 파라미터도 사용합니다.

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
p_from_scn	숫자	디스크에 있는 보관된 재실행 로그의 SCN입니다. 이 값은 p_to_scn에 대해 지정된 값과 같거나 이보다 작아야 합니다.	—	Required	보관된 로그 백업의 시작 SCN입니다.
p_to_scn	숫자	디스크에 있는 보관된 재실행 로그의 SCN입니다. 이 값은 p_from_scn에 대해 지정된 값과 같거나 이보다 커야 합니다.	—	Required	보관된 로그 백업의 종료 SCN입니다.

다음 예에서는 DB 인스턴스의 SCN 범위에서 보관된 재실행 로그를 백업합니다.

```

BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_scn(
        p_owner          => 'MYDBADMIN',
        p_directory_name => 'MYDIRECTORY',
        p_from_scn       => 1533835,
        p_to_scn         => 1892447,
        p_parallel       => 4,
        p_rman_to_dbms_output => FALSE);
END;
/

```

시퀀스 번호 범위에서 보관된 재실행 로그 백업

Amazon RDS 프로시저인 `rdsadmin.rdsadmin_rman_util.backup_archivelog_sequence`를 사용해 시퀀스 번호 범위를 지정하여 Amazon RDS Oracle DB 인스턴스에 대해 특정 보관된 재실행 로그를 백업할 수 있습니다. 시퀀스 번호 범위에서는 백업할 보관된 재실행 로그를 지정합니다.

이 프로시저에서는 RMAN 작업을 위해 다음과 같은 공통 파라미터를 사용합니다.

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

이 프로시저에서는 다음과 같은 추가 파라미터도 사용합니다.

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
<code>p_from_sequence</code>	숫자	디스크에 있는 보관된 재실행 로그의 시퀀스 번호입니다. 이 값은 <code>p_to_sequence</code> 에 대해 지정된 값과 같거나 이보다 작아야 합니다.	—	Required	보관된 로그 백업의 시작 시퀀스 번호입니다.
<code>p_to_sequence</code>	숫자	디스크에 있는 보관된 재실행 로그의 시퀀스 번호입니다. 이 값은 <code>p_from_sequence</code> 에 대해 지정된 값과 같거나 이보다 커야 합니다.	—	Required	보관된 로그 백업의 종료 시퀀스 번호입니다.

다음 예에서는 DB 인스턴스의 시퀀스 번호 범위 범위에서 보관된 재실행 로그를 백업합니다.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_sequence(
        p_owner          => 'MYDBADMIN',
        p_directory_name => 'MYDIRECTORY',
        p_from_sequence  => 11160,
        p_to_sequence    => 11160,
        p_parallel       => 4,
        p_rman_to_dbms_output => FALSE);
END;
/
```

전체 데이터베이스 백업 수행

Amazon RDS 프로시저인 `rdsadmin.rdsadmin_rman_util.backup_database_full`을 사용해 백업에 포함된 데이터 파일의 모든 블록을 백업할 수 있습니다.

이 프로시저에서는 RMAN 작업을 위해 다음과 같은 공통 파라미터를 사용합니다.

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archived_logs`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

이 프로시저는 다음 Oracle DB용 Amazon RDS 엔진 버전에서 지원됩니다.

- 11.2.0.4.v19 이상의 11.2 버전
- 12.1.0.2.v15 이상의 12.1 버전
- 12.2.0.1.ru-2019-01.rur-2019-01.r1 이상의 12.2 버전

다음 예에서는 파라미터에 지정된 값을 사용하여 완전한 DB 인스턴스 백업을 수행합니다.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_database_full(
        p_owner          => 'MYDBADMIN',
        p_directory_name => 'MYDIRECTORY',
        p_parallel       => 4,
        p_section_size_mb => 10,
        p_rman_to_dbms_output => FALSE);
END;
/
```

증분 데이터베이스 백업 수행

Amazon RDS 프로시저인 `rdsadmin.rdsadmin_rman_util.backup_database_incremental`을 사용해 DB 인스턴스에 대한 증분 백업을 수행할 수 있습니다.

증분 백업에 대한 자세한 내용은 Oracle 설명서의 [증분 백업](#)을 참조하십시오.

이 프로시저에서는 RMAN 작업을 위해 다음과 같은 공통 파라미터를 사용합니다.

- p_owner
- p_directory_name
- p_label
- p_parallel
- p_section_size_mb
- p_include_archived_logs
- p_include_controlfile
- p_optimize
- p_compress
- p_rman_to_dbms_output

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

이 프로시저는 다음 Oracle DB용 Amazon RDS 엔진 버전에서 지원됩니다.

- 11.2.0.4.v19 이상의 11.2 버전
- 12.1.0.2.v15 이상의 12.1 버전
- 12.2.0.1.ru-2019-01.rur-2019-01.r1 이상의 12.2 버전

이 프로시저에서는 다음과 같은 추가 파라미터도 사용합니다.

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
p_level	숫자	0, 1	0	선택	0을 지정하여 전체 증분 백업을 활성화합니다. 1을 지정하여 비누적 증분 백업을 활성화합니다.

다음 예에서는 지정된 값을 파라미터에 사용하여 DB 인스턴스에 대한 증분 백업을 수행합니다.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_database_incremental(
        p_owner          => 'MYDBADMIN',
        p_directory_name => 'MYDIRECTORY',
        p_level          => 1,
        p_parallel       => 4,
        p_section_size_mb => 10,
        p_rman_to_dbms_output => FALSE);
END;
/
```

테이블스페이스 백업 수행

Amazon RDS 프로시저 `rdsadmin.rdsadmin_rman_util.backup_tablespace`를 사용해 DB 인스턴스 테이블스페이스를 수행할 수 있습니다.

이 프로시저에서는 RMAN 작업을 위해 다음과 같은 공통 파라미터를 사용합니다.

- p_owner
- p_directory_name
- p_label
- p_parallel
- p_section_size_mb
- p_include_archived_logs
- p_include_controlfile
- p_optimize
- p_compress
- p_rman_to_dbms_output

자세한 내용은 [RMAN 프로시저용 공통 파라미터 \(p. 859\)](#) 단원을 참조하십시오.

이 프로시저에서는 다음과 같은 추가 파라미터도 사용합니다.

파라미터 이름	데이터 형식	유효한 값	기본값	Required	설명
p_tablespace_name	varchar2	유효한 테이블스페이스 이름입니다.	—	Required	백업할 테이블스페이스의 이름입니다.

이 프로시저는 다음 Oracle DB용 Amazon RDS 엔진 버전에서 지원됩니다.

- 11.2.0.4.v19 이상의 11.2 버전
- 12.1.0.2.v15 이상의 12.1 버전
- 12.2.0.1.ru-2019-01.rur-2019-01.r1 이상의 12.2 버전

다음 예에서는 지정된 값을 파라미터에 사용하여 테이블스페이스 백업을 수행합니다.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_database_incremental(
        p_owner          => 'MYDBADMIN',
        p_directory_name => 'MYDIRECTORY',
        p_tablespace     => 'MYTABLESPACE',
        p_parallel       => 4,
        p_section_size_mb => 10,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Oracle DB 인스턴스에 대한 공통 DBA 기타 작업

이 단원에서는 Oracle을 실행하는 Amazon RDS DB 인스턴스에서 기타 DBA 작업을 수행하는 방법에 대해 살펴보겠습니다. 관리형 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 shell 액세스를 제공하지 않으며, 고급 권한이 필요한 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다.

주제

- 메인 데이터 스토리지 공간에 새로운 디렉터리 생성 (p. 874)
- DB 인스턴스 디렉터리의 파일 목록 표시 (p. 874)
- DB 인스턴스 디렉터리의 파일 목록 읽기 (p. 875)

메인 데이터 스토리지 공간에 새로운 디렉터리 생성

Amazon RDS 프로시저 `rdsadmin.rdsadmin_util.create_directory`를 사용하여 디렉터리를 만들 수 있습니다. 디렉터리를 최대 10,000개까지 만들어 메인 데이터 스토리지 공간에 저장할 수 있습니다.

`create_directory` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
<code>p_directory_name</code>	varchar2	—	Required	새로운 디렉터리의 이름입니다.

다음 예에서는 `product_descriptions`라는 새 디렉터리를 생성합니다.

```
exec rdsadmin.rdsadmin_util.create_directory(p_directory_name => 'product_descriptions');
```

`DBA_DIRECTORIES`에 쿼리를 실행하면 디렉터리 목록을 표시할 수 있습니다. 실제 호스트 경로 이름은 시스템이 자동으로 선택합니다. 다음은 `product_descriptions`라는 디렉터리의 디렉터리 경로를 얻는 예제입니다.

```
select DIRECTORY_PATH
      from DBA_DIRECTORIES
     where DIRECTORY_NAME='product_descriptions';

DIRECTORY_PATH
-----
/rdsdbdata/userdirs/01
```

DB 인스턴스의 마스터 사용자 이름은 새로운 디렉터리에서도 읽기/쓰기 권한이 있으며, 다른 사용자에게 액세스 권한을 부여할 수도 있습니다. 실행 권한은 DB 인스턴스의 디렉터리에서 사용할 수 없습니다. 디렉터리는 메인 데이터 스토리지 공간에 생성되어 일정 공간과 I/O 대역폭을 사용합니다.

생성한 디렉터리는 Oracle `drop directory` 명령을 사용해 삭제할 수 있습니다. 디렉터리를 삭제해도 그 내용은 제거되지 않습니다. `create_directory()` 메서드가 경로 이름을 재사용하므로 삭제한 디렉터리의 파일이 새로 생성한 디렉터리에 나타날 수도 있습니다. 디렉터리를 삭제하려면 먼저 `UTL_FILE.FREMOVE`를 사용해 디렉터리에서 파일을 제거해야 합니다. 자세한 내용은 Oracle 설명서의 [FREMOVE 프로시저](#)를 참조하십시오.

DB 인스턴스 디렉터리의 파일 목록 표시

Amazon RDS 프로시저 `rdsadmin.rds_file_util.listdir`을 이용해 디렉터리의 파일을 나열할 수 있습니다. `listdir` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
<code>p_directory</code>	varchar2	—	Required	목록을 표시할 디렉터리의 이름입니다.

다음 예에서는 `product_descriptions`라는 디렉터리의 파일 목록을 표시합니다.

```
select * from table
(rdsadmin.rds_file_util.listdir(p_directory => 'product_descriptions'));
```

DB 인스턴스 디렉터리의 파일 목록 읽기

Amazon RDS 프로시저 `rdsadmin.rds_file_util.read_text_file`을 사용하여 텍스트 파일을 읽을 수 있습니다. `read_text_file` 프로시저에는 다음과 같은 파라미터가 있습니다.

파라미터 이름	데이터 형식	기본값	Required	설명
p_directory	varchar2	—	Required	파일이 포함된 디렉터리의 이름입니다.
p_filename	varchar2	—	Required	읽을 파일의 이름입니다.

다음 예에서는 `product_descriptions` 디렉터리에서 `rice.txt` 파일을 읽습니다.

```
select * from table
(rdsadmin.rds_file_util.read_text_file(
    p_directory => 'product_descriptions',
    p_filename  => 'rice.txt'));
```

Oracle DB 인스턴스용 도구 및 타사 소프트웨어

이 단원에서는 Amazon RDS의 Oracle DB 인스턴스용 도구 및 타사 소프트웨어에 대한 정보를 제공합니다.

주제

- Oracle용 도구 및 타사 소프트웨어를 호스팅하도록 Amazon RDS 설정 (p. 875)
- Amazon RDS와 함께 Oracle GoldenGate 사용하기 (p. 885)
- Oracle용 Amazon RDS에서 Oracle Repository Creation Utility 사용 (p. 895)
- Amazon RDS에서 Oracle에 Siebel Database 설치 (p. 899)

Oracle용 도구 및 타사 소프트웨어를 호스팅하도록 Amazon RDS 설정

Amazon RDS를 사용하여 다음과 같은 소프트웨어 및 구성 요소를 지원하는 Oracle DB 인스턴스를 호스팅 할 수 있습니다.

- Siebel 고객 관계 관리(CRM)
- Oracle Fusion Middleware 메타데이터 – RCU(Repository Creation Utility)에서 설치

다음 절차를 통해 Oracle용 추가 소프트웨어 및 구성 요소를 호스팅하는 데 사용할 수 있는 Amazon RDS의 Oracle DB 인스턴스를 만들 수 있습니다.

Oracle Database와 함께 사용할 Amazon VPC 생성

다음 절차에서는 Amazon VPC, 프라이빗 서브넷, 보안 그룹을 만듭니다. Amazon RDS DB 인스턴스는 퍼블릭 인터넷이 아닌 미들티어 구성 요소에서만 사용할 수 있어야 하므로 Amazon RDS DB 인스턴스는 프라이빗 서브넷에서 호스팅되고, 이에 따라 보안이 향상됩니다.

Amazon VPC 생성

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. AWS Management 콘솔의 오른쪽 상단 모서리에서 VPC를 위한 AWS 리전을 선택합니다. 이 예에서는 미국 서부(오레곤) 리전을 사용합니다.
3. 좌측 상단 모서리에서 VPC 대시보드를 선택한 다음 VPC 마법사 시작을 선택합니다.
4. Step 1: Select a VPC Configuration 페이지에서 VPC with Public and Private Subnets를 선택한 후 Select를 선택합니다.
5. Step 2: VPC with Public and Private Subnets 페이지에서 다음과 같이 값을 설정합니다.

옵션	값
IPv4 CIDR block	10.0.0.0/16 VPC의 CIDR 블록 선택에 대한 자세한 내용은 VPC Sizing 단원을 참조하십시오.
IPv6 CIDR block	No IPv6 CIDR Block(IPv6 CIDR 블록 없음)
[VPC name]	예를 들어 VPC 이름은 vpc-1 입니다.
Public subnet's IPv4 CIDR(퍼블릭 서브넷의 IPv4 CIDR)	10.0.0.0/24 서브넷 크기 조정에 대한 자세한 내용은 서브넷 크기 조정 을 참조하십시오.
가용 영역	AWS 리전의 가용 영역을 선택합니다.
Public subnet name	예를 들어 퍼블릭 서브넷의 이름은 subnet-public-1 입니다.
Private subnet's IPv4 CIDR(프라이빗 서브넷의 IPv4 CIDR)	10.0.1.0/24 서브넷 크기 조정에 대한 자세한 내용은 서브넷 크기 조정 을 참조하십시오.
가용 영역	AWS 리전의 가용 영역을 선택합니다.
Private subnet name	예를 들어 프라이빗 서브넷의 이름은 subnet-private-1 입니다.
인스턴스 유형	NAT 인스턴스의 인스턴스 유형입니다(예: t2.small). Note 콘솔에 [Instance type]이 표시되지 않으면 [Use a NAT instance instead]를 선택합니다.
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: (65531 IP addresses available)

IPv6 CIDR block: No IPv6 CIDR Block
 Amazon provided IPv6 CIDR block

VPC name:

Public subnet's IPv4 CIDR: (251 IP addresses available)

Availability Zone:

Public subnet name:

Private subnet's IPv4 CIDR: (251 IP addresses available)

Availability Zone:

Private subnet name:

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:

Key pair name:

Service endpoints

Enable DNS hostnames: Yes No

Hardware tenancy:

[Cancel and Exit](#) [Back](#) [Create VPC](#)

- [Create VPC]를 선택합니다.

VPC의 Amazon RDS DB 인스턴스는 다중 AZ 배포를 지원하기 위해 적어도 두 개의 프라이빗 서브넷 또는 적어도 두 개의 퍼블릭 서브넷이 있어야 합니다. 다중 가용 영역 작업에 대한 자세한 내용은 [리전 및 가용 영역 \(p. 99\)](#) 단원을 참조하십시오. 데이터베이스가 프라이빗이므로 두 번째 프라이빗 서브넷을 VPC에 추가합니다.

서브넷을 추가로 생성하는 방법

- AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
- AWS Management 콘솔 오른쪽 상단 모서리에서 VPC를 위한 올바른 AWS 리전에 있음을 확인합니다.
- 좌측 상단 모서리에서 VPC Dashboard를 선택한 다음 Subnets를 선택하고 Create Subnet을 선택합니다.

4. 서브넷 생성 페이지에서 다음 값을 설정합니다.

옵션	값
[Name tag]	예를 들어 두 번째 프라이빗 서브넷의 이름은 subnet-private-2 입니다.
VPC	VPC입니다(예: vpc-1).
가용 영역	AWS 리전의 가용 영역을 선택합니다. Note 첫 번째 프라이빗 서브넷에 대해 선택한 것과 다른 가용 영역을 선택합니다.
CIDR block	10.0.2.0/24

5. [Yes, Create]를 선택합니다.

2개의 프라이빗 서브넷이 모두 동일한 라우팅 테이블을 사용해야 합니다. 다음 절차에서는 라우팅 테이블 일치를 확인하고, 일치하지 않으면 그중 하나를 편집합니다.

서브넷이 동일한 라우팅 테이블을 사용하도록 하려면,

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. AWS Management 콘솔 오른쪽 상단 모서리에서 VPC를 위한 올바른 AWS 리전에 있음을 확인합니다.
3. 왼쪽 상단 모서리에서 VPC Dashboard(VPC 대시보드)를 선택하고, Subnets(서브넷)을 선택한 다음, 첫 번째 프라이빗 서브넷(예: **subnet-private-1**)을 선택합니다.
4. 콘솔 하단에서 다음과 같이 Route Table 탭을 선택합니다.

Route Table: rtb-2cc33354	
Destination	Target
10.0.0.0/16	local
0.0.0.0/0	[Blurred]

5. 라우팅 테이블을 적어둡니다(예: rtb-0d9fc668).
6. 서브넷 목록에서 두 번째 프라이빗 서브넷(예: **subnet-private-2**)을 선택합니다.
7. 콘솔 하단에서 Route Table 탭을 선택합니다.
8. 두 번째 서브넷의 라우팅 테이블과 첫 번째 서브넷의 라우팅 테이블이 동일하지 않은 경우 일치하도록 편집합니다:
 - a. [Edit]를 선택합니다.
 - b. Change to에서 첫 번째 서브넷에 일치하는 라우팅 테이블을 선택합니다.
 - c. Save를 선택합니다.

보안 그룹은 DB 인스턴스에 대한 인바운드 및 아웃바운드 트래픽을 제어하는 가상 방화벽 역할을 합니다. 다음 절차에서는 DB 인스턴스를 위한 보안 그룹을 만듭니다. 보안 그룹에 대한 자세한 내용은 [Security Groups for Your VPC](#)를 참조하십시오.

프라이빗 Amazon RDS DB 인스턴스에 대한 VPC 보안 그룹을 생성하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. AWS Management 콘솔 오른쪽 상단 모서리에서 VPC를 위한 올바른 AWS 리전에 있음을 확인합니다.
3. 좌측 상단 모서리에서 VPC Dashboard를 선택하고, Security Groups를 선택한 다음 Create Security Group을 선택합니다.
4. Create Security Group(보안 그룹 생성) 페이지에서 다음 값을 설정합니다.

옵션	값
[Name tag]	보안 그룹의 이름입니다(예: sgdb-1).
Group name(그룹 이름)	보안 그룹의 이름입니다(예: sgdb-1).
설명	보안 그룹 설명.
VPC	VPC입니다(예: vpc-1).

5. [Yes, Create]를 선택합니다.

다음 절차에서는 DB 인스턴스에 대한 인바운드 트래픽을 제어하는 규칙을 보안 그룹에 추가합니다. 인바운드 규칙에 대한 자세한 내용은 [Security Group Rules](#) 단원을 참조하십시오.

인바운드 규칙을 보안 그룹에 추가하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. AWS Management 콘솔 오른쪽 상단 모서리에서 VPC를 위한 올바른 AWS 리전에 있음을 확인합니다.
3. 왼쪽 상단 모서리에서 VPC Dashboard(VPC 대시보드)를 선택하고, Security Groups(보안 그룹)를 선택한 다음, 보안 그룹(예: **sgdb-1**)을 선택합니다.
4. 콘솔 하단에서 Inbound Rules 탭을 선택한 다음, Edit를 선택합니다.
5. 다음과 같이 이러한 값을 설정합니다.

옵션	값
유형	oracle (1521)
프로토콜	TCP (6)
포트 범위	1521
소스	보안 그룹의 식별자. 상자를 선택하면 보안 그룹의 이름(예: sgdb-1)이 보입니다.



6. Save를 선택합니다.

Oracle DB 인스턴스 생성

Amazon RDS를 사용하여 Oracle DB 인스턴스를 호스팅할 수 있습니다. 다음 절차에서는 Oracle DB 인스턴스를 만들습니다.

Oracle DB 인스턴스를 실행하는 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. AWS Management 콘솔의 오른쪽 상단 모서리에서 DB 인스턴스의 AWS 리전을 선택합니다. VPC와 동일한 AWS 리전을 선택합니다.
3. 데이터베이스를 선택한 후 데이터베이스 생성을 선택합니다.
4. [Select engine] 페이지에서 [Oracle]을 선택한 후 [Oracle Database Enterprise Edition]을 선택합니다.

Select engine

Engine options

- Amazon Aurora
Amazon Aurora
- MySQL

- MariaDB

- PostgreSQL

- Oracle
ORACLE
- Microsoft SQL Server


Oracle

Edition

Oracle Enterprise Edition
Efficient, reliable, and secure database management system that delivers comprehensive high-end capabilities for mission-critical applications and demanding database workloads.

Oracle Standard Edition
Affordable and full-featured database management system supporting up to 32 vCPUs.

Oracle Standard Edition One
Affordable and full-featured database management system supporting up to 16 vCPUs.

Oracle Standard Edition Two
Affordable and full-featured database management system supporting up to 16 vCPUs.
Oracle Database Standard Edition Two is a replacement for Standard Edition and Standard Edition One.

Only enable options eligible for RDS Free Usage Tier [info](#)

[Cancel](#) [Next](#)

5. [Next]를 선택합니다.
6. [Choose use case] 페이지에서 [Production]을 선택한 후 [Next]를 선택합니다.

Note

개발 및 테스팅용 DB 인스턴스인 경우, [Dev/Test]를 선택할 수 있습니다.

7. DB 세부 정보 지정 페이지에서 다음과 같이 다음 값을 설정합니다.

옵션	값
라이선스 모델	bring-your-own-license

옵션	값
DB 엔진 버전	사용할 Oracle 버전입니다. 최신 Oracle 12c 버전을 사용합니다.
DB 인스턴스 클래스	사용할 DB 인스턴스 클래스입니다. 자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 단원을 참조하십시오.
다중 AZ 배포	<p>Create replica in different zone를 선택합니다. 다중 AZ 배포에서는 장애 조치를 위해 다른 가용 영역에 DB 인스턴스의 예비 복제본을 생성합니다. 프로덕션 워크로드에는 다중 AZ를 권장합니다. 다중 가용 영역에 대한 자세한 내용은 리전 및 가용 영역 (p. 99) 단원을 참조하십시오.</p> <p>Note</p> <p>개발 및 테스팅을 위해 No를 선택할 수 있습니다.</p>
스토리지 유형	<p>Provisioned IOPS (SSD) 프로덕션 워크로드를 위해서는 Provisioned IOPS(초당 입/출력 작업 수)를 권장합니다. 스토리지에 대한 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.</p> <p>Note</p> <p>개발 및 테스팅을 위해 General Purpose (SSD)를 선택할 수 있습니다.</p>
할당된 스토리지	데이터베이스에 할당할 스토리지입니다. 데이터베이스에 적어도 스토리지 20GiB를 할당합니다. 경우에 따라 DB 인스턴스에 대해 데이터베이스의 크기보다 많은 양의 스토리지를 할당하면 I/O 성능을 개선할 수 있습니다. 스토리지 할당에 대한 자세한 내용은 Amazon RDS 스토리지 유형 (p. 101) 및 Guidelines for Creating Oracle Database Tablespaces 를 참조하십시오.
프로비저닝된 IOPS	DB 인스턴스용으로 처음에 할당할 프로비저닝된 IOPS의 양입니다. 이 값은 DB 인스턴스용 스토리지 양의 3배에서 10배 사이여야 합니다. 또한 이 값은 1,000의 정수 배수여야 합니다.
	<p>Note</p> <p>개발 및 테스팅을 위해서는 프로비저닝된 IOPS가 필요 없습니다.</p>
DB 인스턴스 식별자	DB 인스턴스의 이름입니다(예: oracle-instance).
Master username	DB 인스턴스의 마스터 사용자 이름입니다(예: oracle_mu).
Master password 및 Confirm password	마스터 사용자 암호로 사용할 인쇄 가능한 ASCII 문자(/, " 및 @ 제외) 8 - 30자를 포함하는 암호입니다. [Confirm Password] 상자에 암호를 다시 입력합니다.

Specify DB details

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](#)

DB engine version [info](#)

DB instance class [info](#)

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](#)

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

Provisioned IOPS [info](#)

Settings

DB instance identifier [info](#)
Specify a name that is unique for all DB instances owned by your AWS account in the current region.

DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance".

Master username [info](#)
Specify an alphanumeric string that defines the login ID for the master user.

Master Username must start with a letter.

Master password [info](#)

Confirm password [info](#)

Master Password must be at least eight characters long, as in "mypassword".

8. [Next]를 선택합니다.
9. 고급 설정 구성 페이지에서 다음과 같이 다음 값을 설정합니다.

옵션	값
Virtual Private Cloud(VPC)	VPC입니다(예: vpc-1).
서브넷 그룹	Create new DB Subnet Group
퍼블릭 액세스 가능성	No
[Availability zone]	[No Preference]
VPC 보안 그룹	Select existing VPC security groups(기존 VPC 보안 그룹 선택)를 선택한 다음, VPC 보안 그룹을 선택합니다(예: sgdb-1).
데이터베이스 이름	예를 들어 데이터베이스 이름은 db1 입니다.
데이터베이스 포트	1521
DB 파라미터 그룹	기본 파라미터 그룹입니다.
옵션 그룹	기본 옵션 그룹입니다.
스냅샷으로 태그 복사	이 옵션을 선택하면 스냅샷을 만들 때 DB 인스턴스 태그를 DB 스냅샷에 복사하게 됩니다. 자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.
Character set name	DB 인스턴스에 대한 문자 세트입니다. 기본값 AL32UTF8은 유니코드 5.0 UTF-8 범용 문자 세트입니다. DB 인스턴스를 만든 후에는 문자 세트를 변경할 수 없습니다.
Enable encryption	Enable Encryption(암호 활성화) 또는 Disable Encryption . [Enable Encryption] 값은 이 DB 인스턴스에 대해 비활성화되어 있는 암호화를 활성화하며, 마스터 키를 선택할 수 있습니다. 자세한 내용은 Amazon RDS 리소스 암호화 (p. 385) 단원을 참조하십시오.
백업 보존 기간	데이터베이스의 자동 백업을 지속할 기간(일)을 설정합니다. 대부분의 DB 인스턴스에서는 이 값을 1 이상으로 설정해야 합니다.
백업 기간	데이터베이스를 백업할 특정 시간을 지정하지 않으려면 기본값으로 No Preference를 사용합니다.
확장 모니터링	DB 인스턴스가 실행되는 운영 체제에 대한 실시간 측정치를 수집하려면 [Enable enhanced monitoring]을 선택합니다. 자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.
마이너 버전 자동 업그레이드	기본 마이너 DB 엔진 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다. Amazon RDS는 유지 관리 기간에 마이너 버전 자동 업그레이드를 수행합니다.
유지 관리 기간	기간 선택을 선택하고 대기 중인 DB 인스턴스의 설정 변경이 적용되는 시간인 30분을 선택합니다. 이 시간이 중요하지 않은 경우 기본 설정 없음을 선택합니다.

10. 마법사 마지막 페이지에서 [View DB instance details]를 선택합니다.

RDS 콘솔에 새 DB 인스턴스의 세부 정보가 표시됩니다. DB 인스턴스를 만들고 사용할 준비가 될 때까지 DB 인스턴스의 상태는 [creating]입니다. 상태가 available로 변경되면 DB 인스턴스에 연결할 수 있습니다.

DB 인스턴스 클래스와 할당된 스토리지에 따라 새 인스턴스를 사용할 수 있을 때까지 몇 분 정도 걸릴 수 있습니다.

추가 Amazon RDS 인터페이스

앞의 절차에서는 AWS Management 콘솔을 사용하여 작업을 수행합니다. Amazon Web Services는 AWS Command Line Interface(AWS CLI) 및 애플리케이션 프로그래밍 인터페이스(API)도 제공합니다. AWS CLI 또는 API를 사용하여 Amazon RDS를 통한 Oracle DB 인스턴스 관리 작업을 포함한 대부분의 Amazon RDS 관리 작업을 자동화할 수 있습니다.

자세한 내용은 [Amazon RDS에 대한 AWS Command Line Interface 참조](#) 및 [Amazon RDS API Reference 단원](#)을 참조하십시오.

관련 주제

- [Amazon RDS 설정 \(p. 5\)](#)
- [Oracle용 Amazon RDS에서 Oracle Repository Creation Utility 사용 \(p. 895\)](#)
- [Amazon RDS에서 Oracle에 Siebel Database 설치 \(p. 899\)](#)
- [VPC에서 DB 인스턴스에 액세스하는 시나리오 \(p. 408\)](#)
- [Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 \(p. 736\)](#)

Amazon RDS와 함께 Oracle GoldenGate 사용하기

Oracle GoldenGate(GoldenGate)는 데이터베이스 간 트랜잭션 데이터를 수집, 복제, 관리하는 데 사용됩니다. 온라인 트랜잭션 처리(OLTP) 시스템에 Oracle 데이터베이스와 함께 사용되는 로그 기반 변경 데이터 캡처(CDC) 및 복제 소프트웨어 패키지입니다. GoldenGate는 원본 데이터베이스에서 가장 최근에 변경된 데이터를 포함하는 주적 파일을 생성한 후 이 파일을 대상 데이터베이스로 푸시합니다. 활성/활성 데이터베이스 복제, 제로 중단 시간 마이그레이션 및 업그레이드, 재해 복구, 데이터 보호, 리전 내/리전 간 복제를 위해 Amazon RDS에서 GoldenGate를 사용할 수 있습니다.

다음은 Amazon RDS에서 GoldenGate로 작업할 때 알아 두어야 할 중요한 사항입니다.

- 사용자는 Amazon RDS와 함께 사용하기 위해 GoldenGate를 설정 및 관리해야 합니다.
- 사용자는 모든 AWS 리전에서 Amazon RDS와 함께 사용하기 위해 GoldenGate 라이선싱(기준 보유 라이선스 사용)을 관리해야 합니다. 자세한 내용은 [Oracle 라이선싱 \(p. 704\)](#) 단원을 참조하십시오.
- Amazon RDS는 Oracle Database Standard Edition Two(SE2), Standard Edition One(SE1), Standard Edition(SE), Enterprise Edition(EE)에서 GoldenGate를 지원합니다.
- Amazon RDS는 GoldenGate 데이터베이스 버전 11.2.0.4, 12.1.0.2 또는 12.2.0.1을 지원합니다.
- Amazon RDS는 Oracle GoldenGate 버전 12.1, 12.2, 12.3을 포함하여 11.2.1 이상 버전을 지원합니다.
- Amazon RDS는 GoldenGate를 사용하여 Oracle 데이터베이스 간 마이그레이션 및 복제를 지원합니다. Amazon에서는 이기종 데이터베이스 간의 마이그레이션이나 복제를 지원하거나 억제하지 않습니다.
- Oracle Transparent Data Encryption(TDE)을 사용하는 Amazon RDS Oracle DB 인스턴스에서 GoldenGate를 사용할 수 있습니다. 복제된 데이터의 무결성을 유지 관리하려면 EBS 암호화된 볼륨 또는 주적 파일 암호화를 사용하여 GoldenGate 허브에서 암호화를 구성해야 합니다. 또한 GoldenGate 허브와 소스 및 대상 데이터베이스 인스턴스 간에 전송되는 데이터에 대한 암호화를 구성해야 합니다. Amazon RDS DB 인스턴스는 [Oracle 보안 소켓 Layer \(p. 810\)](#) 또는 [Oracle 기본 네트워크 암호화 \(p. 807\)](#)을 통하여 암호화를 지원합니다.
- GoldenGate DDL은 통합 캡처 모드를 사용할 때 GoldenGate 버전 12.1 이상에서 지원됩니다.

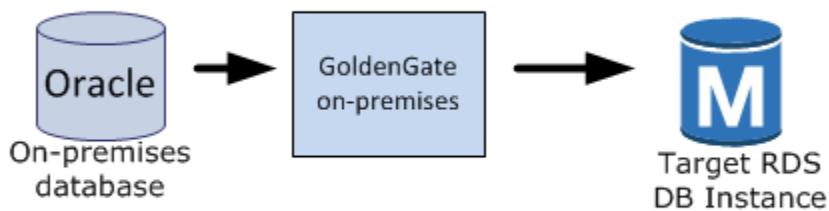
개요

Amazon RDS용 GoldenGate 아키텍처는 세 가지 개별 모듈로 구성됩니다. 원본 데이터베이스는 온프레미스 Oracle 데이터베이스, EC2 인스턴스의 Oracle 데이터베이스 또는 Amazon RDS DB 인스턴스의 Oracle

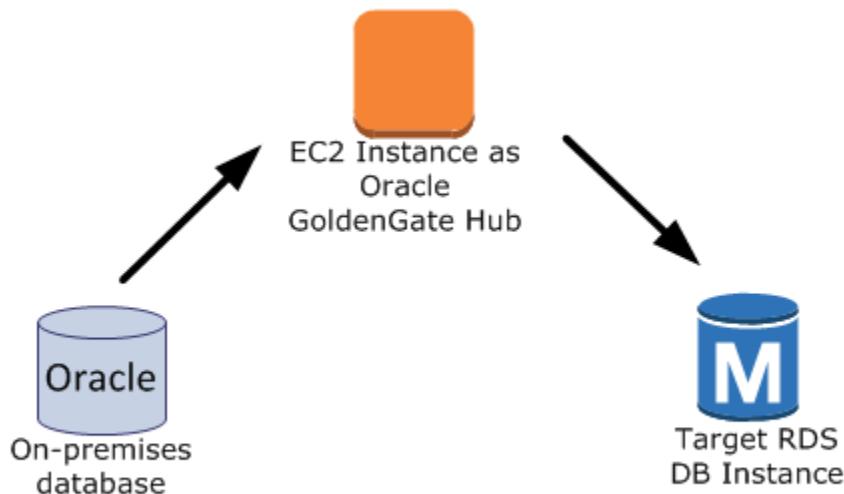
데이터베이스입니다. 이어서, 원본 데이터베이스에서 대상 데이터베이스로 트랜잭션 정보를 이동하는 GoldenGate 허브는 Oracle Database 11.2.0.4, GoldenGate 11.2.1 또는 온프레미스 Oracle 제품이 설치된 EC2 인스턴스입니다. 두 개 이상의 EC2 허브가 있을 수 있으며, 리전 간 복제 시 GoldenGate를 사용하는 경우 허브를 두 개 사용하는 것이 좋습니다. 마지막으로, 대상 데이터베이스는 Amazon RDS DB 인스턴스, EC2 인스턴스 또는 온프레미스 위치에 있습니다.

Amazon RDS에 대한 GoldenGate는 다음과 같은 일반 시나리오를 지원합니다.

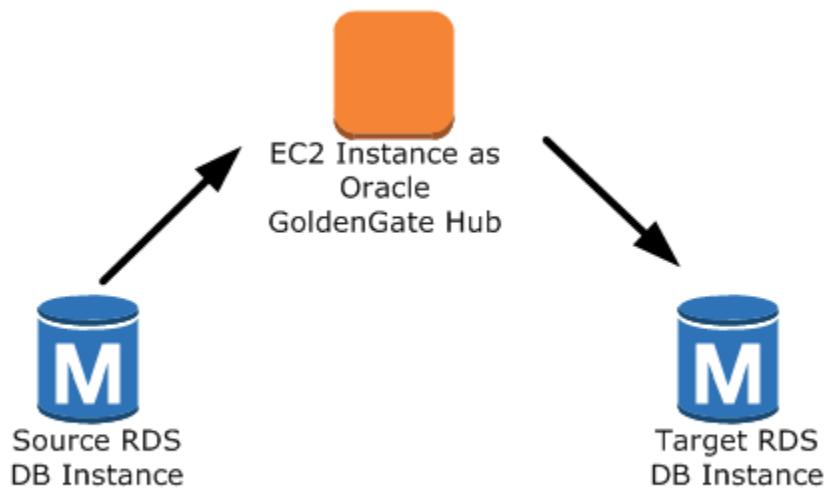
시나리오 1: 대상 Amazon RDS DB 인스턴스에 데이터를 제공하는 온프레미스 Oracle 원본 데이터베이스 및 온프레미스 GoldenGate 허브.



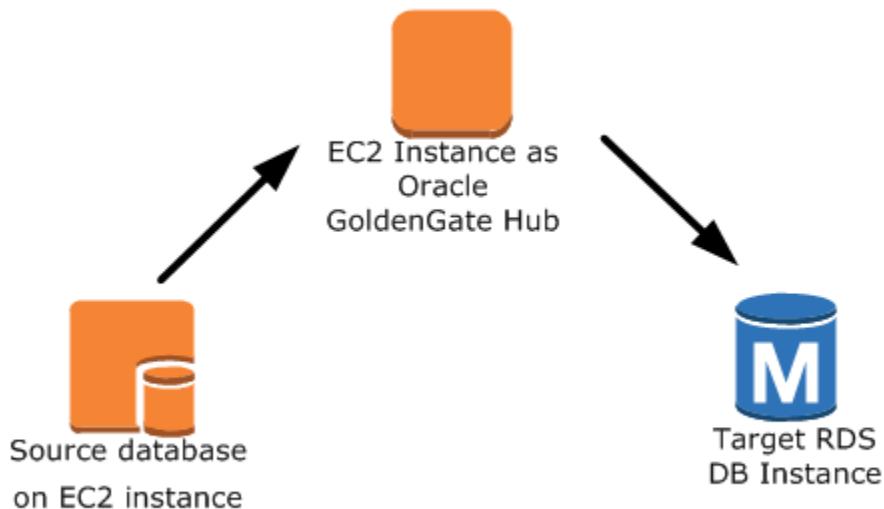
시나리오 2: 대상 Amazon RDS DB 인스턴스에 데이터를 제공하는 Amazon EC2 인스턴스 허브에 연결된 원본 데이터베이스 역할을 하는 온프레미스 Oracle 데이터베이스.



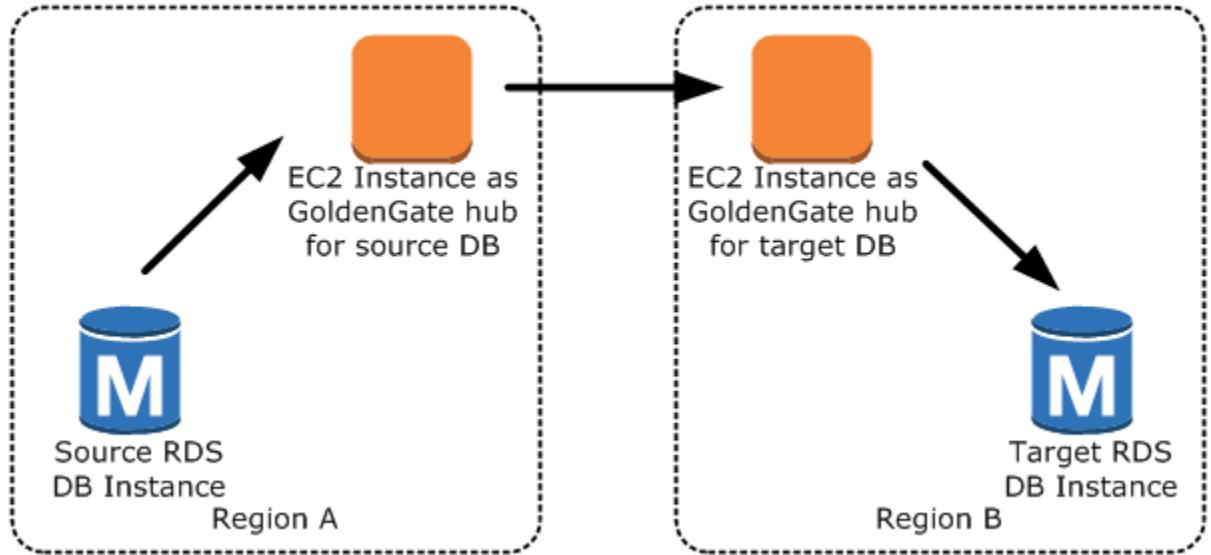
시나리오 3: 대상 Amazon RDS DB 인스턴스에 데이터를 제공하는 Amazon EC2 인스턴스 허브에 연결된 원본 데이터베이스 역할을 하는 Amazon RDS DB 인스턴스의 Oracle 데이터베이스.



시나리오 4: 대상 Amazon RDS DB 인스턴스에 데이터를 제공하는 Amazon EC2 인스턴스 허브에 연결된 원본 데이터베이스 역할을 하는 Amazon EC2 DB 인스턴스의 Oracle 데이터베이스.



시나리오 5: 두 번째 EC2 인스턴스 허브와 같은 리전의 대상 Amazon EC2 DB 인스턴스에 데이터를 제공하는 같은 리전의 Amazon RDS 인스턴스 허브에 연결되거나 다른 리전의 Amazon RDS 인스턴스 허브에 연결된 Amazon EC2 DB 인스턴스의 Oracle 데이터베이스



Note

온프레미스 환경의 GoldenGate 실행에 영향을 미치는 문제가 AWS에서 GoldenGate를 실행하는 데도 영향을 미칩니다. GoldenGate 허브를 모니터링하여 장애 조치가 발생하는 경우 EXTRACT와 REPLICAT가 재개되는지 확인하는 것이 좋습니다. GoldenGate 허브가 Amazon EC2 인스턴스에서 실행되어 Amazon RDS에서 GoldenGate 허브를 관리하지 않으므로 이 허브가 실행 중인지 확인할 수 없습니다.

Amazon RDS에서 GoldenGate를 사용하여 Oracle의 메이저 버전으로 업그레이드할 수 있습니다. 예를 들어, Amazon RDS에서 GoldenGate를 사용하여 Oracle 버전 8 온프레미스 데이터베이스를 Amazon RDS DB 인스턴스에서 버전 11.2.0.4를 실행하는 Oracle 데이터베이스로 업그레이드할 수 있습니다.

Amazon RDS를 사용하는 GoldenGate를 설치하려면 EC2 인스턴스에서 허브를 구성한 후 원본 및 대상 데이터베이스를 구성합니다. 다음 단계에서는 Amazon RDS용으로 GoldenGate를 설치하는 방법을 보여 줍니다. 각 단계는 다음 섹션에서 자세히 설명합니다.

- [EC2에 GoldenGate 허브 설정 \(p. 888\)](#)
- [Amazon RDS에서 GoldenGate와 함께 사용할 원본 데이터베이스 설치 \(p. 889\)](#)
- [Amazon RDS에서 GoldenGate와 함께 사용할 대상 데이터베이스 설치 \(p. 892\)](#)
- [GoldenGate의 EXTRACT 및 REPLICAT 유ти리티로 작업 \(p. 892\)](#)

EC2에 GoldenGate 허브 설정

Amazon EC2 인스턴스에 GoldenGate 허브를 생성하는 몇 가지 단계가 있습니다. 먼저, Oracle DBMS 11g 버전 11.2.0.4 전체가 설치된 EC2 인스턴스를 생성합니다. 이 EC2 인스턴스에는 GoldenGate 11.2.1 소프트웨어와 Oracle 패치 13328193도 설치해야 합니다. GoldenGate 설치에 대한 자세한 내용은 [Oracle 설명서](#) 단원을 참조하십시오.

GoldenGate 허브 역할을 하는 EC2 인스턴스가 원본 데이터베이스의 트랜잭션 정보를 트레일 파일에 저장하고 처리하므로 트레일 파일을 저장할 수 있는 충분한 스토리지를 할당해야 합니다. 또한 EC2에는 처리되는 데이터의 양을 관리할 수 있는 충분한 처리 용량과 트레일 파일에 기록되기 전 트랜잭션 정보를 저장할 수 있는 충분한 메모리도 할당해야 합니다.

다음 작업에서는 Amazon EC2 인스턴스에 GoldenGate 허브를 설치합니다. 각 작업은 이 단원에서 자세히 설명합니다. 작업은 다음과 같습니다.

- tnsname.ora 파일에 별칭 추가
- GoldenGate 하위 디렉터리 생성
- GLOBALS 파라미터 파일 업데이트
- mgr.prm 파일 구성 및 관리자 시작

다음 항목을 tnsname.ora 파일에 추가하여 별칭을 생성합니다. tnsname.ora 파일에 대한 자세한 내용은 [Oracle 설명서](#)를 참조하십시오.

```
$ cat /example/config/tnsnames.ora
TEST=
(DESCRIPTION=
  (ENABLE=BROKEN)
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-test.abcdef12345.us-west-2.rds.amazonaws.com)
     (PORT=8200))
  )
  (CONNECT_DATA=
    (SID=ORCL)
  )
)
```

다음으로 EC2 명령줄 셸과 ggsci(GoldenGate 명령 인터프리터)를 사용하여 GoldenGate 디렉터리에서 하위 디렉터리를 생성합니다. gg 디렉터리 아래에 하위 디렉터리가 생성되며 이러한 하위 디렉터리에는 파라미터, 보고서 및 체크포인트 파일에 대한 디렉터리가 포함됩니다.

```
prompt$ cd /gg
prompt$ ./ggsci
GGSCI> CREATE SUBDIRS
```

EC2 명령줄 셸을 사용하여 GLOBALS 파라미터 파일을 생성합니다. 모든 GoldenGate 프로세스에 영향을 주는 파라미터는 GLOBALS 파라미터 파일에 정의되어 있습니다. 다음 예제에서는 필요한 파일을 생성합니다.

```
$ cd $GGHOME
$ vi GLOBALS
CheckpointTable oggadm1.oggchkpt
```

GoldenGate 허브 설치 및 구성의 마지막 단계는 관리자를 구성하는 것입니다. 다음 줄을 mgr.prm 파일에 추가하고 ggsci를 사용하여 관리자를 시작합니다.

```
PORT 8199
PurgeOldExtracts ./dirdat/*, UseCheckpoints, MINKEEPDAYS 5
```

```
GGSCI> start mgr
```

이러한 단계를 완료했으면 GoldenGate 허브를 사용할 수 있습니다. 다음으로 원본 및 대상 데이터베이스를 설치합니다.

Amazon RDS에서 GoldenGate와 함께 사용할 원본 데이터베이스 설치

원본 데이터베이스에서 버전 11.2.0.4 이상을 실행 중인 경우 GoldenGate에 사용할 원본 데이터베이스를 설치하기 위해 수행할 작업에는 세 가지가 있습니다.

- compatible 파라미터를 11.2.0.4 이상으로 설정합니다.

- ENABLE_GOLDENGATE_REPLICATION 파라미터를 True로 설정합니다. 이 파라미터는 원본 데이터베이스에 대한 보충 로깅을 사용하도록 설정합니다. 원본 데이터베이스가 Amazon RDS DB 인스턴스에 있는 경우 파라미터 그룹을 DB 인스턴스에 배정하고 ENABLE_GOLDENGATE_REPLICATION 파라미터를 true로 설정해야 합니다. ENABLE_GOLDENGATE_REPLICATION 파라미터에 대한 자세한 내용은 [Oracle 설명서](#)를 참조하십시오.
- GoldenGate 원본 데이터베이스에 대한 보관된 다시 실행 로그의 보존 기간을 설정합니다.
- 원본 데이터베이스에서 GoldenGate 사용자 계정을 생성합니다.
- GoldenGate 사용자에게 필요한 권한을 부여합니다.

원본 데이터베이스에서 compatible 파라미터를 11.2.0.4 이상으로 설정해야 합니다. Amazon RDS DB 인스턴스의 Oracle 데이터베이스를 원본 데이터베이스로 사용하려는 경우 compatible 파라미터가 11.2.0.4 이상으로 설정된 파라미터 그룹을 DB 인스턴스와 연결해야 합니다. DB 인스턴스와 연결된 파라미터 그룹에서 compatible 파라미터를 변경하는 경우 변경 내용이 적용되려면 인스턴스를 재부팅해야 합니다. 다음과 같은 Amazon RDS CLI 명령을 사용하여 새 파라미터 그룹을 생성하고 compatible 파라미터를 설정할 수 있습니다. 원본 DB 인스턴스와 새 파라미터 그룹을 연결해야 합니다.

Linux, OS X, Unix의 경우:

```
aws rds create-db-parameter-group \
--db-parameter-group-name example-goldengate \
--description "Parameters to allow GoldenGate" \
--db-parameter-group-family oracle-ee-11.2

aws rds modify-db-parameter-group \
--db-parameter-group-name example-goldengate \
--parameters "ParameterName=compatible, ParameterValue=11.2.0.4, ApplyMethod=pending-reboot"

aws rds modify-db-instance \
--db-instance-identifier example-test \
--db-parameter-group-name example-goldengate \
--apply-immediately

aws rds reboot-db-instance \
--db-instance-identifier example-test
```

Windows의 경우:

```
aws rds create-db-parameter-group ^
--db-parameter-group-name example-goldengate ^
--description "Parameters to allow GoldenGate" ^
--db-parameter-group-family oracle-ee-11.2

aws rds modify-db-parameter-group ^
--db-parameter-group-name example-goldengate ^
--parameters "ParameterName=compatible, ParameterValue=11.2.0.4, ApplyMethod=pending-reboot"

aws rds modify-db-instance ^
--db-instance-identifier example-test ^
--db-parameter-group-name example-goldengate ^
--apply-immediately

aws rds reboot-db-instance ^
--db-instance-identifier example-test
```

항상 파라미터 그룹을 compatible 파라미터와 함께 유지하십시오. DB 스냅샷에서 인스턴스를 복원하는 경우 compatible 파라미터 값 이상의 지정된 파라미터 그룹을 사용하도록 복원된 인스턴스를 수정해야 합니다. 이 작업은 복원 작업 후 가능한 빨리 수행해야 하며 인스턴스를 재부팅해야 합니다.

ENABLE_GOLDENGATE_REPLICATION 파라미터를 True로 설정하는 경우 원본 데이터베이스에 대한 보증 로깅이 사용되며 필요한 GoldenGate 권한이 구성됩니다. 원본 데이터베이스가 Amazon RDS DB 인스턴스에 있는 경우 파라미터 그룹을 DB 인스턴스에 배정하고 ENABLE_GOLDENGATE_REPLICATION 파라미터를 true로 설정해야 합니다. ENABLE_GOLDENGATE_REPLICATION 파라미터에 대한 자세한 내용은 [Oracle 설명서](#)를 참조하십시오.

또한 원본 데이터베이스에서 보관된 다시 실행 로그가 유지되어야 합니다. 예를 들어, 다음 명령은 보관된 다시 실행 로그의 보존 기간을 24시간으로 설정합니다.

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours', 24);
```

로그 보존 기간은 시간 단위로 지정됩니다. GoldenGate가 필요에 따라 원본 인스턴스에서 로그를 복구할 수 있도록, 이 기간은 원본 인스턴스에 대한 잠재적 중단 시간이나 잠재적 통신/네트워킹 문제가 발생할 가능성 있는 기간보다 길어야 합니다. 로그 보존 기간으로 필요한 절대 최소값은 1시간입니다.

로그 보존 설정이 너무 작으면 다음 메시지가 발생합니다.

```
ERROR OGG-02028 Failed to attach to logmining server OGG$<extract_name> error 26927 -  
ORA-26927: altering an outbound server with a remote capture is not allowed.
```

이러한 로그는 DB 인스턴스에 보존되기 때문에, 로그 파일을 수용하기 위해 인스턴스에서 충분한 스토리지를 사용할 수 있는지 확인해야 합니다. 지난 "X"시간 동안 사용한 공간의 양을 확인하려면 다음 쿼리를 사용하고 "X"는 시간으로 바꾸십시오.

```
select sum(blocks * block_size) bytes from v$archived_log  
where next_time>=sysdate-X/24 and dest_id=1;
```

GoldenGate는 데이터베이스 사용자로 실행되며 원본 데이터베이스에 대한 다시 실행 및 보관 로그에 액세스하려면 GoldenGate에 적절한 데이터베이스 권한이 있어야 하므로, 원본 데이터베이스에서 GoldenGate 사용자 계정을 생성해야 합니다. GoldenGate 사용자 계정 권한에 대한 자세한 내용은 [Oracle 설명서](#)의 섹션 4, 섹션 4.4 및 표 4.1을(를) 참조하십시오.

다음 문은 oggadm1이라는 사용자 계정을 생성합니다.

```
CREATE tablespace administrator;  
CREATE USER oggadm1 IDENTIFIED BY "XXXXXX"  
    default tablespace ADMINISTRATOR temporary tablespace TEMP;
```

마지막으로 필요한 권한을 GoldenGate 사용자 계정에 부여합니다. 다음 문은 oggadm1이라는 사용자에게 권한을 부여합니다.

```
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 instance master username> 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;  
  
EXEC DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE (grantee=>'OGGADM1',  
privilege_type=>'capture',  
grant_select_privileges=>true,  
do_grants=>TRUE);
```

Amazon RDS에서 GoldenGate와 함께 사용할 대상 데이터베이스 설치

다음 작업은 GoldenGate를 사용할 대상 DB 인스턴스를 설정합니다.

- compatible 파라미터를 11.2.0.4 이상으로 설정합니다.
- ENABLE_GOLDENGATE_REPLICATION 파라미터를 True로 설정합니다. 대상 데이터베이스가 Amazon RDS DB 인스턴스에 있을 경우 ENABLE_GOLDENGATE_REPLICATION 파라미터가 true로 설정된 DB 인스턴스에 지정된 파라미터 그룹이 있어야 합니다. ENABLE_GOLDENGATE_REPLICATION 파라미터에 대한 자세한 내용은 [Oracle 설명서](#)을(를) 참조하십시오.
- 대상 데이터베이스에서 GoldenGate 사용자 계정 생성 및 관리
- GoldenGate 사용자에게 필요한 권한 부여

GoldenGate는 데이터베이스 사용자로 실행되며 적절한 데이터베이스 권한이 부여되어야 하므로, 대상 데이터베이스에서 GoldenGate 사용자 계정을 생성해야 합니다. 다음 문은 oggadm1이라는 사용자를 생성합니다.

```
create tablespace administrator;
create tablespace administrator_idx;
CREATE USER oggadm1 IDENTIFIED BY "XXXXXX"
    default tablespace ADMINISTRATOR
    temporary tablespace TEMP;
alter user oggadm1 quota unlimited on ADMINISTRATOR;
alter user oggadm1 quota unlimited on ADMINISTRATOR_IDX;
```

마지막으로 필요한 권한을 GoldenGate 사용자 계정에 부여합니다. 다음 문은 oggadm1이라는 사용자에게 권한을 부여합니다.

```
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;

EXEC DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE
  (grantee=>'OGGADM1',privilege_type=>'apply',
  grant_select_privileges=>true, do_grants=>TRUE);
```

GoldenGate의 EXTRACT 및 REPLICAT 유틸리티로 작업

GoldenGate 유틸리티인 EXTRACT와 REPLICAT는 함께 작동하여 트레일 파일을 사용한 증분형 트랜잭션 복제를 통해 원본 및 대상 데이터베이스의 동기 상태를 유지합니다. 원본 데이터베이스에 대해 발생하는 모든 변경 내용은 EXTRACT에 의해 자동으로 감지되고 서식이 지정되어 GoldenGate 온프레미스 또는 EC2 인스

터스 허브의 트레일 파일로 전송됩니다. 초기 로드가 완료된 후에는 REPLICAT 유ти리티에서 이러한 파일의 데이터를 읽어 대상 데이터베이스로 복제합니다.

GoldenGate의 EXTRACT 유ти리티 실행

EXTRACT 유ти리티는 원본 데이터베이스에서 데이터를 검색 및 변환하고 트레일 파일로 출력합니다. EXTRACT는 트랜잭션 세부 정보를 메모리나 임시 디스크 스토리지의 대기열에 저장합니다. 트랜잭션이 원본 데이터베이스로 커밋되는 경우, EXTRACT는 라우팅을 위해 모든 트랜잭션 세부 정보를 GoldenGate 온프레미스 또는 EC2 인스턴스 허브로 플러시하고 대상 데이터베이스로 플러시합니다.

다음 작업에서는 EXTRACT 유ти리티를 활성화하고 시작합니다.

- GoldenGate 허브(온프레미스 또는 EC2 인스턴스)에서 EXTRACT 파라미터 파일을 구성합니다. 다음 목록은 예제 EXTRACT 파라미터 파일을 보여 줍니다.

```
EXTRACT EABC
SETENV (ORACLE_SID=ORCL)
SETENV (NLSLANG=AL32UTF8)

USERID oggadm1@TEST, PASSWORD XXXXXX
EXTTRAIL /path/to/goldengate/dirdat/ab

IGNOREREPPLICATES
GETAPPLOPS
TRANLOGOPTIONS EXCLUDEUSER OGGADM1

TABLE EXAMPLE.TABLE;
```

- GoldenGate 허브에서 GoldenGate 명령줄 인터페이스(ggsci)를 시작합니다. 원본 데이터베이스에 로그인합니다. 다음 예제에서는 로그인 형식을 보여 줍니다.

```
dblogin userid <user>@<db tnsname>
```

- 데이터베이스의 체크포인트 테이블을 추가합니다.

```
add checkpointtable
```

- 데이터베이스 테이블에 대한 보충 로깅을 사용하기 위해 트랜잭션 데이터를 추가합니다.

```
add trandata <user>.<table>
```

또는 데이터베이스의 모든 테이블에 대해 보충 로깅을 사용하기 위해 트랜잭션 데이터를 추가할 수도 있습니다.

```
add trandata <user>.*
```

- ggsci 명령줄에서 다음 명령을 사용하여 EXTRACT 유ти리티를 활성화합니다.

```
add extract <extract name> tranlog, INTEGRATED tranlog, begin now
add exttrail <path-to-trail-from-the param-file>
    extract <extractname-from-paramfile>,
    MEGABYTES Xm
```

- EXTRACT 유ти리티를 데이터베이스에 등록하여 보관 로그가 삭제되지 않도록 합니다. 이렇게 하면 필요할 경우 이전의 커밋되지 않은 트랜잭션을 복구할 수 있습니다. EXTRACT 유ти리티를 데이터베이스에 등록하려면 다음 명령을 사용합니다.

```
register EXTRACT <extract process name>, DATABASE
```

- EXTRACT 유ти리티를 시작하려면 다음 명령을 사용합니다.

```
start <extract process name>
```

GoldenGate의 REPLICAT 유ти리티 실행

REPLICAT 유ти리티는 트레일 파일의 트랜잭션 정보를 대상 데이터베이스로 "푸시"하는 데 사용됩니다.

다음 작업에서는 REPLICAT 유ти리티를 활성화하고 시작합니다.

- GoldenGate 허브(온프레미스 또는 EC2 인스턴스)에서 REPLICAT 파라미터 파일을 구성합니다. 다음 목록은 예제 REPLICAT 파라미터 파일을 보여 줍니다.

```
REPLICAT RABC
SETENV (ORACLE_SID=ORCL)
SETENV (NLSSLANG=AL32UTF8)

USERID oggadm1@TARGET, password XXXXXX

ASSUMETARGETDEFS
MAP EXAMPLE.TABLE, TARGET EXAMPLE.TABLE;
```

- GoldenGate 명령줄 인터페이스(ggsci)를 시작합니다. 대상 데이터베이스에 로그인합니다. 다음 예제에서는 로그인 형식을 보여 줍니다.

```
dblogin userid <user>@<db tnsname>
```

- ggsci 명령줄을 사용하여 체크포인트 테이블을 추가합니다. 표시된 사용자는 대상 테이블 스키마 소유자가 아니라 GoldenGate 사용자 계정이어야 합니다. 다음 예제에서는 gg_checkpoint라는 체크포인트 테이블을 생성합니다.

```
add checkpointtable <user>.gg_checkpoint
```

- REPLICAT 유ти리티를 활성화하려면 다음 명령을 사용합니다.

```
add replicat <replicat name> EXTTRAIL <extract trail file> CHECKPOINTTABLE
<user>.gg_checkpoint
```

- REPLICAT 유ти리티를 시작하려면 다음 명령을 사용합니다.

```
start <replicat name>
```

Amazon RDS에 GoldenGate 사용 시 문제 해결 문제

이 단원에서는 Amazon RDS에 GoldenGate를 사용할 때 가장 일반적으로 발생하는 문제를 설명합니다.

주제

- [로그 보존 \(p. 894\)](#)
- [GoldenGate는 제대로 구성된 것으로 보이나 복제가 작동하지 않음 \(p. 895\)](#)

로그 보존

로그 보존 기능을 활성화해야 합니다. 로그 보존 기능을 활성화하지 않거나 보존 값이 너무 작은 경우에는 다음 메시지가 나타납니다.

```
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.
```

GoldenGate는 제대로 구성된 것으로 보이나 복제가 작동하지 않음

기존 테이블의 경우 GoldenGate가 작업을 수행할 SCN을 GoldenGate에 알려 주어야 합니다. 이 문제를 해결하려면 다음과 같이 하십시오.

- GoldenGate 명령줄 인터페이스(ggsci)를 시작합니다. 원본 데이터베이스에 로그인합니다. 다음 예제에서는 로그인 형식을 보여 줍니다.

```
dblogin userid <user>@<db tnsname>
```

- ggsci 명령줄을 사용하여 EXTRACT 프로세스에 대한 시작 SCN을 설정합니다. 다음 예는 추출을 위한 SCN을 223274로 설정합니다.

```
ALTER EXTRACT <extract process name> SCN 223274  
start <extract process name>
```

- 대상 데이터베이스에 로그인합니다. 다음 예제에서는 로그인 형식을 보여 줍니다.

```
dblogin userid <user>@<db tnsname>
```

- ggsci 명령줄을 사용하여 REPLICAT 프로세스에 대한 시작 SCN을 설정합니다. 다음 예제에서는 REPLICAT을 위한 SCN을 223274로 설정합니다.

```
start <replicat process name> atcsn 223274
```

Oracle용 Amazon RDS에서 Oracle Repository Creation Utility 사용

Amazon RDS를 사용하여 Fusion Middleware 구성 요소를 지원하는 스키마를 보관하는 Oracle DB 인스턴스를 호스팅할 수 있습니다. Fusion Middleware 구성 요소를 사용하려면 데이터베이스에 이 구성 요소의 스키마를 만들어 채워야 합니다. 스키마는 Oracle RCU(Repository Creation Utility)를 사용하여 만들고 채웁니다.

Fusion Middleware 구성 요소에 대한 스키마를 Amazon RDS DB 인스턴스에 저장할 수 있습니다. 다음은 제대로 설치되도록 확인된 스키마 목록입니다.

- 분석(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)
- 포털 및 서비스(WEBCENTER)
- 포틀릿 생산자(PORTLET)
- 서비스 테이블(STB)

- SOA 인프라(SOAINFRA)
- User Messaging Service(UCSUMS)
- WebLogic Services(WLS)

라이선스 및 버전

Amazon RDS는 Oracle RCU(Repository Creation Utility) 버전 12c만 지원합니다. 다음 구성에서 RCU를 사용할 수 있습니다.

- Oracle 데이터베이스 12.2.0.1을 사용하는 RCU 12c
- Oracle Database 12.1.0.2.v4 이상을 사용하는 RCU 12c
- Oracle Database 11.2.0.4.v8 이상을 사용하는 RCU 12c

RCU를 사용하려면 Oracle Fusion Middleware용 라이선스가 있어야 합니다. 또한 리포지토리를 호스팅하는 Oracle Database용 Oracle 라이선싱 지침을 따라야 합니다. 자세한 내용은 Oracle 설명서에서 [Oracle Fusion Middleware 라이선싱 정보 사용자 매뉴얼](#)을 참조하십시오.

Fusion MiddleWare는 Oracle Database Enterprise Edition 및 Standard Edition(SE, SE One 또는 SE Two)의 리포지토리를 지원합니다. 온라인 인덱스 다시 빌드가 필요한 파티셔닝 및 설치를 수행해야 하는 프로덕션 설치용 Enterprise Edition을 사용하는 것이 좋습니다.

Oracle DB 인스턴스를 생성하기 전에 배포할 구성 요소를 지원해야 하는 Oracle Database 버전을 확인하십시오. 인증표에서 배포할 Fusion Middleware 구성 요소 및 버전에 대한 요구 사항을 확인할 수 있습니다. 자세한 내용은 Oracle 설명서에서 [Oracle Fusion Middleware 지원 시스템 구성](#)을 참조하십시오.

Amazon RDS는 필요한 경우 Oracle Database 버전 업그레이드를 지원합니다. 자세한 내용은 [DB 인스턴스 엔진 버전 업그레이드 \(p. 118\)](#) 단원을 참조하십시오.

시작하기 전에

시작하려면 Amazon VPC가 필요합니다. Amazon RDS DB 인스턴스는 퍼블릭 인터넷이 아니라 Fusion Middleware 구성 요소에서만 사용해야 하므로 Amazon RDS DB 인스턴스는 프라이빗 서브넷에서 호스팅되어 보안이 향상됩니다. Oracle DB 인스턴스와 함께 사용하도록 Amazon VPC를 만드는 방법에 대한 자세한 내용은 [Oracle Database와 함께 사용할 Amazon VPC 생성 \(p. 875\)](#) 단원을 참조하십시오.

시작하려면 Oracle DB 인스턴스도 필요합니다. Fusion Middleware 메타데이터와 함께 사용하도록 Oracle DB 인스턴스를 만드는 방법에 대한 자세한 내용은 [Oracle DB 인스턴스 생성 \(p. 880\)](#) 단원을 참조하십시오.

권장 사항

다음은 이 시나리오에서 DB 인스턴스를 작업하는 경우 알아 두어야 할 권장 사항입니다.

- 프로덕션 워크로드에는 다중 다중 AZ를 사용하는 것이 좋습니다. 다중 가용 영역 작업에 대한 자세한 내용은 [리전 및 가용 영역 \(p. 99\)](#) 단원을 참조하십시오.
- 보안을 추가하려면 유휴 데이터를 암호화할 수 있는 TDE(Transparent Data Encryption)를 사용하는 것이 좋습니다. 고급 보안 옵션이 제공되는 Enterprise Edition 라이선스를 보유하고 있는 경우 TDE 옵션을 사용하여 유휴 시 암호화를 활성화할 수 있습니다. 자세한 내용은 [Oracle Transparent Data Encryption \(p. 827\)](#) 단원을 참조하십시오.

Amazon RDS는 모든 데이터베이스 버전에 유휴 시 암호화 옵션도 제공합니다. 자세한 내용은 [Amazon RDS 리소스 암호화 \(p. 385\)](#) 단원을 참조하십시오.

- 애플리케이션 서버와 Amazon RDS DB 인스턴스 간에 통신할 수 있도록 VPC 보안 그룹을 구성합니다. Fusion Middleware 구성 요소를 호스팅하는 애플리케이션 서버는 Amazon EC2 또는 온프레미스에 있을 수 있습니다.

Oracle Repository Creation Utility 사용

Oracle RCU(Repository Creation Utility)를 사용하여 Fusion Middleware 구성 요소를 지원하도록 스키마를 만들어 채울 수 있습니다.

한 단계로 명령줄을 사용하여 RCU 실행

스키마를 채우기 전에 편집할 필요가 없는 경우 RCU를 한 단계로 실행할 수 있습니다. 편집이 필요한 경우 다음 단원에서 여러 단계에 걸쳐 RCU 실행에 대한 내용을 참조하십시오.

명령줄 파라미터 `-silent`를 사용하면 자동 모드에서 RCU를 실행할 수 있습니다. 자동 모드로 RCU를 실행하는 경우 암호가 들어 있는 텍스트 파일을 만들면 명령줄에 암호를 입력하지 않아도 됩니다. 첫 행에 `dbUser`의 암호가 있고 다음 행에 각 구성 요소에 대한 암호가 나오는 텍스트 파일을 만듭니다. RCU 명령의 마지막 파라미터로 암호 파일의 이름을 지정합니다.

Example

다음은 한 단계로 SOA 인프라 구성 요소(및 종속 항목)의 스키마를 만들어 채우는 예제입니다.

Linux, OS X, Unix의 경우:

```
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
```

자세한 내용은 Oracle 설명서에서 [명령줄에서 Repository Creation Utility 실행](#)을 참조하십시오.

여러 단계에 걸쳐 명령줄을 사용하여 RCU 실행

스키마 스크립트를 수동으로 편집해야 하는 경우 여러 단계에 걸쳐 RCU를 실행할 수 있습니다.

- 스키마에 대한 스크립트를 생성하려면 `-generateScript` 명령줄 파라미터를 사용하여 Prepare Scripts for System Load 모드에서 RCU를 실행합니다.
- 생성된 스크립트 `script_systemLoad.sql`을 수동으로 편집하고 실행합니다.
- 스키마를 채우려면 `-dataLoad` 명령줄 파라미터를 사용하여 Perform Product Load 모드에서 RCU를 다시 실행합니다.
- 생성된 정리 스크립트 `script_postDataLoad.sql`을 실행합니다.

명령줄 파라미터 `-silent`를 사용하면 자동 모드에서 RCU를 실행할 수 있습니다. 자동 모드로 RCU를 실행하는 경우 암호가 들어 있는 텍스트 파일을 만들면 명령줄에 암호를 입력하지 않아도 됩니다. 첫 행에

dbUser의 암호가 있고 다음 행에 각 구성 요소에 대한 암호가 나오는 텍스트 파일을 만듭니다. RCU 명령의 마지막 파라미터로 암호 파일의 이름을 지정합니다.

Example

다음은 SOA 인프라 구성 요소(및 종속 항목)의 스키마 스크립트를 만드는 예제입니다.

Linux, OS X, Unix의 경우:

```
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
```

이제 생성된 스크립트를 편집하고, Oracle DB 인스턴스에 연결하고, 스크립트를 실행할 수 있습니다. 생성된 스크립트의 이름은 script_systemLoad.sql입니다. Oracle DB 인스턴스 연결에 대한 자세한 내용은 [샘플 Oracle DB 인스턴스에 연결 \(p. 37\)](#) 단원을 참조하십시오.

다음은 SOA 인프라 구성 요소(및 종속 항목)의 스키마를 채우는 예제입니다.

Linux, OS X, Unix의 경우:

```
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
```

완료하려면 Oracle DB 인스턴스에 연결하고 정리 스크립트를 실행합니다. 생성된 스크립트의 이름은 script_postDataLoad.sql입니다.

자세한 내용은 Oracle 설명서에서 [명령줄에서 Repository Creation Utility 실행](#)을 참조하십시오.

대화형 모드에서 RCU 실행

RCU 그래픽 사용자 인터페이스를 사용하려면 대화형 모드로 RCU를 실행합니다. 대화형 모드로 RCU를 실행하려면 `-interactive` 파라미터를 넣고 `-silent` 파라미터는 생략합니다. 자세한 내용은 Oracle 설명서에서 [Repository Creation Utility 화면 이해](#)를 참조하십시오.

Example

다음은 대화형 모드에서 RCU를 시작하고 연결 정보를 미리 채우는 예제입니다.

Linux, OS X, Unix의 경우:

```
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
```

알려진 문제

다음은 RCU 작업에 관련된 알려진 문제와 문제 해결 제안 사항입니다.

- Oracle Managed Files(OMF) – Amazon RDS는 OMF 데이터 파일을 사용하여 스토리지 관리를 간소화합니다. 크기 및 익스텐트 관리와 같은 테이블스페이스 속성을 사용자 지정할 수 있습니다. 하지만 RCU를 실행할 때 데이터 파일 이름을 지정하면 테이블스페이스 코드에 ORA-20900 오류가 발생하게 됩니다. RCU는 다음 방법으로 OMF와 함께 사용할 수 있습니다.
 - RCU 12.2.1.0 이상에서 `-honorOMF` 명령줄 파라미터를 사용합니다.
 - RCU 12.1.0.3 이상에서 여러 단계를 사용하여 생성된 스크립트를 편집합니다. 자세한 내용은 [여러 단계에 걸쳐 명령줄을 사용하여 RCU 실행 \(p. 897\)](#) 단원을 참조하십시오.
- SYSDBA – Amazon RDS는 관리형 서비스이므로 Oracle DB 인스턴스에 대한 전체 SYSDBA 액세스 권한이 없습니다. 하지만 RCU 12c는 낮은 권한을 가진 사용자를 지원합니다. 대부분의 경우 마스터 사용자 권한으로 리포지토리를 생성할 수 있습니다. 일부 경우에 RCU에서 SYS 객체 권한을 부여하려고 하면 ORA-01031 오류가 발생할 수 있습니다. 다시 시도하고 RDSADMIN_UTIL.GRANT_SYS_OBJECT() 저장 프로시저를 실행하거나 AWS Support에 문의할 수 있습니다.
- Enterprise Scheduler Service 삭제 – RCU를 사용하여 Enterprise Scheduler Service 리포지토리를 삭제하는 경우 RCU에 `Error: Component drop check failed` 오류가 발생할 수 있습니다.

관련 주제

- [Oracle 라이선싱 \(p. 704\)](#)

Amazon RDS에서 Oracle에 Siebel Database 설치

Amazon RDS를 사용하여 Oracle DB 인스턴스에서 Siebel Database를 호스팅할 수 있습니다. Siebel Database는 Siebel 고객 관계 관리(CRM) 애플리케이션 아키텍처의 일부입니다. 예시는 [Generic Architecture of Siebel Business Application](#)을 참조하십시오.

다음 주제는 Amazon RDS의 Oracle DB 인스턴스에서 Siebel Database를 설정하는 데 도움이 됩니다. 또한 Amazon Web Services를 사용하여 Siebel CRM 애플리케이션 아키텍처에서 요구하는 다른 구성 요소를 지원하는 방법도 설명합니다.

Note

Amazon RDS에서 Oracle에 Siebel Database를 설치하려면 마스터 사용자 계정을 사용해야 합니다. SYSDBA 권한은 필요 없으며, 마스터 사용자 권한으로 충분합니다. 자세한 내용은 [마스터 사용자 계정 권한 \(p. 402\)](#) 단원을 참조하십시오.

라이선스 및 버전

Amazon RDS에 Siebel Database를 설치하려면 Oracle 데이터베이스 라이선스와 Siebel 라이선스를 사용해야 합니다. DB 인스턴스 클래스와 Oracle 데이터베이스 에디션에 적합한 Oracle 데이터베이스 라이선스(소프트웨어 업데이트 라이선스 및 지원 포함)가 있어야 합니다. 자세한 내용은 [Oracle 라이선싱 \(p. 704\)](#) 단원을 참조하십시오.

이 시나리오의 경우 Oracle Database Enterprise Edition이 Siebel이 인증한 유일한 에디션입니다. Amazon RDS는 Siebel CRM 15.0 또는 16.0 버전을 지원합니다. Oracle 12c, 버전 12.1.0.2.0을 사용합니다. 아래의 절차에서는 Siebel CRM 버전 15.0과 Oracle 12.1.0.2 또는 12.2.0.1을 사용합니다. 자세한 내용은 [Amazon RDS 기반 Oracle 12c \(p. 709\)](#) 단원을 참조하십시오.

Amazon RDS는 데이터베이스 버전 업그레이드를 지원합니다. 자세한 내용은 [DB 인스턴스 엔진 버전 업그레이드 \(p. 118\)](#) 단원을 참조하십시오.

시작하기 전에

시작하려면 Amazon VPC가 필요합니다. Amazon RDS DB 인스턴스는 퍼블릭 인터넷이 아닌 Siebel Enterprise Server에서만 사용 가능해야 하므로 Amazon RDS DB 인스턴스는 프라이빗 서브넷에서 호스팅되고 이로써 보안이 향상됩니다. Siebel CRM과 함께 사용할 Amazon VPC를 만드는 방법에 대한 정보는 [Oracle Database와 함께 사용할 Amazon VPC 생성 \(p. 875\)](#) 단원을 참조하십시오.

시작하려면 Oracle DB 인스턴스도 필요합니다. Siebel CRM과 함께 사용하도록 Oracle DB 인스턴스를 만드는 방법에 대한 자세한 내용은 [Oracle DB 인스턴스 생성 \(p. 880\)](#) 단원을 참조하십시오.

Siebel Database 설치 및 구성

Oracle DB 인스턴스를 만든 후에 Siebel Database를 설치할 수 있습니다. 테이블 소유자 및 관리자 계정을 만들어 데이터베이스를 설치하고, 저장된 절차와 기능을 설치한 다음 Siebel Database 구성 마법사를 실행합니다. 자세한 내용은 [Installing the Siebel Database on the RDBMS](#) 단원을 참조하십시오.

Siebel Database 구성 마법사를 실행하려면 마스터 사용자 계정을 사용해야 합니다. SYSDBA 권한은 필요 없으며, 마스터 사용자 권한으로 충분합니다. 자세한 내용은 [마스터 사용자 계정 권한 \(p. 402\)](#) 단원을 참조하십시오.

Siebel Database에 다른 Amazon RDS 기능 사용

Oracle DB 인스턴스를 만든 후, 다른 Amazon RDS 기능을 사용하여 Siebel Database를 사용자 지정할 수 있습니다.

Oracle Statspack 통계 수집 옵션

DB 옵션 그룹의 옵션을 사용해 Oracle DB 인스턴스에 기능을 추가할 수 있습니다. Oracle DB 인스턴스를 만들 때는 기본 DB 옵션 그룹을 사용했습니다. 데이터베이스에 기능을 추가하고 싶다면 DB 인스턴스를 위한 새로운 옵션 그룹을 만들 수 있습니다.

Siebel Database에서 성능 통계를 수집하고 싶다면 Oracle Statspack 기능을 추가할 수 있습니다. 자세한 내용은 [Oracle Statspack \(p. 822\)](#) 단원을 참조하십시오.

즉시 적용되는 옵션 변경 사항도 있고, DB 인스턴스의 다음 번 유지 관리 기간 중에 적용되는 옵션 변경 사항도 있습니다. 자세한 내용은 [옵션 그룹 작업 \(p. 148\)](#) 단원을 참조하십시오. 사용자 지정 옵션 그룹을 생성한 후에는 DB 인스턴스를 수정해 옵션 그룹에 연결하십시오. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

파라미터로 성능 튜닝

DB 파라미터 그룹의 파라미터 사용을 통해 DB 엔진 구성을 관리합니다. Oracle DB 인스턴스를 만들 때는 기본 DB 파라미터 그룹을 사용했습니다. 데이터베이스 구성은 사용자 지정하고 싶다면 DB 인스턴스를 위한 새로운 파라미터 그룹을 만들 수 있습니다.

파라미터를 변경하는 경우, 파라미터 유형에 따라 변경 사항이 즉시 적용되거나 DB 인스턴스를 수동으로 재부팅한 후에 적용될 수 있습니다. 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오. 사용자 지정 파라미터 그룹을 생성한 후에는 DB 인스턴스를 수정해 파라미터 그룹에 연결하십시오. 자세한 내용은 [Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 \(p. 742\)](#) 단원을 참조하십시오.

Siebel CRM에 맞게 Oracle DB 인스턴스를 최적화하기 위해 일부 파라미터를 사용자 지정할 수 있습니다. 다음 표에는 몇 가지 권장되는 파라미터 설정이 나와 있습니다. Siebel CRM 성능 튜닝에 대한 자세한 내용은 [Siebel CRM Performance Tuning Guide](#)를 참조하십시오.

파라미터 이름	기본값	Siebel CRM 성능 최적화 지침
_always_semi_join	TRUE	OFF
_b_tree_bitmap_p	TRUE	FALSE
_like_with_bind_	FALSE	quality TRUE
_no_or_expansion	FALSE	FALSE
_optimizer_join_	TRUE	sanity_check TRUE
_optimizer_max_p	2000	iterations 100
_optimizer_sortmerge_	TRUE	join_enabled FALSE
_partition_view_	ENABLED	FALSE
open_cursors	300	최소 2000.

스냅샷 만들기

Siebel Database를 만든 후 Amazon RDS의 스냅샷 기능을 이용해 데이터베이스를 복사할 수 있습니다. 자세한 내용은 [DB 스냅샷 생성 \(p. 203\)](#) 및 [DB 스냅샷에서 복원 \(p. 205\)](#) 단원을 참조하십시오.

다른 Siebel CRM 구성 요소에 대한 지원

Amazon Web Services를 사용하여 Siebel Database 외에 Siebel CRM 애플리케이션 아키텍처의 다른 구성 요소도 지원할 수 있습니다. 다음 표에 추가 Siebel CRM 구성 요소를 위한 Amazon AWS의 지원 옵션이 설명되어 있습니다.

Siebel CRM 구성 요소	Amazon AWS 지원
Siebel Enterprise (Siebel 서버 1개 이상)	Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 Siebel 서버를 호스팅할 수 있습니다. Amazon EC2를 사용하여 필요한 만큼 많거나 적은 수의 가상 서버를 시작할 수 있습니다. Amazon EC2를 사용하여 손쉽게 규모를 확장 또는 축소하여 수요 변화에 대처할 수 있습니다. 자세한 내용은 Amazon EC2란 무엇입니까? 단원을 참조하십시오. 서버를 DB 인스턴스와 같은 VPC에 배치하고 VPC 보안 그룹을 사용해 데이터베이스에 액세스할 수 있습니다. 자세한

Siebel CRM 구성 요소	Amazon AWS 지원
	내용은 VPC에서 Amazon RDS DB 인스턴스를 사용한 작업 (p. 415) 단원을 참조하십시오.
웹 서버 (Siebel Web Server Extensions 사용)	복수의 EC2 인스턴스에 복수의 웹 서버를 설치할 수 있습니다. 그런 다음 Elastic Load Balancing을 사용해 수신 트래픽을 인스턴스에 분산할 수 있습니다. 자세한 내용은 Elastic Load Balancing란 무엇입니까? 단원을 참조하십시오.
Siebel Gateway Name Server	EC2 인스턴스에서 Siebel Gateway Name Server를 호스팅할 수 있습니다. 그러면 서버를 DB 인스턴스와 같은 VPC에 배치하고 VPC 보안 그룹을 사용해 데이터베이스에 액세스할 수 있습니다. 자세한 내용은 VPC에서 Amazon RDS DB 인스턴스를 사용한 작업 (p. 415) 단원을 참조하십시오.

관련 주제

- Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스에서 데이터베이스에 연결 (p. 736)

Oracle 데이터베이스 엔진 출시 정보

Amazon RDS for Oracle DB 인스턴스 업데이트로 인스턴스를 최신으로 유지합니다. 업데이트를 적용하면 Oracle과 Amazon 모두에서 회귀 테스트를 거쳐 안정적인 공통 버전의 데이터베이스 소프트웨어를 DB 인스턴스에서 실행할 수 있습니다. 개별 DB 인스턴스에 대한 일회용 패치 적용은 지원하지 않습니다.

Oracle 버전 12.2.0.1

Oracle 버전 12.2.0.1용 Amazon RDS의 경우, Amazon RDS는 RU(Release Updates) 및 RUR(Release Updates Revisions)의 버그 수정을 포함시킵니다.

다음 RU 및 RUR은 Oracle 버전 12.2.0.1용 Amazon RDS에 적용됩니다.

- [12.2.0.1.ru-2018-10.rur-2018-10.r1 \(p. 904\)](#)

Oracle 버전 12.1.0.2 및 11.2.0.4

Oracle 버전 12.1.0.2 및 11.2.0.4용 Amazon RDS의 경우, Amazon RDS는 분기별 데이터베이스 PSU(Patch Set Updates)를 통해 Oracle의 버그 수정을 포함시킵니다. Oracle과 Amazon 모두에서 회귀 테스트를 거쳐 안정적인 공통 버전의 데이터베이스 소프트웨어를 DB 인스턴스에서 실행할 수 있습니다. 개별 DB 인스턴스에 대한 일회용 패치 적용은 지원하지 않습니다.

Oracle 버전 12.1.0.2 및 11.2.0.4용 Amazon RDS에 적용되는 Oracle PSU(Patch Set Updates)를 찾으려면 다음 표를 참조하십시오.

PSU	버전 12.1.0.2	버전 11.2.0.4
2018년 10월	12.1.0.2.v14 (p. 906)	11.2.0.4.v18 (p. 925)
2018년 7월	12.1.0.2.v13 (p. 908)	11.2.0.4.v17 (p. 926)
2018년 4월	12.1.0.2.v12 (p. 910)	11.2.0.4.v16 (p. 928)

PSU	버전 12.1.0.2	버전 11.2.0.4
2018년 1월	12.1.0.2.v11 (p. 912)	11.2.0.4.v15 (p. 929)
2017년 10월	12.1.0.2.v10 (p. 913)	11.2.0.4.v14 (p. 931)
2017년 7월	12.1.0.2.v9 (p. 915)	11.2.0.4.v13 (p. 932)
2017년 4월	12.1.0.2.v8 (p. 916)	11.2.0.4.v12 (p. 933)
2017년 1월	12.1.0.2.v7 (p. 918)	11.2.0.4.v11 (p. 935)
2016년 10월	12.1.0.2.v6 (p. 919)	11.2.0.4.v10 (p. 936)
2016년 7월	12.1.0.2.v5 (p. 920)	11.2.0.4.v9 (p. 937)
2016년 4월	12.1.0.2.v4 (p. 921)	11.2.0.4.v8 (p. 938)
2016년 1월	12.1.0.2.v3 (p. 922)	11.2.0.4.v7 (p. 939)
2015년 10월	12.1.0.2.v2 (p. 923)	11.2.0.4.v6 (p. 941) 11.2.0.4.v5 (p. 941)
2015년 4월	12.1.0.2.v1 (p. 924)	11.2.0.4.v4 (p. 942)
2014년 10월	—	11.2.0.4.v3 (p. 943)
2014년 7월	—	11.2.0.4.v2 (p. 943) (사용되지 않음)
2014년 1월	—	11.2.0.4.v1 (p. 944)

주제

- [데이터베이스 엔진: 12.2.0.1 \(p. 903\)](#)
- [데이터베이스 엔진: 12.1.0.2 \(p. 905\)](#)
- [데이터베이스 엔진: 11.2.0.4 \(p. 924\)](#)

데이터베이스 엔진: 12.2.0.1

Oracle 12c 버전 12.2.0.1의 경우 Oracle은 Oracle 데이터베이스 업데이트를 출시하는 방법을 변경했습니다. Oracle에서는 PSU(Patch Set Updates) 대신 RU(Release Updates)와 RUR(Release Updates Revisions)을 지원합니다. RU에는 최적화 프로그램 변경 사항, 기능 추가 사항 및 보안 수정 사항이 포함됩니다. RUR에는 분기별로 진행되는 두 번의 이전 패치 주기에 대한 보안 수정 사항만 포함됩니다. 이 새로운 시스템을 통해 각 업데이트 시 설치하는 기능을 더 세부적으로 제어할 수 있습니다.

Oracle 12c 버전 12.2.0.1 버전에서 명명 규칙도 변경되었습니다. 이전 버전에서 Oracle용 Amazon RDS는 [oracle-version.vpatch-version](#)의 PSU 명명 규칙을 사용했습니다. [patch-version](#)은 Oracle PSU에 해당했습니다. 예를 들어 Oracle 버전 12.1.0.2.v13용 Amazon RDS에서 버전 번호의 v13 부분이 Oracle PSU에 해당합니다.

Oracle 12c 버전 12.2.0.1 명명 규칙이 RU와 RUR 업데이트를 모두 설명합니다. 예를 들어 사용 가능한 첫 번째 Oracle 버전용 Amazon RDS는 12.2.0.1.ru-2018-10.rur-2018-10.r1입니다. 이 예에서 12.2는 메이저 버전이고, 0.1은 마이너 버전입니다. 개정 버전에는 다음 부분이 있습니다.

- ru-2018-10 – 10월 RU
- rur-2018-10 – 10월 RU의 10월 RUR

- r1 – 내부 Amazon RDS 개정으로, 이를 통해 Amazon RDS가 기존 RU/RUR의 긴급 패치 간에 구별할 수 있음

새로운 Oracle 데이터베이스 버전 관리 시스템에 대한 자세한 내용은 Upgrade your Database – NOW! 블로그의 [Differences between PSU / BP and RU / RUR](#) 및 Oracle-Help 블로그의 [RU and RUR patches for Oracle 12.2](#) 게시물을 참조하십시오.

버전 12.2.0.1.ru-2018-10.rur-2018-10.r1

버전 12.2.0.1.ru-2018-10.rur-2018-10.r1에 다음에 대한 지원이 추가됩니다.

- 2018년 10월 릴리스 업데이트: 12.2.0.1.181016(28662603)

Oracle 릴리스 업데이트 12.2.0.1.181016, 2018년 10월 릴리스

버그 수정: 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,

데이터베이스 엔진: 12.1.0.2

데이터베이스 엔진 12.1.0.2에 사용할 수 있는 버전은 다음과 같습니다.

- [버전 12.1.0.2.v14 \(p. 906\)](#)
- [버전 12.1.0.2.v13 \(p. 908\)](#)

- 버전 12.1.0.2.v12 (p. 910)
- 버전 12.1.0.2.v11 (p. 912)
- 버전 12.1.0.2.v10 (p. 913)
- 버전 12.1.0.2.v9 (p. 915)
- 버전 12.1.0.2.v8 (p. 916)
- 버전 12.1.0.2.v7 (p. 918)
- 버전 12.1.0.2.v6 (p. 919)
- 버전 12.1.0.2.v5 (p. 920)
- 버전 12.1.0.2.v4 (p. 921)
- 버전 12.1.0.2.v3 (p. 922)
- 버전 12.1.0.2.v2 (p. 923)
- 버전 12.1.0.2.v1 (p. 924)

버전 12.1.0.2.v14

버전 12.1.0.2.v14에 다음에 대한 지원이 추가됩니다.

- 패치 28259833: Oracle Database 패치 세트 업데이트 12.1.0.2.181016
- 패치 28440711: Oracle JVM 패치 세트 업데이트 12.1.0.2.181016
- 패치 28125601: RDBMS용 DSTv32(TZDATA2018E)
- 패치 28127287: OJVM용 DSTv32(TZDATA2018E)
- 패치 17969866: Oracle GoldenGate - Oracle RDBMS Server 권장 패치
- 패치 20394750: Oracle GoldenGate - Oracle RDBMS Server 권장 패치
- 패치 21171382: DBMS_STATS 패치
- 패치 28697469: JSON 데이터베이스 패치
- 패치 20033733: KGL 힙 크기 패치

Oracle 패치 28259833, 2018년 10월 릴리스

버그 수정: 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, 27199245, 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,

버전 12.1.0.2.v13

버전 12.1.0.2.v13에 다음에 대한 지원이 추가됩니다.

- 패치 27547329: Oracle Database 패치 세트 업데이트 12.1.0.2.180717
- 패치 27923320: Oracle JVM 패치 세트 업데이트 12.1.0.2.180717
- 패치 28125601: RDBMS용 DSTv32(TZDATA2018E)
- 패치 28127287: OJVM용 DSTv32(TZDATA2018E)
- 패치 17969866: Oracle GoldenGate - Oracle RDBMS Server 권장 패치
- 패치 20394750: Oracle GoldenGate - Oracle RDBMS Server 권장 패치
- 패치 21171382: DBMS_STATS 패치
- 패치 28307069: JSON 데이터베이스 패치
- 패치 20033733: KGL 힙 크기 패치

Oracle 패치 27547329, 2018년 7월 릴리스 완료

버그 수정됨: 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 21649497, 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

버전 12.1.0.2.v12

버전 12.1.0.2.v12에 다음에 대한 지원이 추가됩니다.

- 패치 27338041: 데이터베이스 패치 세트 업데이트 12.1.0.2.180417
- 패치 27475603: OJVM 패치 세트 업데이트 12.1.0.2.180417
- 패치 27015449: RDBMS - PROACTIVE DSTV31 UPDATE - TZDATA2017C
- 패치 27015468: PROACTIVE DSTV31 UPDATE - TZDATA2017C - NEED OJVM FIX
- 패치 17969866: Oracle GoldenGate - Oracle RDBMS Server 권장 패치
- 패치 20394750: Oracle GoldenGate - Oracle RDBMS Server 권장 패치
- 패치 21171382: AUTO DOP COMPUTES A HIGH DOP UNNECESSARILY
- 패치 27666699: JSON 데이터베이스 패치
- 패치 20033733: PART :IMC: HIT ORA 600[KGL-HEAP-SIZE-EXCEEDED]

Oracle 패치 27338041, 2018년 4월 릴리스

버그 수정: 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, 2189588, 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, 18799063, 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

버전 12.1.0.2.v11

버전 12.1.0.2.v11에 다음에 대한 지원이 추가됩니다.

- 패치 26925311: DATABASE PATCH SET UPDATE 12.1.0.2.180116
- 패치 27001733: OJVM PATCH SET UPDATE 12.1.0.2.180116
- 패치 27015449: RDBMS - PROACTIVE DSTV31 UPDATE - TZDATA2017C
- 패치 27015468: PROACTIVE DSTV31 UPDATE - TZDATA2017C - NEED OJVM FIX
- 패치 17969866: Oracle GoldenGate - Oracle RDBMS Server 권장 패치
- 패치 20394750: Oracle GoldenGate - Oracle RDBMS Server 권장 패치
- 패치 21171382: AUTO DOP COMPUTES A HIGH DOP UNNECESSARILY
- 패치 27315904: JSON 데이터베이스 패치
- 패치 20033733: ORA 600[KGL-HEAP-SIZE-EXCEEDED]

Oracle 패치 26925311, 2018년 1월 릴리스

버그 수정: 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, 19188927, 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

버전 12.1.0.2.v10

버전 12.1.0.2.v10에 다음에 대한 지원이 추가됩니다.

- Oracle 2017 10월 PSU, 데이터베이스 PSU(패치 26713565) + OJVM 구성 요소 PSU(패치 26635845) 조합
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 17969866)
- DBMS_STATS AUTO DOP이 불필요하게 HIGH DOP을 계산(패치 21171382)
- JSON 번들 패치(패치 26750145)
- KGL 힙 크기 패치(패치 20033733)
- 시간대 파일 DSTv30(패치 25881255, OJVM 패치 25881271)

Oracle 패치 26713565, 2017년 10월 릴리스

버그 수정: 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

버전 12.1.0.2.v9

버전 12.1.0.2.v9에 다음에 대한 지원이 추가됩니다.

- Oracle 2017 7월 PSU, 데이터베이스 PSU(패치 26609783) + OJVM 구성 요소 PSU(패치 26027162) 조합
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 17969866)
- DBMS_STATS AUTO DOP이 불필요하게 HIGH DOP을 계산(패치 21171382)
- JSON 번들 패치(패치 26083365)
- KGL 힙 크기 패치(12.1.0.2에 대한 패치 20033733)
- 시간대 파일 DSTv30(패치 25881255, OJVM 패치 25881271)
- RMAN 논리적 확인 유ти리티를 사용하는 [DB 인스턴스 파일 확인 \(p. 862\)](#)에 대한 지원 추가
- [DB 인스턴스의 기본 에디션 설정 \(p. 850\)](#)에 대한 지원 추가

Oracle 패치 26609783, 2017년 7월 릴리스

버그 수정: 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, 20588502, 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

버전 12.1.0.2.v8

버전 12.1.0.2.v8에 다음에 대한 지원이 추가됩니다.

- Oracle 패치 25433980, 데이터베이스 PSU(패치 25171037) + OJVM 구성 요소 PSU(패치 25437695) 조합
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 17969866 for 12.1.0.2)
- Oracle 양식 패치 18307021 for 12.1.0.2
- DBMS_STATS 패치(12.1.0.2에 대한 패치 21171382)

- JSON 번들 패치(12.1.0.2에 대한 패치 25531469)
- KGL 힙 크기 패치(12.1.0.2에 대한 패치 20033733)
- 12.1.0.2.v5, v6 및 v7로 업그레이드 후 PSU 적용에 영향을 미친 버그를 수정했습니다.
- 시간대 파일 DSTv28(패치 24701840)
- DBMS_CHANGE_NOTIFICATION 패키지에 대한 지원을 추가합니다.
- XSTREAM 패키지 및 보기에 대한 지원을 추가합니다(추가 라이선스가 필요할 수 있음).

Oracle 패치 25171037, 2017년 4월 릴리스

버그 수정: 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, 20588502, 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

버전 12.1.0.2.v7

버전 12.1.0.2.v7에 다음에 대한 지원이 추가됩니다.

- Oracle 패치 24917069, 데이터베이스 PSU(패치 24732082) + OJVM 구성 요소 PSU(패치 24917972) 조합
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 17969866 for 12.1.0.2)
- Oracle 양식 패치 18307021 for 12.1.0.2
- DBMS_STATS 패치(12.1.0.2에 대한 패치 21171382)
- JSON 번들 패치(12.1.0.2에 대한 패치 25089615)
- KGL 힐 크기 패치(12.1.0.2에 대한 패치 20033733)

Oracle 패치 24917069, 2017년 1월 릴리스

버그 수정: 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

버전 12.1.0.2.v6

버전 12.1.0.2.v6에 다음에 대한 지원이 추가됩니다.

- Oracle 패치 24433133, 데이터베이스 PSU(패치 24006101) + OJVM 구성 요소 PSU(패치 24315824) 조합
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 17969866 for 12.1.0.2)
- Oracle 양식 패치 18307021 for 12.1.0.2
- DBMS_STATS 패치(12.1.0.2에 대한 패치 21171382)
- JSON 번들 패치(12.1.0.2에 대한 패치 24568656)
- 12c 업그레이드 스크립트가 고객 디렉터리를 누락하게 만든 버그를 수정함
- 고객이 DIAG 로그 디렉터리를 사용할 수 있게 함

기준: Oracle Database Patch Set Update 12.1.0.2.161018(패치 24006101, 2016년 10월 릴리스)

수정된 버그: 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

버전 12.1.0.2.v5

버전 12.1.0.2.v5에 다음에 대한 지원이 추가됩니다.

- Oracle 패치 23615289, 데이터베이스 PSU(패치 23054246) + OJVM 구성 요소 PSU(패치 23177536) 조합
- 시간대 파일 DSTv26(패치 22873635 for 12.1.0.2)
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 17969866 for 12.1.0.2)
- Oracle 양식 패치 18307021 for 12.1.0.2
- 사용자 정의 암호 확인 기능 만들기 추가. 자세한 내용은 [사용자 지정 암호 확인 함수 생성 \(p. 841\)](#) 단원을 참조하십시오.
- SYS 소유 보기에 대한 암시적 재컴파일을 차단하는 버그 수정

기준: Oracle Database Patch Set Update 12.1.0.2.160719(패치 23054246, 2016년 7월 릴리스)

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

버전 12.1.0.2.v4

버전 12.1.0.2.v4에 다음에 대한 지원이 추가됩니다.

- Oracle PSU 12.1.0.2.160419(22291127)
- 시간대 파일 DSTv25(패치 22037014)
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 17969866)
- 마스터 사용자가 EM_EXPRESS_BASIC 및 EM_EXPRESS_ALL 역할을 부여할 수 있는 기능 추가
- 마스터 사용자가 RDSADMIN.RDSADMIN_UTIL.GRANT_SYS_OBJECT 프로시저를 사용하는 부여 옵션을 통해 SYS 객체에 대한 권한을 부여할 수 있는 기능 추가
- Oracle Fusion Middleware RCU(Repository Creation Utility)에서 만든 가장 일반적인 스키마를 지원하도록 마스터 사용자 권한 추가

기준: Oracle Database Patch Set Update 12.1.0.2.160419(패치 22291127, 2016년 4월 릴리스)

버그 수정: 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

버전 12.1.0.2.v3

버전 12.1.0.2.v3에 다음에 대한 지원이 추가됩니다.

- Oracle PSU 12.1.0.2.160119(21948354).
- 시간대 파일 DSTv25(패치 22037014 for 12.1.0.2). 빌드 시에 DSTv25의 백포트(backport)가 제공되지 않아 12.1.0.1에는 DSTv24, 패치 20875898(12.1.0.1.v3에서 변경되지 않음)이 포함되어 있습니다.
- 고객이 데이터베이스에서 10개를 초과하는 디렉터리 객체를 생성할 수 없는 문제 해결.
- 고객이 ADUMP 및 BDUMP 디렉터리 객체에 대한 읽기 권한을 다시 부여할 수 없는 문제 해결.

기준: Oracle Database Patch Set Update 12.1.0.2.160119(패치 21948354, 2016년 1월 릴리스)

버그 수정: 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

버전 12.1.0.2.v2

버전 12.1.0.2.v2에 다음에 대한 지원이 추가됩니다.

- Oracle PSU 12.1.0.2.5(21359755)
- 2015년 4월 PSU 이후 나온 일광 절약 시간 패치인 패치 20875898: DST-24가 포함됩니다.

기준: Oracle Database Patch Set Update 12.1.0.2.5(패치 21359755, 2015년 10월 릴리스)

버그 수정: 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

버전 12.1.0.2.v1

버전 12.1.0.2.v1에 다음에 대한 지원이 추가됩니다.

- Oracle PSU 12.1.0.2.3(20299023)
- 인 메모리 옵션을 사용하여 데이터의 일부를 성능에 최적화된 인메모리 컬럼 형식으로 저장할 수 있습니다.
- Oracle Database에서 추가 Oracle Text 기술 자료를 설치합니다. 예제 미디어(영어 및 프랑스어)
- RDSADMIN.RDSADMIN_DBMS_REPAIR를 통해 DBMS_REPAIR에 대한 액세스 제공
- 마스터 사용자에게 ALTER DATABASE LINK, ALTER PUBLIC DATABASE LINK, EXEMPT ACCESS POLICY, EXEMPT IDENTITY POLICY, EXEMPT REDACTION POLICY 권한 부여

Note

버전 12.1.0.2.v1은 Enterprise Edition만 지원합니다.

기준: Oracle Database Patch Set Update 12.1.0.2.3(패치 20299023, 2015년 4월 릴리스)

버그 수정: 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

관련 주제

- Oracle DB 엔진 업그레이드 (p. 752)
- Amazon RDS의 Oracle (p. 702)

데이터베이스 엔진: 11.2.0.4

데이터베이스 엔진 11.2.0.4에 사용할 수 있는 버전은 다음과 같습니다.

- 버전 11.2.0.4.v18 (p. 925)
- 버전 11.2.0.4.v17 (p. 926)
- 버전 11.2.0.4.v16 (p. 928)
- 버전 11.2.0.4.v15 (p. 929)
- 버전 11.2.0.4.v14 (p. 931)

- 버전 11.2.0.4.v13 (p. 932)
- 버전 11.2.0.4.v12 (p. 933)
- 버전 11.2.0.4.v11 (p. 935)
- 버전 11.2.0.4.v10 (p. 936)
- 버전 11.2.0.4.v9 (p. 937)
- 버전 11.2.0.4.v8 (p. 938)
- 버전 11.2.0.4.v7 (p. 939)
- 버전 11.2.0.4.v6 (p. 941)
- 버전 11.2.0.4.v5 (p. 941)
- 버전 11.2.0.4.v4 (p. 942)
- 버전 11.2.0.4.v3 (p. 943)
- 버전 11.2.0.4.v2(사용되지 않음) (p. 943)
- 버전 11.2.0.4.v1 (p. 944)

버전 11.2.0.4.v18

버전 11.2.0.4.v18에 다음에 대한 지원이 추가됩니다.

- 패치 28204707: Oracle Database 패치 세트 업데이트 11.2.0.4.181016
- 패치 28440700: Oracle JVM 패치 세트 업데이트 11.2.0.4.181016
- 패치 28125601: RDBMS용 DSTv32(TZDATA2018E)
- 패치 27015468: OJVM용 DSTv32(TZDATA2018E)
- 패치 27216420: Oracle GoldenGate – Oracle RDBMS Server 권장 패치
- 이제 데이터베이스 Patch Set Update에 패치 27659043 및 19692824 포함

Oracle 패치 28204707, 2018년 10월 릴리스

버그 수정: 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, 20299015, 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

버전 11.2.0.4.v17

버전 11.2.0.4.v17은 다음에 대한 지원을 추가합니다.

- 패치 27734982: Oracle Database 패치 세트 업데이트 11.2.0.4.180717
- 패치 27923163: Oracle JVM 패치 세트 업데이트 11.2.0.4.180717

- 패치 28125601: RDBMS용 DSTv32(TZDATA2018E)
- 패치 27015468: OJVM용 DSTv32(TZDATA2018E)
- 패치 27216420: Oracle GoldenGate – Oracle RDBMS Server 권장 패치
- 패치 27659043: MES 번들 405
- 패치 19692824: OEL 7에서는 DBCONTROL이 제공되지 않음

Oracle 패치 27734982, 2018년 7월 릴리스

버그 수정: 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

버전 11.2.0.4.v16

버전 11.2.0.4.v16에 다음에 대한 지원이 추가됩니다.

- 패치 27338049: 데이터베이스 패치 세트 업데이트 11.2.0.4.180417
- 패치 27475598: OJVM 패치 세트 업데이트 11.2.0.4.180417
- 패치 27015449: RDBMS - PROACTIVE DSTV31 UPDATE - TZDATA2017C
- 패치 27015468: PROACTIVE DSTV31 UPDATE - TZDATA2017C - NEED OJVM FIX
- 패치 27216420: Oracle GoldenGate – Oracle RDBMS Server 권장 패치
- 패치 27659043: RDBMS 11.2.0.4.180116 PSU 기반의 MES 405 번들
- 패치 19692824: OEL 7에서는 DBCONTROL이 제공되지 않음
- DBMS_ADVANCED_REWRITE 패키지에 대한 지원을 추가합니다.
- 11.2.0.4.v15의 새 DB 인스턴스에서 DBA_LOCKS와 관련 보기가 11.2.0.4.v15로 업그레이드할 때 생성되지 않는 버그를 수정했습니다. 이제 11.2.0.4.v16 이상의 새로운 또는 업그레이드된 DB 인스턴스에 보기가 생성됩니다.

Oracle 패치 27338049, 2018년 4월 릴리스

버그 수정: 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

버전 11.2.0.4.v15

버전 11.2.0.4.v15에 다음에 대한 지원이 추가됩니다.

- 패치 26925576: 데이터베이스 패치 세트 업데이트 11.2.0.4.180116
- 패치 26925532: OJVM 패치 세트 업데이트 11.2.0.4.180116

- 패치 27015449: RDBMS - PROACTIVE DSTV31 UPDATE - TZDATA2017C
- 패치 27015468: PROACTIVE DSTV31 UPDATE - TZDATA2017C - NEED OJVM FIX
- 패치 27216420: Oracle GoldenGate – Oracle RDBMS Server 권장 패치
- 패치 27244661: RDBMS 11.2.0.4.180116 PSU 기반의 MES 405 번들
- 패치 19692824: OEL 7에서는 DBCONTROL이 제공되지 않음
- DBA_LOCKS 및 관련 보기에 대한 지원 추가

Oracle 패치 26925576, 2018년 1월 릴리스

버그 수정: 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, 19121551 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

버전 11.2.0.4.v14

버전 11.2.0.4.v14에 다음에 대한 지원이 추가됩니다.

- Oracle 2017 10월 PSU, 데이터베이스 PSU(패치 26392168) + OJVM 구성 요소 PSU(패치 26635834) 조합
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 26950781)
- RSA 마이크로 에디션 제품군 번들(패치 26963526)
- 시간대 파일 DSTv30(패치 25881255, OJVM 패치 25881271)

Oracle 패치 26392168, 2017년 10월 릴리스

버그 수정: 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

버전 11.2.0.4.v13

버전 11.2.0.4.v13에 다음에 대한 지원이 추가됩니다.

- Oracle 2017 7월 PSU, 데이터베이스 PSU(패치 26609445) + OJVM 구성 요소 PSU(패치 26027154) 조합
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 26554712)
- RSA 마이크로 에디션 제품군 번들(패치 26770426)
- 시간대 파일 DSTv30(패치 25881255, OJVM 패치 25881271)
- RMAN 논리적 확인 유ти리티를 사용하는 [DB 인스턴스 파일 확인 \(p. 862\)](#)에 대한 지원 추가
- [DB 인스턴스의 기본 에디션 설정 \(p. 850\)](#)에 대한 지원 추가

Oracle 패치 26609445, 2017년 7월 릴리스

버그 수정: 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, 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, 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

버전 11.2.0.4.v12

버전 11.2.0.4.v12에 다음에 대한 지원이 추가됩니다.

- Oracle 패치 25440428, 데이터베이스 PSU(패치 24732075) + OJVM 구성 요소 PSU(패치 25434033) 조합

- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 25734992)
- MES 번들(패치 24975421 for 11.2.0.4)
- 시간대 파일 DSTv28(패치 24701840)
- DBMS_CHANGE_NOTIFICATION 패키지에 대한 지원을 추가합니다.
- XSTREAM 패키지 및 보기에 대한 지원을 추가합니다(추가 라이선스가 필요할 수 있음).

Oracle 패치 24732075, 2017년 4월 릴리스

버그 수정: 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

버전 11.2.0.4.v11

버전 11.2.0.4.v11에 다음에 대한 지원이 추가됩니다.

- Oracle 패치 24918033, 데이터베이스 PSU(패치 24006111) + OJVM 구성 요소 PSU(패치 24917954) 조합
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 24491261)
- MES 번들(패치 24975421 for 11.2.0.4)

Oracle 패치 24918033, 2017년 1월 릴리스

버그 수정: 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, 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, 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

버전 11.2.0.4.v10

버전 11.2.0.4.v10에 다음에 대한 지원이 추가됩니다.

- Oracle 패치 24436313, 데이터베이스 PSU(패치 24006111) + OJVM 구성 요소 PSU(패치 24315821) 조합
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 24491261)
- MES 번들(패치 24975421 for 11.2.0.4)

기준: Oracle Database Patch Set Update 11.2.0.4.161018(패치 24006111, 2016년 10월 릴리스)

수정된 버그: 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, 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

버전 11.2.0.4.v9

버전 11.2.0.4.v9에 다음에 대한 지원이 추가됩니다.

- Oracle 패치 23615392, 데이터베이스 PSU(패치 23054359) + OJVM 구성 요소 PSU(패치 23177551) 조합
- 시간대 파일 DSTv26(패치 22873635 for 11.2.0.4)
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 24320398 for 11.2.0.4.160719)
- MES 번들(패치 22695784 for 11.2.0.4)
- 사용자 정의 암호 확인 기능 만들기 추가. 자세한 내용은 [사용자 지정 암호 확인 함수 생성 \(p. 841\)](#) 단원을 참조하십시오.
- SYS 소유 보기에 대한 암시적 재컴파일을 차단하는 버그 수정

기준: Oracle Database Patch Set Update 11.2.0.4.160719(패치 23054359, 2016년 7월 릴리스)

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

버전 11.2.0.4.v8

버전 11.2.0.4.v8에 다음에 대한 지원이 추가됩니다.

- Oracle PSU 11.2.0.4.160419(22502456)
- 시간대 파일 DSTv25(패치 22037014)
- Oracle에서 권장하는 Oracle GoldenGate용 RDBMS 패치(패치 22576728)
- MES 번들(패치 22695784 for 11.2.0.4)
- 마스터 사용자가 RDSADMIN.RDSADMIN_UTIL.GRANT_SYS_OBJECT 프로시저를 사용하는 부여 옵션을 통해 SYS 객체에 대한 권한을 부여할 수 있는 기능 추가
- Oracle Fusion Middleware RCU(Repository Creation Utility)에서 만든 가장 일반적인 스키마를 지원하도록 마스터 사용자 권한 추가

기준: Oracle Database Patch Set Update 11.2.0.4.160419(패치 22502456, 2016년 4월 릴리스)

버그 수정: 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

버전 11.2.0.4.v7

버전 11.2.0.4.v7에 다음에 대한 지원이 추가됩니다.

- Oracle PSU 11.2.0.4.160119(21948347)
- 시간대 파일 DSTv25 - 11.2.0.4 및 12.1.0.2용 패치 22037014(빌드 시에 DSTv25의 백포트(backport)가 제 공되지 않아 12.1.0.1에는 DSTv24, 패치 20875898(12.1.0.1.v3에서 변경되지 않음)이 포함되어 있음)
- 고객이 데이터베이스에서 10개를 초과하는 디렉터리 객체를 생성할 수 없는 문제 해결
- 고객이 ADUMP 및 BDUMP 디렉터리 객체에 대한 읽기 권한을 다시 부여할 수 없는 문제 해결

기준: Oracle Database Patch Set Update 11.2.0.4.160119(패치 21948347, 2016년 1월 릴리스)

버그 수정: 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

버전 11.2.0.4.v6

버전 11.2.0.4.v6에 다음에 대한 지원이 추가됩니다.

- Standard Edition 및 Standard Edition One에 대해 SSL 암호화 활성화

버전 11.2.0.4.v5

버전 11.2.0.4.v5에 다음에 대한 지원이 추가됩니다.

- Oracle PSU 11.2.0.4.8(패치 21352635)
- 2015년 4월 PSU 이후 나온 일광 절약 시간 패치인 패치 20875898: DST-24가 포함됩니다.

기준: Oracle Database Patch Set Update 11.2.0.4.8(패치 21352635, 2015년 10월 릴리스)

버그 수정: 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

버전 11.2.0.4.v4

버전 11.2.0.4.v4에 다음에 대한 지원이 추가됩니다.

- Oracle PSU 11.2.0.4.6(20299013)
- Oracle Database에서 추가 Oracle Text 기술 자료를 설치합니다. 예제 미디어(영어 및 프랑스어)
- RDSADMIN.RDSADMIN_DBMS_REPAIR를 통해 DBMS_REPAIR에 대한 액세스 제공
- 마스터 사용자에게 ALTER DATABASE LINK, ALTER PUBLIC DATABASE LINK, EXEMPT ACCESS POLICY, EXEMPT IDENTITY POLICY, EXEMPT REDACTION POLICY 권한 부여

기준: Oracle Database Patch Set Update 11.2.0.4.6(패치 20299013, 2015년 4월 릴리스)

버그 수정: 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 19271443, 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

버전 11.2.0.4.v3

버전 11.2.0.4.v3에 다음에 대한 지원이 추가됩니다.

- Oracle PSU 11.2.0.4.4(19121551)
- 최신 DST 파일(DSTv23 - 패치 19396455, 2014년 10월 릴리스). 이 패치는 기본적으로 새로운 인스턴스에서만 함께 제공됩니다.

기준: Oracle Database Patch Set Update 11.2.0.4.4(패치 19121551, 2014년 10월 릴리스)

버그 수정: 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

버전 11.2.0.4.v2(사용되지 않음)

버전 11.2.0.4.v2에 다음에 대한 지원이 추가됩니다.

- Oracle PSU 11.2.0.4.3(18522509)
- 실패한 분산 트랜잭션을 삭제할 수 있는 DBMS_TRANSACTION에 대한 사용자 액세스
- 최신 DST 파일(DSTv22 - 패치 18759211, 2014년 6월 릴리스) 이 패치는 기본적으로 새로운 Oracle DB 인스턴스에서만 함께 제공됩니다.

- DBA 역할에 DBMS_REPUTIL 권한 부여(11.2.0.4 업그레이드 시 모든 사용자에게서 제거됨)
- DBMS_TRANSACTION, v\$pending_xatrans\$ 및 v\$xatrans\$에 관한 권한 부여
- 사용자 객체 이름에 "SYSTEM"이 포함되어 있는 경우 DDL 명령을 이용해 문제 해결
- 스키마 객체 설치를 통해 XA 트랜잭션을 지원함으로써 외부 트랜잭션 관리자도 트랜잭션 관리 가능
- 임시 SYS 및 SYSTEM 객체의 잘라버림 허용으로 LogMiner 같은 도구의 올바른 작동 가능

기준: Oracle Database Patch Set Update 11.2.0.4.3(패치 18522509, 2014년 7월 릴리스)

버그 수정: 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

버전 11.2.0.4.v1

버전 11.2.0.4.v1에 다음에 대한 지원이 추가됩니다.

- Oracle PSU 11.2.0.4.1
- 메인 데이터 스토리지 공간에 새로운 디렉터리 생성 (p. 874)

기준: Oracle Database Patch Set Update 11.2.0.4.1(2014년 1월 릴리스)

버그 수정: 17432124, 16850630, 17551709, 13944971, 17811447, 13866822, 17811429, 16069901 16721594, 17443671, 17478514, 17612828, 17610798, 17239687, 17501491 17446237, 16450169, 17811438, 17288409, 17811456, 12905058, 17088068 16285691, 17332800

관련 주제

- Oracle DB 엔진 업그레이드 (p. 752)
- Amazon RDS의 Oracle (p. 702)

Amazon RDS의 PostgreSQL

Amazon RDS는 여러 PostgreSQL 버전을 실행하는 DB 인스턴스를 지원합니다. DB 인스턴스 및 DB 스냅샷, 특정 시점으로 복원 및 백업을 만들 수 있습니다. PostgreSQL을 실행하는 DB 인스턴스는 다중 AZ 배포, 읽기 전용 복제본(버전 9.3.5 이상), 프로비저닝된 IOPS를 지원하며, VPC 내부에서 생성될 수 있습니다. 또한, SSL(Secure Socket Layer)을 사용하여 PostgreSQL을 실행하는 DB 인스턴스에 연결할 수 있습니다.

DB 인스턴스를 생성하기 전에 이 안내서의 [Amazon RDS 설정 \(p. 5\)](#) 섹션에 있는 단계를 완료해야 합니다.

클라이언트 컴퓨터에서 표준 SQL 클라이언트 애플리케이션을 사용하여 인스턴스에 대한 명령을 실행할 수 있습니다. 이런 애플리케이션으로는 PostgreSQL용으로 널리 사용되는 오픈 소스 관리 및 개발 도구인 pgAdmin 또는 PostgreSQL 설치 시 포함되는 명령줄 유ти리티인 psql이 포함됩니다. 관리형 서비스 환경을 제공하기 위해 Amazon RDS은 DB 인스턴스에 대한 호스트 액세스를 제공하지 않으며 고급 권한이 필요한 특정 시스템 절차 및 테이블에 대한 액세스를 제한합니다. Amazon RDS는 표준 SQL 클라이언트 애플리케이션을 사용하여 DB 인스턴스의 데이터베이스에 대한 액세스를 지원합니다. Amazon RDS은 텔넷 또는 SSH(Secure Shell)를 사용하여 DB 인스턴스에 대한 직접 호스트 액세스를 허용하지 않습니다.

PostgreSQL용 Amazon RDS는 다수의 산업 표준을 준수합니다. 예를 들면, PostgreSQL 데이터베이스용 Amazon RDS를 사용하여 HIPAA 준수 애플리케이션을 구축하고 AWS와 체결한 이행필 비즈니스 제휴 계약(BAA)에 따라 보호 대상 건강 정보(PHI)를 비롯한 의료 관련 정보를 저장할 수 있습니다. 또한 PostgreSQL용 Amazon RDS는 FedRAMP(연방 위험 및 인증 관리 프로그램) 보안 요건을 충족합니다. PostgreSQL용 Amazon RDS는 FedRAMP 공동 승인 위원회(JAB)로부터 AWS GovCloud(미국 서부) 리전 내에서 행사할 수 있는 FedRAMP HIGH Baseline 수준의 잠정적 운영 권한(P-ATO)을 취득했습니다. 지원되는 규정 준수 표준에 대한 자세한 내용은 [AWS 클라우드 규정 준수](#)를 참조하십시오.

PostgreSQL 데이터를 DB 인스턴스로 가져오려면 [Amazon RDS에서 PostgreSQL로 데이터 가져오기 \(p. 970\)](#) 섹션의 정보를 따릅니다.

주제

- [Amazon RDS의 PostgreSQL을 위한 공통 관리 작업 \(p. 945\)](#)
- [PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스 만들기 \(p. 947\)](#)
- [PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 \(p. 953\)](#)
- [PostgreSQL 데이터베이스 엔진을 실행 중인 DB 인스턴스 수정 \(p. 956\)](#)
- [Amazon RDS용 PostgreSQL DB 엔진 업그레이드 \(p. 963\)](#)
- [PostgreSQL 읽기 전용 복제본 작업 \(p. 967\)](#)
- [Amazon RDS에서 PostgreSQL로 데이터 가져오기 \(p. 970\)](#)
- [PostgreSQL을 위한 일반 DBA 작업 \(p. 984\)](#)
- [데이터베이스 미리 보기 환경 작업 \(p. 1011\)](#)
- [PostgreSQL용 Amazon RDS 버전과 확장 \(p. 1014\)](#)

Amazon RDS의 PostgreSQL을 위한 공통 관리 작업

다음은 PostgreSQL용 Amazon RDS DB 인스턴스로 수행하는 일반적인 관리 작업과 각 작업에 해당하는 문서 링크입니다.

작업 영역	관련 문서
처음 사용 시 Amazon RDS 설정	Amazon RDS 설정 (p. 5)

작업 영역	관련 문서
방화벽도 함께 생성됩니다. 따라서 DB 인스턴스에 액세스하기 위한 알맞은 IP 주소와 네트워크 구성으로 보안 그룹을 만들어야 합니다.	
Amazon RDS DB 인스턴스 이해 프로덕션 목적으로 DB 인스턴스를 만들 경우에는 Amazon RDS에서 인스턴스 클래스, 스토리지 유형 및 프로비저닝된 IOPS이 작동하는 방식을 이해해야 합니다.	DB 인스턴스 클래스 선택 (p. 77) Amazon RDS 스토리지 유형 (p. 101) 프로비저닝된 IOPS SSD 스토리지 (p. 103)
지원되는 PostgreSQL 버전 찾기 Amazon RDS는 여러 버전의 PostgreSQL을 지원합니다.	지원되는 PostgreSQL 데이터베이스 버전 (p. 1014)
고가용성 및 장애 조치 지원 설정 프로덕션 DB 인스턴스에서는 다중 AZ 배포를 사용해야 합니다. 다중 AZ 배포는 DB 인스턴스를 위해 향상된 가용성, 데이터 내구성 및 내결함성을 제공합니다.	Amazon RDS를 위한 고가용성(다중 AZ) (p. 109)
Amazon Virtual Private Cloud(VPC) 네트워크 이해 AWS 계정에 기본 VPC가 있는 경우에는 DB 인스턴스가 기본 VPC 내부에 자동으로 생성됩니다. 경우에 따라 계정에 기본 VPC가 없을 수 있으며 VPC에 DB 인스턴스가 필요할 수 있습니다. 이러한 경우에는 DB 인스턴스를 생성하기 전에 VPC 및 서브넷 그룹을 생성하십시오.	EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 (p. 407) VPC에서 Amazon RDS DB 인스턴스를 사용한 작업 (p. 415)
Amazon RDS PostgreSQL로 데이터 가져오기 다양한 도구를 사용하여 Amazon RDS의 PostgreSQL DB 인스턴스로 데이터를 가져올 수 있습니다.	Amazon RDS에서 PostgreSQL로 데이터 가져오기 (p. 970)
읽기 전용 복제본(마스터 및 대기) 설정 Amazon RDS의 PostgreSQL에서는 마스터 인스턴스와 동일한 AWS 리전과 다른 AWS 리전 모두에서 읽기 전용 복제본을 지원합니다.	읽기 전용 복제본 작업 (p. 135) PostgreSQL 읽기 전용 복제본 작업 (p. 967) 다른 AWS 리전에서 읽기 전용 복제본 만들기 (p. 141)
보안 그룹 이해 기본적으로, DB 인스턴스와 함께 인스턴스에 대한 액세스를 막는 방화벽도 생성됩니다. 따라서 DB 인스턴스에 액세스하기 위한 알맞은 IP 주소와 네트워크 구성으로 보안 그룹을 만들어야 합니다. 일반적으로 DB 인스턴스가 EC2-Classic 플랫폼을 기반으로 하는 경우에는 DB 보안 그룹을 생성해야 합니다. 그렇지 않고 DB 인스턴스가 EC2-VPC 플랫폼을 기반으로 하는 경우에는 VPC 보안 그룹을 생성해야 합니다.	EC2-VPC 또는 EC2-Classic 플랫폼을 사용 중인지 확인 (p. 407) 보안 그룹을 통한 액세스 제어 (p. 390)
파라미터 그룹 및 기능 설정 DB 인스턴스에 특정 데이터베이스 파라미터가 필요할 경우, 파라미터 그룹을 만든 후 DB 인스턴스를 만들어야 합니다.	DB 파라미터 그룹 작업 (p. 161)

작업 영역	관련 문서
PostgreSQL을 위한 공통 DBA 작업 수행 PostgreSQL DBA의 일반적인 작업은 다음과 같습니다. <ul style="list-style-type: none">• 역할 생성 (p. 984)• PostgreSQL 데이터베이스 액세스 관리 (p. 985)• PostgreSQL 파라미터 작업 (p. 985)• Amazon RDS에서 PostgreSQL Autovacuum 사용 (p. 994)• PostgreSQL DB 인스턴스에 대한 감사 로깅 (p. 1002)• PostGIS 작업 (p. 1005)• pgBadger를 사용한 PostgreSQL의 로그 분석 (p. 1007)• 아웃바운드 네트워크 액세스에 사용자 지정 DNS 서버 사용 (p. 1009)	PostgreSQL을 위한 일반 DBA 작업 (p. 984)
PostgreSQL DB 인스턴스에 연결 보안 그룹을 만들고 이를 DB 인스턴스에 연결한 후, pgAdmin III와 같은 표준 SQL 클라이언트 애플리케이션을 사용하여 DB 인스턴스에 연결할 수 있습니다.	PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 (p. 953) PostgreSQL DB 인스턴스와 함께 SSL 사용 (p. 1059)
DB 인스턴스 백업 및 복원 DB 인스턴스를 구성하여 자동 백업을 생성하거나 수동 스냅샷을 생성한 다음 백업 또는 스냅샷에서 인스턴스를 복원할 수 있습니다.	Amazon RDS DB 인스턴스 백업 및 복원 (p. 196)
DB 인스턴스 활동 및 성능 모니터링 CloudWatch Amazon RDS 측정치, 이벤트 및 향상된 모니터링 기능을 통해 PostgreSQL DB 인스턴스를 모니터링할 수 있습니다.	DB 인스턴스 측정치 보기 (p. 236) Amazon RDS 이벤트 보기 (p. 305)
PostgreSQL 데이터베이스 버전 업그레이드 PostgreSQL DB 인스턴스의 메이저 버전과 마이너 버전을 모두 업그레이드할 수 있습니다.	PostgreSQL DB 인스턴스 업그레이드 (p. 1059) PostgreSQL 메이저 버전 업그레이드 (p. 964)
로그 파일 작업 PostgreSQL DB 인스턴스의 로그 파일에 액세스할 수 있습니다.	PostgreSQL 데이터베이스 로그 파일 (p. 331)
PostgreSQL DB 인스턴스에 대한 모범 사례 이해 Amazon RDS의 PostgreSQL 사용에 대한 몇 가지 모범 사례를 찾아보십시오.	PostgreSQL로 작업하기 위한 모범 사례 (p. 72)

PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스 만들기

Amazon RDS의 기본 빌딩 블록은 DB 인스턴스입니다. PostgreSQL 데이터베이스를 실행하는 환경입니다.

Important

DB 인스턴스를 생성하거나 DB 인스턴스에 연결하려면 먼저 [Amazon RDS 설정 \(p. 5\)](#) 섹션의 작업을 완료해야 합니다.

PostgreSQL DB 인스턴스 생성

PostgreSQL DB 인스턴스 시작 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. AWS Management 콘솔의 오른쪽 위 모서리에서 DB 인스턴스를 만들 AWS 리전을 선택합니다.
3. 탐색 창에서 데이터베이스를 선택합니다.
탐색 창이 닫혀 있는 경우 왼쪽 상단의 메뉴 아이콘을 선택하여 여십시오.
4. 데이터베이스 생성을 선택하여 엔진 선택 페이지를 열고 시작합니다.

Select engine

Engine options

Amazon Aurora

 PostgreSQL

 MySQL

 MariaDB

 Oracle

 Microsoft SQL Server


PostgreSQL
PostgreSQL is a powerful, open-source object-relational database system with a strong reputation of reliability, stability, and correctness.

- High reliability and stability in a variety of workloads.
- Advanced features to perform in high-volume environments.
- Vibrant open-source community that releases new features multiple times per year.
- Supports multiple extensions that add even more functionality to the database.
- Supports up to 5 Read Replicas per instance, within a single Region or cross-region.
- The most Oracle-compatible open-source database.

If you want to create PostgreSQL 11 in the Preview environment, click [here](#)

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**

5. 엔진 선택 페이지에서 PostgreSQL 아이콘을 선택한 후 다음을 선택합니다.
6. 다음으로 [Use case] 페이지에 생성하려는 DB 인스턴스를 프로덕션에 사용할지 묻는 메시지가 나타납니다. 사용할 계획이라면 [Production]을 선택합니다. 이 옵션을 선택하면 다음 단계에서 다음 설정이 모두 미리 선택됩니다.

- 다중 AZ 장애 조치 옵션
- 프로비저닝된 IOPS 스토리지 옵션
- 삭제 방지 활성화 옵션

마쳤으면 [Next]를 선택합니다.

7. [Specify DB Details] 페이지에서 DB 인스턴스 정보를 지정합니다. 마쳤으면 [Next]를 선택합니다.

이 파라미터의 경우...	수행할 작업
라이선스 모델	PostgreSQL에는 라이선스 모델이 하나밖에 없습니다. postgresql-license 를 선택하여 PostgreSQL 일반 라이선스 계약을 사용합니다.
DB 엔진 버전	사용할 PostgreSQL 버전을 선택합니다.
DB 인스턴스 클래스	2GiB 메모리, 1ECU(1ECU의 가상 코어 하나), 64비트 플랫폼 및 보통 I/O 용량과 동일한 구성에 대해 db.t2.small 을 선택합니다. 모든 DB 인스턴스 클래스 옵션에 대한 자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 단원을 참조하십시오.
다중 AZ 배포	장애 조치를 위해 다른 가용 영역에 DB 인스턴스의 예비 복제본을 생성하려면 Yes 를 선택합니다. 이때 고가용성을 유지하려면 프로덕션 워크로드를 위한 다중 AZ를 권장합니다. 개발 및 테스팅을 위해 No 를 선택할 수 있습니다. 자세한 내용은 Amazon RDS를 위한 고가용성(다중 AZ) (p. 109) 단원을 참조하십시오.
[Storage Type]	스토리지 유형 General Purpose (SSD) 를 선택합니다. 스토리지에 대한 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.
할당된 스토리지	데이터베이스에 대해 스토리지 20GiB를 할당하려면 20 을 입력합니다. 경우에 따라 DB 인스턴스에 대해 데이터베이스의 크기보다 많은 양의 스토리지를 할당하면 I/O 성능을 개선할 수 있습니다. 스토리지 할당에 대한 자세한 내용은 Amazon Relational Database Service 기능 을 참조하십시오.
DB 인스턴스 식별자	선택한 AWS 리전의 계정에 대해 고유한 DB 인스턴스의 이름을 입력합니다. 예를 들어, postgresql-test 처럼 선택한 AWS 리전과 DB 엔진을 포함하는 등 알기 쉬운 이름을 지정할 수 있습니다.
마스터 사용자 이름	영숫자 문자를 사용해 DB 인스턴스에 로그온하기 위해 마스터 사용자 이름으로 사용할 이름을 입력합니다. 마스터 사용자 이름에 부여된 기본 권한에 대한 자세한 내용은 PostgreSQL용 Amazon RDS 버전과 확장 (p. 1014) 단원을 참조하십시오.
[Master Password] 및 [Confirm Password]	마스터 암호에 인쇄 가능한 ASCII 문자(/," 및 @ 제외)로 구성된 암호(8~128자)를 입력하고 [Confirm Password] 상자에 해당 암호를 다시 입력합니다.

8. [Configure Advanced Settings] 페이지에서 Amazon RDS가 PostgreSQL DB 인스턴스를 시작하는 데 필요한 추가 정보를 입력합니다. 표는 예제 DB 인스턴스의 설정을 보여 줍니다. DB 인스턴스 정보를 지정한 다음 데이터베이스 생성을 선택합니다.

이 파라미터의 경우...	수행할 작업
VPC	이 설정은 현재 사용 중인 플랫폼에 따라 다릅니다. AWS를 처음 사용하는 경우에는 기본 VPC를 선택합니다. 이전 E2-Classic 플랫폼에서 VPC를 사용하지 않는 DB 인스턴스를 만드는 경우에는 [Not in VPC]를 선택합니다. VPC에 대한 자세한 내용은 Amazon Virtual Private Cloud(VPC) 및 Amazon RDS (p. 406) 단원을 참조하십시오.
[Subnet Group]	이 설정은 현재 사용 중인 플랫폼에 따라 다릅니다. AWS를 처음 사용하는 고객은 default 를 선택합니다. 이 DB 서브넷 그룹은 계정에 대해 생성된 기본 DB 서브넷 그룹입니다. 이전 E2-Classic 플랫폼에서 DB 인스턴스를 만들어서 특정 VPC에 배포하려면 해당 VPC로 생성된 DB 서브넷 그룹을 선택합니다. VPC에 대한 자세한 내용은 Amazon Virtual Private Cloud(VPC) 및 Amazon RDS (p. 406) 단원을 참조하십시오.
Publicly Accessible	Yes 를 선택하여 VPC 외부에서도 액세스할 수 있게 DB 인스턴스에 퍼블릭 IP 주소를 할당합니다. VPC 내부에서만 DB 인스턴스에 액세스할 수 있게 하려면 No 를 선택합니다. 퍼블릭 액세스가 불가능하도록 DB 인스턴스를 숨기는 방법에 대한 자세한 내용은 VPC에 있는 DB 인스턴스를 인터넷에서 숨기기 (p. 416) 단원을 참조하십시오.
가용 영역	가용 영역을 지정하지 않으려면 기본값으로 No Preference 를 사용합니다.
[VPC Security Group]	AWS를 처음 사용하는 경우에는 기본 VPC를 선택합니다. 이전에 VPC 보안 그룹을 만들었으면 해당 VPC 보안 그룹을 선택합니다. RDS 콘솔에서 새 VPC 보안 그룹 생성을 선택하는 경우 브라우저에서 검색된 IP 주소에서 DB 인스턴스에 액세스하도록 허용하는 발신 규칙을 사용하여 새 보안 그룹이 생성됩니다.
데이터베이스 이름	데이터베이스의 이름을 최대 63자의 영숫자 문자로 입력합니다. 이름을 제공하지 않으면 기본 "postgres" 데이터베이스가 생성됩니다. 추가 데이터베이스를 생성하려면 DB 인스턴스에 연결한 다음 SQL 명령어 <code>CREATE DATABASE</code> 를 사용하십시오. DB 인스턴스 연결에 대한 자세한 내용은 PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스에 연결하기 (p. 953) 단원을 참조하십시오.
데이터베이스 포트	데이터베이스에 액세스하는 데 사용할 포트를 지정합니다. PostgreSQL 설치의 기본값은 포트 5432로 설정됩니다.
DB 파라미터 그룹	고유한 파라미터 그룹을 만들지 않았다면 기본값을 사용합니다.

이 파라미터의 경우...	수행할 작업
옵션 그룹	고유한 옵션 그룹을 만든 경우가 아니면 기본값을 사용합니다.
스냅샷으로 태그 복사	스냅샷을 생성할 때 DB 인스턴스 태그를 DB 스냅샷에 복사하려면 이 옵션을 선택합니다. 자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.
[Enable Encryption]	이 DB 인스턴스에 대해 비활성화되어 있는 암호화를 활성화하려면 [Yes]를 선택합니다. 자세한 내용은 Amazon RDS 리소스 암호화 (p. 385) 단원을 참조하십시오.
백업 보존 기간	데이터베이스에서 자동 백업을 지속할 일수를 설정합니다. 테스트를 위해 이 값을 1로 설정할 수 있습니다.
백업 기간	데이터베이스를 백업할 특정 시간을 지정하지 않으려면 기본값으로 No Preference 를 사용합니다.
확장 모니터링 활성화	실시간 OS 모니터링을 사용하려면 Yes 를 선택합니다. Amazon RDS는 DB 인스턴스가 실행되는 운영 체제(OS)에 대한 측정치를 실시간으로 제공합니다. Amazon CloudWatch Logs이 제공하는 프리 티어를 초과하는 Enhanced Monitoring에 대해서만 비용이 청구됩니다.
역할 모니터링	기본 IAM 역할을 사용하려면 Default 를 선택합니다.
Granularity	매분마다 인스턴스를 모니터링하려면 60 을 선택합니다.
마이너 버전 자동 업그레이드	기본 마이너 DB 엔진 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다. Amazon RDS는 유지 관리 기간에 마이너 버전 자동 업그레이드를 수행합니다.
유지 관리 기간	대기 중인 DB 인스턴스의 설정 변경을 적용하려면 30 분으로 선택합니다. 시간이 중요하지 않은 경우 No Preference 를 선택합니다.
삭제 방지 활성화	DB 인스턴스가 삭제되지 않도록 방지하려면, 삭제 방지를 활성화합니다. AWS Management 콘솔을 사용하여 프로덕션 DB 인스턴스를 생성할 경우 기본적으로 삭제 방지가 활성화됩니다. 자세한 내용은 DB 인스턴스 삭제 (p. 128) 단원을 참조하십시오.

9. 최종 페이지에서 데이터베이스 생성을 선택합니다.
10. Amazon RDS 콘솔의 DB 인스턴스 목록에 새로운 DB 인스턴스가 나타납니다. DB 인스턴스를 만들고 사용할 준비가 될 때까지 DB 인스턴스의 상태는 *creating*입니다. 상태가 *available*로 변경되면 DB 인스턴스에 연결할 수 있습니다. DB 인스턴스 클래스와 할당된 저장소에 따라 새 인스턴스를 사용할 수 있을 때까지 몇 분 정도 걸릴 수 있습니다.

The screenshot shows the AWS RDS Databases console. At the top, there's a navigation bar with 'RDS' and 'Databases'. Below it is a search bar labeled 'Filter databases'. The main area is a table titled 'Databases' with columns: DB Name, Role, Engine, Region & AZ, Size, Status, and CPU. The table lists several database instances:

DB Name	Role	Engine	Region & AZ	Size	Status	CPU
mymariadb	Instance	MariaDB	us-west-2a	db.t2.small	Available	
mysql-instance1	Master	MySQL	us-west-2a	db.t2.small	Available	
testmysql-rr	Replica	MySQL	us-west-2c	db.t2.small	Available	
mysqlserver-instance1	Instance	SQL Server Express Edition	us-west-2b	db.t2.small	Available	
oracle-instance1	Instance	Oracle Enterprise Edition	us-west-2c	db.t2.small	Available	
postgres-test	Instance	PostgreSQL	us-west-2b	db.t2.small	Available	
testauroramysql-cl	Regional	Aurora MySQL	us-west-2	2 instances	Available	
testauroramysql	Writer	Aurora MySQL	us-west-2c	db.r4.xlarge	Available	
testauroramysql-us-west-2a	Reader	Aurora MySQL	us-west-2a	db.r4.xlarge	Available	

CLI

PostgreSQL DB 인스턴스를 생성하려면 AWS CLI `create-db-instance` 명령을 다음 파라미터와 함께 사용합니다.

- `--db-instance-identifier`
- `--allocated-storage`
- `--db-instance-class`
- `--engine`
- `--master-username`
- `--master-user-password`

Example

Linux, OS X, Unix의 경우:

```
aws rds create-db-instance
  --db-instance-identifier pgdbinstance \
  --allocated-storage 20 \
  --db-instance-class db.t2.small \
  --engine postgres \
  --master-username masterawsuser \
  --master-user-password masteruserpassword
```

Windows의 경우:

```
aws rds create-db-instance
  --db-instance-identifier pgdbinstance ^
  --allocated-storage 20 ^
  --db-instance-class db.t2.small ^
  --engine postgres ^
  --master-username masterawsuser ^
  --master-user-password masteruserpassword
```

이 명령은 다음과 유사한 출력을 생성합니다.

```
DBINSTANCE pgdbinstance db.t2.small postgres 20 sa creating 3 **** n 9.3
SECGROUP default active
PARAMGRP default.PostgreSQL9.3 in-sync
```

API

PostgreSQL DB 인스턴스를 생성하려면 Amazon RDS API [CreateDBInstance](#) 명령을 다음 파라미터와 함께 사용합니다.

- Engine = *postgres*
- DBInstanceIdentifier = *pgdbinstance*
- DBInstanceClass = *db.t2.small*
- AllocatedStorage = *20*
- BackupRetentionPeriod = *3*
- MasterUsername = *masterawsuser*
- MasterUserPassword = *masteruserpassword*

Example

```
https://rds.amazonaws.com/
    ?Action=CreateDBInstance
    &AllocatedStorage=20
    &BackupRetentionPeriod=3
    &DBInstanceClass=db.t2.small
    &DBInstanceIdentifier=pgdbinstance
    &DBName=mydatabase
    &DBSecurityGroups.member.1=mysecuritygroup
    &DBSubnetGroup=mydbsubnetgroup
    &Engine=postgres
    &MasterUserPassword=<masteruserpassword>
    &MasterUsername=<masterawsuser>
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2013-09-09
    &X-Amz-Algorithm=AWS4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20140212/us-west-2/rds/aws4_request
    &X-Amz-Date=20140212T190137Z
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
    &X-Amz-Signature=60d520ca0576c191b9eac8dbfe5617ebb6a6a9f3994d96437a102c0c2c80f88d
```

PostgreSQL 데이터베이스 엔진 기반 DB 인스턴스 에 연결하기

Amazon RDS가 DB 인스턴스를 프로비저닝한 후에는 표준 SQL 클라이언트 애플리케이션을 사용해 인스턴스에 연결할 수 있습니다. Amazon RDS DB 인스턴스의 세부 정보를 나열하려면 AWS Management 콘솔, AWS CLI [describe-db-instances](#) 명령 또는 Amazon RDS API [DescribeDBInstances](#) 작업을 사용할 수 있습니다. 다음 정보를 제공해야 연결할 수 있습니다.

- DB 인스턴스의 호스트나 호스트 이름에 대한 예입니다.

```
myinstance.123456789012.us-east-1.rds.amazonaws.com
```

- DB 인스턴스 서버가 수신 대기하는 포트입니다. 예를 들어, PostgreSQL 포트의 기본값은 5432입니다.
- DB 인스턴스의 사용자 이름 및 암호입니다.

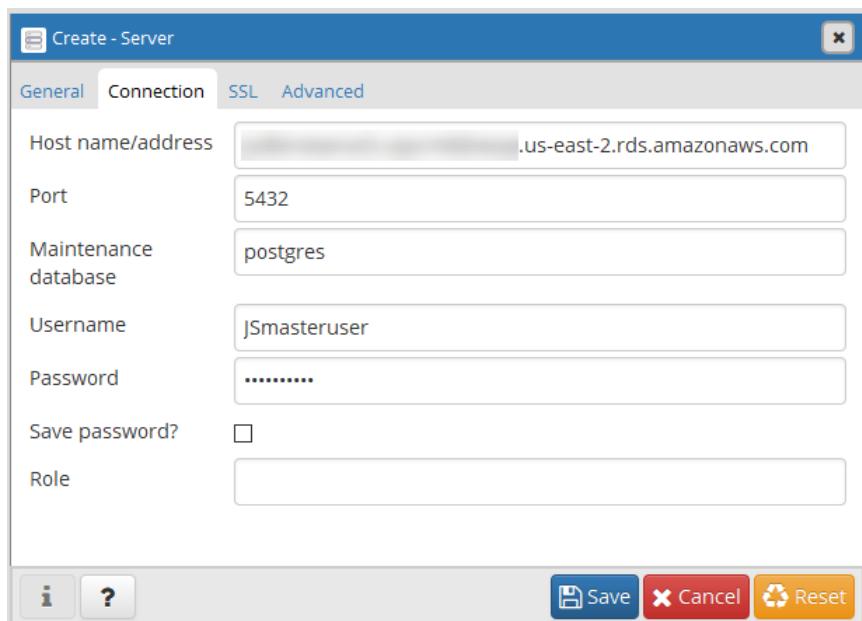
다음은 PostgreSQL DB 인스턴스에 연결하는 두 가지 방법입니다. 첫 번째 예에서는 PostgreSQL의 잘 알려진 오픈 소스 관리 및 개발 도구인 pgAdmin을 사용합니다. 두 번째 예에서는 PostgreSQL 설치에 속하는 명령행 유ти리티인 psql을 사용합니다.

pgAdmin을 사용해 PostgreSQL DB 인스턴스에 연결

또한, 오픈 소스 도구인 pgAdmin을 사용하여 PostgreSQL DB 인스턴스를 연결할 수 있습니다.

pgAdmin을 사용해 PostgreSQL DB 인스턴스에 연결하는 방법

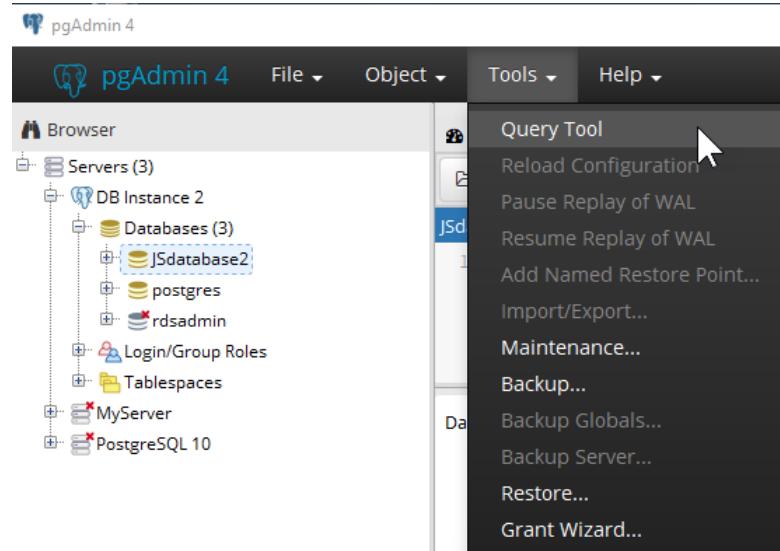
1. <http://www.pgadmin.org/>에서 pgAdmin을 설치하십시오. 클라이언트 컴퓨터에 PostgreSQL의 로컬 인스턴스가 없어도 pgAdmin을 다운로드하고 사용할 수 있습니다.
2. 클라이언트 컴퓨터에서 pgAdmin 애플리케이션을 실행합니다.
3. [Dashboard] 탭에서 [Add New Server]를 선택합니다.
4. [Create - Server] 대화 상자에서 pgAdmin의 서버를 식별하기 위해 [General] 탭에 이름을 입력합니다.
5. [Connection] 탭에서 사용자 DB 인스턴스에 있는 다음 정보를 입력합니다.
 - Host(호스트)에 엔드포인트를 입력합니다(예: mypostgresql.c6c8dntfzzhgv0.us-east-2.rds.amazonaws.com).
 - [Port]에 할당된 포트를 입력합니다.
 - [username]에서 DB 인스턴스를 생성할 때 입력했던 사용자 이름을 입력합니다.
 - Password(암호)에 DB 인스턴스를 생성할 때 입력했던 암호를 입력합니다.



6. Save를 선택합니다.

연결 문제는 [연결 문제 해결 \(p. 956\)](#)를 참조하십시오.

7. pgAdmin 브라우저에서 데이터베이스에 액세스 하려면, [Servers], DB 인스턴스, [Databases]를 확장합니다. DB 인스턴스의 데이터베이스 이름을 선택합니다.



8. SQL 명령을 입력할 수 있는 패널을 열려면, [Tools], [Query Tool]를 선택합니다.

psql을 사용해 PostgreSQL DB 인스턴스에 연결

psql 명령줄 유틸리티의 로컬 인스턴스를 사용하여 PostgreSQL DB 인스턴스에 연결할 수 있습니다. 클라이언트 컴퓨터에 PostgreSQL 또는 psql 클라이언트를 설치해야 할 수도 있습니다. psql을 사용해 PostgreSQL DB 인스턴스에 연결하려면 호스트 정보 및 액세스 자격 증명을 제공해야 합니다.

다음 형식 중 하나를 사용하여 Amazon RDS의 PostgreSQL DB 인스턴스에 연결합니다. 연결이 되면 암호를 입력하라는 메시지가 표시됩니다. 배치 작업이나 스크립트에는 --no-password 옵션을 사용합니다.

Unix는 다음 형식을 사용합니다.

```
psql \
--host=<DB instance endpoint> \
--port=<port> \
--username=<master user name> \
--password \
--dbname=<database name>
```

Windows는 다음 형식을 사용합니다.

```
psql ^
--host=<DB instance endpoint> ^
--port=<port> ^
--username=<master user name> ^
--password ^
--dbname=<database name>
```

예를 들어 다음 명령은 가상 자격 증명을 사용해 mypostgresql이라는 PostgreSQL DB 인스턴스에서 mypgdb라는 데이터베이스에 연결합니다.

```
psql --host=mypostgresql.c6c8mwvfdgv0.us-west-2.rds.amazonaws.com --port=5432 --
username=awsuser --password --dbname=mpgdb
```

연결 문제 해결

DB 인스턴스에 연결할 수 없도록 만드는 가장 많은 오류가 `Could not connect to server: Connection timed out.`입니다. 이 오류 메시지가 표시되면 다음을 조치합니다.

- 사용한 호스트 이름이 DB 인스턴스 엔드포인트이고 포트 번호가 올바른지 확인하십시오.
- DB 인스턴스의 퍼블릭 액세스 가능성을 [Yes]로 설정하십시오.
- DB 인스턴스에 할당된 보안 그룹에 모든 연결이 통과할 수 있는 방화벽을 통한 액세스를 허용하는 데 필요 한 규칙이 있는지 확인하십시오. 기본값 포트 5432를 사용해 DB 인스턴스를 생성했는데, 기업 방화벽 규 칙이 기업 장치의 해당 포트 연결을 차단하는 경우를 예로 들 수 있습니다.

이 오류를 수정하려면 다른 포트를 사용하도록 DB 인스턴스를 수정해야 합니다. 또 DB 인스턴스에 적용 된 보안 그룹의 새로운 포트에 대한 연결이 허용되어야 합니다.

- 애플리케이션이 실행되고 있는 디바이스 또는 Amazon EC2 인스턴스의 연결을 승인하지 않는 보안 그룹 을 사용하여 DB 인스턴스를 생성했는지 여부를 확인합니다. 연결이 되도록 만들려면, 생성 때 DB 인스턴 스에 할당한 보안 그룹이 DB 인스턴스 액세스를 허용해야 합니다. 예를 들어, DB 인스턴스가 VPC 내부 에서 생성된 경우 이 인스턴스의 VPC 보안 그룹에서 연결 권한을 부여해야 합니다. 또한, DB 인스턴스가 VPC 외부에서 생성된 경우에는 이 인스턴스의 데이터베이스 보안 그룹에서 이 연결을 승인해야 합니다.

지금까지 가장 많이 발생하고 있는 연결 문제는 DB 인스턴스에 할당된 보안 그룹의 액세스 규칙과 관련이 있 습니다. DB 인스턴스를 만들 때 기본 DB 보안 그룹을 사용한 경우 인스턴스에 액세스할 수 있도록 허용하는 규칙이 보안 그룹에 없을 확률이 큽니다. Amazon RDS 보안 그룹에 대한 자세한 내용은 [보안 그룹을 통한 액 세스 제어 \(p. 390\)](#) 단원을 참조하십시오.

PostgreSQL 데이터베이스 엔진을 실행 중인 DB 인 스탄스 수정

추가 스토리지를 더하거나 DB 인스턴스 클래스를 변경하는 것과 같은 작업을 완수하기 위해 DB 인스턴스의 설정을 변경할 수 있습니다. 이 주제에서는 Amazon RDS PostgreSQL DB 인스턴스를 수정하는 과정을 안내 하고, PostgreSQL 인스턴스에 대한 설정을 설명합니다. Amazon RDS DB 인스턴스의 이름 바꾸기, 재부팅, 삭제, 태그 지정 또는 업그레이드와 같은 추가 작업에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 수명 주기 \(p. 111\)](#) 단원을 참조하십시오. 프로덕션 인스턴스를 변경하기 전에 테스트 인스턴스에서 변경 사항을 테스트하면 변경 사항이 미칠 영향을 더 정확하게 이해하는 데 도움이 됩니다. 이는 특히 데이터베이스 버전 을 업그레이드할 때 중요합니다.

변경 사항을 즉시 적용하거나 DB 인스턴스의 다음 유지 관리 기간 중에 적용할 수 있습니다. 경우에 따라 변 경 사항을 즉시 적용하면 인스턴스가 중단될 수도 있습니다. DB 인스턴스 설정 변경 시 [Apply Immediately] 옵션이 미치는 영향에 대한 자세한 내용은 [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#) 단원을 참조하십시오.

AWS Management 콘솔

PostgreSQL DB 인스턴스를 수정하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔 을 엽니다.
2. 탐색 창에서 데이터베이스를 선택한 다음 변경하려는 DB 인스턴스를 선택합니다.
3. [Modify]를 선택합니다. [Modify DB Instance] 페이지가 나타납니다.
4. 원하는 설정을 모두 변경합니다. 각 설정에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 설 정 \(p. 958\)](#) 단원을 참조하십시오.
5. 원하는 대로 모두 변경되었으면 [Continue]를 선택합니다.
6. 변경 사항을 즉시 적용하려면 즉시 적용을 선택합니다. 일부의 경우 이 옵션을 선택하면 중단이 발생할 수 있습니다. 자세한 내용은 [즉시 적용 파라미터 사용 \(p. 112\)](#) 단원을 참조하십시오.

7. 확인 페이지에서 변경 내용을 검토합니다. 변경 내용이 정확할 경우 [Modify DB Instance]를 선택하여 변경 내용을 저장합니다.

그렇지 않으면 [Back]를 선택하여 변경 내용을 편집하거나 [Cancel]을 선택하여 변경 내용을 취소합니다.

CLI

PostgreSQL DB 인스턴스를 수정하려면 AWS CLI 명령 [modify-db-instance](#)를 사용합니다.

Example

다음은 백업 보존 기간을 1주일(7일)로 설정하고 마이너 버전 자동 업그레이드를 비활성화하여 pgdbinstance를 수정하는 코드입니다. 이 변경 사항은 다음 유지 관리 기간에 적용됩니다.

파라미터

- `--db-instance-identifier` - DB 인스턴스의 이름
- `--backup-retention-period`—자동 백업을 보존할 일수
- `--auto-minor-version-upgrade` - 마이너 버전 자동 업그레이드를 허용합니다. 마이너 버전 자동 업그레이드를 허용하지 않으려면 `--no-auto-minor-version-upgrade`를 사용합니다.
- `--no-apply-immediately`—변경 사항이 다음 유지 관리 기간에 적용됩니다. 변경 사항을 바로 적용하려면 `--apply-immediately`를 사용합니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
  --db-instance-identifier pgdbinstance \
  --backup-retention-period 7 \
  --auto-minor-version-upgrade \
  --no-apply-immediately
```

Windows의 경우:

```
aws rds modify-db-instance ^
  --db-instance-identifier pgdbinstance ^
  --backup-retention-period 7 ^
  --auto-minor-version-upgrade ^
  --no-apply-immediately
```

API

PostgreSQL DB 인스턴스를 수정하려면 [ModifyDBInstance](#) 작업을 사용합니다.

Example

다음은 백업 보존 기간을 1주일(7일)로 설정하고 마이너 버전 자동 업그레이드를 비활성화하여 pgdbinstance를 수정하는 코드입니다. 이 변경 사항은 다음 유지 관리 기간에 적용됩니다.

파라미터

- `DBInstanceIdentifier` - DB 인스턴스의 이름
- `BackupRetentionPeriod`—자동 백업을 보존할 일수
- `AutoMinorVersionUpgrade=true` - 마이너 버전 자동 업그레이드를 허용합니다. 마이너 버전 자동 업그레이드를 허용하지 않으려면 값을 `false`로 설정합니다.

- **ApplyImmediately=false**—변경 사항이 다음 유지 관리 기간에 적용됩니다. 변경 사항을 바로 적용하려면 값을 **true**로 설정합니다.

```
https://rds.us-east-1.amazonaws.com/
?Action=ModifyDBInstance
&ApplyImmediately=false
&AutoMinorVersionUpgrade=true
&BackupRetentionPeriod=7
&DBIdentifier=mydbinstance
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2013-09-09
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20131016/us-east-1/rds/aws4_request
&X-Amz-Date=20131016T233051Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=087a8eb41cb1ab0fc9ec1575f23e73757fffc6a1e42d7d2b30b9cc0be988cff97
```

PostgreSQL DB 인스턴스 설정

다음 표에는 수정이 가능한 설정, 수정이 불가능한 설정, 변경 사항을 적용할 수 있는 시점, 변경 사항으로 인한 DB 인스턴스 가동 중지 여부에 대한 자세한 내용이 나와 있습니다.

설정	설정 설명	변경이 발생할 때	가동 중지 참고 사항
할당된 스토리지	<p>DB 인스턴스에 할당할 스토리지(단위: GB). 할당된 스토리지를 늘릴 수만 있고 줄일 수는 없습니다.</p> <p>DB 인스턴스 상태가 스토리지 최적화이거나 지난 6시간 동안 DB 인스턴스에 할당된 스토리지가 수정된 경우 할당된 스토리지를 수정할 수 없습니다.</p> <p>허용 가능한 최대 스토리지는 스토리지 유형에 따라 다릅니다. 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.</p>	[Apply Immediately]가 true 로 설정되는 즉시 변경됩니다. [Apply Immediately]가 false 로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.	가동 중지 없음. 변경 도중 성능이 저하될 수 있습니다.
マイ너 버전 자동 업그레이드	기본 마이너 DB 엔진 버전 업그레이드가 제공되면 DB 인스턴스가 자동으로 이를 받을 수 있게 하려면 마이너 버전 자동 업그레이드 사용을 선택합니다. Amazon RDS는 유지 관리 기간에 마이너 버전 자동 업그레이드를 수행합니다.	-	-
백업 보존 기간	<p>자동 백업을 보존할 일수. 자동 백업을 비활성화하기 위해 백업 보존 기간을 0으로 설정합니다.</p> <p>자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되고 이 설정을 0이 아닌 값에서 다른 0이 아닌 값으로 변경하면 비동기 방식이지만 최</p>	백업 보존 기간을 0에서 0이 아닌 값으로 또는 0이 아닌 값에서 0으로 변경할 경우 인스턴스가 종단됩니다.

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
		대한 빠른 시간 내에 변경 사항이 적용됩니다. 이때 적용되지 않을 경우, 다음 유지 관리 기간에 변경 사항이 적용됩니다.	
백업 기간	<p>데이터베이스의 자동 백업이 실행되는 기간. 백업 기간은 국제 표준시(UTC)의 시작 시간과 시간 단위의 지속 기간으로 구성됩니다.</p> <p>자세한 내용은 백업 작업 (p. 196) 단원을 참조하십시오.</p>	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다.	-
인증 기관	사용할 인증서.	-	-
스냅샷으로 태그 복사	<p>DB 인스턴스 태그가 있는 경우 이 옵션은 DB 스냅샷을 만들 때 태그를 복사합니다.</p> <p>자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-
데이터베이스 포트	<p>데이터베이스에 액세스하는 데 사용할 포트.</p> <p>포트 값은 DB 인스턴스의 옵션 그룹에서 옵션에 대해 지정한 포트 값과 일치하지 않아야 합니다.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply Immediately] 설정을 무시합니다.	DB 인스턴스가 즉시 재부팅됩니다.
DB 엔진 버전	사용할 PostgreSQL 데이터베이스 엔진의 버전입니다. 프로덕션 DB 인스턴스를 업그레이드하려면 먼저 테스트 인스턴스에서 업그레이드 프로세스를 테스트하여 업그레이드 시간을 확인하고 애플리케이션을 검증하는 것이 좋습니다.	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다.
DB 인스턴스 클래스	<p>사용할 DB 인스턴스 클래스.</p> <p>자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	이 변경 도중 인스턴스가 중단됩니다.

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
DB 인스턴스 식별자	<p>DB 인스턴스 식별자 이 값은 소문자 문자열로 저장됩니다.</p> <p>DB 인스턴스 이름 바꾸기의 영향에 대한 자세한 내용은 DB 인스턴스 이름 바꾸기 (p. 121) 단원을 참조하십시오.</p>	[Apply Immediately]가 true로 설정되는 즉시 변경됩니다. [Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.	이 변경 도중 인스턴스가 종단됩니다. DB 인스턴스가 재부팅됩니다.
DB 파라미터 그룹	<p>DB 인스턴스와 연결할 파라미터 그룹.</p> <p>자세한 내용은 DB 파라미터 그룹 작업 (p. 161) 단원을 참조하십시오.</p>	파라미터 그룹 변경 사항은 즉시 적용됩니다.	<p>이 변경 도중 인스턴스가 종단되지 않습니다. 파라미터 그룹을 변경하는 경우 일부 파라미터에 대한 변경 내용은 재부팅 없이 DB 인스턴스에 즉시 적용됩니다. 다른 파라미터에 대한 변경 내용은 DB 인스턴스를 재부팅한 후에만 적용됩니다.</p> <p>자세한 내용은 DB 클러스터에서 DB 인스턴스 (p. 123) 단원을 참조하십시오.</p>
삭제 방지 활성화	DB 인스턴스가 삭제되지 않도록 방지하려면, 삭제 방지를 활성화합니다. 자세한 내용은 DB 인스턴스 삭제 (p. 128) 단원을 참조하십시오.	-	-
확장 모니터링	<p>DB 인스턴스가 실행되는 운영 체제에 대한 실시간 측정치 수집을 활성화하려면 [Yes].</p> <p>자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.</p>	-	-
IAM DB 인증	<p>이 DB 인스턴스에 대한 IAM 데이터베이스 인증을 활성화하기 위한 [Enable IAM DB authentication].</p> <p>자세한 내용은 MySQL 및 PostgreSQL을 위한 IAM 데이터베이스 인증 (p. 369) 단원을 참조하십시오.</p>	-	-
라이선스 모델	PostgreSQL의 라이선스 모델은 하나뿐이기 때문에 라이선스 모델을 변경 할 수 없습니다.	-	-
로그 내보내기	<p>Amazon CloudWatch Logs에 게시할 PostgreSQL 데이터베이스 로그 파일의 유형을 선택합니다.</p> <p>자세한 내용은 PostgreSQL 데이터베이스 로그 파일 (p. 331) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply immediately] 설정을 무시합니다.	-

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
유지 관리 기간	<p>시스템 유지 관리를 실행하는 기간. 시스템 유지 관리는 업그레이드를 포함합니다(해당할 경우). 유지 관리 기간은 국제 표준시(UTC)의 시작 시간과 시간 단위의 지속 기간으로 구성됩니다.</p> <p>이 기간을 현재 시간으로 설정하려면 대기 중인 변경 사항이 모두 적용될 수 있도록 현재 시간과 기간 종료 시간 사이에 최소 30분 이상 필요합니다.</p> <p>자세한 내용은 Amazon RDS 유지 관리 기간 (p. 116) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply Immediately] 설정을 무시합니다.	인스턴스가 종단될 수 있는 작업이 하나 이상 대기 중이고, 유지 관리 기간이 현재 시간을 포함하여 변경된 경우 대기 중인 작업들이 즉시 적용되고 인스턴스가 종단됩니다.
다중 AZ 배포	<p>다중 가용 영역에 DB 인스턴스를 배포하려면 [Yes], 그렇지 않으면 [No].</p> <p>자세한 내용은 리전 및 가용 영역 (p. 99) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다.</p> <p>[Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	-
새 마스터 암호	마스터 사용자의 암호. 암호는 8~30자 의 영숫자 문자로 구성되어야 합니다.	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이 설정은 [Apply Immediately] 설정을 무시합니다.	-
옵션 그룹	<p>PostgreSQL DB 인스턴스에 아무런 옵션도 사용할 수 없습니다.</p> <p>자세한 내용은 옵션 그룹 작업 (p. 148) 단원을 참조하십시오.</p>	-	-
퍼블릭 액세스 가능성	<p>DB 인스턴스에 퍼블릭 IP 주소를 부여하려면(즉 VPC 외부에서 액세스할 수 있음) [Yes]를 선택합니다. 공개적으로 액세스가 가능하려면 DB 인스턴스도 VPC의 퍼블릭 서브넷에 있어야 합니다. VPC 내부에서만 DB 인스턴스에 액세스할 수 있게 하려면 [No].</p> <p>자세한 내용은 VPC에 있는 DB 인스턴스를 인터넷에서 숨기기 (p. 416) 단원을 참조하십시오.</p>	변경 사항이 즉시 적용됩니다. 이 설정은 [Apply Immediately] 설정을 무시합니다.	-
보안 그룹	<p>DB 인스턴스와 연결할 보안 그룹.</p> <p>자세한 내용은 DB 보안 그룹 작업 (EC2-Classic 플랫폼) (p. 394) 단원을 참조하십시오.</p>	비동기 방식이지만 최대한 빠른 시간 내에 변경 사항이 적용됩니다. 이 설정은 [Apply Immediately] 설정을 무시합니다.	-

설정	설정 설명	변경이 발생할 때	자동 중지 참고 사항
스토리지 유형	<p>사용할 스토리지 유형. 자세한 내용은 Amazon RDS 스토리지 유형 (p. 101) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다. [Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	<p>다음과 같이 변경하면 프로세스가 시작되는 동안 잠시 작동이 중단됩니다. 그 이후에 변경 사항이 적용되는 동안 데이터베이스를 정상적으로 사용할 수 있습니다.</p> <ul style="list-style-type: none"> • [General Purpose (SSD)] → [Magnetic]. • 범용(SSD)에서 프로비저닝된 IOPS(SSD)로. DB 인스턴스가 단일 AZ이고 사용자 지정 파라미터 그룹을 사용하는 경우에만 중단이 발생합니다. 다른 AZ DB 인스턴스가 중단되지는 않습니다. • [Magnetic] → [General Purpose (SSD)]. • [Magnetic] → [Provisioned IOPS (SSD)]. • [Provisioned IOPS (SSD)] → [Magnetic]. • 프로비저닝된 IOPS(SSD)에서 범용 (SSD)로. DB 인스턴스가 단일 AZ이고 사용자 지정 파라미터 그룹을 사용하는 경우에만 중단이 발생합니다. 다른 AZ DB 인스턴스가 중단되지는 않습니다.
[Subnet Group]	<p>DB 인스턴스에 대한 서브넷 그룹. 이 설정을 사용하여 DB 인스턴스를 다른 VPC로 이동할 수 있습니다. DB 인스턴스가 VPC 내에 있지 않을 경우 이 설정을 사용하여 DB 인스턴스를 VPC 내로 이동할 수 있습니다.</p> <p>자세한 내용은 VPC에 있지 않은 DB 인스턴스를 VPC로 이동 (p. 420) 단원을 참조하십시오.</p>	<p>[Apply Immediately]가 true로 설정되는 즉시 변경됩니다. [Apply Immediately]가 false로 설정되면 다음 유지 관리 기간에 변경 사항을 적용합니다.</p>	<p>이 변경 도중 인스턴스가 중단됩니다. DB 인스턴스가 재부팅됩니다.</p>

Amazon RDS용 PostgreSQL DB 엔진 업그레이드

Amazon RDS에서 새 데이터베이스 엔진 버전을 지원하는 경우, DB 인스턴스를 새 버전으로 업그레이드할 수 있습니다. PostgreSQL DB 인스턴스의 업그레이드에는 메이저 버전 업그레이드와 마이너 버전 업그레이드라는 두 가지 업그레이드가 있습니다.

메이저 버전 업그레이드에는 기존 애플리케이션과 호환되지 않는 데이터베이스 변경 사항이 포함될 수 있습니다. 따라서 DB 인스턴스의 메이저 버전 업그레이드를 수동으로 수행해야 합니다. DB 인스턴스를 수정하여 메이저 버전 업그레이드를 시작할 수 있습니다. 그러나 주 버전 업그레이드를 수행하기 전에 [PostgreSQL 메이저 버전 업그레이드 \(p. 964\)](#)에 설명된 단계를 수행하는 것이 좋습니다.

반대로 마이너 버전 업그레이드에는 기존 애플리케이션과 호환되는 변경 사항만 포함됩니다. DB 인스턴스를 수정하여 마이너 버전 업그레이드를 수동으로 시작할 수 있습니다. 또는 DB 인스턴스를 생성하거나 수정할 때 마이너 버전 자동 업그레이드 옵션을 활성화할 수 있습니다. 이렇게 하면 Amazon RDS에서 새 버전을 테스트 및 승인한 후 DB 인스턴스가 자동으로 업그레이드됩니다. 자세한 내용은 [PostgreSQL 마이너 버전 자동 업그레이드 \(p. 967\)](#) 단원을 참조하십시오.

주제

- [PostgreSQL 업그레이드 개요 \(p. 963\)](#)
- [PostgreSQL 메이저 버전 업그레이드 \(p. 964\)](#)
- [PostgreSQL 마이너 버전 자동 업그레이드 \(p. 967\)](#)

PostgreSQL 업그레이드 개요

DB 인스턴스를 안전하게 업그레이드하기 위해 Amazon RDS는 [PostgreSQL 설명서](#)에 있는 pg_upgrade 유ти리티를 사용합니다.

백업 보존 기간이 0보다 큰 경우 업그레이드 프로세스 종에 Amazon RDS가 두 개의 DB 스냅샷을 사용합니다. 첫 번째 DB 스냅샷은 업그레이드 변경 이전 DB 인스턴스의 스냅샷입니다. 업그레이드가 데이터베이스에 맞지 않는 경우에는 이 스냅샷을 복구하여 이전 버전의 DB 인스턴스를 생성할 수 있습니다. 두 번째 DB 스냅샷은 업그레이드 완료 이후에 캡처됩니다.

Note

DB 인스턴스에 대한 백업 보존 기간을 0보다 큰 수로 설정하는 경우에만 Amazon RDS가 업그레이드 프로세스 중에 DB 스냅샷을 캡처합니다. 백업 보존 기간을 변경하려면 [PostgreSQL 데이터베이스 엔진을 실행 중인 DB 인스턴스 수정 \(p. 956\)](#) 단원을 참조하십시오.

DB 인스턴스를 다른 AZ 배포로 생성한 경우에는 기본 라이터 DB 인스턴스와 예비 DB 인스턴스가 모두 업그레이드됩니다. 라이터 DB 인스턴스와 예비 DB 인스턴스가 모두 동시에 업그레이드되므로 업그레이드가 끝날 때까지 작동 중단을 겪게 됩니다.

업그레이드가 완료되면 이전 버전의 데이터베이스 엔진으로 되돌릴 수 없습니다. 이때 이전 버전으로 되돌리려면 업그레이드 전에 캡처한 DB 스냅샷을 복구하여 새로운 DB 인스턴스를 생성해야 합니다.

PostgreSQL 업그레이드는 PostgreSQL 확장 버전을 업그레이드하지 않습니다. 확장 버전을 업그레이드하려면 ALTER EXTENSION UPDATE 명령을 사용합니다. 예를 들어, PostgreSQL DB 엔진을 9.4.x에서 9.5.12로 업그레이드할 때 PostGIS를 업그레이드하려면 다음 명령을 실행합니다.

```
ALTER EXTENSION POSTGIS UPDATE TO '2.2.5'
```

지원되는 PostgreSQL 확장 버전 목록은 [Amazon RDS에서 지원되는 PostgreSQL 확장 기능 및 모듈 \(p. 1030\)](#) 단원을 참조하십시오.

Note

Amazon RDS PostgreSQL DB 인스턴스에서 PostGIS 확장 버전을 실행하는 경우 업그레이드하기 전에 PostGIS 설명서의 [PostGIS 업그레이드 지침](#)을 따르십시오.

PostgreSQL 메이저 버전 업그레이드

메이저 버전 업그레이드에는 이전 버전의 데이터베이스와 호환되지 않는 데이터베이스 변경 사항이 포함될 수 있습니다. 이 기능을 사용하면 기존 애플리케이션이 올바르게 작동하지 않을 수 있습니다.

따라서 Amazon RDS는 자동으로 메이저 버전 업그레이드를 적용하지 않습니다. 메이저 버전 업그레이드를 수행하려면 DB 인스턴스를 수동으로 수정합니다. 모든 업그레이드는 프로덕션 DB 인스턴스에 적용하기 전에 철저하게 테스트하여 애플리케이션이 올바르게 작동하는지 확인해야 합니다. PostgreSQL 메이저 버전 업그레이드를 수행할 때 [메이저 버전 업그레이드를 수행하는 방법 \(p. 965\)](#)에 설명된 단계를 따르는 것이 좋습니다.

읽기 전용 복제본은 메이저 버전 업그레이드를 수행할 수 없지만 읽기 전용 복제본의 원본 인스턴스는 가능합니다. 읽기 전용 복제본의 원본 인스턴스가 메이저 버전 업그레이드를 거치면 해당 원본 인스턴스의 모든 읽기 전용 복제본은 이전 엔진 버전으로 유지됩니다. 이 경우 읽기 전용 복제본은 원본 인스턴스에서 수행된 변경 사항을 더 이상 복제할 수 없습니다. 인스턴스가 다른 메이저 버전으로 업그레이드된 이후에는 읽기 전용 복제본을 승격하거나, 삭제한 후 다시 만드는 것이 좋습니다. 자세한 내용은 [PostgreSQL 읽기 전용 복제본 작업 \(p. 967\)](#) 단원을 참조하십시오.

여러 메이저 버전 선택

PostgreSQL 데이터베이스를 다음 메이저 버전으로 업그레이드 할 수 있습니다. 일부 PostgreSQL 데이터베이스 버전에서는 업그레이드 할 때 상위 메이저 버전으로 건너뛸 수 있습니다. 다음 표에는 원본 PostgreSQL 데이터베이스 버전과 업그레이드에 사용할 수 있도록 연결된 대상 메이저 버전을 나열되어 있습니다.

소스 버전	메이저 버전 대상	추가 메이저 버전 대상(PostGIS 확장 없음)
9.3.x	9.4.x	
9.3.23	9.4.x, 9.5.13	9.6.9
9.3.24	9.4.x, 9.5.14	9.6.10
9.3.25	9.4.x, 9.5.15	9.6.11
9.4.x	9.5.x	
9.4.20	9.5.x	11.1
9.4.21	9.5.x	10.7, 11.2
9.5.x	9.6.x	
9.5.15	9.6.x	11.1
9.5.16	9.6.x	10.7, 11.2
9.6.x	10.x	
9.6.11	10.x	11.1
9.6.12	10.x	11.2
10.x	11.x	

Important

업그레이드 대상은 원본 마이너 버전 이상과 동시에 릴리스된 상위 버전으로 활성화됩니다.

데이터베이스에서 PostGIS 확장을 사용하는 경우 일부 원본에서 대상 조합으로 메이저 버전을 건너뛸 수 없습니다. 이러한 상황에서는 원하는 대상 버전에 도달할 때까지 단계별로 다음 메이저 버전으로 업그레이드하십시오.

메이저 버전 업그레이드를 수행하는 방법

Amazon RDS PostgreSQL DB 인스턴스를 업그레이드할 때는 다음 프로세스를 따르는 것이 좋습니다.

- 버전 호환 가능 파라미터 그룹 준비 – 사용자 지정 파라미터 그룹을 사용하는 경우 두 가지 옵션이 있습니다. 새 DB 엔진 버전에 대한 기본 파라미터 그룹을 지정할 수 있습니다. 또는 새 DB 엔진 버전에 대한 사용자 지정 파라미터 그룹을 직접 만들 수도 있습니다.

새 파라미터 그룹을 DB 인스턴스와 연결하려면 업그레이드가 완료된 후 데이터베이스를 재부팅하십시오. 파라미터 그룹 변경 사항을 적용하기 위해 인스턴스를 재부팅해야 하는 경우, 인스턴스의 파라미터 그룹 상태가 `pending-reboot`로 표시됩니다. 인스턴스의 파라미터 그룹 상태는 콘솔에서 보거나 `describe-db-instances` 같은 `describe` 명령을 사용하여 볼 수 있습니다.

- 지원되지 않는 사용 확인:
 - 준비된 트랜잭션 – 업그레이드하기 전에 열려 있는 준비된 트랜잭션을 모두 커밋하거나 룰백합니다.

다음 쿼리를 사용하여 인스턴스에 열려 있는 준비된 트랜잭션이 없음을 확인할 수 있습니다.

```
SELECT count(*) FROM pg_catalog.pg_prepared_xacts;
```

- 행 데이터 유형 – RDS PostgreSQL 9.3 인스턴스를 업그레이드하는 경우 업그레이드를 시도하기 전에 `line` 데이터 형식의 사용을 모두 제거하십시오. PostgreSQL 버전 9.4 이하에는 `line` 데이터 형식이 완전히 구현되지 않았습니다.

`line` 데이터 형식이 사용되지 않음을 확인하려면 업그레이드 할 각 데이터베이스에서 다음 쿼리를 사용합니다.

```
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 = 'pg_catalog.line'::pg_catalog.regtype
AND c.relnamespace = n.oid
AND n.nspname !~ '^pg_temp_'
AND n.nspname !~ '^pg_toast_temp_'
AND n.nspname NOT IN ('pg_catalog', 'information_schema');
```

Note

인스턴스의 모든 데이터베이스를 나열하려면 다음 쿼리를 사용합니다.

```
SELECT d.datname FROM pg_catalog.pg_database d WHERE d.datallowconn = true;
```

- Reg* 데이터 형식 – 업그레이드를 시도하기 전에 `reg*` 데이터 형식의 사용을 모두 제거하십시오. `regtype` 및 `regclass` 이외에는 `reg*` 데이터 형식을 업그레이드할 수 없습니다. `pg_upgrade` 유ти리티는 Amazon RDS에서 업그레이드를 수행하는 데 사용하는 이 데이터 형식을 유지할 수 없습니다.

지원되지 않는 `reg*` 데이터 형식이 사용되지 않음을 확인하려면 각 데이터베이스에 다음 쿼리를 사용합니다.

```
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');
```

- VACUUM 수행 – 가동 중단을 줄이려면 DB 인스턴스를 업그레이드하기 전에 VACUUM 작업을 수행하십시오. VACUUM 작업을 수행하지 않으면 업그레이드 프로세스가 오래 걸릴 수 있습니다. 다른 메이저 버전으로 업그레이드할 때 pg_upgrade 유ти리티는 각 데이터베이스를 vacuum하기 때문입니다.
- 백업 수행 – 메이저 버전 업그레이드를 하기 전에 백업을 수행하여 데이터베이스에 대해 알려진 복원 지점을 생성하는 것이 좋습니다. 백업 보존 기간이 0보다 큰 경우 업그레이드 프로세스는 업그레이드 전과 후에 DB 인스턴스의 DB 스냅샷을 생성합니다. 백업 보존 기간을 변경하려면 [PostgreSQL 데이터베이스 엔진을 실행 중인 DB 인스턴스 수정 \(p. 956\)](#) 단원을 참조하십시오. 백업을 수동으로 수행하려면 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.
- [프로덕션 테스트 실행](#) – 데이터베이스에서 메이저 버전 업그레이드를 수행하기 전에 프로덕션 데이터베이스의 복제본에서 메이저 버전 업그레이드를 테스트하는 것이 좋습니다. 종복 테스트 인스턴스를 만들려면 최근 스냅샷에서 데이터베이스를 복원하거나 데이터베이스를 최근 복원 가능 시간으로 시점 복원할 수 있습니다. 자세한 내용은 [스냅샷에서 복구 \(p. 206\)](#) 또는 [DB 인스턴스를 지정된 시간으로 복원 \(p. 221\)](#) 단원을 참조하십시오. 업그레이드 수행에 대한 자세한 내용은 [엔진 버전 수동 업그레이드 \(p. 119\)](#) 단원을 참조하십시오.
- [프로덕션 인스턴스 업그레이드](#) – 메이저 버전 업그레이드의 테스트 실행이 성공한 경우 확신을 가지고 프로덕션 데이터베이스를 업그레이드해도 됩니다. 자세한 내용은 [엔진 버전 수동 업그레이드 \(p. 119\)](#) 단원을 참조하십시오.

Note

업그레이드 프로세스 중에 인스턴스를 특정 시점으로 복원할 수 없습니다. Amazon RDS가 업그레이드를 수행하면 인스턴스 자동 백업이 수행됩니다. 업그레이드를 시작하기 이전의 시간이나 인스턴스 자동 백업을 완료한 이후의 시간으로 시점 복원을 수행할 수 있습니다.

진행 중인 업그레이드에 대한 정보는 Amazon RDS를 사용하여 pg_upgrade 유ти리티가 생성하는 두 개의 로그인 pg_upgrade_internal.log 및 pg_upgrade_server.log를 볼 수 있습니다. Amazon RDS는 이러한 로그의 파일 이름에 타임스탬프를 추가합니다. 다른 로그와 마찬가지로 이러한 로그를 볼 수 있습니다. 자세한 내용은 [Amazon RDS 데이터베이스 로그 파일 \(p. 306\)](#) 단원을 참조하십시오.

메이저 버전 업그레이드 중에 public 및 template1 데이터베이스와 인스턴스의 모든 데이터베이스에 있는 public 스키마의 이름이 일시적으로 변경됩니다. 이러한 객체는 원래 이름 뒤에 임의의 문자열이 추가된 상태로 로그에 표시됩니다. 그러면 locale 및 owner와 같은 사용자 정의 설정이 메이저 버전 업그레이드 중에 유지되도록 문자열이 추가됩니다. 업그레이드가 완료되면 객체의 이름이 원래 이름으로 다시 변경됩니다.

Note

업그레이드를 완료한 후에는 ANALYZE 작업을 실행해 pg_statistic 테이블을 새로 고쳐야 합니다.

메이저 버전 업그레이드를 완료한 후 비슷한 워크로드로 업그레이드된 데이터베이스에서 애플리케이션을 테스트하여 모든 것이 예상대로 작동하는지 확인해 보십시오. 업그레이드를 확인한 후 이 테스트 인스턴스를 삭제할 수 있습니다.

PostgreSQL 마이너 버전 자동 업그레이드

DB 인스턴스를 생성하거나 수정할 때 마이너 버전 자동 업그레이드 옵션을 활성화하면 자동으로 인스턴스가 업그레이드될 수 있습니다. Amazon RDS에 의해 마이너 업그레이드가 테스트되고 승인된 후에는 마이너 버전 업그레이드가 자동으로 수행됩니다. 자세한 내용은 [마이너 엔진 버전 자동 업그레이드 \(p. 120\)](#) 단원을 참조하십시오. 마이너 버전 업그레이드를 수동으로 수행하려면 [엔진 버전 수동 업그레이드 \(p. 119\)](#) 단원을 참조하십시오.

PostgreSQL DB 인스턴스가 읽기 전용 복제본을 사용하는 경우 원본 인스턴스를 업그레이드하기 전에 읽기 전용 복제본부터 모두 업그레이드해야 합니다. DB 인스턴스를 다른 AZ 배포로 생성한 경우에는 라이터 복제본과 예비 복제본이 모두 업그레이드되어 업그레이드가 끝날 때까지 인스턴스를 사용할 수 없습니다.

PostgreSQL 읽기 전용 복제본 작업

일반적으로 읽기 전용 복제본을 사용하여 Amazon RDS DB 인스턴스 간 복제를 구성합니다. 읽기 전용 복제본에 대한 일반적인 정보는 [읽기 전용 복제본 작업 \(p. 135\)](#) 단원을 참조하십시오.

이 단원에는 PostgreSQL의 읽기 전용 복제본 작업에 대한 자세한 정보가 포함되어 있습니다.

주제

- [PostgreSQL을 사용한 읽기 전용 복제본 구성 \(p. 967\)](#)
- [PostgreSQL 읽기 전용 복제본 모니터링 \(p. 968\)](#)
- [PostgreSQL을 사용한 읽기 전용 복제본 제한 \(p. 968\)](#)
- [PostgreSQL 읽기 전용 복제본을 통한 복제 중단 \(p. 968\)](#)
- [PostgreSQL 읽기 전용 복제본의 문제 해결 \(p. 969\)](#)

PostgreSQL을 사용한 읽기 전용 복제본 구성

Amazon RDS PostgreSQL은 버전 9.3.5 이상부터 PostgreSQL의 기본 스트리밍 복제 기능을 사용하여 원본(Postgres 용어로 "마스터") DB 인스턴스의 읽기 전용 복제본을 생성합니다. 이 읽기 전용 복제본(PostgreSQL 용어로 "스탠바이") DB 인스턴스는 마스터 DB 인스턴스에서 비동기식으로 생성된 물리적 복제본입니다. 생성 방법은 이렇습니다. 먼저 Write Ahead Log(WAL) 데이터를 전송할 수 있도록 원본 DB 인스턴스와 읽기 전용 복제본 사이에 특별한 연결 채널이 만들어집니다. 그런 다음 PostgreSQL이 그때그때 데이터베이스 변경 사항을 비동기식으로 스트리밍합니다.

PostgreSQL은 "복제" 역할을 사용하여 스트리밍 복제를 실행합니다. 이 역할은 권한이 부여되기는 하지만 그렇다고 데이터까지 수정하지는 못합니다. PostgreSQL은 단일 프로세스를 통해 복제를 처리합니다.

임의의 DB 인스턴스를 원본 DB 인스턴스로 사용하려면 백업 보존 기간을 0이 아닌 다른 값으로 설정하여 원본 DB 인스턴스의 자동 백업을 활성화해야 합니다.

PostgreSQL 읽기 전용 복제본을 생성할 때는 마스터 DB 인스턴스를 중단할 필요가 없습니다. Amazon RDS가 원본 DB 인스턴스와 읽기 전용 복제본에 필요한 파라미터와 권한을 설정하기 때문에 서비스 중단은 발생하지 않습니다. 원본 DB 인스턴스를 캡처한 스냅샷이 읽기 전용 복제본이 됩니다. 읽기 전용 복제본을 삭제 하더라도 중단은 발생하지 않습니다.

원본 DB 인스턴스 하나에서 최대 5개까지 읽기 전용 복제본을 생성할 수 있습니다. 효과적인 복제를 위해서는 읽기 전용 복제본도 각각 원본 DB 인스턴스와 동일한 용량의 컴퓨팅 파워와 스토리지 리소스를 가져야 합니다. 원본 DB 인스턴스를 확장하면 읽기 전용 복제본도 확장해야 합니다.

Amazon RDS는 읽기 전용 복제본에서 파라미터가 호환되지 않아 읽기 전용 복제본을 시작하지 못할 경우에는 호환되지 않는 파라미터를 모두 무시합니다. 예를 들어 `max_connections` 파라미터 값이 읽기 전용 복

제본보다 원본 DB 인스턴스에서 더 크다고 가정하겠습니다. 이 경우 가 원본 DB 인스턴스의 파라미터 값이 동일하도록 읽기 전용 복제본의 파라미터를 업데이트합니다.

반면 PostgreSQL DB 인스턴스는 원본과 읽기 전용 복제본 인스턴스 모두 `ssl` 파라미터를 1로 설정하여 암호화가 가능하도록 함으로써 안전하게 연결합니다.

읽기 전용 복제본은 단일 AZ 또는 다중 AZ DB 인스턴스 배포를 통해서도 생성할 수 있습니다. 다중 AZ 배포는 중요 데이터의 내구성과 가용성을 개선하는 데 효과적이지만 읽기 전용 쿼리를 실행하는 데 보조로 사용할 수는 없습니다. 그 대신 읽기 전용 쿼리 부하를 줄일 목적으로 트래픽이 많은 다중 AZ DB 인스턴스에서 읽기 전용 복제본을 생성할 수 있습니다. 다중 AZ 배포의 원본 인스턴스가 보조 인스턴스로 장애 조치된 경우에는 연결된 모든 읽기 전용 복제본이 자동으로 전환되어 보조(이제는 기본) 인스턴스를 복제 원본으로 사용합니다. 자세한 내용은 [Amazon RDS를 위한 고가용성\(다중 AZ\) \(p. 109\)](#) 단원을 참조하십시오.

읽기 전용 복제본을 Multi-AZ DB 인스턴스를 생성할 수 있습니다. Amazon RDS는 복제본에 대한 장애 조치 지원을 위해 다른 가용 영역에 복제본의 대기를 생성합니다. 읽기 전용 복제본을 다중 AZ DB 인스턴스로 생성하는 작업은 원본 데이터베이스가 다중 AZ DB 인스턴스인지 여부와는 독립적입니다.

`postgres_fdw` 확장을 사용하여 원격 서버에서 데이터로 액세스하려면 읽기 전용 복제본도 원격 서버로 액세스해야 합니다. `postgres_fdw` 사용에 관한 자세한 내용은 [postgres_fdw 확장으로 외부 데이터 액세스 \(p. 1009\)](#) 단원을 참조하십시오.

PostgreSQL 읽기 전용 복제본 모니터링

PostgreSQL 읽기 전용 복제본의 경우 Amazon RDS `ReplicaLag` 지표를 보면서 Amazon CloudWatch의 복제 지연을 모니터링할 수 있습니다. `ReplicaLag` 지표는 `SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS slave_lag`의 값을 보고합니다.

PostgreSQL을 사용한 읽기 전용 복제본 제한

PostgreSQL 읽기 전용 복제본에 대한 제한 사항은 다음과 같습니다.

- 각 PostgreSQL 읽기 전용 복제본은 읽기 전용이라서 쓰기가 가능한 읽기 전용 복제본으로 만들 수 없습니다.
- 다른 읽기 전용 복제본에서 읽기 전용 복제본을 생성할 수 없습니다(즉, Cascading 방식의 읽기 전용 복제본은 생성할 수 없습니다).
- PostgreSQL 읽기 전용 복제본은 새로운 원본 DB 인스턴스로 승격할 수 없습니다. 하지만 읽기 전용 복제본은 자동으로 새로운 원본 DB 인스턴스가 되지 않습니다. 읽기 전용 복제본을 승격하면 WAL 통신을 수신하지 못하고 더 이상 읽기 전용 인스턴스의 역할을 하지 못합니다. 승격된 읽기 전용 복제본은 새로운 원본 DB 인스턴스와 다름없기 때문에 예정되어 있는 모든 복제를 그대로 진행하도록 준비해야 합니다.
- PostgreSQL 읽기 전용 복제본은 원본 DB 인스턴스에서 사용자 트랜잭션이 없는 경우 최대 5분까지 복제 지연을 보고합니다.

PostgreSQL 읽기 전용 복제본을 통한 복제 중단

몇몇 상황에서는 PostgreSQL 원본 DB 인스턴스가 읽기 전용 복제본을 이용한 복제를 우발적으로 중단할 수 있습니다. 이러한 상황은 다음과 같습니다.

- `max_wal_senders` 파라미터가 너무 낮게 설정되어 읽기 전용 복제본의 수에 충분한 데이터를 제공하지 못하는 경우 복제가 중단됩니다.
- PostgreSQL 파라미터인 `wal_keep_segments`는 데이터를 읽기 전용 복제본으로 보낼 때 유지할 WAL 파일 수를 결정합니다. 이 파라미터 값에 따라 유지할 로그 수가 결정됩니다. 이 파라미터 값이 너무 낮으면 읽기 전용 복제본이 너무 멀리 뒤처지면서 스트리밍 복제가 중단될 수 있습니다. 이 경우에는 Amazon RDS가 복제 오류를 보고한 후 원본 DB 인스턴스에 보관된 WAL 로그를 재실행하여 읽기 전용 복제본의

복구를 시작합니다. 이 복구 프로세스는 읽기 전용 복제본이 스트리밍 복제를 이어갈 만큼 충분히 따라잡을 때까지 계속됩니다. 자세한 내용은 [PostgreSQL 읽기 전용 복제본의 문제 해결 \(p. 969\)](#) 단원을 참조하십시오.

- PostgreSQL 읽기 전용 복제본은 원본 DB 인스턴스 엔드포인트가 변경되면 재부팅해야 합니다.

읽기 전용 복제본에 데이터를 전송하는 WAL 스트림이 중단되면 PostgreSQL이 복구 모드로 전환되면서 보관 중인 WAL 파일을 사용해 읽기 전용 복제본을 복구합니다. 복구 프로세스가 완료되면 PostgreSQL이 스트리밍 복제를 다시 구성합니다.

PostgreSQL 읽기 전용 복제본의 문제 해결

PostgreSQL은 리전 간 복제에 복제 슬롯을 사용하기 때문에 동일한 리전의 복제 문제와 리전 간 복제 문제를 해결하는 방법도 다릅니다.

AWS 리전 내 PostgreSQL 읽기 전용 복제본의 문제 해결

PostgreSQL 파라미터인 `wal_keep_segments`는 데이터를 읽기 전용 복제본으로 보낼 때 유지할 Write Ahead Log(WAL) 파일 수를 결정합니다. 이 파라미터 값에 따라 유지할 로그 수가 결정됩니다. 이 파라미터 값이 너무 낮으면 읽기 전용 복제본이 너무 멀리 뒤처지면서 스트리밍 복제가 중단될 수 있습니다. 이 경우에는 Amazon RDS가 복제 오류를 보고한 후 원본 DB 인스턴스에 보관된 WAL 로그를 재실행하여 읽기 전용 복제본의 복구를 시작합니다. 이 복구 프로세스는 읽기 전용 복제본이 스트리밍 복제를 이어갈 만큼 충분히 따라잡을 때까지 계속됩니다.

Amazon RDS가 보관 중인 WAL 파일을 재실행하여 이러한 상태의 읽기 전용 복제본을 복구할 때는 읽기 전용 복제본의 PostgreSQL 로그가 표시됩니다.

```
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
```

일정 시간이 지나면 Amazon RDS가 복제본에 보관된 WAL 파일을 재실행하여 충분히 따라잡은 후 읽기 전용 복제본이 스트리밍을 재개할 수 있습니다. 이때부터는 PostgreSQL이 스트리밍을 재개하고 다음과 비슷한 라인을 로그 파일에 기록합니다.

```
2014-11-07 19:41:36 UTC::@:[24714]:LOG: started streaming WAL from primary at 1B/
B6000000
on timeline 1
```

로그의 체크포인트 정보를 보면 유지해야 할 WAL 파일 수를 결정할 수 있습니다. PostgreSQL 로그는 각 체크포인트마다 다음과 같은 정보를 표시합니다. 아래 로그 문에서 "# recycled" 트랜잭션 로그 파일을 보면 일정 시간에 재사용되는 트랜잭션 파일 수를 확인 후 이 정보를 사용해 `wal_keep_segments` 파라미터를 설정할 수 있습니다.

```
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

예를 들어 PostgreSQL 로그의 "checkpoint completed" 로그 문에서 5분 안에 재사용되는 파일 수가 35개라고 표시되어 있다고 가정하겠습니다. 이러한 사용 패턴에서는 읽기 전용 복제본이 5분간 이용하는 트랜잭션 파일 수는 35개입니다. 원본 DB 인스턴스의 `wal_keep_segments` 파라미터 값이 32로 기본 설정되어 있는 경우에는 읽기 전용 복제본이 비스트리밍 상태로 5분을 넘길 수 없습니다.

AWS 리전 간 PostgreSQL 읽기 전용 복제본의 문제 해결

PostgreSQL(버전 9.4.7 및 9.5.2에 한함)은 물리적인 복제 슬롯을 사용하여 원본 DB 인스턴스의 Write Ahead Log(WAL) 보존을 관리합니다. Amazon RDS가 리전 간 읽기 전용 복제본 인스턴스마다 물리적 복제 슬롯을 생성하여 연동시킵니다. 2개의 Amazon CloudWatch 지표인 `oldest Replication Slot Lag`와 `Transaction Logs Disk Usage`를 통해 수신되는 WAL 데이터와 관련하여 가장 지체된 복제본이 얼마나 오래되었는지, 그리고 현재 WAL 데이터에 얼마나 많은 스토리지가 사용되고 있는지 알 수 있습니다. `Transaction Logs Disk Usage` 값은 리전 간 읽기 전용 복제본이 많이 지체될 수록 크게 증가합니다.

DB 인스턴스의 워크로드에서 대용량의 WAL 데이터가 생성되는 경우에는 원본 DB 인스턴스와 읽기 전용 복제본의 DB 인스턴스 클래스를 변경해야 할 수 있습니다. 이 경우 복제본이 뒤처지지 않도록 High(10Gbps) 네트워크 성능의 클래스로 변경합니다. Amazon CloudWatch 지표인 *Transaction Logs Generation*을 보면 워크로드에서 발생하는 WAL 데이터 비율을 쉽게 이해할 수 있습니다.

리전 간 읽기 전용 복제본의 상태를 알고 싶으면 아래 예제와 같이 원본 인스턴스에 대한 pg_replication_slots를 쿼리를 실행하면 됩니다.

```
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_uzwlholddgpbblksc6hgwnk | physical |       |
 | 12598 |       |           | 4E/95000060
(1 row)
```

Amazon RDS에서 PostgreSQL로 데이터 가져오기

Amazon RDS로 이동하려는 기존 PostgreSQL 배포가 있다고 가정하십시오. 작업의 복잡성은 데이터베이스의 크기와 전송하는 데이터베이스 객체 유형에 따라 다릅니다. 예를 들어 프로시저 및 트리거와 함께 기가바이트 규모의 데이터세트가 데이터베이스에 저장되어 있다고 가정하겠습니다. 이러한 데이터베이스는 트리거나 프로시저 없이 근소한 메가바이트 규모의 테스트 데이터가 저장된 데이터베이스보다 더욱 복잡할 가능성이 높습니다.

다음과 같은 경우에는 기본 PostgreSQL 데이터베이스 마이그레이션 도구를 사용하는 것이 좋습니다.

- 같은 유형의 마이그레이션을 수행합니다. 이 경우 대상 데이터베이스와 동일한 데이터베이스 엔진으로 데이터베이스에서 마이그레이션합니다.
 - 전체 데이터베이스를 마이그레이션합니다.
 - 기본 도구를 사용하여 최소한의 가동 중지로 시스템을 마이그레이션할 수 있습니다.

대부분의 다른 경우 AWS Database Migration Service(AWS DMS)를 사용하여 데이터베이스 마이그레이션을 수행하는 것이 가장 좋은 방법입니다. AWS DMS는 가동 중지 없이 데이터베이스를 마이그레이션할 수 있으며 대부분의 데이터베이스 엔진에서는 대상 데이터베이스로 전환할 준비가 될 때까지 지속적으로 복제를 계속할 수 있습니다. AWS DMS를 사용하여 둘일하거나 다른 데이터베이스 엔진으로 마이그레이션

할 수 있습니다. 원본 데이터베이스와 다른 데이터베이스 엔진으로 마이그레이션하는 경우 AWS Schema Conversion Tool(AWS SCT)를 사용하여 마이그레이션할 수 있습니다. AWS SCT를 사용하여 AWS DMS로 마이그레이션되지 않은 스키마 객체를 마이그레이션합니다. AWS DMS에 대한 자세한 내용은 [AWS Database Migration Service란?](#)을 참조하십시오.

가져오기에 대해서만 다음 설정을 포함하도록 DB 파라미터 그룹을 수정합니다. 파라미터 설정을 테스트하여 DB 인스턴스 크기에 가장 효율적인 설정을 찾아야 합니다. 또한 가져오기가 완료된 후 이들 파라미터를 프로덕션 값으로 되돌려야 합니다.

DB 인스턴스 설정을 다음과 같이 수정합니다.

- DB 인스턴스 백업 비활성화(backup_retention을 0으로 설정).
- Multi-AZ 비활성화.

다음 설정을 포함하도록 DB 파라미터 그룹을 수정합니다. 이들 설정은 데이터를 가져올 때만 사용해야 합니다. 파라미터 설정을 테스트하여 DB 인스턴스 크기에 가장 효율적인 설정을 찾아야 합니다. 또한 가져오기가 완료된 후 이들 파라미터를 프로덕션 값으로 되돌려야 합니다.

파라미터	가져오기 시 권장 값	설명
<code>maintenance_work_mem</code>	524288, 1048576, 2097152 또는 4194304(KB). 이들 설정은 512MB, 1GB, 2GB 및 4GB에 해당합니다.	이 설정의 값은 호스트 크기에 따라 달라집니다. 이 파라미터는 CREATE INDEX 문에서 사용되며 각 병렬 명령은 이 크기의 메모리를 사용할 수 있습니다. 이 값을 너무 높게 설정하여 메모리가 부족해지지 않도록 최선의 값을 계산합니다.
<code>checkpoint_segments</code>	256	이 설정의 값은 더 많은 디스크 공간을 소비하지만 WAL 로그에서는 경합이 감소합니다. PostgreSQL 버전 9.5.x 및 9.6.x에서는 이 값이 <code>max_wal_size</code> 입니다.
<code>checkpoint_timeout</code>	1800	이 설정의 값은 WAL 교체 빈도를 줄일 수 있습니다.
<code>synchronous_commit</code>	Off	쓰기 속도를 높이려면 이 설정을 비활성화합니다. 이 파라미터를 고면 서버 충돌 시 데이터 손실 위험이 증가합니다(FSYNC를 고지 않음).
<code>wal_buffers</code>	8192	이 값은 8KB 단위입니다. 이는 다시 WAL 생성 속도에 도움이 됩니다.
<code>autovacuum</code>	Off	리소스를 사용하지 않으므로 데이터를 로드하는 동안에는 PostgreSQL auto-vacuum 파라미터를 비활성화합니다.

이런 설정과 함께 `pg_dump -Fc(압축)` 또는 `pg_restore -j(병렬)` 명령을 사용합니다.

Note

PostgreSQL의 `pg_dumpall` 명령을 사용하려면 DB 인스턴스를 생성할 때 부여되지 않은 `super_user` 권한이 필요하며, 따라서 이 명령은 데이터 가져오기에 사용할 수 없습니다.

주제

- [Amazon EC2 인스턴스에서 PostgreSQL 데이터베이스 가져오기 \(p. 972\)](#)
- [\copy 명령을 사용하여 PostgreSQL DB 인스턴스의 테이블로 데이터 가져오기 \(p. 973\)](#)
- [PostgreSQL DB 인스턴스용 RDS로 Amazon S3 데이터 가져오기 \(p. 974\)](#)

Amazon EC2 인스턴스에서 PostgreSQL 데이터베이스 가져오기

Amazon EC2 인스턴스 상의 PostgreSQL Server에 데이터가 있는데 이를 PostgreSQL DB 인스턴스로 이동하려는 경우 다음 프로세스를 사용할 수 있습니다. 아래 목록에 수행할 단계가 나와 있습니다. 각 단계에 대해서는 다음에 이어지는 섹션에서 자세히 설명합니다.

1. 로드할 데이터를 포함한 pg_dump를 사용하여 파일 만들기
2. 대상 DB 인스턴스 만들기
3. psql을 사용하여 DB 인스턴스에서 데이터베이스를 만들고 데이터 로드
4. DB 인스턴스의 DB 스냅샷 만들기

1단계: 로드할 데이터를 포함하여 pg_dump로 파일 생성

pg_dump 유ти리티는 COPY 명령을 사용하여 PostgreSQL 데이터베이스의 스키마와 데이터 덤프를 만듭니다. pg_dump에 의해 생성되는 덤프 스크립트는 같은 이름을 가진 데이터베이스로 데이터를 로드하고 테이블, 인덱스 및 외래 키를 다시 만듭니다. pg_restore 명령과 -d 파라미터를 사용하여 데이터를 다른 이름의 데이터베이스로 복원할 수 있습니다.

데이터 덤프를 만들기 전, 대상 DB 인스턴스에서 개수를 확인할 수 있도록 행 개수를 확인하기 위해 덤프할 테이블을 쿼리해야 합니다.

다음 명령을 실행하면 mydb2라는 데이터베이스에 대해 mydb2dump.sql이라는 덤프 파일이 생성됩니다.

```
prompt>pg_dump dbname=mydb2 -f mydb2dump.sql
```

2단계: 대상 DB 인스턴스 생성

Amazon RDS 콘솔, AWS CLI 또는 API를 사용하여 대상 PostgreSQL DB 인스턴스를 만듭니다. 백업 보존 설정을 0으로 지정하여 인스턴스를 만들고 다중 AZ를 비활성화합니다. 그러면 데이터를 더 빠르게 가져올 수 있습니다. 데이터를 덤프하려면 먼저 인스턴스에 데이터베이스를 생성해야 합니다. 데이터베이스 이름은 덤프 데이터가 저장된 데이터베이스와 동일하게 지정할 수 있습니다. 그렇지 않으면 다른 이름으로 지정하는 것도 가능합니다. 이 경우에는 pg_restore 명령과 -d 파라미터를 사용하여 데이터를 새로운 이름의 데이터베이스로 복원할 수 있습니다.

예를 들어 다음 명령을 사용하여 데이터베이스를 덤프 및 복원하고 이름을 바꿀 수 있습니다.

```
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
```

3단계: psql을 사용하여 DB 인스턴스에서 데이터베이스를 만들고 데이터 로드

pg_dump 명령을 실행할 때 사용한 것과 같은 연결을 사용하여 대상 DB 인스턴스에 연결하고 데이터베이스를 다시 만들 수 있습니다. psql을 사용할 때는 마스터 사용자 이름과 마스터 암호를 사용하여 DB 인스턴스에 데이터베이스를 만들 수 있습니다.

다음 예제에서는 psql과 mydb2dump.sql이라는 덤프 파일을 사용하여 mypginstance라는 PostgreSQL DB 인스턴스에 mydb2라는 데이터베이스를 만들니다.

Linux, OS X, Unix의 경우:

```
psql \
-f mydb2dump.sql \
--host mypginstance.c6c8mntzhgv0.us-west-2.rds.amazonaws.com \
--port 8199 \
--username myawsuser \
--password password \
--dbname mydb2
```

Windows의 경우:

```
psql ^
-f mydb2dump.sql ^
--host mypginstance.c6c8mntzhgv0.us-west-2.rds.amazonaws.com ^
--port 8199 ^
--username myawsuser ^
--password password ^
--dbname mydb2
```

4단계: DB 인스턴스의 DB 스냅샷 생성

데이터가 DB 인스턴스로 로드된 것을 확인했으면, 대상 PostgreSQL DB 인스턴스의 DB 스냅샷을 만드는 것이 좋습니다. DB 스냅샷은 DB 인스턴스를 알려진 상태로 복원하는 데 사용할 수 있는 DB 인스턴스의 완전한 백업입니다. 로드 직후에 생성된 DB 스냅샷은 사고 발생 시 데이터를 다시 로드하지 않아도 됩니다. 또한 이러한 스냅샷을 사용하여 새 DB 인스턴스를 시드할 수 있습니다. DB 스냅샷 생성에 대한 정보는 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.

\copy 명령을 사용하여 PostgreSQL DB 인스턴스의 테이블로 데이터 가져오기

psql 프롬프트에서 \copy 명령을 실행하여 PostgreSQL DB 인스턴스의 테이블로 데이터를 가져올 수 있습니다. 이 테이블은 DB 인스턴스에 이미 존재하고 있어야 합니다. \copy 명령에 대한 자세한 내용은 [PostgreSQL 문서](#) 단원을 참조하십시오.

Note

\copy 명령은 삽입된 행의 개수와 같은 작업 확인 기능을 제공하지 않습니다. PostgreSQL은 오류로 인해 copy 명령이 실패할 경우 오류 메시지를 제공합니다.

원본 테이블의 데이터에서 .csv 파일을 만들고, psql을 사용하여 PostgreSQL 인스턴스의 대상 데이터베이스에 로그온한 후, 다음 명령을 실행합니다. 이 예제에서는 source-table을 원본 테이블 이름으로, source-table.csv를 .csv 파일로, target-db를 대상 데이터베이스로 사용합니다.

```
target-db=> \copy source-table from 'source-table.csv' with DELIMITER ',';
```

클라이언트 명령 프롬프트에서 다음 명령을 실행할 수도 있습니다. 이 예제에서는 source-table을 원본 테이블 이름으로, source-table.csv를 .csv 파일로, target-db를 대상 데이터베이스로 사용합니다.

Linux, OS X, Unix의 경우:

```
$psql target-db \
-U <admin user> \
-p <port> \
```

```
-h <DB instance name> \
-c "\copy source-table from 'source-table.csv' with DELIMITER ',' "
```

Windows의 경우:

```
$psql target-db ^
-U <admin user> ^
-p <port> ^
-h <DB instance name> ^
-c "\copy source-table from 'source-table.csv' with DELIMITER ',' "
```

PostgreSQL DB 인스턴스용 RDS로 Amazon S3 데이터 가져오기

PostgreSQL DB 인스턴스용 Amazon RDS에 속하는 테이블로 Amazon S3의 데이터를 가져올 수 있습니다. 이를 위해서는 Amazon RDS가 제공하는 `aws_s3` PostgreSQL 확장을 사용할 수 있습니다.

Amazon S3를 이용한 데이터 저장에 대한 자세한 내용은 Amazon Simple Storage Service 시작 안내서의 [버킷 생성](#)을 참조하십시오. 파일을 Amazon S3 버킷에 업로드하는 방법에 관한 지침은 Amazon Simple Storage Service 시작 안내서의 [버킷에 객체 추가](#)를 참조하십시오.

Note

데이터베이스에서는 PostgreSQL 버전 11.1 이상을 실행하여 Amazon S3에서 PostgreSQL용 Amazon RDS로 가져오기를 수행해야 합니다.

주제

- [Amazon S3 데이터 수집 및 가져오기 \(p. 974\)](#)
- [IAM 역할을 사용해 Amazon S3 파일에 액세스 \(p. 976\)](#)
- [보안 자격 증명을 사용해 Amazon S3 파일에 액세스 \(p. 979\)](#)
- [Amazon S3 파일 형식 처리 \(p. 979\)](#)
- [함수 참조 \(p. 981\)](#)

Amazon S3 데이터 수집 및 가져오기

Amazon S3 버킷에 저장된 데이터를 PostgreSQL 데이터베이스 테이블로 가져오려면 다음 정보를 수집한 후 설명된 대로 `aws_s3.table_import_from_s3` 함수를 사용하십시오.

RDS PostgreSQL로 S3 데이터를 가져오려면

1. `psql`을 시작하고 다음 명령을 사용해 필수 Amazon RDS PostgreSQL 확장을 설치합니다. 여기에는 `aws_s3` 및 `aws_commons` 확장이 포함됩니다.

```
psql=> CREATE EXTENSION aws_s3 CASCADE;
NOTICE: installing required extension "aws_commons"
```

`aws_s3` 확장에서는 Amazon S3 데이터를 가져오는 데 사용하는 [aws_s3.table_import_from_s3 \(p. 981\)](#) 함수를 제공합니다.

2. 가져오고려는 Amazon S3 파일에 액세스할 수 있는 권한을 제공합니다. 이를 위해서는 파일이 있는 Amazon S3 버킷에 액세스할 수 있는 AWS Identity and Access Management(IAM) 역할을 생성해야 합니다. 그런 다음 이 역할을 RDS PostgreSQL DB 인스턴스에 할당합니다.

IAM 역할을 설정하여 액세스 권한을 제공하는 방법에 관한 자세한 내용은 [IAM 역할을 사용해 Amazon S3 파일에 액세스 \(p. 976\)](#) 단원을 참조하십시오.

3. 다음 절차에 따라 `table_import_from_s3` 함수에 대한 필수 정보를 수집하십시오.

- a. 가져오려는 Amazon S3 파일에 대해 다음 정보를 얻으십시오.

- 버킷 이름 – 버킷은 Amazon S3 객체 또는 파일을 위한 컨테이너입니다.
- 파일 경로 – 파일 경로를 통해 파일이 Amazon S3 버킷에 배치됩니다.
- AWS 리전 – AWS 리전은 Amazon S3 버킷이 있는 위치입니다.

이 정보를 얻는 방법에 대해 알아보려면 Amazon Simple Storage Service 시작 안내서에서 [객체 보기](#)를 참조하십시오.

- b. 다음 예에 나와 있는 것처럼 `aws_commons.create_s3_uri (p. 983)` 함수를 사용해 이 파일 정보를 저장할 `aws_commons._s3_uri_1` 구조를 생성하십시오.

```
psql=> SELECT aws_commons.create_s3_uri(
    'sample_s3_bucket',
    'sample.csv',
    'us-east-1'
) AS s3_uri \gset
```

나중에 `s3_info` 파라미터에서 [aws_s3.table_import_from_s3 \(p. 981\)](#) 함수에 대한 호출을 위해 이 `aws_commons._s3_uri_1` 구조를 제공하십시오.

- c. 데이터를 어떤 PostgreSQL 데이터베이스 테이블에 배치할지 식별합니다. 예를 들어 다음은 샘플 `t1` 데이터베이스 테이블입니다.

```
psql=> CREATE TABLE t1 (bid bigint PRIMARY KEY, name varchar(80));
```

4. 준비 작업을 완료한 후에는 [aws_s3.table_import_from_s3 \(p. 981\)](#) 함수를 사용해 Amazon S3 데이터를 가져오십시오.

다음은 `psql`을 사용하는 일반적인 PostgreSQL의 예입니다.

```
psql=> SELECT aws_s3.table_import_from_s3(
    't1',
    '',
    '(format csv)',
    :'s3_uri'
);
```

파라미터는 다음과 같습니다.

- `t1` – 데이터를 복사할 PostgreSQL DB 인스턴스의 테이블에 지정된 이름입니다.
- `''` – 데이터베이스 테이블의 열 목록입니다(선택 사항). 이 파라미터를 사용해 S3 데이터 중 어떤 열이 어떤 테이블 열에 들어가는지 표시할 수 있습니다. 열을 지정하지 않으면 모든 열이 테이블에 복사됩니다. 열 목록 사용에 대한 예시는 [사용자 지정 구분 기호를 사용하는 파일 가져오기 \(p. 979\)](#) 단원을 참조하십시오.
- `(format csv)` – PostgreSQL COPY 인수입니다. 복사 프로세스에서는 [PostgreSQL COPY 명령의 인수 및 형식](#)을 사용합니다. 앞의 예에서 COPY 명령은 쉼표로 구분된 값(CSV) 파일 형식을 사용해 데이터를 복사합니다. 더 많은 예제는 [Amazon S3 파일 형식 처리 \(p. 979\)](#)를 참조하십시오.
- `s3_uri` – Amazon S3 파일을 식별하는 정보가 포함된 구조입니다.

이 함수에 대한 자세한 내용은 [aws_s3.table_import_from_s3 \(p. 981\)](#) 단원을 참조하십시오.

IAM 역할을 사용해 Amazon S3 파일에 액세스

Amazon S3 파일에서 데이터를 로드하기 전에 RDS PostgreSQL DB 인스턴스에 Amazon S3에 액세스할 수 있는 권한을 부여하십시오. 이를 위해서는 Amazon S3 버킷에 액세스할 수 있는 IAM 역할을 생성한 후 이 역할을 DB 인스턴스에 할당해야 합니다. 이렇게 하면 추가 자격 증명 정보를 제공하거나 관리할 필요가 없습니다.

IAM 역할을 통해 Amazon S3에 액세스할 수 있는 권한을 RDS PostgreSQL DB 인스턴스에 부여하려면

1. RDS PostgreSQL DB 인스턴스가 Amazon S3에 액세스하도록 허용하는 버킷 및 객체 권한을 제공하는 IAM 정책을 생성합니다.

정책에 다음과 같은 필수 작업을 포함하여 Amazon S3 버킷에서 Amazon RDS로의 파일 전송을 허용합니다.

- GetObject
- ListBucket

RDS PostgreSQL에 대한 IAM 정책 생성에 대한 자세한 내용은 [IAM 데이터베이스 액세스를 위한 IAM 정책 생성 및 사용 \(p. 372\)](#) 단원을 참조하십시오.

다음 AWS CLI 명령은 이 옵션으로 rds-s3-integration-policy라는 IAM 정책을 만듭니다. 이 정책은 your-s3-bucket-arn이라는 버킷에 액세스할 수 있는 권한을 부여합니다.

Note

정책을 만든 후에 정책의 Amazon 리소스 이름(ARN)을 기록하십시오. 가져오기 프로세스의 후속 단계에 이 ARN이 필요합니다.

Example

Linux, OS X, Unix의 경우:

```
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"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:s3:::your-s3-bucket-arn",  
                "arn:aws:s3:::your-s3-bucket-arn/*"  
            ]  
        }  
    ]  
}'
```

Windows의 경우:

```
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"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:s3:::your-s3-bucket-arn",  
                "arn:aws:s3:::your-s3-bucket-arn/*"  
            ]  
        }  
    ]  
}'
```

- Amazon RDS가 Amazon S3 버킷에 액세스하기 위해 사용자 대신 가정할 수 있는 IAM 역할을 만듭니다. 자세한 내용은 IAM 사용 설명서의 역할을 만들어 [IAM 사용자에게 권한 위임](#) 단원을 참조하십시오.

다음 예에서는 AWS CLI 명령을 사용해 `rds-s3-integration-role`이라는 역할을 생성하는 방법을 보여줍니다.

Example

Linux, OS X, Unix의 경우:

```
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"  
            }  
        ]  
    }'
```

Windows의 경우:

```
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"  
            }  
        ]  
    }'
```

- 생성한 IAM 역할에 생성한 IAM 정책을 연결합니다.

다음 AWS CLI 명령은 앞서 생성한 정책을 `rds-s3-integration-role`이라는 역할에 연결합니다. `your-policyarn`을 이전 단계에서 기록한 정책 ARN으로 교체합니다.

Example

Linux, OS X, Unix의 경우:

```
aws iam attach-role-policy \
--policy-arn your-policy-arn \
--role-name rds-s3-integration-role
```

Windows의 경우:

```
aws iam attach-role-policy ^
--policy-arn your-policy-arn ^
--role-name rds-s3-integration-role
```

4. 아래 설명된 대로 AWS Management 콘솔 또는 AWS CLI를 사용해 PostgreSQL DB 인스턴스에 IAM 역할을 추가합니다.

콘솔

콘솔을 사용하여 PostgreSQL DB 인스턴스에 대해 IAM 역할을 추가하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 세부 정보를 표시하고자 하는 PostgreSQL DB 인스턴스 이름을 선택합니다.
3. Connectivity & security(연결성 및 보안) 탭에 있는 Manage IAM roles(IAM 역할 관리) 섹션의 Add IAM roles to this instance(이 인스턴스에 IAM 역할 추가)에서 추가할 역할을 선택합니다.
4. Feature(기능)에서 s3Import를 선택합니다.
5. [Add role]을 선택합니다.

AWS CLI

CLI를 사용하여 PostgreSQL DB 인스턴스에 대해 IAM 역할을 추가하려면

- 다음 명령을 사용해 mydbinstance라는 PostgreSQL DB 인스턴스에 역할을 추가합니다. `your-role-arn`을 이전 단계에서 기록한 정책 ARN으로 교체합니다. --feature-name 옵션의 값에 대해 s3Import를 사용합니다.

Example

Linux, OS X, Unix의 경우:

```
aws rds add-role-to-db-instance \
--db-instance-identifier mydbinstance \
--feature-name s3Import \
--role-arn your-role-arn \
--region us-west-2
```

Windows의 경우:

```
aws rds add-role-to-db-instance ^
--db-instance-identifier mydbinstance ^
```

```
--feature-name s3Import ^
--role-arn your-role-arn ^
--region us-west-2
```

보안 자격 증명을 사용해 Amazon S3 파일에 액세스

IAM 역할을 사용해 Amazon S3 파일에 액세스하는 대신에 [aws_s3.table_import_from_s3 \(p. 981\)](#) 함수 호출의 `credentials` 파라미터에 있는 액세스 키와 비밀 키를 제공할 수 있습니다.

`credentials` 파라미터는 AWS 자격 증명을 포함하는 `aws_commons._aws_credentials_1` 유형의 구조입니다. 다음과 같이 [aws_commons.create_aws_credentials \(p. 983\)](#) 함수를 사용해 `aws_commons._aws_credentials_1` 구조에서 액세스 키와 비밀 키를 설정하십시오.

```
psql=> SELECT aws_commons.create_aws_credentials(
    '<sample_access_key>', '<sample_secret_key>', '')
AS creds \gset
```

다음과 같이 `aws_commons._aws_credentials_1` 구조를 생성한 후 [aws_s3.table_import_from_s3 \(p. 981\)](#) 함수를 사용해 데이터를 가져옵니다.

```
psql=> SELECT aws_s3.table_import_from_s3(
    't', '', '(format csv)',
    :'s3_uri',
    :'creds'
);
```

또는 [aws_commons.create_aws_credentials \(p. 983\)](#) 함수 호출 인라인을 `aws_s3.table_import_from_s3` 함수 호출에 포함할 수 있습니다.

```
psql=> SELECT aws_s3.table_import_from_s3(
    't', '', '(format csv)',
    :'s3_uri',
    aws_commons.create_aws_credentials('<sample_access_key>', '<sample_secret_key>', '')
);
```

Amazon S3 파일 형식 처리

다음 예에서는 가져오기를 수행할 때 다양한 종류의 파일을 지정하는 방법을 보여줍니다.

주제

- 사용자 지정 구분 기호를 사용하는 파일 가져오기 ([p. 979](#))
- 압축(gzip) 파일 가져오기 ([p. 980](#))
- 인코딩된 파일 가져오기 ([p. 981](#))

Note

다음 예에서는 Amazon S3 파일에 액세스할 수 있는 권한을 제공하기 위해 IAM 역할 방법을 사용합니다. 따라서 `aws_s3.table_import_from_s3` 함수 호출에는 자격 증명 파라미터가 없습니다.

사용자 지정 구분 기호를 사용하는 파일 가져오기

다음 예에서는 사용자 지정 구분 기호를 사용하는 파일을 가져오는 방법을 보여줍니다. 또한 [aws_s3.table_import_from_s3 \(p. 981\)](#) 함수의 `column_list` 파라미터를 사용해 데이터베이스 테이블에 서 데이터를 배치할 곳을 제어하는 방법을 보여줍니다.

이 예에서는 다음 정보가 Amazon S3 파일의 파이프로 구분된 열에 정리되어 있다고 가정합니다.

```
1|foo1|bar1|elephant1
2|foo2|bar2|elephant2
3|foo3|bar3|elephant3
4|foo4|bar4|elephant4
...
```

사용자 지정 구분 기호를 사용하는 파일을 가져오려면

1. 가져온 데이터에 대해 데이터베이스에서 테이블을 생성합니다.

```
psql=> CREATE TABLE test (a text, b text, c text, d text, e text);
CREATE TABLE
```

2. [aws_s3.table_import_from_s3 \(p. 981\)](#) 함수의 다음과 같은 형식을 사용해 Amazon S3 파일에서 데이터를 가져옵니다.

[aws_commons.create_s3_uri \(p. 983\)](#) 함수 호출 인라인을 `aws_s3.table_import_from_s3` 함수 호출에 포함하여 파일을 지정할 수 있습니다.

```
psql=> SELECT aws_s3.table_import_from_s3(
    'test',
    'a,b,d,e',
    'DELIMITER ''|''',
    aws_commons.create_s3_uri('sampleBucket', 'pipeDelimitedSampleFile', 'us-east-2')
);
```

이제 데이터는 다음 열의 테이블에 있습니다.

```
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
```

압축(gzip) 파일 가져오기

다음 예에서는 gzip으로 압축된 Amazon S3에서 파일을 가져오는 방법을 보여줍니다.

파일이 다음 Amazon S3 메타데이터를 포함하고 있는지 확인합니다.

- 키: Content-Encoding
- 값: gzip

이 값을 Amazon S3 메타데이터에 추가하는 것에 관한 자세한 내용은 Amazon Simple Storage Service 콘솔 사용 설명서의 [S3 객체에 메타데이터를 추가하려면 어떻게 해야 합니까?](#)를 참조하십시오.

아래와 같이 gzip 파일을 RDS PostgreSQL DB 인스턴스로 가져옵니다.

```
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'
```

) ;

인코딩된 파일 가져오기

다음 예에서는 Windows-1252 인코딩이 있는 Amazon S3에서 파일을 가져오는 방법을 보여줍니다.

```
psql=> SELECT aws_s3.table_import_from_s3(
    'test_table', '', 'encoding ''WIN1252'''',
    aws_commons.create_s3_uri('sampleBucket', 'SampleFile', 'us-east-2')
);
```

함수 참조

Functions

- [aws_s3.table_import_from_s3 \(p. 981\)](#)
- [aws_commons.create_s3_uri \(p. 983\)](#)
- [aws_commons.create_aws_credentials \(p. 983\)](#)

aws_s3.table_import_from_s3

Amazon S3 데이터를 RDS PostgreSQL 테이블로 가져옵니다. aws_s3 확장은 aws_s3.table_import_from_s3 함수를 제공합니다.

세 가지 필수 파라미터는 table_name, column_list 및 options입니다. 이 파라미터에서는 데이터베이스 테이블을 식별하고 데이터가 테이블로 복사되는 방식을 지정합니다.

다음 파라미터도 사용할 수 있습니다.

- s3_info 파라미터는 가져올 Amazon S3 파일을 지정합니다. 이 파라미터를 사용하는 경우 IAM 역할에서 PostgreSQL DB 인스턴스에 대해 Amazon S3에 액세스할 수 있는 권한을 제공합니다.

```
aws_s3.table_import_from_s3 (
    table_name text,
    column_list text,
    options text,
    s3_info aws_commons._s3_uri_1
)
```

- credentials 파라미터에서는 Amazon S3에 액세스할 수 있는 자격 증명을 지정합니다. 이 파라미터를 사용할 때는 IAM 역할을 사용하지 마십시오.

```
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
)
```

aws_s3.table_import_from_s3 파라미터는 다음 표에 설명되어 있습니다.

파라미터	설명
table_name	데이터를 가져올 필수 텍스트 문자열로서, PostgreSQL 데이터베이스 테이블의 이름을 포함합니다.

파라미터	설명
column_list	데이터를 복사할 PostgreSQL 데이터베이스 테이블 열의 목록(선택 사항)을 포함하는 필수 텍스트 문자열입니다. 문자열이 비어 있는 경우 테이블의 모든 열이 사용됩니다. 관련 예시는 사용자 지정 구분 기호를 사용하는 파일 가져오기 (p. 979) 단원을 참조하십시오.
옵션	PostgreSQL COPY 명령에 대한 인수를 포함하는 필수 텍스트 스트링입니다. 이 인수에서는 데이터가 PostgreSQL 테이블에 복사되는 방식을 지정합니다. 자세한 내용은 PostgreSQL COPY 설명서 를 참조하십시오.
s3_info	S3 객체에 대한 다음 정보를 포함하는 aws_commons._s3_uri_1 복합 키입니다. <ul style="list-style-type: none"> • bucket – 파일을 포함하는 Amazon S3 버킷 이름입니다. • file_path – 파일의 Amazon S3 경로입니다. • region – 파일이 위치한 AWS 리전입니다. aws_commons._s3_uri_1 복합 구조를 생성하려면 aws_commons.create_s3_uri (p. 983) 단원을 참조하십시오.
자격 증명	가져오기 작업에 사용할 다음 자격 증명을 포함하는 aws_commons._aws_credentials_1 복합 유형입니다. <ul style="list-style-type: none"> • 액세스 키 • 비밀 키 • 세션 토큰 aws_commons._aws_credentials_1 복합 구조를 생성하려면 aws_commons.create_aws_credentials (p. 983) 단원을 참조하십시오.

대체 파라미터

s3_info 및 credentials 파라미터 대신에 확장 파라미터 집합을 사용하면 테스트에 도움이 됩니다. 다음은 aws_s3.table_import_from_s3 함수에 대한 추가 구문 변형입니다.

- s3_info 파라미터를 사용해 Amazon S3 파일을 식별하는 대신 bucket, file_path 및 region 파라미터의 조합을 사용하십시오. IAM 역할은 이러한 형식의 함수를 통해 PostgreSQL DB 인스턴스에 대해 Amazon S3에 액세스할 수 있는 권한을 제공합니다.

```
aws_s3.table_import_from_s3 (
    table_name text,
    column_list text,
    options text,
    bucket text,
    file_path text,
    region text
)
```

- credentials 파라미터를 사용해 Amazon S3 액세스를 지정하는 대신 access_key, session_key 및 session_token 파라미터의 조합을 사용하십시오.

```
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  
)
```

이러한 대체 파라미터에 대한 설명을 다음 표에서 확인하십시오.

파라미터	설명
버킷	파일이 들어 있는 Amazon S3 버킷의 이름이 포함된 텍스트 문자열입니다.
file_path	파일의 Amazon S3 경로가 포함된 텍스트 문자열입니다.
region	파일이 위치한 AWS 리전이 포함된 텍스트 문자열입니다.
액세스 키	가져오기 작업에 사용할 액세스 키가 포함된 텍스트 문자열입니다. 기본값은 NULL입니다.
secret_key	가져오기 작업에 사용할 비밀 키가 포함된 텍스트 문자열입니다. 기본값은 NULL입니다.
session_token(선택 사항)	가져오기 작업에 사용할 세션 키가 포함된 텍스트 문자열입니다. 기본값은 NULL입니다.

aws_commons.create_s3_uri

Amazon S3 파일 정보를 저장할 aws_commons._s3_uri_1 구조를 생성합니다.

aws_s3.table_import_from_s3 (p. 981) 함수의 s3_info 파라미터에서 aws_commons.create_s3_uri 함수의 결과를 사용합니다. 함수 구문은 다음과 같습니다.

```
aws_commons.create_s3_uri(  
    bucket text,  
    file_path text,  
    region text  
)
```

aws_commons.create_s3_uri 함수 파라미터는 다음 표에 설명되어 있습니다.

파라미터	설명
버킷	파일의 Amazon S3 버킷 이름이 포함된 필수 텍스트 문자열입니다.
file_path	파일의 Amazon S3 경로가 포함된 필수 텍스트 문자열입니다.
region	파일이 위치한 AWS 리전이 포함된 필수 텍스트 문자열입니다.

aws_commons.create_aws_credentials

aws_commons._aws_credentials_1 구조에서 액세스 키와 비밀 키를 설정합니다. aws_s3.table_import_from_s3 (p. 981) 함수의 credentials 파라미터에서

aws_commons.create_aws_credentials 함수의 결과를 사용합니다. 함수 구문은 다음과 같습니다.

```
aws_commons.create_aws_credentials(  
    access_key text,  
    secret_key text,  
    session_token text
```

)

`aws_commons.create_aws_credentials` 함수 파라미터는 다음 표에 설명되어 있습니다.

파라미터	설명
액세스 키	Amazon S3 파일 가져오기에 사용할 액세스 키가 포함된 필수 텍스트 문자열입니다. 기본값은 NULL입니다.
secret_key	Amazon S3 파일 가져오기에 사용할 비밀 키가 포함된 필수 텍스트 문자열입니다. 기본값은 NULL입니다.
session_token	Amazon S3 파일 가져오기에 사용할 세션 토큰이 포함된 텍스트 문자열(선택 사항)입니다. 기본값은 NULL입니다. 선택 사항인 <code>session_token</code> 을 제공하는 경우 임시 자격증명을 사용할 수 있습니다.

PostgreSQL을 위한 일반 DBA 작업

이번 단원에서는 PostgreSQL 데이터베이스 엔진을 실행하는 DB 인스턴스의 몇 가지 공통 DBA 작업에 대한 Amazon RDS 구현에 대해 설명합니다. 관리형 서비스 환경을 제공하기 위해 Amazon RDS는 DB 인스턴스에 대해 shell 액세스를 제공하지 않으며, 고급 권한을 필요로 하는 특정 시스템 절차와 테이블에 대한 액세스를 제한합니다.

Amazon RDS의 PostgreSQL 로그 파일 작업 방법에 대한 자세한 내용은 [PostgreSQL 데이터베이스 로그 파일 \(p. 331\)](#) 단원을 참조하십시오.

주제

- [역할 생성 \(p. 984\)](#)
- [PostgreSQL 데이터베이스 액세스 관리 \(p. 985\)](#)
- [PostgreSQL 파라미터 작업 \(p. 985\)](#)
- [Amazon RDS에서 PostgreSQL Autovacuum 사용 \(p. 994\)](#)
- [PostgreSQL DB 인스턴스에 대한 감사 로깅 \(p. 1002\)](#)
- [pgaudit 확장 작업 \(p. 1003\)](#)
- [pg_repack 확장 작업 \(p. 1004\)](#)
- [PostGIS 작업 \(p. 1005\)](#)
- [pgBadger를 사용한 PostgreSQL의 로그 분석 \(p. 1007\)](#)
- [pg_config의 내용 보기 \(p. 1007\)](#)
- [orafce 확장 작업 \(p. 1008\)](#)
- [postgres_fdw 확장으로 외부 데이터 액세스 \(p. 1009\)](#)
- [아웃바운드 네트워크 액세스에 사용자 지정 DNS 서버 사용 \(p. 1009\)](#)
- [암호 관리 제한 \(p. 1010\)](#)

역할 생성

DB 인스턴스를 생성할 때 마스터 사용자 시스템 계정은 `rds_superuser` 역할에 할당됩니다.

`rds_superuser` 역할은 PostgreSQL 수퍼유저 역할(로컬 인스턴스에 사용자 정의로 명명된 `postgres`)과 마찬가지로 사전에 정의된 Amazon RDS 역할이지만 일부 제한 사항이 있습니다. PostgreSQL 수퍼유저 역할과 마찬가지로 `rds_superuser` 역할에는 DB 인스턴스에 대한 최대의 권한이 있습니다. 사용자에게 DB 인스턴스에 대한 최대의 액세스 권한이 필요하지 않다면 사용자에게 이 역할을 할당하면 안 됩니다.

`rds_superuser` 역할은 다음을 수행할 수 있습니다.

- Amazon RDS와 함께 사용할 수 있는 확장 기능을 추가합니다. 자세한 내용은 [지원되는 PostgreSQL 기능 \(p. 1054\)](#) 및 [PostgreSQL 문서](#)를 참조하십시오.
- 테이블스페이스 생성 및 삭제를 포함한 테이블스페이스 관리. 자세한 내용은 PostgreSQL 문서에서 [테이블스페이스](#) 단원을 참조하십시오.
- `pg_stat_activity` 명령을 사용하여 `rds_superuser` 역할이 할당되지 않은 모든 사용자를 보고, `pg_terminate_backend` 및 `pg_cancel_backend` 명령을 사용하여 연결 중단
- `rds_superuser` 역할이 아닌 모든 역할에 대해 `rds_replication` 역할을 부여하고 취소합니다. 자세한 내용은 PostgreSQL 문서에서 [GRANT](#) 단원을 참조하십시오.

다음은 사용자를 생성한 후 사용자에게 `rds_superuser` 역할을 부여하는 방법을 나타낸 예제입니다. 여기서는 `rds_superuser` 같은 사용자 정의 역할을 부여해야 합니다.

```
create role testuser with password 'testuser' login;
CREATE ROLE
grant rds_superuser to testuser;
GRANT ROLE
```

PostgreSQL 데이터베이스 액세스 관리

PostgreSQL용 Amazon RDS에서 어느 사용자가 어느 데이터베이스에 연결할 권한이 있는지 관리할 수 있습니다. 다른 PostgreSQL 환경에서 `pg_hba.conf` 파일을 수정하여 이러한 관리를 수행하는 경우도 있습니다. Amazon RDS에서 데이터베이스 권한을 대신 사용할 수 있습니다.

PostgreSQL의 새 데이터베이스는 항상 기본 권한 세트를 사용하여 생성됩니다. 기본 권한을 사용하면 `PUBLIC`(모든 사용자)이 데이터베이스에 연결하고 연결된 동안 임시 테이블을 생성할 수 있습니다.

Amazon RDS의 주어진 데이터베이스에 연결할 수 있는 사용자를 제어하려면 먼저 기본 `PUBLIC` 권한을 취소합니다. 그런 다음 권한을 다시 세부적으로 허용합니다. 다음 예제 코드에서는 작업 방법을 보여줍니다.

```
psql> revoke all on database <database-name> from public;
psql> grant connect, temporary on database <database-name> to <user/role name>;
```

PostgreSQL 데이터베이스의 권한에 대한 자세한 내용은 PostgreSQL 설명서에서 [GRANT](#) 명령을 참조하십시오.

PostgreSQL 파라미터 작업

`postgresql.conf` 파일에서 로컬 PostgreSQL 인스턴스의 PostgreSQL 파라미터를 설정하면 DB 인스턴스의 DB 파라미터 그룹에 유지됩니다. 기본 파라미터 그룹을 사용해 DB 인스턴스를 생성하면 파라미터 설정은 `default.postgres9.6`이라는 이름의 파라미터 그룹에 저장됩니다.

DB 인스턴스를 생성하면 연동되어 있는 DB 파라미터 그룹의 파라미터가 로드됩니다. 여기서 파라미터 값은 변경할 수 있습니다. 또한 변경할 수 있는 보안 권한이 있는 경우에는 `ALTER DATABASE`, `ALTER ROLE` 및 `SET` 명령을 사용해서도 파라미터 값 변경이 가능합니다. 호스트에 대한 액세스 권한이 없기 때문에 명령줄 `postgres` 명령 또는 `env PGOPTIONS` 명령은 사용할 수 없습니다.

PostgreSQL 파라미터 설정은 경우에 따라 추적이 어렵기도 합니다. 다음은 현재 파라미터 설정과 기본 값을 표시하는 명령입니다.

```
select name, setting, boot_val, reset_val, unit
from pg_settings
order by name;
```

출력 값에 대한 자세한 설명은 PostgreSQL 문서의 [pg_settings](#) 항목을 참조하십시오.

`max_connections`, `shared_buffers` 또는 `effective_cache_size`의 메모리를 너무 크게 설정하면 PostgreSQL 인스턴스가 시작되지 않습니다. 익숙하지 않은 단위를 사용하는 파라미터도 있습니다. 예를 들어 `shared_buffers`는 서버에서 사용하는 8KB 공유 메모리 버퍼의 수치를 설정합니다.

다음은 인스턴스를 시작하려고 하지만 잘못된 파라미터 설정으로 인해 시작되지 않을 때 `postgres.log` 파일로 출력되는 오류입니다.

```
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.
```

PostgreSQL 파라미터는 정적과 동적, 두 가지 유형이 있습니다. 정적 파라미터를 적용하려면 DB 인스턴스를 재부팅해야 합니다. 동적 파라미터는 바로 적용됩니다. 다음 표는 PostgreSQL DB 인스턴스에서 변경할 수 있는 파라미터와 각 파라미터 유형을 나타냅니다.

파라미터 이름	적용 유형	설명
<code>application_name</code>	동적	애플리케이션 이름이 통계 및 로그에 표시되도록 설정합니다.
<code>array_nulls</code>	동적	어레이의 NULL 요소 입력을 활성화합니다.
<code>authentication_timeout</code>	동적	클라이언트 인증 완료를 위한 최대 허용 시간을 설정합니다.
<code>autovacuum</code>	동적	<code>autovacuum</code> 서브프로세스를 시작합니다.
<code>autovacuum_analyze_scale_factor</code>	동적	분석 전 삽입, 업데이트 또는 삭제되는 튜플 수를 <code>reltuples</code> 변수 값으로 지정합니다.
<code>autovacuum_analyze_threshold</code>	동적	분석 전 삽입, 업데이트 또는 삭제되는 튜플의 최소 수를 지정합니다.
<code>autovacuum_naptime</code>	동적	<code>autovacuum</code> 실행 간 절전 시간을 지정합니다.
<code>autovacuum_vacuum_cost_delay</code>	동적	<code>autovacuum</code> 에서 <code>vacuum</code> 코스트 지연 시간(밀리초)을 지정합니다.
<code>autovacuum_vacuum_cost_limit</code>	동적	<code>autovacuum</code> 에서 지연 시간 이전에 이용 가능한 <code>vacuum</code> 코스트 값을 지정합니다.
<code>autovacuum_vacuum_scale_factor</code>	동적	<code>vacuum</code> 전 업데이트 또는 삭제되는 튜플 수를 <code>reltuples</code> 변수 값으로 지정합니다.
<code>autovacuum_vacuum_threshold</code>	동적	<code>vacuum</code> 전 업데이트 또는 삭제되는 튜플의 최소 수를 지정합니다.
<code>backslash_quote</code>	동적	문자열 리터럴에서 백슬래시(\)의 허용 여부를 설정합니다.
<code>bgwriter_delay</code>	동적	라운드 사이에 백그라운드 라이터의 절전 시간을 지정합니다.
<code>bgwriter_lru_maxpages</code>	동적	백그라운드 라이터가 라운드마다 LRU 페이지를 작성할 최대 수를 지정합니다.

파라미터 이름	적용 유형	설명
<code>bgwriter_lru_multiplier</code>	동적	라운드마다 해제할 평균 버퍼 사용량의 배수를 지정합니다.
<code>bytea_output</code>	동적	바이트의 출력 형식을 설정합니다.
<code>check_function_bodies</code>	동적	CREATE FUNCTION 도중 함수 본문을 검사합니다.
<code>checkpoint_completion_target</code>	동적	체크포인트 도중 변경된 버퍼 플러시에 사용된 시간으로 체크포인트 간격의 분수 값입니다.
<code>checkpoint_segments</code>	동적	로그 세그먼트에서 자동 write-ahead log(WAL) 체크포인트의 최대 간격을 설정합니다.
<code>checkpoint_timeout</code>	동적	자동 WAL 체크포인트 사이의 최대 시간을 설정합니다.
<code>checkpoint_warning</code>	동적	체크포인트 세그먼트가 이 파라미터 값보다 더 빨리 채워지는 경우 경고를 활성화합니다.
<code>client_encoding</code>	동적	클라이언트 문자 세트 인코딩을 설정합니다.
<code>client_min_messages</code>	동적	클라이언트에게 보여지는 메시지 수준을 설정합니다.
<code>commit_delay</code>	동적	트랜잭션 커밋부터 디스크에 대한 WAL 플러시까지 지연 시간(밀리초)을 설정합니다.
<code>commit_siblings</code>	동적	<code>commit_delay</code> 실행 전에 동시에 열려 있는 트랜잭션 최소 개수를 설정합니다.
<code>constraint_exclusion</code>	동적	planner가 제약 조건을 사용하여 쿼리를 최적화하도록 활성화합니다.
<code>cpu_index_tuple_cost</code>	동적	인덱스 스캔 중 각 인덱스 항목을 처리하는 데 따른 planner의 예상 코스트를 설정합니다.
<code>cpu_operator_cost</code>	동적	각 연산자 또는 함수 호출을 처리하는 데 따른 planner의 예상 코스트를 설정합니다.
<code>cpu_tuple_cost</code>	동적	각 투플(행)을 처리하는 데 따른 planner의 예상 코스트를 설정합니다.
<code>cursor_tuple_fraction</code>	동적	planner가 예상하는 검색할 커서 행의 분수 값을 설정합니다.
<code>datestyle</code>	동적	날짜와 시간 값에 대한 표시 형식을 설정합니다.
<code>deadlock_timeout</code>	동적	교착 상태 여부를 확인하기 이전 잠금 대기 시간을 설정합니다.
<code>debug_pretty_print</code>	동적	구문과 실행 계획 트리를 들여쓰기 하여 표시합니다.
<code>debug_print_parse</code>	동적	각 쿼리의 구문 분석 트리를 기록합니다.
<code>debug_print_plan</code>	동적	각 쿼리의 실행 계획을 기록합니다.
<code>debug_print_rewritten</code>	동적	각 쿼리에서 재작성된 구문 분석 트리를 기록합니다.
<code>default_statistics_target</code>	동적	기본 통계 대상을 설정합니다.
<code>default_tablespace</code>	동적	테이블과 인덱스를 생성할 기본 테이블스페이스를 설정합니다.

파라미터 이름	적용 유형	설명
default_transaction_deferrable	동적	새로운 트랜잭션의 기본 deferrable 상태를 설정합니다.
default_transaction_isolation	동적	새로운 트랜잭션마다 트랜잭션 격리 수준을 설정합니다.
default_transaction_read_only	동적	새로운 트랜잭션의 기본 읽기 전용 상태를 설정합니다.
default_with_oids	동적	새로운 테이블을 생성할 때 OID가 기본적으로 포함됩니다.
effective_cache_size	동적	디스크 캐시 크기에 대한 planner의 가정을 설정합니다.
effective_io_concurrency	동적	디스크 하위 시스템에서 효율적으로 동시에 처리할 수 있는 요청 수를 지정합니다.
enable_bitmapscan	동적	planner가 비트맵 스캔 계획을 사용할 수 있도록 활성화합니다.
enable_hashagg	동적	planner가 해시된 집계 계획을 사용할 수 있도록 활성화합니다.
enable_hashjoin	동적	planner가 해시 조인 계획을 사용할 수 있도록 활성화합니다.
enable_indexscan	동적	planner가 인덱스 스캔 계획을 사용할 수 있도록 활성화합니다.
enable_material	동적	planner가 구체화를 사용할 수 있도록 활성화합니다.
enable_mergejoin	동적	planner가 병합 조인 계획을 사용할 수 있도록 활성화합니다.
enable_nestloop	동적	planner가 중첩 루프 조인 계획을 사용할 수 있도록 활성화합니다.
enable_seqscan	동적	planner가 순차적 스캔 계획을 사용할 수 있도록 활성화합니다.
enable_sort	동적	planner가 명시적 정렬 단계를 사용할 수 있도록 활성화합니다.
enable_tidscan	동적	planner가 TID 스캔 계획을 사용할 수 있도록 활성화합니다.
escape_string_warning	동적	일반 문자열 리터럴의 백슬래시(\) 이스케이프에 대해 경고합니다.
extra_float_digits	동적	부동 소수점으로 표시할 자릿수를 설정합니다.
fromCollapse_limit	동적	서브 쿼리가 축소되지 않는 FROM 목록 크기를 설정합니다.
fsync	동적	업데이트를 디스크와 강제로 동기화합니다.
full_page_writes	동적	체크포인트 후 최초 변경 시 전체 페이지를 WAL에 기입합니다.
geqo	동적	유전적 쿼리 최적화를 활성화합니다.

파라미터 이름	적용 유형	설명
geqo_effort	동적	GEQO: 다른 GEQO 파라미터의 기본값을 설정하는 데 사용됩니다.
geqo_generations	동적	GEQO: 알고리즘의 반복 횟수입니다.
geqo_pool_size	동적	GEQO: 모집단의 개체 수입니다.
geqo_seed	동적	GEQO: 무작위 경로 선택을 위한 시드(seed)를 지정합니다.
geqo_selection_bias	동적	GEQO: 모집단 내 선택적 압력을 지정합니다.
geqo_threshold	동적	GEQO가 사용되는 FROM 항목의 임계값을 설정합니다.
gin_fuzzy_search_limit	동적	정확한 GIN 기준 검색에 허용되는 최대 결과 수를 설정합니다.
hot_standby_feedback	동적	핫 스탠바이가 피드백 메시지를 기본 또는 업스트림 스탠바이로 전송하는지 여부를 결정합니다.
intervalstyle	동적	간격 값에 대한 표시 형식을 설정합니다.
joinCollapse_limit	동적	JOIN 구문이 결합되지 않는 FROM 목록 크기를 설정합니다.
lc_messages	동적	메시지 표시 언어를 설정합니다.
lc_monetary	동적	통화 금액의 형식으로 사용할 로캘을 설정합니다.
lc_numeric	동적	숫자의 형식으로 사용할 로캘을 설정합니다.
lc_time	동적	날짜와 시간 값의 형식으로 사용할 로캘을 설정합니다.
log_autovacuum_min_duration	동적	autovacuum 작업이 기록되는 최소 실행 시간을 설정합니다.
log_checkpoints	동적	각 체크포인트를 기록합니다.
log_connections	동적	성공한 연결을 모두 기록합니다.
log_disconnections	동적	지속 시간을 포함해 세션 종료를 기록합니다.
log_duration	동적	완료된 개별 SQL 문의 지속 시간을 기록합니다.
log_error_verbosity	동적	기록된 메시지의 세부 사항을 설정합니다.
log_executor_stats	동적	실행기 성능 통계를 서버 로그에 기록합니다.
log_filename	동적	로그 파일의 이름 패턴을 설정합니다.
log_hostname	동적	연결 로그에 호스트 이름을 기록합니다.
log_lock_waits	동적	오랜 잠금 대기 시간을 기록합니다.
log_min_duration_statement	동적	문이 기록되는 최소 실행 시간을 설정합니다.
log_min_error_statement	동적	이 수준 이상으로 오류 원인이 되는 모든 문을 기록합니다.

파라미터 이름	적용 유형	설명
<code>log_min_messages</code>	동적	기록되는 메시지 수준을 설정합니다.
<code>log_parser_stats</code>	동적	구문 분석기 성능 통계를 서버 로그에 기록합니다.
<code>log_planner_stats</code>	동적	planner 성능 통계를 서버 로그에 기록합니다.
<code>log_rotation_age</code>	동적	N분 후에 자동 로그 파일 로테이션이 일어납니다.
<code>log_rotation_size</code>	동적	N킬로바이트 후에 자동 로그 파일 로테이션이 일어납니다.
<code>log_statement</code>	동적	기록할 문 유형을 설정합니다.
<code>log_statement_stats</code>	동적	누적 성능 통계를 서버 로그에 기록합니다.
<code>log_temp_files</code>	동적	이 킬로바이트 수치보다 큰 임시 파일의 사용을 기록합니다.
<code>maintenance_work_mem</code>	동적	유지 관리 작업에 사용할 최대 메모리를 설정합니다.
<code>max_stack_depth</code>	동적	최대 스택 깊이(KB)를 설정합니다.
<code>max_standby_archive_delay</code>	동적	핫 스탠바이 서버가 아카이브 WAL 데이터를 처리할 때 큐리 취소까지 걸리는 최대 지연 시간을 설정합니다.
<code>max_standby_streaming_delay</code>	동적	핫 스탠바이 서버가 스트리밍 WAL 데이터를 처리할 때 큐리 취소까지 걸리는 최대 지연 시간을 설정합니다.
<code>max_wal_size</code>	정적	검사 절차를 트리거하는 WAL 크기를 설정합니다. PostgreSQL 버전 9.6 이하의 경우 <code>max_wal_size</code> 는 16MB 단위입니다. PostgreSQL 버전 10 이상의 경우 <code>max_wal_size</code> 는 1MB 단위입니다.
<code>min_wal_size</code>	정적	WAL을 축소할 최소 크기를 설정합니다. PostgreSQL 버전 9.6 이하의 경우 <code>min_wal_size</code> 는 16MB 단위입니다. PostgreSQL 버전 10 이상의 경우 <code>min_wal_size</code> 는 1MB 단위입니다.
<code>quote_all_identifiers</code>	동적	SQL 조작 생성 시 모든 식별자에 인용 부호(")를 추가합니다.
<code>random_page_cost</code>	동적	비순차적으로 가져온 디스크 페이지에 대한 planner의 예상 코스트를 설정합니다.
<code>rds.adaptive_autovacuum</code>	동적	트랜잭션 ID 임계값이 초과될 때마다 autovacuum 파라미터를 자동으로 조정합니다.
<code>rds.log_retention_period</code>	동적	Amazon RDS가 N분보다 오래된 PostgreSQL 로그를 삭제하도록 로그 보존을 설정합니다.
<code>rds.restrict_password_commands</code>	정적	암호를 관리할 수 있는 사람을 <code>rds_password 역할</code> 이 있는 사용자로 제한합니다. 암호 제한을 활성화하려면 이 파라미터를 1로 설정합니다. 기본값은 0입니다.
<code>search_path</code>	동적	스키마로 한정되지 않은 이름의 스키마 검색 순서를 설정합니다.
<code>seq_page_cost</code>	동적	순차적으로 가져온 디스크 페이지에 대한 planner의 예상 코스트를 설정합니다.

파라미터 이름	적용 유형	설명
session_replication_role	동적	트리거 및 다시 쓰기 규칙에 대한 세션 동작을 설정합니다.
sql_inheritance	동적	다양한 명령에서 서브테이블이 기본적으로 상속됩니다.
ssl_renegotiation_limit	동적	암호화 키를 재협상하기 전에 전송 및 수신할 트래픽 양을 설정합니다.
standard_conforming_strings	동적	... 문자열에서 백슬래시가 리터럴로 처리됩니다.
statement_timeout	동적	모든 문에 허용되는 최대 지속 시간을 설정합니다.
synchronize_seqscans	동적	동기 방식의 순차적 스캔을 활성화합니다.
synchronous_commit	동적	현재 트랜잭션 동기화 수준을 설정합니다.
tcp_keepalives_count	동적	TCP keepalive의 최대 재전송 횟수를 지정합니다.
tcp_keepalives_idle	동적	TCP keepalive의 실행 주기를 지정합니다.
tcp_keepalives_interval	동적	TCP keepalive의 재전송 주기를 지정합니다.
temp_buffers	동적	각 세션에서 사용하는 임시 버퍼의 최대 수를 설정합니다.
temp_tablespaces	동적	임시 테이블 및 정렬 파일에 사용할 테이블스페이스를 설정합니다.
timezone	동적	타임스탬프를 표시 및 해석할 시간대를 설정합니다.
track_activities	동적	명령 실행에 대한 정보를 수집합니다.
track_counts	동적	데이터베이스 작업에 관한 통계를 수집합니다.
track_functions	동적	데이터베이스 작업에 관한 함수 수준 통계를 수집합니다.
track_io_timing	동적	데이터베이스 I/O 작업에 관한 시간 통계를 수집합니다.
transaction_deferrable	동적	잠재적 직렬화 오류 없이 실행될 때까지 직렬화가 가능한 읽기 전용 트랜잭션의 지연 여부를 결정합니다.
transaction_isolation	동적	현재 트랜잭션 격리 수준을 설정합니다.
transaction_read_only	동적	현재 트랜잭션의 읽기 전용 상태를 설정합니다.
transform_null_equals	동적	expr=NULL을 expr IS NULL로 처리합니다.
update_process_title	동적	프로세스 제목을 업데이트하여 활성 SQL 명령을 표시합니다.
vacuum_cost_delay	동적	vacuum 코스트 지연 시간(밀리초)을 지정합니다.
vacuum_cost_limit	동적	지연 시간 이전에 이용 가능한 vacuum 코스트 값을 지정합니다.
vacuum_cost_page_dirty	동적	vacuum으로 페이지 변경 시 부과되는 vacuum 코스트를 지정합니다.

파라미터 이름	적용 유형	설명
vacuum_cost_page_hit	동적	버퍼 캐시에서 발견되는 페이지에 대한 vacuum 코스트를 지정합니다.
vacuum_cost_page_miss	동적	버퍼 캐시에서 발견되지 않는 페이지에 대한 vacuum 코스트를 지정합니다.
vacuum_defer_cleanup_age	동적	vacuum 및 hot cleanup을 연기해야 하는 트랜잭션 수를 지정합니다(있는 경우).
vacuum_freeze_min_age	동적	vacuum에서 테이블 행을 동결해야 하는 최소 기간을 지정합니다.
vacuum_freeze_table_age	동적	vacuum에서 전체 테이블을 스캔하여 튜플을 동결해야 하는 기간을 지정합니다.
wal_writer_delay	동적	WAL 플러시 사이에 WAL 작성기의 절전 시간을 지정합니다.
work_mem	동적	쿼리 작업 공간에 사용할 최대 메모리를 설정합니다.
xmlbinary	동적	XML에서 바이너리 값의 인코딩 방식을 설정합니다.
xmloption	동적	암시적 구문 분석 및 직렬화 작업에서 XML 데이터를 문서 또는 내용 조각으로 간주할지 여부를 설정합니다.
autovacuum_freeze_max_age	정적	트랜잭션 ID 랙어라운드를 방지하기 위한 테이블의 autovacuum 기간을 지정합니다.
autovacuum_max_workers	정적	autovacuum 작업자 프로세스를 동시에 실행할 수 있는 최대 수를 설정합니다.
max_connections	정적	동시에 접속할 수 있는 최대 수를 설정합니다.
max_files_per_process	정적	서버 프로세스마다 파일을 동시에 열 수 있는 최대 수를 설정합니다.
max_locks_per_transaction	정적	하나의 트랜잭션에서 사용할 수 있는 최대 잠금 횟수를 설정합니다.
max_pred_locks_per_transaction	정적	하나의 트랜잭션에서 사용할 수 있는 최대 술어 (predicate) 잠금 횟수를 설정합니다.
max_prepared_transactions	정적	트랜잭션을 동시에 준비할 수 있는 최대 수를 설정합니다.
shared_buffers	정적	서버에서 사용할 공유 메모리 버퍼의 수를 설정합니다.
ssl	정적	SSL 연결을 활성화합니다.
temp_file_limit	정적	임시 파일이 증가할 수 있는 최대 크기(KB)를 설정합니다.
track_activity_query_size	정적	pg_stat_activity.current_query에 예약되는 크기(바이트)를 설정합니다.
wal_buffers	정적	WAL 기능을 위해 공유 메모리에서 사용할 디스크 페이지 버퍼 수를 설정합니다.

Amazon RDS는 모든 파라미터에서 기본 PostgreSQL 단위를 사용합니다. 다음 표는 PostgreSQL 기본 단위 및 각 파라미터에 대한 값을 나타냅니다.

파라미터 이름	단위
effective_cache_size	8KB
segment_size	8KB
shared_buffers	8KB
temp_buffers	8KB
wal_buffers	8KB
wal_segment_size	8KB
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

파라미터 이름	단위
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
wal_receiver_status_interval	s

Amazon RDS에서 PostgreSQL Autovacuum 사용

PostgreSQL 데이터베이스용 autovacuum 기능을 사용하여 PostgreSQL DB 인스턴스의 상태를 유지 관리하는 것이 좋습니다. Autovacuum은 VACUUM 및 ANALYZE 명령의 실행을 자동화합니다. Autovacuum은 삽입되고 업데이트되거나 삭제된 튜플 수가 많은 테이블이 있는지 확인합니다. 그런 다음 Autovacuum은 PostgreSQL 데이터베이스에서 폐기된 데이터 또는 튜플을 제거하여 스토리지를 회수합니다.

Autovacuum은 모든 새로운 Amazon RDS PostgreSQL DB 인스턴스에 대해 기본적으로 활성화되며, 관련 Autovacuum 구성 파라미터는 적절히 기본적으로 설정됩니다. 기본값이 일반적으로 설정되어 있으므로 특정 워크로드에 맞게 파라미터를 조정하면 유용하게 사용할 수 있습니다. 다음 단원에는 필요한 autovacuum 조정을 수행하는 데 도움이 되는 정보가 나와 있습니다.

주제

- [Autovacuum에 메모리 할당 \(p. 994\)](#)
- [트랜잭션 ID 랙어라운드의 가능성 감소 \(p. 995\)](#)
- [데이터베이스의 테이블을 Vacuum해야 하는지 여부를 결정 \(p. 996\)](#)
- [현재 Autovacuum을 수행할 수 있는 테이블 결정 \(p. 997\)](#)
- [현재 Autovacuum이 실행 중인지 여부 및 실행 기간 확인 \(p. 997\)](#)
- [수동 vacuum freeze 수행 \(p. 999\)](#)
- [Autovacuum이 실행 중인 경우 테이블 인덱스 다시 지정 \(p. 1000\)](#)
- [autovacuum에 영향을 주는 기타 파라미터 \(p. 1001\)](#)
- [테이블 수준 Autovacuum 파라미터 설정 \(p. 1001\)](#)
- [Autovacuum 로깅 \(p. 1002\)](#)

Autovacuum에 메모리 할당

autovacuum 성능에 영향을 미치는 가장 중요한 파라미터 중 하나는 `maintenance_work_mem` 파라미터입니다. 이 파라미터는 autovacuum에서 데이터베이스 테이블을 스캔하고 vacuum되는 모든 행 ID를 보관하는데 사용할 수 있는 메모리를 얼마큼 할당할지를 결정합니다. `maintenance_work_mem` 파라미터 값을 너무 낮게 설정하면 vacuum 프로세스가 테이블을 여러 번 스캔해야 작업이 완료될 수 있습니다. 이러한 다중 스캔은 성능에 부정적인 영향을 줄 수 있습니다.

`maintenance_work_mem` 파라미터 값을 결정하는 계산을 할 때 다음 두 가지 사항을 유의하십시오.

- 이 파라미터의 기본 단위는 킬로바이트(KB)입니다.

- `maintenance_work_mem` 파라미터는 `autovacuum_max_workers` 파라미터와 함께 작동합니다. 작은 테이블이 많이 있는 경우에는 `autovacuum_max_workers`를 더 많이 할당하고 `maintenance_work_mem`을 더 적게 할당합니다. 큰 테이블이 많이 있는 경우(100GB 이상)에는 메모리를 더 많이 할당하고 작업자 프로세스를 더 적게 할당합니다. 가장 큰 테이블에서 성공적으로 작업을 수행하려면 충분한 메모리를 할당해 두어야 합니다. 각각의 `autovacuum_max_workers`는 할당된 메모리를 사용할 수 있습니다. 따라서 작업자 프로세스와 메모리를 합한 양이 할당하려는 전체 메모리 양과 같도록 해야 합니다.

일반적으로 큰 호스트의 경우 `maintenance_work_mem` 파라미터를 1~2기가바이트 사이(1,048,576 ~ 2,097,152KB) 값으로 설정합니다. 매우 큰 호스트의 경우 파라미터를 2~4기가바이트 사이(2,097,152 ~ 4,194,304KB) 값으로 설정합니다. 이 파라미터에 설정하는 값은 워크로드에 따라 달라져야 합니다. Amazon RDS는 이 파라미터의 기본값이 다음과 같이 계산된 킬로바이트 값이 되도록 업데이트했습니다.

`GREATEST({DBInstanceClassMemory/63963136*1024}, 65536)`를 선택하십시오.

트랜잭션 ID 랩어라운드의 가능성 감소

경우에 따라 Autovacuum과 관련된 파라미터 그룹 설정이 트랜잭션 ID 랩어라운드를 방지하기에 충분히 공격적이지 않을 수 있습니다. 이를 해결하기 위해 PostgreSQL용 Amazon RDS는 Autovacuum 파라미터 값을 자동으로 조정하는 메커니즘을 제공합니다. 적응형 autovacuum 파라미터 튜닝은 PostgreSQL 버전 9.4 이상을 위한 RDS의 기능입니다. `TransactionID wraparound`에 대한 자세한 설명은 PostgreSQL 설명서에 나와 있습니다.

동적 파라미터 `rds.adaptive_autovacuum`이 ON으로 설정된 RDS PostgreSQL 인스턴스의 경우 적응형 autovacuum 파라미터 튜닝이 기본적으로 활성화됩니다. 이 설정을 항상 활성화해 놓는 것이 좋습니다. 그러나 적응형 Autovacuum 파라미터 튜닝을 꺼려면 `rds.adaptive_autovacuum` 파라미터를 0 또는 OFF로 설정합니다.

RDS가 autovacuum 파라미터를 조정하더라도 트랜잭션 ID 랩어라운드는 계속 가능합니다. 트랜잭션 ID 랩어라운드에 대한 Amazon CloudWatch 정보를 구현하는 것이 좋습니다. 자세한 내용은 Amazon RDS에서 블로그 게시물 [PostgreSQL용 Amazon RDS의 트랜잭션 ID 랩어라운드용 조기 경고 시스템 구현](#)을 참조하십시오.

적응형 Autovacuum 파라미터 튜닝을 활성화하면 RDS는 CloudWatch 지표 `MaximumUsedTransactionIDs`가 다음 값 중 큰 값에 도달하면 Autovacuum 파라미터를 조정하기 시작합니다.

- 750,000,000
- Autovacuum 파라미터 `autovacuum_freeze_max_age`

테이블이 계속 트랜잭션 ID 랩어라운드 방향으로 향하면 RDS는 Autovacuum의 파라미터를 계속 조정합니다. 이러한 각각의 조정은 랩어라운드를 피하기 위해 Autovacuum에 더 많은 리소스를 할애합니다. RDS는 다음 Autovacuum 관련 파라미터를 업데이트합니다.

- `autovacuum_vacuum_cost_delay`
- `autovacuum_vacuum_cost_limit`
- `autovacuum_work_mem`
- `autovacuum_naptime`

RDS는 새 값이 Autovacuum을 보다 공격적으로 만드는 경우에만 이러한 파라미터를 수정합니다. 파라미터는 DB 인스턴스의 메모리에서 수정됩니다. 파라미터 그룹의 값은 변경되지 않습니다. 현재 인 메모리 설정을 보려면 PostgreSQL `SHOW` SQL 명령을 사용하십시오.

RDS가 이러한 Autovacuum 파라미터를 수정하면 RDS API를 통해 AWS Management 콘솔(<https://console.aws.amazon.com/rds/>)에서 볼 수 있는 영향 받은 DB 인스턴스에 대한 이벤트를 생성합니다.

MaximumUsedTransactionIDs CloudWatch 지표가 임계값 미만의 값을 반환하면 RDS는 메모리의 Autovacuum 관련 파라미터를 파라미터 그룹에 지정된 값으로 다시 설정하고, 이 변경에 해당하는 다른 이벤트를 생성합니다.

데이터베이스의 테이블을 Vacuum해야 하는지 여부를 결정

다음 쿼리를 사용하여 데이터베이스의 vacuum되지 않은 트랜잭션 수를 표시할 수 있습니다. 데이터베이스 pg_database 행의 datfrozenxid 열은 해당 데이터베이스에 나타나는 정상 트랜잭션 ID의 하한값입니다. 이 열은 데이터베이스 내 테이블 단위 refrozenxid 값 중 최소값입니다.

```
SELECT datname, age(datfrozenxid) FROM pg_database ORDER BY age(datfrozenxid) desc limit 20;
```

예를 들어 앞의 쿼리를 실행하면 다음과 같은 결과가 나올 수 있습니다.

datname	age
mydb	1771757888
template0	1721757888
template1	1721757888
rdsadmin	1694008527
postgres	1693881061

(5 rows)

데이터베이스의 수명이 20억 트랜잭션 ID에 도달하면 트랜잭션 ID(XID) 랩어라운드가 발생하고 데이터베이스는 읽기 전용이 됩니다. 이 쿼리를 사용하면 지표를 생성하고 쿼리가 하루에 몇 번 실행되도록 할 수 있습니다. 기본적으로 autovacuum은 트랜잭션 수명을 200,000,000([autovacuum_freeze_max_age](#)) 미만으로 유지하도록 설정됩니다.

샘플 모니터링 전략은 다음과 같습니다.

- autovacuum_freeze_max_age 값을 2억 개 트랜잭션으로 설정하십시오.
- 테이블이 5억 개의 vacuum되지 않은 트랜잭션에 도달하면 낮은 심각도 경보가 트리거됩니다. 이 값은 타당한 값이지만 autovacuum이 계속 수행되고 있지 않음을 나타낼 수 있습니다.
- 테이블 수명이 10억이 되면 조치를 취해야 할 경보로 처리되어야 합니다. 성능상의 이유로 수명을 autovacuum_freeze_max_age에 더 가깝게 유지하려는 경우가 대부분입니다. 다음 권장 사항을 사용하여 조사하는 것이 좋습니다.
- 테이블이 15억 개의 vacuum되지 않은 트랜잭션에 도달하면 높은 심각도 경보가 트리거됩니다. 데이터베이스가 트랜잭션 ID를 사용하는 속도에 따라 이 경보는 시스템에서 autovacuum을 실행할 시간이 부족함을 나타낼 수 있습니다. 이 경우 즉시 이를 해결하는 것이 좋습니다.

테이블이 지속적으로 이 임계값을 위반하면 autovacuum 파라미터를 추가로 수정해야 합니다. 기본적으로 수동 VACUUM을 사용하면(비용에 따른 지연이 비활성화됨)은 기본 autovacuum을 사용할 때보다 더 적극적이지만 시스템 전체에 더 많이 침입할 수 있는 상태이기도 합니다.

다음과 같이 하는 것이 좋습니다.

- 모니터링 메커니즘을 숙지하고 활성화하여 가장 오래된 트랜잭션의 수명을 확인합니다.

트랜잭션 ID 랩어라운드에 대해 경고하는 프로세스 생성에 대한 자세한 내용은 AWS Database 블로그 게시물 [PostgreSQL용 Amazon RDS의 트랜잭션 ID 랩어라운드용 조기 경고 시스템 구현](#)을 참조하십시오.

- 더 많이 사용되는 테이블의 경우 autovacuum을 사용하는 것 이외에 유지 관리 기간 동안 수동 vacuum freeze를 정기적으로 수행합니다. 수동 vacuum freeze 수행에 대한 자세한 내용은 [수동 vacuum freeze 수행 \(p. 999\)](#) 단원을 참조하십시오.

현재 Autovacuum을 수행할 수 있는 테이블 결정

vacuum를 수행해야 하는 테이블이 하나이거나 두 개인 경우가 많습니다. `realfrozenid` 값이 `autovacuum_freeze_max_age`의 트랜잭션 수보다 큰 테이블은 항상 Autovacuum의 대상이 됩니다. 그렇지 않은 경우 VACUUM이 "vacuum 임계값"을 초과하여 튜플 수가 더 이상 사용되지 않는 경우 테이블이 vacuum됩니다.

`autovacuum 임계값`은 다음과 같이 정의되어 있습니다.

```
Vacuum-threshold = vacuum-base-threshold + vacuum-scale-factor * number-of-tuples
```

데이터베이스에 연결되어 있는 상태에서 다음 쿼리를 실행하여 autovacuum이 vacuum 가능한 대상으로 분류하는 테이블 목록을 확인합니다.

```
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(realfrozenid) 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 cvsfs on cvsfs.param = 'autovacuum_vacuum_scale_factor' and
c.oid = cvsfs.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(realfrozenid) >= coalesce(cfma.value::float,
autovacuum_freeze_max_age::float)
    or
    coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
coalesce(cvsfs.value::float, autovacuum_vacuum_scale_factor::float) *
c.reltuples <= n_dead_tup
    -- or 1 = 1
)
ORDER BY age(realfrozenid) DESC LIMIT 50;
```

현재 Autovacuum이 실행 중인지 여부 및 실행 기간 확인

테이블을 수동으로 vacuum해야 하는 경우 autovacuum이 현재 실행 중인지 확인해야 합니다. 실행 중이면 더 효율적으로 실행되도록 파라미터를 수정하거나 VACUUM을 수동으로 실행할 수 있도록 autovacuum을 종료해야 합니다.

다음 쿼리를 사용하여 autovacuum이 실행 중인지 여부와 얼마 동안 실행되고 있는지, 다른 세션에 대해 대기하고 있는지 확인합니다.

Amazon RDS PostgreSQL 9.6+ 이상을 사용하면 다음 쿼리를 사용합니다.

```
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;
```

쿼리를 실행하면 다음과 유사한 출력이 표시됩니다.

```
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
like '%VACUUM%'
+
|           |       |       |           |                           | ORDER BY
xact_start;
+
```

Amazon RDS PostgreSQL 9.6 이하(9.3.12 이상, 9.4.7 이상, 9.5.2 이상인 경우만 해당)의 버전을 사용한다면 다음 쿼리를 사용합니다.

```
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;
```

쿼리를 실행하면 다음과 유사한 출력이 표시됩니다.

```
datname | usename | 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;
+ | | | | |
```

몇 가지 문제로 인해 autovacuum 세션이 오래(며칠간) 실행될 수 있습니다. 이 문제는 대부분 [maintenance_work_mem](#) 파라미터 값이 테이블 크기 또는 업데이트 속도에 대해 너무 낮게 설정된 경우입니다.

다음 공식을 사용하여 [maintenance_work_mem](#) 파라미터 값을 설정하는 것이 좋습니다.

```
GREATEST({DBInstanceClassMemory/63963136*1024},65536)
```

짧은 기간 동안 실행되는 autovacuum 세션에서도 문제를 표시할 수 있습니다.

- 워크로드에 [autovacuum_max_workers](#)가 충분하지 않다고 표시될 수 있습니다. 이 경우 작업자 수를 명시해야 합니다.
- 인덱스 손상(autovacuum에 충돌이 발생하여 동일한 관계에서 다시 시작되지만 진행되지 않음)이 있다고 표시될 수 있습니다. 이 경우 수동 vacuum freeze verbose [__table__](#)을 실행하여 정확한 원인을 확인합니다.

수동 vacuum freeze 수행

vacuum 프로세스가 실행되고 있는 테이블에서 수동 vacuum을 수행하려는 경우가 있습니다. 이 작업은 수명이 20억 개 트랜잭션에 도달하거나 모니터링 종인 임계값을 초과한 테이블을 파악해 둔 경우 유용합니다.

다음 단계는 지침이며 이 프로세스를 여러 가지로 변형할 수 있습니다. 예를 들어 테스트 중에 [maintenance_work_mem](#) 파라미터 값이 너무 작게 설정되었고 테이블에 작업을 즉시 수행해야 한다고 가정해보십시오. 그러나 지금은 인스턴스를 반송하고 싶지 않을 수도 있습니다. 이전 세션의 쿼리를 사용하여 어떤 테이블이 문제이고 오랜 기간 동안 실행 중인 autovacuum 세션이 있는지 확인합니다. [maintenance_work_mem](#) 파라미터 설정도 변경해야 하지만 즉시 조치를 취해 문제가 되는 테이블을 vacuum해야 하기도 합니다. 이 경우 어떤 작업을 수행해야 하는지가 다음 절차에 나와 있습니다.

vacuum freeze를 수동으로 수행하려면

- vacuum 할 테이블이 포함되어 있는 데이터베이스에 세션 두 개를 엽니다. 두 번째 세션의 경우 "screen"을 사용하거나 연결이 끊긴 경우 세션을 유지하는 다른 유ти리티를 사용합니다.
- 첫 번째 세션에서는 테이블에서 실행 중인 autovacuum 세션의 PID를 가져옵니다. 이 작업을 수행하여 실행 중인 rdsadmin 프로세스를 완전히 보려면 Amazon RDS PostgreSQL 9.3.12 이상, 9.4.7 이상 또는 9.5.2 이상을 실행해야 합니다.

다음 쿼리를 실행하여 autovacuum 세션의 PID를 가져옵니다.

```
SELECT datname, username, pid, waiting, current_timestamp - xact_start
AS xact_runtime, query
FROM pg_stat_activity WHERE upper(query) LIKE '%VACUUM%' ORDER BY
xact_start;
```

- 세션 2에서 이 작업에 필요한 메모리 양을 계산합니다. 이 예제에서는 이 작업에 메모리를 최대 2GB까지 사용할 수 있는 것으로 보고 현재 세션의 [maintenance_work_mem](#)을 2GB로 설정합니다.

```
set maintenance_work_mem='2 GB';
SET
```

- 세션 2에서 테이블에 대한 vacuum freeze verbose 명령을 실행하십시오. 현재 PostgreSQL에서 이 작업에 대한 진행률 보고가 없음에도 작업을 확인할 수 있기 때문에 상세 정보 표시 설정이 유용하게 사용됩니다.

```
\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. 세션 1에서 autovacuum이 차단된 경우 pg_stat_activity에서 vacuum 세션에 대한 대기를 나타내는 "T"를 확인할 수 있습니다. 이 경우 다음과 같이 autovacuum 프로세스를 종료해야 합니다.

```
SELECT pg_terminate_backend('the_pid');
```

6. 이때 세션이 시작됩니다. 이 테이블이 작업 목록에서 가장 상위에 있을 것이므로 autovacuum이 즉시 다시 시작된다는 점을 알아 두어야 합니다. 세션 2에서 vacuum freeze verbose 명령을 시작한 다음 세션 1에서 autovacuum 프로세스를 종료합니다.

Autovacuum이 실행 중인 경우 테이블 인덱스 다시 지정

인덱스가 손상되면 autovacuum은 계속해서 테이블을 처리하려 하고 실패합니다. 이 경우 수동 vacuum을 시도하면 다음과 비슷한 오류 메시지가 표시됩니다.

```
mydb=# vacuum freeze pgbench_branches;
ERROR: index "pgbench_branches_test_index" contains unexpected
       zero page at block 30521
HINT: Please REINDEX it.
```

인덱스가 손상된 상태에서 테이블에 대해 autovacuum을 실행하려고 하면 이미 실행 중인 autovacuum 세션이 있음을 확인하게 됩니다. "REINDEX" 명령을 실행하면 테이블에 대한 단독 잠금을 해제합니다. 쓰기 작업과 해당 특정 인덱스를 사용하는 읽기 작업도 차단됩니다.

테이블에서 autovacuum을 실행할 때 테이블 인덱스를 다시 지정하려면

1. vacuum 할 테이블이 포함되어 있는 데이터베이스에 세션 두 개를 엽니다. 두 번째 세션의 경우 "screen"을 사용하거나 연결이 끊긴 경우 세션을 유지하는 다른 유ти리티를 사용합니다.
2. 첫 번째 세션에서는 테이블에서 실행 중인 autovacuum 세션의 PID를 가져옵니다. 이 작업을 수행하여 실행 중인 rdsadmin 프로세스를 완전히 보려면 Amazon RDS PostgreSQL 9.3.12 이상, 9.4.7 이상 또는 9.5.20 이상을 실행해야 합니다.

다음 쿼리를 실행하여 autovacuum 세션의 PID를 가져옵니다.

```
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;
```

3. 세션 2에서 reindex 명령을 실행합니다.

```
\timing on
Timing is on.
reindex index pgbench_branches_test_index;
REINDEX
Time: 9.966 ms
```

4. 세션 1에서 autovacuum이 차단된 경우 pg_stat_activity에서 vacuum 세션에 대한 대기를 나타내는 "T"를 확인할 수 있습니다. 이 경우 autovacuum 프로세스를 종료해야 합니다.

```
select pg_terminate_backend('the_pid');
```

5. 이때 세션이 시작됩니다. 이 테이블이 작업 목록에서 가장 상위에 있을 것으로 autovacuum이 즉시 다시 시작된다는 점을 알아 두어야 합니다. 세션 2에서 명령을 시작한 다음 세션 1에서 autovacuum 프로세스를 종료합니다.

autovacuum에 영향을 주는 기타 파라미터

이 쿼리를 사용하면 autovacuum 및 해당 동작에 직접 영향을 주는 일부 파라미터 값이 표시됩니다.
[autovacuum 파라미터](#)는 PostgreSQL 설명서에 자세히 설명되어 있습니다.

```
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');
```

모두 autovacuum에 영향을 주지만 가장 중요한 사항 몇 가지는 다음과 같습니다.

- [maintenance_mork_mem](#)
- [autovacuum_freeze_max_age](#)
- [autovacuum_max_workers](#)
- [autovacuum_vacuum_cost_delay](#)
- [Autovacuum_vacuum_cost_limit](#)

테이블 수준 Autovacuum 파라미터 설정

Autovacuum이 관련된 [스토리지 파라미터](#)를 테이블 수준에서 설정할 수 있습니다. 이 방법은 전체 데이터베이스의 동작을 변경하는 방법보다 더 나을 수 있습니다. 큰 테이블에 적극적인 설정을 지정해야 하지만 autovacuum이 모든 테이블에서 이와 같은 방식으로 작동하지 않도록 하려는 경우가 있을 수 있습니다.

이 쿼리를 사용하면 현재 어떤 테이블에 테이블 수준 옵션을 사용 중인지가 표시됩니다.

```
SELECT relname, reloptions  
FROM pg_class  
WHERE reloptions IS NOT null;
```

이 쿼리가 유용한 경우는 테이블 하나가 나머지 테이블보다 훨씬 더 큰 경우입니다. 300GB 테이블 하나와 1GB 미만의 테이블 30개가 있다고 가정하십시오. 이 경우 더 큰 테이블에 특정 파라미터를 설정하여 전체 시스템의 동작이 변경되지 않도록 할 수 있습니다.

```
ALTER TABLE mytable set (autovacuum_vacuum_cost_delay=0);
```

이렇게 하면 시스템의 리소스를 더 많이 사용하는 대신 이 테이블의 비용에 따른 autovacuum 지연이 비활성화됩니다. 일반적으로 autovacuum은 autovacuum_cost_limit에 도달할 때마다 autovacuum_vacuum_cost_delay에서 일시 정지됩니다. [비용에 따른 vacuum 수행](#)에 대한 자세한 내용은 PostgreSQL 설명서에서 확인할 수 있습니다.

Autovacuum 로깅

기본적으로 postgresql.log에는 autovacuum 프로세스에 대한 정보가 없습니다. PostgreSQL 9.4.5 이상을 사용 중인 경우 rds.force_autovacuum_logging_level 파라미터를 설정하면 autovacuum 작업자 작업에서 생성된 PostgreSQL 오류 로그의 출력을 확인할 수 있습니다. 허용되는 값은 disabled, debug5, debug4, debug3, debug2, debug1, info, notice, warning, error, log, fatal, 및 panic입니다. 다른 허용 가능한 값은 로그에 방대한 양의 정보를 추가할 수 있기 때문에 기본값은 disabled로 설정됩니다.

rds.force_autovacuum_logging_level 파라미터 값을 log로 설정하고 log_autovacuum_min_duration파라미터 값을 1,000 ~ 5000밀리초로 설정하는 것이 좋습니다. 이 값을 5,000으로 설정하면 Amazon RDS가 5초 이상 걸리는 모든 활동을 로그에 씁니다. 또한 애플리케이션 잠금으로 인해 Autovacuum이 의도적으로 표를 건너뛸 때 "vacuum skipped" 메시지가 표시됩니다. 문제를 해결 중이어서 더 많은 정보가 필요한 경우 debug1 또는 debug3과 같은 다른 로깅 레벨 값을 사용할 수 있습니다. 이 설정을 사용하면 오류 로그 파일에 매우 자세한 내용이 기록되므로 짧은 기간일 경우 이 디버그 파라미터를 사용합니다. 디버그 설정에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

Note

PostgreSQL 버전 9.4.7 이상은 rds_superuser 계정이 pg_stat_activity에서 autovacuum 세션을 볼 수 있도록 허용하여 autovacuum 세션의 가시성이 향상되었습니다. 예: 명령의 실행을 차단하거나, 수동으로 실행한 vacuum 명령보다 느리게 실행되는 autovacuum 세션을 식별 및 종료 가능

PostgreSQL DB 인스턴스에 대한 감사 로깅

PostgreSQL DB 인스턴스에 발생하는 로그 활동을 설정하는 파라미터가 몇 가지 있습니다. 이러한 파라미터에는 다음이 포함됩니다.

- log_statement 파라미터는 PostgreSQL 데이터베이스에서 사용자 활동 로그에 사용할 수 있습니다. 자세한 내용은 [PostgreSQL 데이터베이스 로그 파일 \(p. 331\)](#) 단원을 참조하십시오.
- rds.force_admin_logging_level 파라미터는 RDS 내부 사용자(rdsadmin)가 DB 인스턴스의 데이터베이스에서 수행한 작업을 기록하고 PostgreSQL 오류 로그에 출력을 씁니다. 허용되는 값은 disabled, debug5, debug4, debug3, debug2, debug1, info, notice, warning, error, log, fatal 및 panic입니다. 기본값은 disabled입니다.
- rds.force_autovacuum_logging_level 파라미터는 DB 인스턴스의 모든 데이터베이스에서 autovacuum 작업자 작업을 기록하고 PostgreSQL 오류 로그에 출력을 씁니다. 허용되는 값은 disabled, debug5, debug4, debug3, debug2, debug1, info, notice, warning, error, log, fatal 및 panic입니다. 기본값은 disabled입니다. rds.force_autovacuum_logging_level에 대한 Amazon RDS 권장 설정은 LOG입니다.

1000 또는 5000에서 log_autovacuum_min_duration 값을 설정합니다. 이 값을 5000으로 설정하면 5초 이상 걸리는 작업을 로그에 쓰고 "vacuum skipped" 메시지를 표시합니다. 이 파라미터에 대한 자세한 내용은 [PostgreSQL로 작업하기 위한 모범 사례 \(p. 72\)](#) 단원을 참조하십시오.

pgaudit 확장 작업

pgaudit 확장은 PostgreSQL 버전 9.6.3 이상 및 버전 9.5.7 버전 이상의 Amazon RDS에 대한 세부적인 세션 및 객체 감사 로깅을 제공합니다. 이 확장을 사용하여 세션 감사 또는 객체 감사를 활성화할 수 있습니다.

세션 감사를 사용하면 다양한 소스의 감사 이벤트를 기록할 수 있으며 사용 가능한 경우 정규화된 명령 텍스트를 포함할 수 있습니다. 예를 들어 세션 감사를 사용하여 pgaudit.log를 'READ'로 설정하면 데이터베이스에 연결된 모든 READ 문을 기록할 수 있습니다.

객체 감사를 사용하면 특정 명령을 사용하도록 감사 로깅을 구체화할 수 있습니다. 예를 들어 특정 테이블 수에서 READ 작업에 대한 로깅을 지정할 수 있습니다.

pgaudit 확장을 사용하여 객체 기반 로깅을 사용하려면

1. rds_pgaudit라는 특정 데이터베이스를 생성합니다. 다음 명령을 실행해 역할을 생성합니다.

```
CREATE ROLE rds_pgaudit;
CREATE ROLE
```

2. DB 인스턴스와 연결되어 있는 파라미터 그룹을 수정하여 pgaudit를 포함하는 공유된 사전 로드를 사용하고 pgaudit.role을 설정합니다. pgaudit.role은 rds_pgaudit 역할로 설정해야 합니다.

다음 명령은 사용자 지정 파라미터 그룹을 수정합니다.

```
aws rds modify-db-parameter-group
  --db-parameter-group-name rds-parameter-group-96
  --parameters
    "ParameterName=pgaudit.role,ParameterValue=rds_pgaudit,ApplyMethod=pending-reboot"
    "ParameterName=shared_preload_libraries,ParameterValue=pgaudit,ApplyMethod=pending-reboot"
  --region us-west-2
```

3. DB 인스턴스가 파라미터 그룹에 대한 변경 사항을 가져오도록 인스턴스를 재부팅합니다. 다음 명령은 DB 인스턴스를 재부팅합니다.

```
aws rds reboot-db-instance --db-instance-identifier rds-test-instance --region us-west-2
```

4. 다음 명령을 실행하여 pgaudit가 초기화되었는지 확인합니다.

```
show shared_preload_libraries;
shared_preload_libraries
-----
rdsutils,pgaudit
(1 row)
```

5. 다음 명령을 실행하여 pgaudit 확장을 생성합니다.

```
CREATE EXTENSION pgaudit;
CREATE EXTENSION
```

6. 다음 명령을 실행하여 pgaudit.role이 rds_pgaudit로 설정되었는지 확인합니다.

```
show pgaudit.role;
pgaudit.role
-----
rds_pgaudit
```

감사 로깅을 테스트하려면 감사하기로 선택한 여러 명령을 실행합니다. 예를 들어 다음과 같은 명령을 실행할 수 있습니다.

```
CREATE TABLE t1 (id int);
CREATE TABLE
GRANT SELECT ON t1 TO rds_pgaudit;
GRANT
select * from t1;
id
-----
(0 rows)
```

데이터베이스 로그에는 다음과 유사한 항목이 포함됩니다.

```
...
2017-06-12 19:09:49 UTC:...:rds_test@postgres:[11701]:LOG: AUDIT:
OBJECT,1,1,READ,SELECT,TABLE,public.t1,select * from t1;
...
```

로그 확인에 대한 자세한 내용은 [Amazon RDS 데이터베이스 로그 파일 \(p. 306\)](#) 단원을 참조하십시오.

pg_repack 확장 작업

pg_repack 확장을 사용하여 테이블과 인덱스에서 부풀림을 제거할 수 있습니다. 이 확장은 Amazon Amazon RDS의 PostgreSQL 버전 9.6.3 이상을 지원합니다. pg_repack 확장에 대한 자세한 내용은 [GitHub 프로젝트 설명서](#)를 참조하십시오.

pg_repack 확장을 사용하려면

1. 다음 명령을 실행하여 PostgreSQL DB 인스턴스용 Amazon RDS에 pg_repack 확장을 설치합니다.

```
CREATE EXTENSION pg_repack;
```

2. pg_repack 클라이언트 유틸리티를 사용하여 데이터베이스에 연결합니다. rds_superuser 권한이 있는 데이터베이스 역할을 사용하여 데이터베이스에 연결합니다. 다음 연결 예제에서

rds_test 역할은 rds_superuser 권한을 가지며, 사용되는 데이터베이스 엔드포인트는 rds-test-instance.cw7jjfgdr4on8.us-west-2.rds.amazonaws.com입니다.

```
pg_repack -h rds-test-instance.cw7jjfgdr4on8.us-west-2.rds.amazonaws.com -U rds_test -k postgres
```

-k 옵션을 사용하여 연결합니다. -a 옵션은 지원되지 않습니다.

- pg_repack 클라이언트의 응답은 다시 구성된 DB 인스턴스 상의 테이블에 대한 정보를 제공합니다.

```
INFO: repacking table "pgbench_tellers"  
INFO: repacking table "pgbench_accounts"  
INFO: repacking table "pgbench_branches"
```

PostGIS 작업

PostGIS는 공간 정보를 저장하고 관리하기 위해 PostgreSQL을 확장한 것입니다. PostGIS에 대해 잘 알지 못하는 경우 [PostGIS Introduction](#)에서 일반 개요를 볼 수 있습니다.

PostGIS 확장 모듈을 사용하려면 그 전에 몇 가지 설정을 수행해야 합니다. 해야 할 설정은 다음과 같습니다. 각 단계는 이번 단원에서 더욱 자세히 설명하겠습니다.

- DB 인스턴스를 생성할 때 사용한 마스터 사용자 이름으로 DB 인스턴스에 연결합니다.
- PostGIS 확장 모듈을 로드합니다.
- 확장 모듈의 소유권을 rds_superuser 역할로 이전합니다.
- 객체 소유권을 rds_superuser 역할로 이전합니다.
- 확장 모듈을 테스트합니다.

1단계: DB 인스턴스를 생성할 때 사용한 마스터 사용자 이름으로 DB 인스턴스에 연결합니다.

먼저 DB 인스턴스를 생성할 때 사용한 마스터 사용자 이름으로 DB 인스턴스에 연결합니다. 해당 이름은 rds_superuser 역할에 자동으로 할당됩니다. 나머지 단계를 수행하는데 필요한 rds_superuser 역할이 필요합니다.

다음은 SELECT를 사용하여 현재 사용자를 표시한 예제입니다. 여기서 현재 사용자는 DB 인스턴스 생성 시 선택한 마스터 사용자 이름이어야 합니다.

```
select current_user;  
current_user  
-----  
myawsuser  
(1 row)
```

2단계: PostGIS 확장 모듈을 로드합니다.

CREATE EXTENSION 문을 사용하여 PostGIS 확장 모듈을 로드합니다. 반드시 확장 모듈을 로드해야 합니다. 그런 다음 \dn psql 명령을 사용하여 PostGIS 스키마의 소유자를 표시합니다.

```
create extension postgis;
```

```
CREATE EXTENSION
create extension fuzzystrmatch;
CREATE EXTENSION
create extension postgis_tiger_geocoder;
CREATE EXTENSION
create extension postgis_topology;
CREATE EXTENSION
\dn
    List of schemas
   Name      |  Owner
-----+-----
public    | myawsuser
tiger     | rdsadmin
tiger_data | rdsadmin
topology   | rdsadmin
(4 rows)
```

3단계: 확장 모듈의 소유권을 rds_superuser 역할로 이전합니다.

ALTER SCHEMA 문을 사용하여 스키마 소유권을 rds_superuser 역할로 이전합니다.

```
alter schema tiger owner to rds_superuser;
ALTER SCHEMA
alter schema tiger_data owner to rds_superuser;
ALTER SCHEMA
alter schema topology owner to rds_superuser;
ALTER SCHEMA
\dn
    List of schemas
   Name      |  Owner
-----+-----
public    | myawsuser
tiger     | rds_superuser
tiger_data | rds_superuser
topology   | rds_superuser
(4 rows)
```

4단계: 객체 소유권을 rds_superuser 역할로 이전합니다.

다음 함수를 사용하여 PostGIS 객체 소유권을 rds_superuser 역할로 이전합니다. psql 프롬프트에서 다음 문을 실행하여 함수를 생성합니다.

```
CREATE FUNCTION exec(text) returns text language plpgsql volatile AS $$ BEGIN EXECUTE $1;
RETURN $1; END; $$;
```

그런 다음 이 쿼리로 실행 함수를 실행하면 이 함수가 해당하는 문을 실행하여 권한을 변경합니다.

```
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;
```

5단계: 확장 모듈을 테스트합니다.

다음 명령을 사용하여 tiger를 검색 경로에 추가합니다.

```
SET search_path=public,tiger;
```

다음 SELECT 문을 사용하여 tiger 모듈을 테스트합니다.

```
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)
```

다음 SELECT 문을 사용하여 topology 모듈을 테스트합니다.

```
select topology.createtopology('my_new_topo',26986,0.5);
createtopology
-----
      1
(1 row)
```

pgBadger를 사용한 PostgreSQL의 로그 분석

pgbadger 등의 로그 분석기를 사용하여 PostgreSQL 로그를 분석할 수 있습니다. pgbadger 문서에는 %I 패턴(세션/프로세스의 로그 라인)은 접두사에 포함되어야 한다고 나와 있습니다. 그러나 현재 rds_log_line_prefix를 pgbadger에 파라미터로 사용하더라도 여전히 보고서를 생성해야 합니다.

예를 들어 다음 명령은 pgbadger를 사용하여 2014-02-04 일자의 Amazon RDS PostgreSQL 로그 파일을 정확한 형식으로 나타내고 있습니다.

```
./pgbadger -p '%t:%r:%u%@%d:[%p]:' postgresql.log.2014-02-04-00
```

pg_config의 내용 보기

PostgreSQL 버전 9.6.1에서는 새로 보기 pg_config를 이용해 현재 설치된 PostgreSQL 버전의 컴파일 시간 구성 파라미터를 볼 수 있습니다. 다음 예제와 같이 pg_config 함수를 호출하여 볼 수 있습니다.

```
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-openssl' '--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 -lxmll -lssl -lcrypto -lz -lreadline -lrt -lcrypt
-lldl -lm
VERSION           | PostgreSQL 9.6.1
(23 rows)
```

바로 보기에 액세스하려고 하면 요청이 실패합니다.

```
select * from pg_config;
ERROR: permission denied for relation pg_config
```

orafce 확장 작업

orafce 확장은 상용 데이터베이스에서 흔히 사용되는 함수를 제공하며, 상용 데이터베이스를 PostgreSQL로 더 쉽게 포팅할 수 있게 합니다. Amazon RDS PostgreSQL 버전 9.6.6 이상에서 이 확장을 지원합니다. orafce에 대한 자세한 내용은 [GitHub에서 orafce 프로젝트](#)를 참조하십시오.

Note

PostgreSQL용 Amazon RDS는 orafce 확장의 일부분인 `utl_file` 패키지를 지원하지 않습니다. 이는 `utl_file` 스키마 함수가 기본 호스트에 대한 수퍼유저 권한을 필요로 하는 운영 체제 텍스트 파일의 읽기 및 쓰기 작업을 제공하기 때문입니다.

orafce 확장을 사용하려면

1. DB 인스턴스를 생성할 때 사용한 마스터 사용자 이름으로 DB 인스턴스에 연결합니다.

Note

동일한 인스턴스의 다른 데이터베이스에서 orafce를 활성화하려는 경우 연결을 시작한 후 `/c dbname psql` 명령을 사용하여 마스터 데이터베이스에서 변경하십시오.

2. `CREATE EXTENSION` 문을 사용하여 orafce 확장을 활성화합니다.

```
CREATE EXTENSION orafce;
```

3. `ALTER SCHEMA` 문을 사용하여 oracle 스키마 소유권을 `rds_superuser` 역할로 이전합니다.

```
ALTER SCHEMA oracle OWNER TO rds_superuser;
```

Note

oracle 스키마의 소유자 목록을 보려면 \dn psql 명령을 사용합니다.

postgres_fdw 확장으로 외부 데이터 액세스

postgres_fdw 확장으로 원격 데이터베이스에 있는 테이블의 데이터에 액세스할 수 있습니다. PostgreSQL DB 인스턴스에서 원격 연결을 설정하는 경우 읽기 전용 복제본에도 액세스할 수 있습니다.

postgres_fdw로 원격 데이터베이스 서버에 액세스하려면

1. postgres_fdw 확장을 설치합니다.

```
CREATE EXTENSION postgres_fdw;
```

2. CREATE SERVER로 외부 데이터 서버를 생성합니다.

```
CREATE SERVER foreign_server
FOREIGN DATA WRAPPER postgres_fdw
OPTIONS (host 'xxx.xx.xxx.xx', port '5432', dbname 'foreign_db');
```

3. 원격 서버에 사용할 역할 식별을 위하여 사용자 매핑을 생성합니다.

```
CREATE USER MAPPING FOR local_user
SERVER foreign_server
OPTIONS (user 'foreign_user', password 'password');
```

4. 원격 서버에서 테이블을 매핑할 테이블을 생성합니다.

```
CREATE FOREIGN TABLE foreign_table (
    id integer NOT NULL,
    data text)
SERVER foreign_server
OPTIONS (schema_name 'some_schema', table_name 'some_table');
```

아웃바운드 네트워크 액세스에 사용자 지정 DNS 서버 사용

PostgreSQL용 Amazon RDS는 DB 인스턴스에서 아웃바운드 네트워크 액세스를 지원하고, 고객이 소유한 사용자 지정 DNS 서버에서의 DNS(Domain Name Service) 확인을 허용합니다. 사용자 지정 DNS 서버를 통해 Amazon RDS DB 인스턴스에서 전체 주소 도메인 이름만을 확인할 수 있습니다.

주제

- 사용자 지정 DNS 확인 활성화 (p. 1010)
- 사용자 지정 DNS 확인 비활성화 (p. 1010)

- 사용자 지정 DNS 서버 설정 (p. 1010)

사용자 지정 DNS 확인 활성화

고객 VPC에서 DNS 확인을 활성화하려면 사용자 지정 DB 파라미터 그룹을 RDS PostgreSQL 인스턴스에 연결하고, rds.custom_dns_resolution 파라미터를 1로 설정하여 켜고, 변경이 발생하도록 DB 인스턴스를 다시 시작해야 합니다.

사용자 지정 DNS 확인 비활성화

고객 VPC에서 DNS 확인을 비활성화하려면 사용자 지정 DB 파라미터 그룹의 rds.custom_dns_resolution 파라미터를 0으로 설정하여 끈 후, 변경이 발생하도록 DB 인스턴스를 다시 시작해야 합니다.

사용자 지정 DNS 서버 설정

사용자 지정 DNS 이름 서버를 설정한 후 변경 사항이 DB 인스턴스에 전파되는 데 최대 30분이 걸립니다. 변경 사항이 DB 인스턴스에 전파된 후 DNS 조회를 필요로 하는 모든 아웃바운드 네트워크 트래픽은 포트 53을 통해 DNS 서버를 쿼리합니다.

Amazon RDS PostgreSQL DB 인스턴스의 사용자 지정 DNS 서버를 설정하려면 다음과 같이 하십시오.

- VPC에 연결된 DHCP 옵션 세트에서 DNS 이름 서버의 IP 주소에 대해 domain-name-servers 옵션을 설정합니다. 자세한 내용은 [DHCP 옵션 세트](#) 단원을 참조하십시오.

Note

domain-name-servers 옵션은 최대 4개의 값을 받아들이지만 Amazon RDS DB 인스턴스는 첫 번째 값만을 사용합니다.

- DNS 서버가 DNS 이름, Amazon EC2 프라이빗 DNS 이름, 고객별 DNS 이름을 비롯한 모든 조회 쿼리를 확인할 수 있는지 확인합니다. 아웃바운드 네트워크 트래픽에 DNS 서버가 처리할 수 없는 DNS 조회가 포함된 경우, DNS 서버에 적절한 업스트림 DNS 공급자가 구성되어 있어야 합니다.
- 512바이트 이하의 UDP(User Datagram Protocol) 응답을 생성하도록 DNS 서버를 구성하십시오.
- 1,024바이트 이하의 TCP(Transmission Control Protocol) 응답을 생성하도록 DNS 서버를 구성하십시오.
- 포트 53을 통한 Amazon RDS DB 인스턴스로부터의 인바운드 트래픽을 허용하도록 DNS 서버를 구성하십시오. DNS 서버가 Amazon VPC에 있는 경우, VPC에는 포트 53에서 UDP 및 TCP 트래픽을 허용하는 인바운드 규칙이 포함된 보안 그룹이 있어야 합니다. DNS 서버가 Amazon VPC에 없는 경우, 포트 53에서 UDP 및 TCP 트래픽을 허용하는 적절한 방화벽 화이트리스트가 있어야 합니다.

자세한 내용은 [VPC의 보안 그룹 및 규칙 추가 및 제거](#) 단원을 참조하십시오.

- 포트 53을 통한 아웃바운드 트래픽을 허용하도록 Amazon RDS DB 인스턴스의 VPC를 구성하십시오. VPC에는 포트 53에서 UDP 및 TCP 트래픽을 허용하는 아웃바운드 규칙이 포함된 보안 그룹이 있어야 합니다.

자세한 내용은 [VPC의 보안 그룹 및 규칙 추가 및 제거](#) 단원을 참조하십시오.

- Amazon RDS DB 인스턴스와 DNS 서버 간 라우팅 경로가 DNS 트래픽을 허용하도록 올바로 구성되어야 합니다.

Amazon RDS DB 인스턴스와 DNS 서버가 같은 VPC에 있지 않은 경우, 그 사이에 피어링 연결을 구축해야 합니다. 자세한 내용은 [VPC 피어링이란?](#)을 참조하십시오.

암호 관리 제한

데이터베이스 사용자 암호를 관리할 수 있는 사람을 특정 역할로 제한할 수 있습니다. 이렇게 하면 클라이언트 측의 암호 관리를 더 잘 제어할 수 있습니다.

정적 파라미터 `rds.restrict_password_commands`로 제한된 암호 관리를 활성화하고 `rds_password`라는 역할을 사용합니다. `rds.restrict_password_commands` 파라미터를 1로 설정하면, `rds_password` 역할의 멤버인 사용자만 특정 SQL 명령을 실행할 수 있습니다. 제한된 SQL 명령은 데이터베이스 사용자 암호와 암호 만료 시간을 수정하는 명령입니다.

제한된 암호 관리를 사용하려면 DB 인스턴스가 PostgreSQL 10.6 이상용 Amazon RDS를 실행해야 합니다. `rds.restrict_password_commands` 파라미터는 정적이므로 이 파라미터를 변경하려면 데이터베이스를 다시 시작해야 합니다.

데이터베이스에 제한된 암호 관리가 활성화되어 있을 때 제한된 SQL 명령을 실행하려고 하면 ERROR: must be a member of `rds_password` to alter passwords 오류가 표시됩니다.

다음은 제한된 암호 관리가 활성화되어 있을 때 제한되는 몇 가지 SQL 명령 예입니다.

```
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;
```

`RENAME TO`가 포함된 일부 `ALTER ROLE` 명령도 제한될 수 있습니다. 이러한 명령이 제한될 수 있는 이유는 MD5 암호가 있는 PostgreSQL 역할 이름을 바꾸면 암호가 지워지기 때문입니다.

`rds_superuser` 역할에는 기본적으로 `rds_password` 역할의 멤버십이 있으므로 변경할 수 없습니다. GRANT SQL 명령을 사용하여 다른 역할에 `rds_password` 역할의 멤버십을 제공할 수 있습니다. 암호 관리에만 사용하는 몇 가지 역할에만 `rds_password` 멤버십을 제공하는 것이 좋습니다. 이러한 역할에는 다른 역할을 수정할 `CREATEROLE` 속성이 필요합니다.

만료 및 클라이언트 측에 필요한 복잡성 등의 암호 요구 사항을 확인해야 합니다. 자체 클라이언트 측 유ти리티를 사용하여 암호 관련 변경을 제한하는 것이 좋습니다. 이 유ти리티에는 `rds_password`의 멤버이며 `CREATEROLE` 역할 속성이 있는 역할이 있어야 합니다.

데이터베이스 미리 보기 환경 작업

Amazon RDS에 DB 인스턴스를 만들 때 그 기반이 되는 PostgreSQL 버전이 테스트를 완료하였고 Amazon에서 완전히 지원된다는 사실을 알아두십시오. PostgreSQL 커뮤니티는 새로운 버전과 새 확장을 지속적으로 릴리스합니다. 완전한 지원을 받기 전에 새 PostgreSQL 버전과 확장을 써볼 수 있습니다. 이를 위해 데이터베이스 미리 보기 환경에서 새 DB 인스턴스를 만들 수 있습니다.

데이터베이스 미리 보기 환경의 DB 인스턴스는 프로덕션 환경의 DB 인스턴스와 유사합니다. 그러나 다음과 같은 중요한 요인을 기억해 두십시오.

- 모든 DB 인스턴스는 생성 60일 후 백업 및 스냅샷과 함께 삭제됩니다.
- Amazon VPC 서비스 기반의 Virtual Private Cloud(VPC)에서만 DB 인스턴스를 생성할 수 있습니다.
- M4, T2, R4 인스턴스 유형만 생성할 수 있습니다. RDS 인스턴스 클래스에 대한 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.
- DB 인스턴스에는 AWS Support의 도움말을 사용할 수 없습니다. 질문은 [RDS 데이터베이스 미리 보기 환경 포럼](#)에 올릴 수 있습니다.
- 범용 SSD와 프로비저닝된 IOPS SSD 스토리지만 사용할 수 있습니다.
- DB 인스턴스의 스냅샷을 프로덕션 환경으로 복제할 수 없습니다.
- 미리 보기 환경에서는 아래 설명과 같이 일부 Amazon RDS 기능을 사용할 수 없습니다.

주제

- 미리 보기 환경에서 지원하지 않는 기능 (p. 1012)
- 미리 보기 환경에서 지원되는 PostgreSQL 확장 (p. 1012)
- 미리 보기 환경에서 새 DB 인스턴스 생성 (p. 1013)

미리 보기 환경에서 지원하지 않는 기능

다음 기능은 미리 보기 환경에서 사용할 수 없습니다.

- 교차 리전 스냅샷 복제
- 교차 리전 읽기 전용 복제본
- 다음 지원 확장표에 없는 확장

미리 보기 환경에서 지원되는 PostgreSQL 확장

데이터베이스 미리 보기 환경에서 지원되는 PostgreSQL 확장 목록은 다음과 같습니다.

확장	버전
amcheck	1.1
블룸	1.0
btree_gin	1.3
btree_gist	1.5
citext	1.5
cube	1.4
dblink	1.2
dict_int	1.0
dict_xsyn	1.0
earthdistance	1.1
fuzzystrmatch	1.1
hstore	1.5
hstore_plper	1.0
intagg	1.1
antarray	1.2
isn	1.2
log_fdw	1.0
ltree	1.1
pg_buffercache	1.3
pg_freespacemap	1.2

확장	버전
pg_prewarm	1.2
pg_stat_statements	1.5
pg_trgm	1.4
pg_visibility	1.2
pgcrypto	1.3
pgrowlocks	1.2
pgstattuple	1.5
plperl	1.0
plpgsql	1.0
pltcl	1.0
postgres_fdw	1.0
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

미리 보기 환경에서 새 DB 인스턴스 생성

다음 절차를 이용하여 미리 보기 환경에서 DB 인스턴스를 새로 만듭니다.

미리 보기 환경에서 DB 인스턴스를 생성하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/rds/>에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택합니다.
3. Switch to database preview environment(데이터베이스 미리 보기 환경으로 전환)을 선택합니다.

Database Preview Environment

Get early access to new DB engine versions, before they're generally available. The RDS database preview environment lets you work with upcoming beta, release candidate, and early production versions of PostgreSQL engines. Preview environment instances are fully functional, so you can easily test new features and functionality with your applications.

[Info](#)

[Preview PostgreSQL in US EAST \(Ohio\)](#)

데이터베이스 미리 보기 환경으로 바로 이동할 수도 있습니다.

[Note](#)

API 또는 CLI로 데이터베이스 미리 보기 환경에 인스턴스를 만들고 싶은 경우 엔드포인트는 `rds-preview.us-east-2.amazonaws.com`입니다.

4. [PostgreSQL DB 인스턴스 생성 \(p. 948\)](#)에 서술한 절차를 계속합니다.

PostgreSQL용 Amazon RDS 버전과 확장

Amazon RDS는 여러 PostgreSQL 에디션을 실행하는 DB 인스턴스를 지원합니다. 이 섹션을 보고 Amazon RDS에서 PostgreSQL로 작업하는 방법을 확인하십시오. PostgreSQL DB 인스턴스에 대한 제한도 알고 있어야 합니다.

새 DB 인스턴스를 생성할 때는 현재 지원되는 모든 PostgreSQL 버전을 지정할 수 있습니다. 메이저 버전 (예: PostgreSQL 10) 및 지정된 메이저 버전에 대해 지원되는 모든 마이너 버전을 지정할 수 있습니다. 버전이 지정되지 않은 경우 Amazon RDS는 지원되는 버전(보통 최신 버전)을 기본값으로 설정합니다. 메이저 버전이 지정되었지만 마이너 버전이 지정되지 않은 경우, Amazon RDS는 고객이 지정한 메이저 버전의 최근 릴리스를 기본값으로 설정합니다. 지원되는 버전 목록과 새로 만든 DB 인스턴스의 기본값을 보려면 `describe-db-engine-versions` AWS CLI 명령을 사용합니다.

DB 인스턴스로 PostgreSQL 데이터를 가져오는 자세한 방법은 [Amazon RDS에서 PostgreSQL로 데이터 가져오기 \(p. 970\)](#) 단원을 참조하십시오.

주제

- [지원되는 PostgreSQL 데이터베이스 버전 \(p. 1014\)](#)
- [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#)

지원되는 PostgreSQL 데이터베이스 버전

Amazon RDS에서는 다음 PostgreSQL 버전을 지원합니다.

주제

- Amazon RDS의 PostgreSQL 버전 11.2 (p. 1016)
- Amazon RDS의 PostgreSQL 버전 11.1 (p. 1016)
- 데이터베이스 미리 보기 환경에서 Amazon RDS의 PostgreSQL 버전 11 (p. 1017)
- Amazon RDS의 PostgreSQL 버전 10.7 (p. 1017)
- Amazon RDS의 PostgreSQL 버전 10.6 (p. 1017)
- Amazon RDS의 PostgreSQL 버전 10.5 (p. 1018)
- Amazon RDS의 PostgreSQL 버전 10.4 (p. 1018)
- Amazon RDS의 PostgreSQL 버전 10.3 (p. 1019)
- Amazon RDS의 PostgreSQL 버전 10.1 (p. 1019)
- Amazon RDS의 PostgreSQL 버전 9.6.12 (p. 1020)
- Amazon RDS의 PostgreSQL 버전 9.6.11 (p. 1020)
- Amazon RDS의 PostgreSQL 버전 9.6.10 (p. 1020)
- Amazon RDS의 PostgreSQL 버전 9.6.9 (p. 1021)
- Amazon RDS의 PostgreSQL 버전 9.6.8 (p. 1021)
- Amazon RDS의 PostgreSQL 버전 9.6.6 (p. 1021)
- Amazon RDS의 PostgreSQL 버전 9.6.5 (p. 1021)
- Amazon RDS의 PostgreSQL 버전 9.6.3 (p. 1022)
- Amazon RDS의 PostgreSQL 버전 9.6.2 (p. 1022)
- Amazon RDS의 PostgreSQL 버전 9.6.1 (p. 1022)
- Amazon RDS의 PostgreSQL 버전 9.5.16 (p. 1023)
- Amazon RDS의 PostgreSQL 버전 9.5.15 (p. 1023)
- Amazon RDS의 PostgreSQL 버전 9.5.14 (p. 1024)
- Amazon RDS의 PostgreSQL 버전 9.5.13 (p. 1024)
- Amazon RDS의 PostgreSQL 버전 9.5.12 (p. 1024)
- Amazon RDS의 PostgreSQL 버전 9.5.10 (p. 1024)
- Amazon RDS의 PostgreSQL 버전 9.5.9 (p. 1024)
- Amazon RDS의 PostgreSQL 버전 9.5.7 (p. 1024)
- Amazon RDS의 PostgreSQL 버전 9.5.6 (p. 1025)
- Amazon RDS의 PostgreSQL 버전 9.5.4 (p. 1025)
- Amazon RDS의 PostgreSQL 버전 9.5.2 (p. 1025)
- Amazon RDS의 PostgreSQL 버전 9.4.21 (p. 1026)
- Amazon RDS의 PostgreSQL 버전 9.4.20 (p. 1027)
- Amazon RDS의 PostgreSQL 버전 9.4.19 (p. 1027)
- Amazon RDS의 PostgreSQL 버전 9.4.18 (p. 1027)
- Amazon RDS의 PostgreSQL 버전 9.4.17 (p. 1027)
- Amazon RDS의 PostgreSQL 버전 9.4.15 (p. 1027)
- Amazon RDS의 PostgreSQL 버전 9.4.14 (p. 1027)
- Amazon RDS의 PostgreSQL 버전 9.4.12 (p. 1027)
- Amazon RDS의 PostgreSQL 버전 9.4.11 (p. 1028)
- Amazon RDS의 PostgreSQL 버전 9.4.9 (p. 1028)
- Amazon RDS의 PostgreSQL 버전 9.4.7 (p. 1028)
- Amazon RDS의 PostgreSQL 버전 9.3.25 (p. 1028)
- Amazon RDS의 PostgreSQL 버전 9.3.24 (p. 1029)
- Amazon RDS의 PostgreSQL 버전 9.3.23 (p. 1029)
- Amazon RDS의 PostgreSQL 버전 9.3.22 (p. 1029)
- Amazon RDS의 PostgreSQL 버전 9.3.20 (p. 1029)

- [Amazon RDS의 PostgreSQL 버전 9.3.19 \(p. 1029\)](#)
- [Amazon RDS의 PostgreSQL 버전 9.3.17 \(p. 1030\)](#)
- [Amazon RDS의 PostgreSQL 버전 9.3.16 \(p. 1030\)](#)
- [Amazon RDS의 PostgreSQL 버전 9.3.14 \(p. 1030\)](#)
- [Amazon RDS의 PostgreSQL 버전 9.3.12 \(p. 1030\)](#)

Amazon RDS의 PostgreSQL 버전 11.2

PostgreSQL 버전 11.2에는 릴리스 11.1의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. PostgreSQL 11.2의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

이 버전에는 또한 다음과 같은 변경 사항이 포함됩니다.

- 새로운 [pgTAP](#) 확장 버전 1.0.
- Amazon S3 가져오기를 지원합니다. 자세한 내용은 [PostgreSQL DB 인스턴스용 RDS로 Amazon S3 데이터 가져오기 \(p. 974\)](#) 단원을 참조하십시오.
- 특정한 이전 PostgreSQL 버전에서 PostgreSQL 11.2로 여러 주요 버전 업그레이드가 가능합니다. 자세한 내용은 [PostgreSQL 메이저 버전 업그레이드 \(p. 964\)](#) 단원을 참조하십시오.

PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 11.1

PostgreSQL 버전 11에는 여러 개선 사항이 포함되어 있으며 관련 설명은 [PostgreSQL 11.1 출시](#)에 포함되었습니다. 이 버전에는 프로시저 내에 임베디드 트랜잭션을 허용하는 SQL 저장 프로시저가 포함되어 있습니다. 이 버전에는 파티셔닝과 병렬화 및 많은 유용한 성능 향상에 대한 주요 개선 사항도 포함되어 있습니다. 예를 들어, 열 기본값에 대해 NULL이 아닌 상수를 사용하면 이제 ALTER TABLE 명령을 사용하여 테이블 다시 쓰기 없이 열을 추가할 수 있습니다.

PostgreSQL 버전 11.1에는 릴리스 11.1의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 [PostgreSQL 릴리스 11.1 설명서](#)를 참조하십시오. 이 버전의 몇 가지 변경 사항은 다음과 같습니다.

- 파티셔닝 – 파티셔닝 개선 사항에는 해시 파티셔닝 지원, 기본 파티션 생성 지원, 키 열 업데이트를 기반으로 하는 다른 파티션으로의 동적 행 이동이 포함됩니다.
- 성능 – 성능 향상에는 인덱스 작성 중 병렬화, 구체화된 보기, 해시 조인 및 순차 스캔이 포함되어 작업 수 행이 향상됩니다.
- 저장 프로시저 – 이제 SQL 저장 프로시저에 임베디드 트랜잭션 지원이 추가되었습니다.
- JIT(Just-In-Time) 기능 지원 – RDS PostgreSQL 11 인스턴스는 JIT 기능으로 생성되어 표현식 평가 속도를 높입니다. 이 기능을 활성화하려면 `jit`을 ON으로 설정하십시오.
- 세그먼트 크기 – write-ahead logging(WAL) 세그먼트 크기가 16MB에서 64MB로 변경되었습니다.
- Autovacuum 개선 – 중요한 로깅을 제공하기 위해 `log_autovacuum_min_duration` 파라미터가 10초로 설정되고 `rds.force_autovacuum_logging` 파라미터는 기본적으로 ON으로 설정됩니다. autovacuum 효과를 높이기 위해, 더 큰 기본값을 제공하도록 `autovacuum_max_workers` 및 `autovacuum_vacuum_cost_limit` 파라미터의 값이 호스트 메모리 용량을 기반으로 계산됩니다.
- 향상된 트랜잭션 제한 시간 – 파라미터 `idle_in_transaction_session_timeout`이 12시간으로 설정됩니다. 12시간 이상 유지 상태였던 세션은 종료됩니다.
- 성능 지표 – `pg_stat_statements` 모듈은 기본적으로 `shared_preload_libraries`에 포함되어 있습니다. 이렇게 하면 생성 직후에 인스턴스를 재부팅하지 않아도 됩니다. 그러나 `CREATE`

EXTENSION pg_stat_statements; 문을 실행하려는 경우에도 이 기능이 필요합니다. 또한 기본적으로 track_io_timing이 활성화되어 pg_stat_statements에 보다 세부적인 데이터를 추가합니다.

- tsearch2 모듈은 더 이상 지원되지 않습니다. – 애플리케이션에서 tsearch2 함수를 사용하는 경우 핵심 PostgreSQL 엔진에서 제공하는 것과 동일한 함수를 사용하도록 업데이트하십시오. tsearch2 모듈에 대한 자세한 내용은 [PostgreSQL tsearch2 단원](#)을 참조하십시오.
- chkpass 모듈은 더 이상 지원되지 않습니다. – chkpass 모듈에 대한 자세한 내용은 [PostgreSQL chkpass 단원](#)을 참조하십시오.
- RDS PostgreSQL 11.1의 확장 업데이트에는 다음이 포함됩니다.
 - pgaudit은 1.3.0으로 업데이트됩니다.
 - pg_hint_plan은 1.3.2로 업데이트됩니다.
 - pglogical은 2.2.1로 업데이트됩니다.
 - plcoffee는 2.3.8로 업데이트됩니다.
 - plv8은 2.3.8로 업데이트됩니다.
 - PostGIS는 2.5.1로 업데이트됩니다.
 - prefix는 1.2.8로 업데이트됩니다.
 - wal2json은 해시 9e962bad로 업데이트됩니다.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

데이터베이스 미리 보기 환경에서 Amazon RDS의 PostgreSQL 버전 11

Note

Amazon RDS 기반 PostgreSQL 버전 11이 프로덕션 환경에서 릴리스되었습니다. 데이터베이스 미리 보기 환경에는 더 이상 지원되지 않습니다.

PostgreSQL 버전 11에는 여러 개선 사항이 포함되어 있으며 관련 설명은 [PostgreSQL 11 출시!](#)에 나와 있습니다.

데이터베이스 미리 보기 환경에 대한 자세한 내용은 [the section called “데이터베이스 미리 보기 환경 작업” \(p. 1011\)](#) 단원을 참조하십시오. 콘솔에서 미리 보기 환경에 액세스하려면 <https://console.aws.amazon.com/rds-preview/>를 선택합니다.

Amazon RDS의 PostgreSQL 버전 10.7

PostgreSQL 버전 10.7에는 릴리스 10.6의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 10.7의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

이 버전에는 또한 다음과 같은 변경 사항이 포함됩니다.

- Amazon S3 가져오기를 지원합니다. 자세한 내용은 [PostgreSQL DB 인스턴스용 RDS로 Amazon S3 데이터 가져오기 \(p. 974\)](#) 단원을 참조하십시오.
- 특정한 이전 PostgreSQL 버전에서 PostgreSQL 10.7로 여러 주요 버전 업그레이드가 가능합니다. 자세한 내용은 [PostgreSQL 메이저 버전 업그레이드 \(p. 964\)](#) 단원을 참조하십시오.

PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 10.6

PostgreSQL 버전 10.6에는 릴리스 10.5의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. PostgreSQL 10.6의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

이 버전에는 또한 다음과 같은 변경 사항이 포함됩니다.

- `rds.restrict_password_commands` 파라미터와 `rds_password` 역할을 새롭게 도입했습니다. `rds.restrict_password_commands` 파라미터가 활성화된 경우, `rds_password` 역할이 있는 사용자만 사용자 암호 및 암호 만료 변경을 수행할 수 있습니다. 암호 관련 작업을 제한된 역할로 제한하면 클라이언트 측의 암호 복잡성 요구 사항 등의 정책을 구현할 수 있습니다. `rds.restrict_password_commands` 파라미터는 정적이므로 변경하려면 데이터베이스를 다시 시작해야 합니다. 자세한 내용은 [암호 관리 제한 \(p. 1010\)](#) 단원을 참조하십시오.
- 논리적 디코딩 플러그인 `wal2json`이 9e962ba를 커밋하도록 업데이트되었습니다.

PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Note

PostgreSQL용 Amazon RDS가 다음 메이저 릴리스에서 `tsearch2` 확장을 제거한다고 발표했습니다. 8.3 이전 텍스트 검색을 사용하는 고객이 동등한 내장 기능으로 마이그레이션할 것을 권장합니다. 마이그레이션에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

Amazon RDS의 PostgreSQL 버전 10.5

PostgreSQL 버전 10.5에는 릴리스 10.4의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 10.5의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

이 버전에는 또한 다음과 같은 변경 사항이 포함됩니다.

- `pglogical` 확장 버전 2.2.0을 지원합니다. 이 확장을 사용하기 위한 사전 조건은 [Amazon RDS의 PostgreSQL에 대한 논리적 복제 \(p. 1054\)](#)에서 설명하는 PostgreSQL에 대한 논리적 복제를 사용하기 위한 사전 조건과 동일합니다.
- `pg_similarity` 확장 버전 1.0을 지원합니다.
- `pageinspect` 확장 버전 1.6을 지원합니다.
- `libprotobuf` PostGIS 구성 요소에 대한 확장 버전 1.3.0을 지원합니다.
- `pg_hint_plan` 버전 1.3.1의 확장에 대한 업데이트.
- `wal2json` 확장을 01c5c1e 버전으로 업데이트.

PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 10.4

PostgreSQL 버전 10.4에는 릴리스 10.3의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 10.4의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

이 버전에는 또한 다음과 같은 변경 사항이 포함됩니다.

- 기본 게시 및 구독 프레임워크를 사용한 PostgreSQL 10 논리적 복제에 대한 지원. RDS PostgreSQL 데이터베이스는 게시자 및 구독자로 작동할 수 있습니다. 데이터베이스 수준 또는 테이블 수준에서 다른 PostgreSQL 데이터베이스로의 복제를 지정할 수 있습니다. 논리적 복제를 통해 게시자와 구독자 데이터베이스는 서로 물리적으로 동일(블록 대 블록)할 필요가 없습니다. 이를 통해 10.4 이상 버전의 서로 다른

데이터베이스 간 데이터 통합, 데이터 백업 및 데이터 복제와 같은 사용 사례가 가능합니다. 자세한 내용은 [Amazon RDS의 PostgreSQL에 대한 논리적 복제 \(p. 1054\)](#) 단원을 참조하십시오.

- 임시 파일 크기 제한은 사용자 구성이 가능합니다. `temp_file_limit` 파라미터를 수정하려면 `rds_superuser` 역할이 필요합니다.
- PostGIS 확장에서 사용하는 GDAL 라이브러리 업데이트. [PostGIS 작업 \(p. 1005\)](#) 단원을 참조하십시오.
- `ip4r` 확장의 2.1.1 버전 업데이트.
- `pg_repack` 확장의 1.4.3 버전 업데이트. [pg_repack 확장 작업 \(p. 1004\)](#) 단원을 참조하십시오.
- `p1v8` 확장의 2.1.2 버전 업데이트.

PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Note

`tsearch2` 확장은 다음 주요 릴리스에서 제거될 예정입니다. 8.3 이전 텍스트 검색을 사용하는 고객이 동등한 내장 기능으로 마이그레이션할 것을 권장합니다. 마이그레이션에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

Amazon RDS의 PostgreSQL 버전 10.3

PostgreSQL 버전 10.3에는 릴리스 10의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 10.3의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

PL/v8 버전 2.1.0이 출시되었습니다. PL/v8을 사용하고 PostgreSQL을 새 PL/v8 버전으로 업그레이드하는 경우 즉시 새로운 확장 기능을 사용하지만 카탈로그 메타데이터는 이 사실을 반영하지 않습니다. 카탈로그 메타데이터를 새 PL/v8 버전과 동기화하는 단계는 [PL/v8 업그레이드 \(p. 1052\)](#) 단원을 참조하십시오.

PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 10.1

PostgreSQL 버전 10.1에는 릴리스 10의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 10.1의 수정 사항에 대한 자세한 내용은 [PostgreSQL 문서](#) 및 [PostgreSQL 10 community announcement](#)를 참조하십시오.

PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL 버전 10.1에는 다음과 같은 변경 사항이 포함됩니다.

- 선언적 테이블 파티셔닝 – PostgreSQL 10은 테이블 파티셔닝을 SQL 구문 및 기본 투플 라우팅을 추가합니다.
- 병렬 쿼리 – 새 PostgreSQL 10.1 인스턴스를 생성하는 경우 병렬 쿼리가 `default_postgres10` 파라미터 그룹에 대해 활성화됩니다. `max_parallel_workers_per_gather` 파라미터는 기본적으로 2로 설정되지만, 특정 워크로드 요구 사항을 지원하도록 수정할 수 있습니다.
- International Components for Unicode(ICU) 지원 – ICU 라이브러리를 사용하여 명시적으로 버전 지정된 콜레이션을 제공할 수 있습니다. 10.1용 Amazon RDS는 ICU 버전 60.2로 컴파일되었습니다. PostgreSQL의 ICU 구현에 대한 자세한 내용은 [Collation Support](#)를 참조하십시오.
- 방대한 페이지 – 방대한 페이지는 현대식 하드웨어 아키텍처의 여러 페이지 크기 기능을 사용하는 Linux 커널의 한 가지 기능입니다. PostgreSQL용 Amazon RDS는 전역 구성 파라미터를 사용하여 방대한 페이

지를 지원합니다. RDS를 사용하여 PostgreSQL 10.1 인스턴스를 생성할 경우 `huge_pages` 파라미터는 `default.postgres10` 파라미터 그룹에 대해 "on"으로 설정됩니다. 특정 워크로드 요구 사항을 지원하도록 이 설정을 수정할 수 있습니다.

- PL/v8 업데이트 – PL/v8은 SQL에서 호출할 수 있는 JavaScript로 함수를 작성할 수 있도록 하는 절차 언어입니다. 이 PostgreSQL 릴리스는 PL/v8의 버전 2.1.0을 지원합니다.
- xlog와 location의 이름 변경 – PostgreSQL 버전 10에서는 "xlog"라는 약자가 "wal"로 바뀌고 "location"이라는 용어는 "lsn"으로 바뀌었습니다. 자세한 내용은 <https://www.postgresql.org/docs/10/static/release-10.html#id-1.11.6.8.4> 단원을 참조하십시오.
- tsearch2 모듈 – Amazon RDS는 PostgreSQL 버전 10에서 tsearch2 모듈을 계속 제공하지만, 다음 메이저 버전 릴리스에서는 제거할 예정입니다. 애플리케이션에서 tsearch2 함수를 사용하는 경우 코어 엔진에서 제공하는 상응하는 함수를 사용하도록 애플리케이션을 업데이트합니다. tsearch2 사용에 대한 자세한 내용은 [tsearch2 모듈](#)을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.6.12

PostgreSQL 버전 9.6.12에는 릴리스 9.6.11의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.6.12의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.6.11

PostgreSQL 버전 9.6.11에는 릴리스 9.6.10의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. PostgreSQL 9.6.11의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

이 버전에서는 논리적 디코딩 플러그인 `wal2json`이 `9e962ba`를 커밋하도록 업데이트되었습니다.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.6.10

PostgreSQL 버전 9.6.10에는 릴리스 9.6.9의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.6.10의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

이 버전에는 다음과 같은 변경 사항이 포함됩니다.

- `pglogical` 확장 버전 2.2.0을 지원합니다. 이 확장을 사용하기 위한 사전 조건은 [Amazon RDS의 PostgreSQL에 대한 논리적 복제 \(p. 1054\)](#)에서 설명하는 PostgreSQL에 대한 논리적 복제를 사용하기 위한 사전 조건과 동일합니다.
- `pg_similarity` 확장 버전 2.2.0을 지원합니다.
- `wal2json` 확장을 01c5c1e 버전으로 업데이트.
- `pg_hint_plan` 버전 1.2.3의 확장에 대한 업데이트.

PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.6.9

PostgreSQL 버전 9.6.9에는 릴리스 9.6.8의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.6.9의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

이 버전에는 다음과 같은 변경 사항이 포함됩니다.

- 임시 파일 크기 제한은 사용자 구성이 가능합니다. `temp_file_limit` 파라미터를 수정하려면 `rds_superuser` 역할이 필요합니다.
- PostGIS 확장에서 사용하는 GDAL 라이브러리 업데이트. [PostGIS 작업 \(p. 1005\)](#) 단원을 참조하십시오.
- `ip4r` 확장의 2.1.1 버전 업데이트.
- `pgaudit` 확장의 1.1.1 버전 업데이트. [pgaudit 확장 작업 \(p. 1003\)](#) 단원을 참조하십시오.
- `pg_repack` 확장의 1.4.3 버전 업데이트. [pg_repack 확장 작업 \(p. 1004\)](#) 단원을 참조하십시오.
- `plv8` 확장의 2.1.2 버전 업데이트.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.6.8

PostgreSQL 버전 9.6.8에는 릴리스 9.6.6의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.6.8의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.6.6

PostgreSQL 버전 9.6.6에는 릴리스 9.6.5의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.6.6의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

이 버전에는 다음과 같은 기능이 포함되어 있습니다.

- `orafce` 확장 버전 3.6.1을 지원합니다. 이 확장에는 상용 데이터베이스에 고유한 함수가 포함되어 있으며, 상용 데이터베이스를 PostgreSQL로 포팅하려는 경우에 유용합니다. Amazon RDS에서 `orafce`를 사용하는 방법에 대한 자세한 내용은 [orafce 확장 작업 \(p. 1008\)](#) 단원을 참조하십시오.
- `prefix` 확장 버전 1.2.6을 지원합니다. 이 확장은 작업자에게 텍스트 접두사 검색을 제공합니다. `prefix`에 대한 자세한 내용은 [GitHub에서 접두사 프로젝트](#)를 참조하십시오.
- PostGIS 버전 2.3.4, pgrouting 버전 2.4.2, 업데이트된 `wal2json` 버전을 지원합니다.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.6.5

PostgreSQL 버전 9.6.5에는 릴리스 9.6.4의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.6.5의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버

전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

이번 버전에는 `pgrouting` 및 `postgresql-hll` 확장과 `decoder_raw` 옵션 모듈에 대한 지원도 추가되었습니다.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.6.3

PostgreSQL 버전 9.6.3에는 여러 새 기능과 버그 수정이 포함되어 있습니다. 이 버전에는 다음과 같은 기능이 포함되어 있습니다.

- 확장 `pg_repack` 버전 1.4.0을 지원합니다. 이 확장을 사용하여 테이블과 인덱스에서 부풀림을 제거할 수 있습니다. Amazon RDS에서 `pg_repack`을 사용하는 방법에 대한 자세한 내용은 [pg_repack 확장 작업 \(p. 1004\)](#) 단원을 참조하십시오.
- 확장 `pgaudit` 버전 1.1.0을 지원합니다. 이 확장은 세부적인 세션 및 객체 감사 로깅을 제공합니다. `pgaudit` 및 Amazon RDS를 함께 사용하는 방법에 대한 자세한 내용은 [pgaudit 확장 작업 \(p. 1003\)](#) 단원을 참조하십시오.
- 논리적 디코딩을 위한 출력 플러그인 `wal2json`을 지원합니다.
- `auto_explain` 모듈을 지원합니다. 이 모듈을 사용하여 느린 문의 로그 실행 계획을 자동으로 기록할 수 있습니다. 다음은 Amazon RDS PostgreSQL 세션 내에서 `auto_explain`을 사용하는 방법을 나타낸 예제입니다.

```
LOAD '$libdir/plugins/auto_explain';
```

`auto_explain` 사용에 대한 자세한 내용은 [PostgreSQL 문서](#)를 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.6.2

PostgreSQL 버전 9.6.2에는 여러 새 기능과 버그 수정이 포함되어 있습니다. 새 버전에는 다음의 확장 버전도 포함되어 있습니다.

- PostGIS 버전 2.3.2
- `pg_freespacemap` 버전 1.1 – FSM(Free Space Map)을 검사하는 기능을 제공합니다. 이 확장 기능은 오버로딩 함수 `pg_freespace`를 제공합니다. 이 함수는 지정된 페이지 또는 릴레이션 내 모든 페이지에 대한 FSM에 기록된 값을 보여 줍니다.
- `pg_hint_plan` 버전 1.1.3 – SQL 문에서 힌트 구를 사용하여 실행 계획을 제어할 수 있습니다.
- `log_fdw` 버전 1.0 – Amazon RDS의 확장 기능을 사용해 데이터베이스 내부에서 데이터베이스 엔진 로그를 로드 및 쿼리할 수 있습니다. 자세한 내용은 [log_fdw 확장 사용 \(p. 1050\)](#) 단원을 참조하십시오.
- 이 버전 릴리스에서는 이제 DB 파라미터 그룹의 `max_worker_processes` 파라미터를 편집할 수 있습니다.

Amazon RDS PostgreSQL 버전 9.6.2는 열거형 값 변경도 지원합니다. 자세한 내용은 [PostgreSQL용 ALTER ENUM \(p. 1058\)](#) 단원을 참조하십시오.

9.6.2의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.6.1

PostgreSQL 버전 9.4.5에는 여러 새 기능과 개선 사항이 포함되어 있습니다. PostgreSQL 9.6.1 수정 및 개선 사항에 대한 자세한 내용은 [PostgreSQL 문서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을

업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오. PostgreSQL 9.6.1에서 Amazon RDS를 이용한 병렬 쿼리 실행과 구문 검색에 대한 자세한 내용은 [AWS Database Blog](#)를 참조하십시오.

PostgreSQL 버전 9.6.1에는 다음과 같은 변경 사항이 포함됩니다.

- **병렬 쿼리 실행:** 다량의 읽기 전용 쿼리의 병렬 실행을 지원하여 순차적 스캔, 해시 조인, 중첩 루프를 허용하며 병렬로 실행되도록 집계합니다. 기본적으로 병렬 쿼리 실행은 활성화되어 있지 않습니다. 병렬 쿼리 실행을 허용하려면 `max_parallel_workers_per_gather` 파라미터를 0보다 큰 수로 정합니다.
- **postgres_fdw extension 업데이트 완료:** 원격 조인, 정렬, 업데이트, 삭제 작업을 지원합니다.
- **PL/v8 업데이트:** PL/v8 언어의 1.5.3 버전을 제공합니다.
- **PostGIS 버전 업데이트:** POSTGIS="2.3.0 r15146" GEOS="3.5.0-CAPI-1.9.0 r4084" PROJ="Rel 4.9.2, 2015년 9월 8일" GDAL="GDAL 2.1.1, 2016년 7월 7일 출시" LIBXML="2.9.1" LIBJSON="0.12" 래스터
- **Vacuum 개선:** vacuum freeze 작업 중 불필요한 페이지 스캔을 피합니다.
- **전체 텍스트 검색은 문구 지원:** 새 연산자 <-> 및 <N>을 이용해 tsquery 입력에서 문구 검색 쿼리 지정 기능을 지원합니다.
- **두 개의 새로운 확장 지원:**
 - `bloom(Bloom 필터 기반의 인덱스 액세스 방법)`
 - `pg_visibility`(테이블의 가시성 맵과 페이지 차원의 가시성 정보를 조사하는 수단 제공)
- **버전 9.6.2 릴리스에서는 이제 PostgreSQL 버전 9.6.1 DB 파라미터 그룹의 `max_worker_processes` 파라미터를 편집할 수 있습니다.**

AWS Management 콘솔, AWS CLI 또는 RDS API를 사용하여 PostgreSQL 9.6.1 데이터베이스 인스턴스를 새로 만들 수 있습니다. 또한 메이저 버전의 업그레이드를 이용해 기존 PostgreSQL 9.5 인스턴스를 버전 9.6.1로 업그레이드할 수도 있습니다. DB 인스턴스를 버전 9.3 또는 9.4에서 9.6으로 업그레이드하려면 먼저 다음 메이저 버전으로 포인트 앤 클릭 방식의 업그레이드를 실행해야 합니다. 각 업그레이드 작업 시에는 잠시 DB 인스턴스를 사용할 수 없습니다.

Amazon RDS의 PostgreSQL 버전 9.5.16

PostgreSQL 버전 9.5.16에는 릴리스 9.5.15의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.5.16의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.5.15

PostgreSQL 버전 9.5.15에는 릴리스 9.5.14의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.5.15의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.5.14

PostgreSQL 버전 9.5.14에는 릴리스 9.5.13의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.5.14의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.5.13

PostgreSQL 버전 9.5.13에는 릴리스 9.5.12의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.5.13의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

이 버전에는 다음과 확장 업데이트가 포함됩니다.

- `pgaudit` 확장의 1.0.6 버전 업데이트. [pgaudit 확장 작업 \(p. 1003\)](#) 단원을 참조하십시오.
- `pg_hint_plan` 확장의 1.1.5 버전 업데이트.
- `plv8` 확장의 2.1.2 버전 업데이트.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.5.12

PostgreSQL 버전 9.5.12에는 버전 9.5.10에서 발견된 문제에 대한 몇 가지 버그 수정 사항이 포함되어 있습니다. 버전 9.5.12의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.5.10

PostgreSQL 버전 9.5.10에는 버전 9.5.9의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.5.10의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.5.9

PostgreSQL 버전 9.5.9에는 버전 9.5.8의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.5.9의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.5.7

PostgreSQL 버전 9.5.7에는 여러 새 기능과 버그 수정이 포함되어 있습니다. 이 버전에는 다음과 같은 기능이 포함되어 있습니다.

- 확장 pgaudit 버전 1.0.5를 지원합니다. 이 확장은 세부적인 세션 및 객체 감사 로깅을 제공합니다. Amazon RDS에서 pgaudit을 사용하는 방법에 대한 자세한 내용은 [pgaudit 확장 작업 \(p. 1003\)](#) 단원을 참조하십시오.
- 논리적 디코딩을 위한 출력 플러그인 wal2json을 지원합니다.
- auto_explain 모듈을 지원합니다. 이 모듈을 사용하여 느린 문의 로그 실행 계획을 자동으로 기록할 수 있습니다. 다음은 Amazon RDS PostgreSQL 세션 내에서 auto_explain을 사용하는 방법을 나타낸 예 제입니다.

```
LOAD '$libdir/plugins/auto_explain';
```

auto_explain 사용에 대한 자세한 내용은 [PostgreSQL 문서](#)를 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.5.6

PostgreSQL 버전 9.5.6에는 여러 새 기능과 버그 수정이 포함되어 있습니다. 새 버전에는 다음의 확장 버전도 포함되어 있습니다.

- PostGIS 버전 2.2.5
- [pg_freespacemap](#) 버전 1.1 – FSM(Free Space Map)을 검사하는 기능을 제공합니다. 이 확장 기능은 오버로딩 함수 pg_freespace를 제공합니다. 이 함수는 지정된 페이지 또는 관련된 모든 페이지에 대한 FSM에 기록된 값을 보여 줍니다.
- [pg_hint_plan](#) 버전 1.1.3 – SQL 문에서 힌트 구를 사용하여 실행 계획을 제어할 수 있습니다.

Amazon RDS PostgreSQL 버전 9.5.6은 열거형 값 변경도 지원합니다. 자세한 내용은 [PostgreSQL용 ALTER ENUM \(p. 1058\)](#) 단원을 참조하십시오.

9.5.6의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.5.4

PostgreSQL 버전 9.5.4은 이전 버전에서 발견된 몇 가지 문제를 수정했습니다. 9.5.4의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL 9.4 이상 버전에서는 논리적 복제 디코딩을 통한 WAL 변경 스트리밍을 지원합니다. Amazon RDS에서는 PostgreSQL 9.4.9 이상 버전과 9.5.4 이상 버전에 대한 논리적 복제를 지원합니다. Amazon RDS의 PostgreSQL 논리적 복제에 대한 자세한 내용은 [Amazon RDS의 PostgreSQL에 대한 논리적 복제 \(p. 1054\)](#) 단원을 참조하십시오.

Amazon RDS용 PostgreSQL 버전 9.5.4로 시작하는 ALTER USER WITH BYPASSRLS 명령을 지원합니다.

PostgreSQL 9.4.9 이상 버전과 9.5.4 이상 버전에서는 이벤트 트리거를 지원하고, Amazon RDS에서는 이러한 버전에 대한 이벤트 트리거를 지원합니다. 마스터 사용자 계정을 사용하여 이벤트 트리거를 생성, 수정, 이름 변경 및 삭제할 수 있습니다. 이벤트 트리거는 DB 인스턴스 레벨에서 수행되므로 인스턴스에 대한 모든 데이터베이스에 적용될 수 있습니다. Amazon RDS의 PostgreSQL 이벤트 트리거에 대한 자세한 내용은 [Amazon RDS의 PostgreSQL에 대한 이벤트 트리거 \(p. 1056\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.5.2

PostgreSQL 버전 9.5.2은 이전 버전에서 발견된 몇 가지 문제를 수정했습니다. 9.5.2의 기능에 대한 자세한 내용은 [PostgreSQL 설명서](#) 단원을 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL 버전 9.5.2는 db.m1 또는 db.m2 DB 인스턴스 클래스를 지원하지 않습니다. PostgreSQL 버전 9.4에서 버전 9.5.2를 실행하는 DB 인스턴스를 이 인스턴스 클래스 중 하나로 업그레이드해야 하는 경우 컴퓨팅을 확장해야 합니다. 확장하려면 PostgreSQL 버전 9.4에서 버전 9.5.2를 실행하는 DB 인스턴스를 업그레이드하기 전에 비교 가능한 db.t2 또는 db.m3 DB 인스턴스 클래스가 필요합니다. DB 인스턴스 클래스에 대한 자세한 내용은 [DB 인스턴스 클래스 선택 \(p. 77\)](#) 단원을 참조하십시오.

네이티브 PostgreSQL 버전 9.5.2에서는 ALTER USER WITH BYPASSRLS 명령을 도입했습니다.

이 릴리스에는 다음을 비롯한 이전 버전의 업데이트도 포함됩니다.

- CVE-2016-2193: 동일한 세션에서 두 개 이상의 ROLE에 대해 쿼리 플랜이 재사용될 수 있는 경우의 문제를 해결합니다. 쿼리 플랜을 재사용하면 쿼리에서 잘못된 RLS(Row Level Security) 정책을 사용할 수 있습니다.
- CVE-2016-3065: BRIN 인덱스 페이지와 함께 pageinspect를 사용하면 발생하는 서버 충돌 버그를 해결합니다. 공격자에게 서버 메모리의 일부가 노출될 수 있기 때문에 이 충돌은 보안 문제로 간주됩니다.

RDS PostgreSQL 9.5의 주요 개선 사항은 다음과 같습니다.

- UPSERT: UPDATE로 전환되거나 무시되는 제한 충돌을 생성하는 INSERT 허용
- GROUP BY 분석 기능인 GROUPING SETS, CUBE, ROLLUP 추가
- 행 수준(row-level) 보안 통제 추가
- 복제 진행 상황을 추적하기 위한 수단 생성(논리적 복제 중에 개별 변경 사항의 출처를 식별하기 위한 방법 등)
- BRIN(Block Range Indexes) 추가
- 정렬을 위한 상당한 성능 개선 추가
- 다중 CPU 머신을 위한 상당한 성능 개선 추가
- PostGIS 2.2.2 - 이 PostGIS 최신 버전을 사용하려면 버전 9.5.2로 업그레이드한 후 ALTER EXTENSION UPDATE 문을 사용하여 업데이트합니다. 예:

ALTER EXTENSION POSTGIS UPDATE TO '2.2.2'

- rds_superuser 계정이 pg_stat_activity에서 autovacuum 세션을 볼 수 있도록 허용하여 autovacuum 세션의 가시성 향상. 예: 명령의 실행을 차단하거나, 수동으로 실행한 vacuum 명령보다 느리게 실행되는 autovacuum 세션을 식별 및 종료 가능

RDS PostgreSQL 버전 9.5.2는 다음과 같은 확장 기능을 새로 제공합니다.

- address_standardizer – 입력 주소를 받아 table, helper lex, gaz tables에 저장된 규칙에 따라 정규화하는 단일 행 주소 구문 분석기.
- hstore_plperl – hstore type for PL/Perl을 위한 변환 제공.
- tsm_system_rows – 테이블 샘플링 메서드 SYSTEM_ROWS 제공. SELECT 명령의 TABLESAMPLE 절에 사용 가능.
- tsm_system_time – 테이블 샘플링 메서드 SYSTEM_TIME 제공. SELECT 명령의 TABLESAMPLE 절에 사용 가능.

Amazon RDS의 PostgreSQL 버전 9.4.21

PostgreSQL 버전 9.4.21에는 릴리스 9.4.20의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다.

9.4.21의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.4.20

PostgreSQL 버전 9.4.20에는 릴리스 9.4.19의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.4.20의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.4.19

PostgreSQL 버전 9.4.19에는 릴리스 9.4.18의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.4.19의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.4.18

PostgreSQL 버전 9.4.18에는 릴리스 9.4.17의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.4.18의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

p1v8 확장은 버전 2.1.2로 업데이트되었습니다. PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.4.17

PostgreSQL 버전 9.4.17에는 릴리스 9.4.15의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.4.17의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.4.15

PostgreSQL 버전 9.4.15에는 릴리스 9.4.14의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.4.15의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.4.14

PostgreSQL 버전 9.4.14에는 릴리스 9.4.12의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.4.14의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.4.12

PostgreSQL 버전 9.4.12은 이전 버전에서 발견된 몇 가지 문제를 수정했습니다.

9.4.12의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.4.11

PostgreSQL 버전 9.4.11은 이전 버전에서 발견된 몇 가지 문제를 수정했습니다.

9.4.11의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL 9.4 이상 버전에서는 논리적 복제 디코딩을 통한 WAL 변경 스트리밍을 지원합니다. Amazon RDS에서는 PostgreSQL 9.4.9 이상 버전과 9.5.4 이상 버전에 대한 논리적 복제를 지원합니다. Amazon RDS의 PostgreSQL 논리적 복제에 대한 자세한 내용은 [Amazon RDS의 PostgreSQL에 대한 논리적 복제 \(p. 1054\)](#) 단원을 참조하십시오.

PostgreSQL 9.4.9 이상 버전과 9.5.4 이상 버전에서는 이벤트 트리거를 지원하고, Amazon RDS에서는 이러한 버전에 대한 이벤트 트리거를 지원합니다. 마스터 사용자 계정을 사용하여 이벤트 트리거를 생성, 수정, 이름 변경 및 삭제할 수 있습니다. 이벤트 트리거는 DB 인스턴스 레벨에서 수행되므로 인스턴스에 대한 모든 데이터베이스에 적용될 수 있습니다. Amazon RDS의 PostgreSQL 이벤트 트리거에 대한 자세한 내용은 [Amazon RDS의 PostgreSQL에 대한 이벤트 트리거 \(p. 1056\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.4.9

PostgreSQL 버전 9.4.9은 이전 버전에서 발견된 몇 가지 문제를 수정했습니다. 9.4.9의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL 9.4 이상 버전에서는 논리적 복제 디코딩을 통한 WAL 변경 스트리밍을 지원합니다. Amazon RDS에서는 PostgreSQL 9.4.9 이상 버전과 9.5.4 이상 버전에 대한 논리적 복제를 지원합니다. Amazon RDS의 PostgreSQL 논리적 복제에 대한 자세한 내용은 [Amazon RDS의 PostgreSQL에 대한 논리적 복제 \(p. 1054\)](#) 단원을 참조하십시오.

PostgreSQL 9.4.9 이상 버전과 9.5.4 이상 버전에서는 이벤트 트리거를 지원하고, Amazon RDS에서는 이러한 버전에 대한 이벤트 트리거를 지원합니다. 마스터 사용자 계정을 사용하여 이벤트 트리거를 생성, 수정, 이름 변경 및 삭제할 수 있습니다. 이벤트 트리거는 DB 인스턴스 레벨에서 수행되므로 인스턴스에 대한 모든 데이터베이스에 적용될 수 있습니다. Amazon RDS의 PostgreSQL 이벤트 트리거에 대한 자세한 내용은 [Amazon RDS의 PostgreSQL에 대한 이벤트 트리거 \(p. 1056\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.4.7

PostgreSQL 버전 9.4.7은 이전 버전에서 발견된 몇 가지 문제를 수정했습니다. 9.4.7의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL 버전 9.4.7은 rds_superuser 계정이 pg_stat_activity에서 autovacuum 세션을 볼 수 있도록 허용하여 autovacuum 세션의 가시성을 향상 시켰습니다. 예: 명령의 실행을 차단하거나, 수동으로 실행한 vacuum 명령보다 느리게 실행되는 autovacuum 세션을 식별 및 종료 가능

Amazon RDS의 PostgreSQL 버전 9.3.25

Note

PostgreSQL은 Amazon RDS에서 2018년 9월 PostgreSQL 9.3을 폐기함을 발표하며 PostgreSQL 9.3에 대한 지원을 중지했습니다. 가능한 한 빨리 PostgreSQL 9.4 이상 버전으로 업그레이드하는 것이 좋습니다. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL 버전 9.3.25에는 릴리스 9.3.24의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다.
9.3.25의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.3.24

PostgreSQL 버전 9.3.24에는 릴리스 9.3.23의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다.
9.3.24의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

Note

PostgreSQL용 Amazon RDS에는 PostgreSQL 버전 9.3.x가 사용되지 않습니다.

PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.3.23

PostgreSQL 버전 9.3.23에는 릴리스 9.3.22의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다.
9.3.23의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

p1v8 확장은 버전 2.1.2로 업데이트되었습니다. PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Note

PostgreSQL용 Amazon RDS에서 2018년 9월 9.3.x 버전이 폐기될 예정이라고 발표했습니다. 편한 때에 9.3.x 데이터베이스를 최신 버전으로 업그레이드할 것을 권장합니다.

Amazon RDS의 PostgreSQL 버전 9.3.22

PostgreSQL 버전 9.3.22에는 릴리스 9.3.20의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다.
9.3.22의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL용 Amazon RDS가 지원하는 확장 기능의 전체 목록은 [Supported PostgreSQL 기능 및 확장자 \(p. 1030\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.3.20

PostgreSQL 버전 9.3.20에는 버전 9.3.19의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.3.20의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.3.19

PostgreSQL 버전 9.3.19에는 버전 9.3.18의 문제에 대한 여러 버그 수정 사항이 포함되어 있습니다. 9.3.19의 수정 사항에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.3.17

PostgreSQL 버전 9.3.17는 이전 버전에서 발견된 몇 가지 버그를 수정했습니다. 이 버전에는 버전 9.3.16과 동일한 확장 구성 요소가 포함되어 있습니다. 버전 9.3.17의 수정 사항 목록은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.3.16

PostgreSQL 버전 9.3.16는 이전 버전에서 발견된 몇 가지 버그를 수정했습니다. 이 버전에는 버전 9.3.14와 동일한 확장 구성 요소가 포함되어 있습니다. 버전 9.3.16의 수정 사항 목록은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.3.14

PostgreSQL 버전 9.3.14는 이전 버전에서 발견된 몇 가지 버그를 수정했습니다. 버전 9.3.14의 수정 사항 목록은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

Amazon RDS의 PostgreSQL 버전 9.3.12

PostgreSQL 버전 9.3.12는 이전 버전에서 발견된 몇 가지 버그를 수정했습니다. 버전 9.3.12의 수정 사항 목록은 [PostgreSQL 설명서](#)를 참조하십시오. PostgreSQL DB 인스턴스의 엔진 버전을 업그레이드하는 방법에 대한 자세한 내용은 [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#) 단원을 참조하십시오.

PostgreSQL 버전 9.3.12는 rds_superuser 계정이 pg_stat_activity에서 autovacuum 세션을 볼 수 있도록 허용하여 autovacuum 세션의 가시성을 향상시켰습니다. 예: 명령의 실행을 차단하거나, 수동으로 실행한 vacuum 명령보다 느리게 실행되는 autovacuum 세션을 식별 및 종료 가능

Supported PostgreSQL 기능 및 확장자

Amazon RDS에서는 대부분의 가장 일반적인 PostgreSQL 확장자 및 기능을 지원합니다.

주제

- [Amazon RDS에서 지원되는 PostgreSQL 확장 기능 및 모듈 \(p. 1030\)](#)
- [PL/v8 업그레이드 \(p. 1052\)](#)
- [지원되는 PostgreSQL 기능 \(p. 1054\)](#)
- [PostgreSQL DB 인스턴스에 대한 제한 \(p. 1058\)](#)
- [PostgreSQL DB 인스턴스 업그레이드 \(p. 1059\)](#)
- [PostgreSQL DB 인스턴스와 함께 SSL 사용 \(p. 1059\)](#)

Amazon RDS에서 지원되는 PostgreSQL 확장 기능 및 모듈

PostgreSQL은 다수의 PostgreSQL 확장 기능 및 모듈을 지원합니다. 여기에서 확장 기능 및 모듈이란 PostgreSQL 엔진에서 제공하는 기능이 더욱 확장된 것을 말합니다. 다음 섹션에서는 Amazon RDS가 주요 PostgreSQL 버전에 대해 지원하는 확장 기능 및 모듈에 대해 설명합니다.

주제

- [Amazon RDS에서 지원되는 PostgreSQL 버전 11.x 확장 기능 및 모듈 \(p. 1031\)](#)
- [Amazon RDS에서 지원되는 PostgreSQL 버전 10.x 확장 기능 및 모듈 \(p. 1033\)](#)
- [Amazon RDS에서 지원되는 PostgreSQL 버전 9.6.x 확장 기능 및 모듈 \(p. 1036\)](#)
- [Amazon RDS에서 지원되는 PostgreSQL 버전 9.5.x 확장 기능 \(p. 1040\)](#)

- Amazon RDS에서 지원되는 PostgreSQL 버전 9.4.x 확장 기능 및 모듈 (p. 1043)
- Amazon RDS에서 지원되는 PostgreSQL 버전 9.3.x 확장 기능 (p. 1046)
- Amazon RDS PostGIS에 대한 PostgreSQL 확장 기능 지원 (p. 1049)
- log_fdw 확장 사용 (p. 1050)

해당 PostgreSQL 버전의 기본 DB 파라미터 그룹에서 Amazon RDS가 지원하는 확장 기능 목록을 확인할 수 있습니다. 또한 다음 예제에서처럼 rds.extensions 파라미터를 표시하면 psql을 사용하여 현재 확장 기능 목록을 확인할 수 있습니다.

```
SHOW rds.extensions;
```

Note

psql에서 rds.extensions 파라미터를 사용하면 마이너 버전 릴리스에서 추가된 파라미터가 잘 못 표시될 수 있습니다.

Amazon RDS에서 지원되는 PostgreSQL 버전 11.x 확장 기능 및 모듈

다음 표에는 현재 Amazon RDS PostgreSQL에서 지원되는 PostgreSQL 버전 11.x의 PostgreSQL 확장 기능 및 모듈이 나와 있습니다. "해당 사항 없음"은 확장 기능 및 모듈이 해당 PostgreSQL 버전에서 지원되지 않는다는 것을 의미합니다. PostgreSQL 확장 기능에 대한 자세한 내용은 [Packaging Related Objects into an Extension 단원](#)을 참조하십시오.

확장	버전 11.1	버전 11.2
address_standardizer	2.4.2	2.4.2
address_standardizer_data_u	2.4.2	2.4.2
블룸	1.0	1.0
btree_gin	1.2	1.2
btree_gist	1.5	1.5
citext	1.4	1.4
cube	1.2	1.2
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
hstore	1.4	1.4
hstore_plperl	1.0	1.0
intagg	1.1	1.1
intarray	1.2	1.2
ip4r	2.3	2.3

확장	버전 11.1	버전 11.2
isn	1.1	1.1
log_fdw — log_fdw 확장 사용 (p. 1050) 참조	1.0	1.0
libprotobuf	1.3.0	1.3.0
ltree	1.1	1.1
orafce	3.6.1	3.6.1
pgaudit	1.3.0	1.3.0
pg_buffercache	1.3	1.3
pg_freespacemap	1.2	1.2
pg_hint_plan	1.3.2	1.3.2
pg_prewarm	1.1	1.1
pg_repack	1.4.3	1.4.3
pg_similarity	1.0	1.0
pg_stat_statements	1.5	1.5
pg_trgm	1.3	1.3
pg_visibility	1.2	1.2
pgcrypto	1.3	1.3
pageinspect	1.6	1.6
pglogical	2.2.1	2.2.1
pgrowlocks	1.2	1.2
pgrouting	2.6.1	2.6.1
pgstattuple	1.5	1.5
pgTAP	NA	1.0
plcoffee	2.3.8	2.3.8
plls	2.3.8	2.3.8
plperl	1.0	1.0
plpgsql	1.0	1.0
pltcl	1.0	1.0
plv8	2.3.8	2.3.8
PostGIS	2.5.1	2.5.1
postgis_tiger_geocoder	2.5.1	2.5.1

확장	버전 11.1	버전 11.2
postgis_topology	2.5.0	2.5.0
postgres_fdw	1.0	1.0
postgresql-hll	2.11	2.11
prefix	1.2.8	1.2.8
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

다음 모듈은 아래와 같이 PostgreSQL 버전 11.x 버전에서 지원됩니다.

모듈	버전 11.1	버전 11.2
amcheck	지원	지원
auto_explain	지원	지원
decoder_raw	지원	지원
ICU	버전 60.2 지원	버전 60.2 지원
test_decoding	지원	지원
wal2json	커밋 해시 9e962bad	커밋 해시 9e962bad

Amazon RDS에서 지원되는 PostgreSQL 버전 10.x 확장 기능 및 모듈

다음 표에는 현재 Amazon RDS PostgreSQL에서 지원되는 PostgreSQL 버전 10의 PostgreSQL 확장 기능 및 모듈이 나와 있습니다. "해당 사항 없음"은 확장 기능 및 모듈이 해당 PostgreSQL 버전에서 지원되지 않는다는 것을 의미합니다. PostgreSQL 확장 기능에 대한 자세한 내용은 [Packaging Related Objects into an Extension](#) 단원을 참조하십시오.

확장	10.1	10.3	10.4	10.5	10.6	10.7
address_standby_feedback	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2
address_standby_feedback	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2
블룸	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
btree_gist	1.5	1.5	1.5	1.5	1.5	1.5

확장	10.1	10.3	10.4	10.5	10.6	10.7
chkpass	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
cube	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
dict_int	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
earthdistance	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
hstore	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
intagg	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
ip4r	2.0	2.0	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
log_fdw — log_fdw 확장 사용 (p. 1050) 참조	1.0	1.0	1.0	1.0	1.0	1.0
libprotobuf	해당사항없음	해당사항없음	해당사항없음	1.3.0	1.3.0	1.3.0
ltree	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
pgaudit	1.2.0	1.2.0	1.2.0	1.2.0	1.2.0	1.2.0
pg_buffercache	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
pg_hint_plan	1.3.0	1.3.0	1.3.0	1.3.1	1.3.1	1.3.1
pg_prewarm	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

확장	10.1	10.3	10.4	10.5	10.6	10.7
pg_similarity	해당사항없음	해당사항없음	해당사항없음	1.0	1.0	1.0
pg_stat_statements	1.5	1.5	1.5	1.5	1.5	1.5
pg_trgm	1.3	1.3	1.3	1.3	1.3	1.3
pg_visibility	1.2	1.2	1.2	1.2	1.2	1.2
pgcrypto	1.3	1.3	1.3	1.3	1.3	1.3
pageinspect	해당사항없음	해당사항없음	해당사항없음	1.6	1.6	1.6
pglogical	해당사항없음	해당사항없음	해당사항없음	2.2.0	2.2.0	2.2.0
pgrowlocks	1.2	1.2	1.2	1.2	1.2	1.2
pgrouting	2.5.2	2.5.2	2.5.2	2.5.2	2.5.2	2.5.2
pgstattuple	1.5	1.5	1.5	1.5	1.5	1.5
plcoffee	2.1.0	2.1.0	2.1.2	2.1.2	2.1.2	2.1.2
plls	2.1.0	2.1.0	2.1.2	2.1.2	2.1.2	2.1.2
plperl	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
pltcl	1.0	1.0	1.0	1.0	1.0	1.0
plv8	2.1.0	2.1.0	2.1.2	2.1.2	2.1.2	2.1.2
PostGIS	2.4.2	2.4.2	2.4.4	2.4.4	2.4.4	2.4.4
postgis_tiger_geometry	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2
postgis_topology	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2
postgres_fdw	1.0	1.0	1.0	1.0	1.0	1.0
postgresql-hll	2.10.2	2.10.2	2.10.2	2.10.2	2.10.2	2.10.2
prefix	1.2.0	1.2.0	1.2.0	1.2.0	1.2.0	1.2.0

확장	10.1	10.3	10.4	10.5	10.6	10.7
sslinfo	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
test_parser	1.0	1.0	1.0	1.0	1.0	1.0
tsearch2(버전 10에서는 사용되지 않습니다)	1.0	1.0	1.0	1.0	1.0	1.0
tsm_system_rdw	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
unaccent	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

[tsearch2](#) 확장자는 버전 10에서는 사용되지 않습니다. PostgreSQL 팀은 PostgreSQL의 다음 메이저 릴리스에서 [tsearch2](#)를 제거할 계획입니다.

다음은 아래와 같이 PostgreSQL 10 버전에서 지원되는 모듈입니다.

모듈	버전 10.1	10.3	10.4	10.5	10.6	10.7
amcheck	지원되지 않음	지원	지원	지원	지원	지원
auto_explain	지원	지원	지원	지원	지원	지원
decoder_raw	지원	지원	지원	지원	지원	지원
ICU	버전 60.2 지원					
test_decoder	지원	지원	지원	지원	지원	지원
wal2json	커밋 해시 5352cc4	커밋 해시 5352cc4	커밋 해시 5352cc4	커밋 해시 01c5c1e	커밋 해시 9e962ba	커밋 해시 9e962ba

Amazon RDS에서 지원되는 PostgreSQL 버전 9.6.x 확장 기능 및 모듈

다음 표에는 현재 Amazon RDS PostgreSQL에서 지원되는 PostgreSQL 버전 9.6.x의 PostgreSQL 확장 기능 및 모듈이 나와 있습니다. "해당 사항 없음"은 확장 기능 및 모듈이 해당 PostgreSQL 버전에서 지원되지 않는다는 것을 의미합니다. PostgreSQL 확장 기능에 대한 자세한 내용은 [Packaging Related Objects into an Extension](#) 단원을 참조하십시오.

확장	9.6	9.6	9.6	9.6.5	9.6.6	9.6.8	9.6.9	9.6.1	9.6.1	9.6.12
address_standard	2.3.0	2.3.1	2.3.2	2.3.2	2.3.3	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4

확장	9.6	9.6	9.6	9.6.5	9.6.6	9.6.8	9.6.9	9.6.1	9.6.1	9.6.12
address_st	2.3	2.3	2.3	2.3	2.3	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4
블룸	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
btree_gin	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
chkpass	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
cube	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
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	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
hstore_plp	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
intarray	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
isn	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
log_fdw	해당사항없음	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
log_fdw 확장 사용 (p. 1050) 참조	해당사항없음	해당사항없음	해당사항없음	해당사항없음	3.6.1	3.6.1	3.6.1	3.6.1	3.6.1	3.6.1
ltree	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
orafce	해당사항없음	해당사항없음	해당사항없음	해당사항없음						
pgaudit	해당사항없음	해당사항없음	1.1	1.1	1.1	1.1.1	1.1.1	1.1.1	1.1.1	1.1.1

확장	9.6.	9.6.	9.6.	9.6.5	9.6.6	9.6.8	9.6.9	9.6.1	9.6.1	9.6.12
pg_buffercache	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pg_freespacemap	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pg_hint_plan	해당사항없음	1.1	1.3	1.3	1.3	1.2.2	1.2.2	1.2.3	1.2.3	1.2.3
pg_prewarm	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pg_repack	해당사항없음	1.4	0.1.4.1	1.4.2	1.4.2	1.4.3	1.4.3	1.4.3	1.4.3	1.4.3
pg_similarity	해당사항없음	해당사항없음	해당사항없음	해당사항없음	해당사항없음	해당사항없음	해당사항없음	1.0	1.0	1.0
pg_stat_statements	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
pg_trgm	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
pg_visibility	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pgcrypto	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
pglogical	해당사항없음	해당사항없음	해당사항없음	해당사항없음	해당사항없음	해당사항없음	해당사항없음	2.2.0	2.2.0	2.2.0
pgrowlocks	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pgrouting	해당사항없음	해당사항없음	해당사항없음	2.3.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2
pgstattuple	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4

확장	9.6	9.6	9.6	9.6.5	9.6.6	9.6.8	9.6.9	9.6.1	9.6.1	9.6.12
plcoffee	1.5.31.5.31.5.31.5.3	1.5.3	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2	
plls	1.5.31.5.31.5.31.5.3	1.5.3	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2	
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
pltcl	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
plv8	1.5.31.5.31.5.31.5.3	1.5.3	2.1.0	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2	
PostGIS	2.3.02.3.22.3.22.3.2	2.3.4	2.3.4	2.3.7	2.3.7	2.3.7	2.3.7	2.3.7	2.3.7	
postgis_tiger	2.3.02.3.22.3.22.3.2	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	
postgis_topo	2.3.02.3.22.3.22.3.2	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4	
postgres_fdw	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
postgresql-hll	해당사항 없음	해당사항 없음	해당사항 없음	해당사항 없음	해당사항 없음	해당사항 없음	해당사항 없음	해당사항 없음	해당사항 없음	2.10.22.10.22.10.22.10.22.10.22.10.2
prefix	해당사항 없음	해당사항 없음	해당사항 없음	해당사항 없음	해당사항 없음	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
tablefunc	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
test_parser	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tsearch2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tsm_system	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tsm_system	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

다음은 아래와 같이 PostgreSQL 9.6 버전에서 지원되는 모듈입니다.

모듈	9.6.1	9.6.2	9.6.3	9.6.5	9.6.8	9.6.9	9.6.10	9.6.11	9.6.12
auto_explain	해당사항 없음	해당사항 없음	지원	지원	지원	지원	지원	지원	지원

모듈	9.6.1	9.6.2	9.6.3	9.6.5	9.6.8	9.6.9	9.6.10	9.6.11	9.6.12
decoder_raw	해당 사항 없음	해당 사항 없음	해당 사항 없음	지원	지원	지원	지원	지원	지원
test_decoding	지원	지원	지원	지원	지원	지원	지원	지원	지원
wal2json	해당 사항 없음	해당 사항 없음	커밋 해시 2828409	커밋 해시 645ab69	커밋 해시 5352cc4	커밋 해시 5352cc4	커밋 해시 01c5c1e	커밋 해시 9e962ba	커밋 해시 9e962ba

Amazon RDS에서 지원되는 PostgreSQL 버전 9.5.x 확장 기능

다음 표에는 현재 Amazon RDS PostgreSQL에서 지원되는 PostgreSQL 버전 9.5.x의 PostgreSQL 확장 기능 및 모듈이 나와 있습니다. "해당 사항 없음"은 확장 기능 및 모듈이 해당 PostgreSQL 버전에서 지원되지 않는다는 것을 의미합니다. PostgreSQL 확장 기능에 대한 자세한 내용은 [Packaging Related Objects into an Extension](#) 단원을 참조하십시오.

확장	9.5	9.5	9.5	9.5	9.5	9.5.1	9.5.1	9.5.1	9.5.1	9.5.16
address_i	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2.5
address_t	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2.5
블룸	해당 사항 없음									
btree_gin	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
chkpass	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
cube	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
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	1.0	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
hstore	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
hstore_plp	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
intarray	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

확장	9.5	9.5	9.5	9.5	9.5	9.5.1	9.5.1	9.5.1	9.5.1	9.5.1	9.5.16
ip4r	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
log_fdw — log_fdw 확장 사용 (p. 105) 참조	해당사항없음										
ltree	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pgaudit	해당사항없음	해당사항없음	해당사항없음		1.0.5	1.0.5	1.0.5	1.0.5	1.0.6	1.0.6	1.0.6
pg_buffercache	1.0	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pg_freespace	해당사항없음	해당사항없음	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pg_hint_plan	해당사항없음	해당사항없음		1.1.3	1.1.3	1.1.3	1.1.3	1.1.3	1.1.5	1.1.5	1.1.5
pg_prewal	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	1.0	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
pg_visibility	해당사항없음										
pgcrypto	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pgrowlock	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pgstattuple	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
plcoffee	1.4.41	1.4.41	1.4.41	1.4.41	1.4.41	1.4.42	1.4.42	1.4.42	1.4.42	1.4.42	2.1.0
plls	1.4.41	1.4.41	1.4.41	1.4.41	1.4.41	1.4.41	1.4.41	1.4.41	1.4.41	1.4.41	2.1.0

확장	9.5	9.5	9.5	9.5	9.5	9.5.1	9.5.1	9.5.1	9.5.1	9.5.1	9.5.16
plperl	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
pltcl	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
plv8	1.4.41	1.4.41	1.4.41	1.4.41	1.4.41	1.4.41	1.4.42	1.0.2	1.2.2	1.2.2	1.2
PostGIS	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2.5
postgis_tiger	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2.5
postgis_topology	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2.5
postgres_10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
sslinfo	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tablefunc	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
test_parser	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tsearch2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tsm_systems淘汰	해당사항없음										
tsm_system淘汰	해당사항없음										
unaccent	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
uuid-ossp	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

다음은 아래와 같이 PostgreSQL 9.5 버전에서 지원되는 모듈입니다.

모듈	9.5.2	9.5.4	9.5.6	9.5.7	9.5.9	9.5.12	9.5.13	9.5.14	9.5.15	9.5.16
auto_explain	해당사항없음	해당사항없음	해당사항없음	해당사항없음	지원	지원	지원	지원	지원	지원
test_decoding	해당사항없음	해당사항없음	해당사항없음	지원	지원	지원	지원	지원	지원	지원

Amazon RDS에서 지원되는 PostgreSQL 버전 9.4.x 확장 기능 및 모듈

다음 표에는 현재 Amazon RDS PostgreSQL에서 지원되는 PostgreSQL 버전 9.4.x의 PostgreSQL 확장 기능 및 모듈이 나와 있습니다. "해당 사항 없음"은 확장 기능 및 모듈이 해당 PostgreSQL 버전에서 지원되지 않는다는 것을 의미합니다. PostgreSQL 확장 기능에 대한 자세한 내용은 [Packaging Related Objects into an Extension](#) 단원을 참조하십시오.

확장	9.4	9.4	9.4	9.4	9.4	9.4	9.4	9.4	9.4	9.4.2	9.4.21
hstore_plperl	해당 사항 없음										
intagg	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
ip4r	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
log_fdw — log_fdw 확장 사용 (p. 1050) 참조	해당 사항 없음										
ltree	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pg_buffercache	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pg_freespacemap	해당 사항 없음										
pg_hint_plan	해당 사항 없음										
pg_prewarm	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	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pg_trgm	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pg_visibility	해당 사항 없음										
pgcrypto	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pgrowlocks	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pgstattuple	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2

다음은 아래와 같이 PostgreSQL 9.4 버전에서 지원되는 모듈입니다.

모듈	9.4.7	9.4.9	9.4.11	9.4.12	9.4.14	9.4.17
test_decoding	해당 사항 없음	해당 사항 없음	해당 사항 없음	지원	지원	지원

Amazon RDS에서 지원되는 PostgreSQL 버전 9.3.x 확장 기능

Note

PostgreSQL용 Amazon RDS에는 PostgreSQL 버전 9.3.x가 사용되지 않습니다. 메이저 버전, 가급적 버전 9.6.x 또는 10.x로 업그레이드하는 것이 좋습니다. [Amazon RDS용 PostgreSQL DB 엔진 업그레이드 \(p. 963\)](#) 단원을 참조하십시오.

다음 표에는 현재 Amazon RDS PostgreSQL에서 지원되는 PostgreSQL 버전 9.3.x의 PostgreSQL 확장 기능이 나와 있습니다. "해당 사항 없음"은 PostgreSQL 버전이 확장 가능하지 않다는 의미입니다. PostgreSQL 확장 기능에 대한 자세한 내용은 [Packaging Related Objects into an Extension](#) 단원을 참조하십시오.

확장	9.3.1	9.3.1	9.3.1	9.3.1	9.3.1	9.3.1	9.3.2	9.3.2	9.3.2	9.3.2	9.3.25
address_standard	해당 사항 없음										
address_standard	해당 사항 없음										
블룸	해당 사항 없음										
btree_gin	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
chkpass	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
citext	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
cube	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
dict_int	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
earthdistance	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
hstore	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
hstore_plperl	해당 사항										

확장	9.3.1	9.3.1	9.3.1	9.3.1	9.3.1	9.3.1	9.3.2	9.3.2	9.3.2	9.3.2	9.3.25
	없음										
intagg	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
ip4r	해당 사항 없음										
isn	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
log_fdw — log_fdw 확장 사용 (p. 1050) 참조	해당 사항 없음										
ltree	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pg_buffercache	해당 사항 없음										
pg_freespacemap	해당 사항 없음										
pg_hint_plan	해당 사항 없음										
pg_prewarm	해당 사항 없음										
pg_stat_statements	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pg_trgm	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1

확장	9.3.1	9.3.1	9.3.1	9.3.1	9.3.1	9.3.2	9.3.2	9.3.2	9.3.2	9.3.25
pg_visibility	해당 사항 없음									
pgcrypto	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
pgrowlocks	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
pgstattuple	해당 사항 없음									
plcoffee	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
plls	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
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
pltcl	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
plv8	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	2.1.0	2.1.2	2.1.2
PostGIS	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8
postgis_tiger_gdal	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8
postgis_topology	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8	2.1.8
postgres_fdw	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
sslinfo	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tablefunc	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
test_parser	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tsearch2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
tsm_system_rows	해당 사항 없음									
tsm_system_time	해당 사항 없음									

확장	9.3.1	9.3.1	9.3.1	9.3.1	9.3.1	9.3.2	9.3.2	9.3.2	9.3.2	9.3.25
unaccent	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
uuid-ossp	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Amazon RDS PostGIS에 대한 PostgreSQL 확장 기능 지원

PostGIS 확장 기능을 사용하려면 다음 명령을 실행하여 확장 기능을 먼저 생성해야 합니다.

```
CREATE EXTENSION POSTGIS;
```

다음 표는 PostgreSQL용 Amazon RDS 버전과 함께 제공되는 PostGIS 구성 요소 버전을 나타낸 것입니다.

PostgreSQL	PostGIS	GEOS	GDAL	PROJ
9.3.12	2.1.8 r13780	3.5.0-CAPI-1.9.0 r4084	1.11.4, 2016년 1월 25일 릴리스됨	릴리스 4.9.2, 2015년 9월 8일
9.3.14	2.1.8 r13780	3.5.0-CAPI-1.9.0 r4084	1.11.5, 2016년 7월 1 일 릴리스됨	릴리스 4.9.2, 2015년 9월 8일
9.3.16	2.1.8 r13780	3.5.0-CAPI-1.9.0 r4084	1.11.5, 2016년 7월 1 일 릴리스됨	릴리스 4.9.2, 2015년 9월 8일
9.3.17	2.1.8 r13780	3.5.0-CAPI-1.9.0 r4084	1.11.5, 2016년 7월 1 일 릴리스됨	릴리스 4.9.2, 2015년 9월 8일
9.4.7	2.1.8 r13780	3.5.0-CAPI-1.9.0 r4084	1.11.4, 2016년 1월 25일 릴리스됨	릴리스 4.9.2, 2015년 9월 8일
9.4.9	2.1.8 r13780	3.5.0-CAPI-1.9.0 r4084	1.11.5, 2016년 7월 1 일 릴리스됨	릴리스 4.9.2, 2015년 9월 8일
9.4.11	2.1.8 r13780	3.5.0-CAPI-1.9.0 r4084	1.11.5, 2016년 7월 1 일 릴리스됨	릴리스 4.9.2, 2015년 9월 8일
9.4.12	2.1.8 r13780	3.5.0-CAPI-1.9.0 r4084	1.11.5, 2016년 7월 1 일 릴리스됨	릴리스 4.9.2, 2015년 9월 8일
9.5.2	2.2.2 r14797	3.5.0-CAPI-1.9.0 r4084	2.0.2, 2016년 1월 26 일 릴리스됨	릴리스 4.9.2, 2015년 9월 8일
9.5.4	2.2.2 r14797	3.5.0-CAPI-1.9.0 r4084	2.0.3, 2016년 7월 1 일 릴리스됨	릴리스 4.9.2, 2015년 9월 8일
9.5.6	2.2.5 r15298	3.5.1-CAPI-1.9.1 r4246	2.0.3, 2016년 7월 1 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
9.5.7	2.2.5 r15298	3.5.1-CAPI-1.9.1 r4246	2.0.3, 2016년 7월 1 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
9.6.1	2.3.0 r15146	3.5.0-CAPI-1.9.0 r4084	2.1.1, 2016년 7월 7 일 릴리스됨	릴리스 4.9.2, 2016년 9월 8일

PostgreSQL	PostGIS	GEOS	GDAL	PROJ
9.6.2	2.3.2 r15302	3.5.1-CAPI-1.9.1 r4246	2.1.3, 2017년 1월 20 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
9.6.3	2.3.2 r15302	3.5.1-CAPI-1.9.1 r4246	2.1.3, 2017년 1월 20 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
9.6.6	2.3.4 r16009	3.6.2-CAPI-1.10.2 4d2925d6	2.1.3, 2017년 1월 20 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
9.6.8	2.3.4 r16009	3.6.2-CAPI-1.10.2 4d2925d6	2.1.3, 2017년 1월 20 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
9.6.9	2.3.7 r16523	3.6.2-CAPI-1.10.2 4d2925d6	2.1.4, 2017년 6월 23 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
9.6.10	2.3.7 r16523	3.6.2-CAPI-1.10.2 4d2925d6	2.1.4, 2017년 6월 23 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
9.6.11	2.3.7 r16523	3.6.2-CAPI-1.10.2 4d2925d6	2.1.4, 2017년 6월 23 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
10.1	2.4.2	3.6.2-CAPI-1.10.2 4d2925d6	2.1.3, 2017년 1월 20 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
10.3	2.4.2	3.6.2-CAPI-1.10.2 4d2925d6	2.1.3, 2017년 1월 20 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
10.4	2.4.4 r16526	3.6.2-CAPI-1.10.2 4d2925d6	2.1.4, 2017년 6월 23 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
10.5	2.4.4 r16526	3.6.2-CAPI-1.10.2 4d2925d6	2.1.4, 2017년 6월 23 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
10.6	2.4.4 r16526	3.6.2-CAPI-1.10.2 4d2925d6	2.1.4, 2017년 6월 23 일 릴리스됨	릴리스 4.9.3, 2016년 9월 15일
11.1	2.5.1 r17027	3.7.0-CAPI-1.11.0 673b9939	2.3.1, 2018년 6월 22 일 릴리스됨	릴리스 5.2.0, 2018년 9월 15일

Note

추가된 PostgreSQL 10.5는 PostGIS 구성 요소에 대한 libprotobuf 확장 버전 1.3.0을 지원합니다.

log_fdw 확장 사용

log_fdw 확장 기능은 Amazon RDS PostgreSQL 버전 9.6.2 이상부터 새롭게 추가되었습니다. 이 확장 기능을 사용하면 SQL 인터페이스에서 데이터베이스 엔진 로그에 액세스할 수 있습니다. RDS에서 기본으로 생성되는 stderr log 로그 파일을 볼 수 있을 뿐 아니라 CSV 로그(log_destination 파라미터를 csvlog로 설정)를 보고 여러 열로 깔끔하게 분할된 데이터로 외부 테이블을 빌드할 수 있습니다.

이 확장 기능은 데이터베이스 로그용 외부 테이블을 간편하게 생성할 수 있게 해주는 새 함수 2개를 제공합니다.

- `list_postgres_log_files()` – 데이터베이스 로그 디렉터리의 파일과 파일 크기(단위: 바이트)를 나열합니다.

- `create_foreign_table_for_log_file(table_name text, server_name text, log_file_name text)` – 현재 데이터베이스에서 지정된 파일에 대해 외부 테이블을 빌드합니다.

`log_fdw`가 생성하는 모든 함수는 `rds_superuser`가 소유합니다. `rds_superuser` 역할의 구성원은 다른 데이터베이스 사용자에게 이러한 함수에 대한 액세스 권한을 부여할 수 있습니다.

다음 예제는 `log_fdw` 확장을 사용하는 방법을 보여 줍니다.

`log_fdw` 확장을 사용하려면

1. `log_fdw` 확장을 가져옵니다.

```
postgres=> CREATE EXTENSION log_fdw;
CREATE EXTENSION
```

2. 로그 서버를 외부 데이터 래퍼로 생성합니다.

```
postgres=> CREATE SERVER log_server FOREIGN DATA WRAPPER log_fdw;
CREATE SERVER
```

3. 로그 파일 목록에서 모든 파일을 선택합니다.

```
postgres=> SELECT * from list_postgres_log_files() order by 1;
```

샘플 응답은 다음과 같습니다.

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. CSV 파일이 아닐 때는 'log_entry' 열 하나만 포함된 테이블을 생성합니다.

```
postgres=> SELECT create_foreign_table_for_log_file('my_postgres_error_log',
    'log_server', 'postgresql.log.2016-08-09-22.csv');
```

샘플 응답은 다음과 같습니다.

```
-----
(1 row)
```

5. 로그 파일의 샘플을 선택합니다. 다음 코드는 로그 시간 및 오류 메시지 설명을 검색합니다.

```
postgres=> SELECT log_time, message from my_postgres_error_log order by 1;
```

샘플 응답은 다음과 같습니다.

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)	

PL/v8 업그레이드

PL/v8을 사용하고 PostgreSQL을 새 PL/v8 버전으로 업그레이드하는 경우 즉시 새로운 확장 기능을 사용하지만 카탈로그 메타데이터는 이 사실을 반영하지 않습니다. 다음 단계는 카탈로그 메타데이터를 새 PL/v8 버전과 동기화합니다. 이 단계는 선택 사항이지만 메타데이터 불일치 경고를 방지하려면 완료하는 것이 좋습니다.

- 업데이트해야 하는 것을 확인합니다.

인스턴스에 연결된 동안 다음 명령을 실행합니다.

```
select * from pg_available_extensions where name in ('plv8','plls','plcoffee');
```

결과에 기본 버전보다 낮은 설치 버전에 대한 값이 포함된 경우 이 절차를 계속 진행하여 확장 기능을 업데이트해야 합니다.

예를 들어 다음 결과 세트는 업데이트해야 함을 나타냅니다.

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)			

- 인스턴스의 스냅샷을 만듭니다.

업그레이드로 모든 PL/v8 함수가 중단됩니다. 만일을 위해 인스턴스의 스냅샷을 만듭니다. 스냅샷이 생성되는 동안 다음 단계를 따라 계속 진행할 수 있습니다.

스냅샷 생성 단계는 [DB 스냅샷 생성 \(p. 203\)](#) 단원을 참조하십시오.

3. 종단하고 다시 생성할 함수의 개수를 가져옵니다.

인스턴스에 있는 PL/v8 함수의 개수를 파악해야 업그레이드 이후 모두 존재하는지 확인할 수 있습니다.

다음 코드는 PL/v8, plcoffee 또는 plls에 쓰여진 함수의 개수를 반환합니다.

```
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. pg_dump를 사용하여 스키마 전용 덤프 파일을 생성합니다.

다음 코드는 /tmp 디렉터리의 클라이언트 머신에 파일을 생성합니다.

```
./pg_dump -Fc --schema-only -U master postgres > /tmp/test.dmp
```

이 예는 다음 플래그를 사용합니다.

- -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"

pg_dump에 대한 자세한 내용은 [pd_dump](#)를 참조하십시오.

5. 덤프 파일에 있는 'CREATE FUNCTION' DDL 문을 추출하십시오.

다음 코드는 함수 생성에 필요한 DDL 문을 추출합니다. 후속 단계에서 이를 사용하여 함수를 다시 생성합니다. 코드는 grep 명령을 사용하여 파일로 문을 추출합니다.

```
./pg_restore -l /tmp/test.dmp | grep FUNCTION > /tmp/function_list/
```

pg_restore에 대한 자세한 내용은 [pg_restore](#)를 참조하십시오.

6. 함수 및 확장 기능을 종단합니다.

다음 코드는 모든 PL/v8 기반 객체를 종단합니다. 캐스케이드 옵션은 모든 종속이 종단되도록 합니다.

```
drop extension plv8 cascade;
```

PostgreSQL 인스턴스에 plcoffee 또는 plls 기반 객체가 포함된 경우 이러한 확장 기능에 대해 이 단계를 반복합니다.

7. 확장 기능을 생성합니다.

다음 코드는 PL/v8, plcoffee 및 plls 확장 기능을 생성합니다.

```
create extension plv8;
```

```
create extension plcoffee;
```

```
create extension plls;
```

8. 덤프 파일과 "드라이버" 파일을 사용하여 함수를 생성합니다.

다음 코드는 이전에 추출한 함수를 다시 생성합니다.

```
./pg_restore -U master -d postgres -Fc -L /tmp/function_list /tmp/test.dmp
```

9. 함수 개수를 확인합니다.

다음 코드를 다시 실행하여 함수가 다시 생성되는 중인지 확인합니다.

```
select * from pg_available_extensions where name in ('plv8','plls','plcoffee');
```

Note

PL/v8 버전 2는 결과 세트에 다음 추가 행을 추가합니다.

proname	nspname	lanname
plv8_version	pg_catalog	plv8

지원되는 PostgreSQL 기능

Amazon RDS에서는 가장 일반적인 PostgreSQL 기능을 대부분 지원합니다. 다음이 포함됩니다.

주제

- [Amazon RDS의 PostgreSQL에 대한 논리적 복제 \(p. 1054\)](#)
- [Amazon RDS의 PostgreSQL에 대한 이벤트 트리거 \(p. 1056\)](#)
- [PostgreSQL용 Amazon RDS의 방대한 페이지 \(p. 1056\)](#)
- [Amazon RDS의 PostgreSQL에 대한 테이블 공간 \(p. 1057\)](#)
- [Amazon RDS의 PostgreSQL용 Autovacuum \(p. 1057\)](#)
- [stats_temp_directory에 대한 RAM 디스크 \(p. 1057\)](#)
- [PostgreSQL용 ALTER ENUM \(p. 1058\)](#)

Amazon RDS의 PostgreSQL에 대한 논리적 복제

PostgreSQL 10.4 이상 버전에서는 RDS가 PostgreSQL 10 논리적 복제에 대한 SQL 구문 계시 및 구독을 지원합니다.

PostgreSQL용 Amazon RDS DB 인스턴스에 대한 논리적 복제를 활성화하려면

1. Amazon RDS의 PostgreSQL 데이터베이스에 대한 논리적 복제를 수행하려면 AWS 사용자 계정에 `rds_superuser` 역할이 필요합니다.
2. `rds.logical_replication` 정적 파라미터를 1로 설정합니다.
3. 게시자 인스턴스(프로덕션)에 대한 보안 그룹의 인바운드 규칙을 수정하여 구독자 인스턴스(복제)의 연결을 허용합니다. 이는 주로 보안 그룹에 구독자의 IP 주소를 포함시키면서 완료됩니다.
4. 정적 `rds.logical_replication` 파라미터 변경이 적용되도록 DB 인스턴스를 다시 시작합니다.

PostgreSQL 논리적 복제에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

논리적 디코딩 및 논리적 복제

PostgreSQL 9.4 이상 버전에서 RDS는 논리적 복제 디코딩을 통한 WAL 변경 스트리밍을 지원합니다. Amazon RDS에서는 PostgreSQL DB 인스턴스 9.4.9 이상 버전과 9.5.4 이상 버전에 대한 논리적 디코딩을 지원합니다. 인스턴스에 대한 논리적 복제 슬롯을 설정하고 해당 슬롯을 통해 데이터베이스 변경을 클라이언트(예: `pg_recvlogical`)에 스트리밍할 수 있습니다. 논리적 복제 슬롯은 데이터베이스 수준에서 생성되며 단일 데이터베이스에 대한 복제 연결을 지원합니다.

가장 일반적인 PostgreSQL 논리적 복제용 클라이언트는 AWS EC2 인스턴스의 사용자 지정 관리형 호스트 또는 AWS Database Migration Service입니다. 논리적 복제 슬롯에서는 스트림 수신기에 대해 전혀 알지 못

하며, 대상이 복제본 데이터베이스일 필요가 없습니다. 논리적 복제 슬롯을 설정하고 슬롯에서 데이터를 읽지 않을 경우 데이터가 DB 인스턴스의 스토리지에 기록되고 빠르게 가득 찰 수 있습니다.

Amazon RDS의 PostgreSQL 논리적 복제 및 논리적 디코딩은 파라미터, 복제 연결 유형 및 보안 역할에 의해 활성화됩니다. PostgreSQL DB 인스턴스의 데이터베이스에 대한 복제 연결을 설정할 수 있는 모든 클라이언트는 논리적 디코딩용 클라이언트가 될 수 있습니다.

PostgreSQL용 Amazon RDS DB 인스턴스에 대한 논리적 디코딩을 활성화하려면

1. 논리적 복제를 활성화하려면 사용자 계정에 `rds_superuser` 역할이 필요합니다. 사용자 계정에는 또한 `rds_replication` 역할이 있어야 논리적 슬롯을 관리하고 논리적 슬롯을 사용하여 데이터를 스트리밍할 수 있는 권한을 부여합니다.
2. `rds.logical_replication` 정적 파라미터를 1로 설정합니다. 이 파라미터를 적용하는 중에 `wal_level`, `max_wal_senders`, `max_replication_slots`, `max_connections` 파라미터 또한 설정합니다. 이러한 파라미터 변경은 WAL 생성을 강화하므로 논리적 슬롯을 사용할 때 `rds.logical_replication` 파라미터만 설정하면 됩니다.
3. 정적 `rds.logical_replication` 파라미터가 적용되도록 DB 인스턴스를 재부팅합니다.
4. 다음 섹션에 설명된 대로 논리적 복제 슬롯을 생성합니다. 이 프로세스에서는 디코딩 플러그인을 지정해야 합니다. 현재 PostgreSQL과 함께 배송되는 `test_decoding` 출력 플러그인을 지원합니다.

PostgreSQL 논리적 디코딩에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

논리적 복제 슬롯 작업

SQL 명령을 사용하여 논리적 슬롯 작업을 수행할 수 있습니다. 예를 들어 다음 명령은 기본 PostgreSQL 출력 플러그인인 `test_decoding`을 사용하여 `test_slot` 논리적 슬롯을 생성합니다.

```
SELECT * FROM pg_create_logical_replication_slot('test_slot', 'test_decoding');
```

다음과 같이 출력됩니다

```
slot_name      | xlog_position
-----+-----
regression_slot | 0/16B1970
(1 row)
```

논리적 슬롯 목록을 보려면 다음 명령을 사용합니다.

```
SELECT * FROM pg_replication_slots;
```

논리적 슬롯을 삭제하려면 다음 명령을 사용합니다.

```
SELECT pg_drop_replication_slot('test_slot');
```

다음과 같이 출력됩니다

```
pg_drop_replication_slot
```

(1 row)

논리적 복제 슬롯 작업에 대한 더 많은 예제는 PostgreSQL 문서의 [Logical Decoding Examples](#)를 참조하십시오.

논리적 복제 슬롯을 생성한 후 스트리밍을 시작할 수 있습니다. 다음 예제에서는 PostgreSQL 배포에 포함된 pg_recvlogical 프로그램을 사용하여 스트리밍 복제 프로토콜에 대한 논리적 디코딩을 제어하는 방법을 보여줍니다. 이렇게 하려면 복제 연결을 허용하도록 클라이언트 인증을 설정해야 합니다.

```
pg_recvlogical -d postgres --slot test_slot -U master
--host sg-postgresql1.c6c8mresaghv0.us-west-2.rds.amazonaws.com
-f - --start
```

Amazon RDS의 PostgreSQL에 대한 이벤트 트리거

PostgreSQL 9.4.9 이상 버전과 9.5.4 이상 버전에서는 이벤트 트리거를 지원하고, Amazon RDS에서는 이러한 버전에 대한 이벤트 트리거를 지원합니다. 마스터 사용자 계정을 사용하여 이벤트 트리거를 생성, 수정, 이름 변경 및 삭제할 수 있습니다. 이벤트 트리거는 DB 인스턴스 레벨에서 수행되므로 인스턴스에 대한 모든 데이터베이스에 적용될 수 있습니다.

예를 들어 다음 코드는 모든 DDL 명령의 끝에 현재 사용자를 인쇄하는 이벤트 트리거를 생성합니다.

```
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();
```

PostgreSQL 이벤트 트리거에 대한 자세한 내용은 PostgreSQL 문서의 [Event Triggers](#) 단원을 참조하십시오.

Amazon RDS에서 PostgreSQL 이벤트 트리거를 사용하는 데 대한 몇 가지 제한 사항이 있습니다. 다음이 포함됩니다.

- 읽기 전용 복제본에 대한 이벤트 트리거를 생성할 수 없지만, 읽기 전용 복제본 마스터에 대한 이벤트 트리거는 생성할 수 있습니다. 그러면 이벤트 트리거가 읽기 전용 복제본으로 복사됩니다. 읽기 전용 복제본으로 복사된 이벤트 트리거는 변경 내용이 마스터에서 푸시되더라도 읽기 전용 복제본에서 발생하지 않습니다. 하지만 읽기 전용 복제본이 승격되면 데이터베이스 작업이 있을 때 기존 이벤트 트리거가 발생합니다.
- 이벤트 트리거를 사용하는 PostgreSQL DB 인스턴스로의 메이저 버전 업그레이드를 수행하려면 인스턴스를 업그레이드하기 전에 이벤트 트리거를 삭제해야 합니다.

PostgreSQL용 Amazon RDS의 방대한 페이지

PostgreSQL용 Amazon RDS는 PostgreSQL 버전 9.4.11 이상, 9.5.6 이상 및 9.6.2 이상에서 다양한 페이지 크기를 지원합니다. 여기에는 4K 및 2MB 페이지 크기도 포함됩니다.

방대한 페이지는 연속 메모리 청크를 대량으로 사용할 때 오버헤드를 줄여줍니다. 애플리케이션에 방대한 페이지를 할당하려면 mmap 또는 SYSV 공유 메모리 호출을 사용합니다. PostgreSQL 데이터베이스용 Amazon RDS에서 방대한 페이지를 활성화하려면 `huge_pages` 파라미터를 사용합니다. 방대한 페이지를 사용하려면 이 파라미터를 'on'으로 설정하십시오.

PostgreSQL 버전 10 이상의 경우, 모든 인스턴스 클래스에서 방대한 페이지가 활성화되어 있습니다. PostgreSQL 버전 10 미만의 경우, db.r4.* , db.m4.16xlarge, db.m5.* 인스턴스 클래스에서 방대한 페이지가 기본적으로 활성화되어 있습니다. 다른 인스턴스 클래스의 경우 방대한 페이지가 기본적으로 비활성화되어 있습니다.

`huge_pages` 파라미터를 "on"으로 설정할 경우 Amazon RDS가 사용 가능한 공유 메모리를 기반으로 방대한 페이지를 사용합니다. DB 인스턴스가 공유 메모리 제약 때문에 방대한 페이지를 사용할 수 없을 경우는 DB 인스턴스 시작을 금지하고 DB 인스턴스 상태를 파라미터 호환 장애 상태로 설정합니다. 이 경우, `huge_pages` 파라미터를 "off"로 설정하여 Amazon RDS가 DB 인스턴스를 시작하도록 허용할 수 있습니다.

`shared_buffers` 파라미터가 방대한 페이지를 사용하는 데 필요한 공유 메모리 풀을 설정하는 관건입니다. `shared_buffers` 파라미터 기본값은 해당 인스턴스의 메모리에서 사용 가능한 총 8K 페이지에 대한 백분율로 설정됩니다. 방대한 페이지를 사용할 경우 이러한 페이지는 함께 배치된 방대한 페이지에 할당됩니다. 공유 메모리 파라미터를 DB 인스턴스 메모리의 90% 이상을 요구하도록 설정할 경우 Amazon RDS가 DB 인스턴스를 파라미터 호환 장애 상태로 설정하므로 주의해야 합니다. PostgreSQL용 공유 메모리 설정에 대한 자세한 내용은 [PostgreSQL 설명서](#)를 참조하십시오.

Note

db.m1, db.m2 및 db.m3 DB 인스턴스 클래스에서는 방대한 페이지가 지원되지 않습니다.

Amazon RDS의 PostgreSQL에 대한 테이블 공간

호환성을 위해 Amazon RDS의 PostgreSQL에서 테이블스페이스가 지원됩니다. 모든 스토리지가 단일 논리 볼륨 상에 있으므로, IO 분할 또는 격리에 테이블스페이스를 사용할 수 없습니다. Amazon은 단일 논리 볼륨이 대부분의 사용 사례에 최선의 설정임을 보여주는 벤치마크 결과와 실용적 경험을 보유하고 있습니다.

Amazon RDS의 PostgreSQL용 Autovacuum

새로운 PostgreSQL DB 인스턴스의 경우 PostgreSQL autovacuum 기능이 기본적으로 켜져 있습니다. Autovacuum은 선택 사항이지만 고지 않는 것이 좋습니다. PostgreSQL용 Amazon RDS와 함께 autovacuum을 사용하는 자세한 방법은 [Amazon RDS에서 PostgreSQL Autovacuum 사용 \(p. 994\)](#) 단원을 참조하십시오.

stats_temp_directory에 대한 RAM 디스크

PostgreSQL용 Amazon RDS의 `rds.pg_stat_ramdisk_size` 파라미터를 사용하여 PostgreSQL `stats_temp_directory` 저장용 RAM 디스크에 할당되는 시스템 메모리를 지정할 수 있습니다. RAM 디스크 파라미터는 Amazon RDS의 모든 PostgreSQL 버전에 사용할 수 있습니다.

특정 워크로드에서 이 파라미터를 설정하면 성능이 향상되고 IO 요구 사항이 감소될 수 있습니다. `stats_temp_directory`에 대한 자세한 내용은 [PostgreSQL 문서](#) 단원을 참조하십시오.

`stats_temp_directory`에 대한 RAM 디스크를 활성화하려면 DB 인스턴스에 사용되는 파라미터 그룹에서 `rds.pg_stat_ramdisk_size` 파라미터를 0이 아닌 값으로 설정합니다. 파라미터 값은 MB 단위입니다. 변경 사항을 적용하려면 DB 인스턴스를 재부팅해야 합니다.

예를 들어 다음 AWS CLI 명령은 RAM 디스크 파라미터를 256MB로 설정합니다.

```
postgres=>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"
```

재부팅 후 다음 명령을 실행하여 stats_temp_directory의 상태를 확인합니다.

```
postgres=>show stats_temp_directory;
```

명령은 다음을 반환합니다.

```
stats_temp_directory  
-----  
/rdsdbramdisk/pg_stat_tmp  
(1 row)
```

PostgreSQL용 ALTER ENUM

PostgreSQL용 Amazon RDS 버전 9.6.2 및 9.5.6 이상은 열거형 값 변경을 지원합니다. 이 기능은 의 다른 버전에서는 제공되지 않습니다.

다음 코드는 열거형 값을 변경하는 예제입니다.

```
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)
```

PostgreSQL DB 인스턴스에 대한 제한

Amazon RDS에서 PostgreSQL에 대한 제한 사항 목록은 다음과 같습니다.

- PostgreSQL DB 인스턴스는 최대 40개까지 보유할 수 있으며,
- 스토리지 제한은 [Amazon RDS DB 인스턴스 스토리지 \(p. 101\)](#) 단원을 참조하십시오.
- Amazon RDS는 시스템 유지 관리를 위해 최대 3개의 연결을 예약합니다. 사용자 연결 파라미터의 값을 지정할 경우 사용할 것으로 예상하는 연결 개수에 3을 더할 필요가 있습니다.

PostgreSQL DB 인스턴스 업그레이드

관리할 수 있는 PostgreSQL DB 인스턴스의 업그레이드 유형은 두 가지가 있습니다:

- OS 업데이트 – 보안 수정 사항이나 OS 변경 사항을 적용하기 위해 때때로 Amazon RDS가 DB 인스턴스의 기본 작업 시스템을 업데이트해야 할 수 있습니다. OS 업데이트는 RDS 콘솔, AWS Command Line Interface(AWS CLI) 또는 RDS API를 사용해 Amazon RDS의 적용 시기를 결정할 수 있습니다.
OS 업데이트에 대한 자세한 내용은 다음 단원을 참조하십시오. [DB 인스턴스의 업데이트 적용 \(p. 114\)](#)
- 데이터베이스 엔진 업그레이드 – Amazon RDS에서 새 데이터베이스 엔진 버전을 지원하는 경우, DB 인스턴스를 새 버전으로 업그레이드할 수 있습니다. 메이저 버전 업그레이드와マイ너 버전 업그레이드라는 두 가지 업그레이드가 있습니다. Amazon RDS는 PostgreSQL DB 인스턴스의 메이저 및 마이너 버전 업그레이드를 둘 다 지원합니다.

PostgreSQL DB 엔진 업그레이드에 대한 자세한 내용은 다음 단원을 참조하십시오. [Amazon RDS용 PostgreSQL DB 엔진 업그레이드 \(p. 963\)](#)

PostgreSQL DB 인스턴스와 함께 SSL 사용

Amazon RDS는 PostgreSQL DB 인스턴스를 위한 SSL 암호화를 지원합니다. SSL을 사용하여 애플리케이션과 PostgreSQL DB 인스턴스 사이의 PostgreSQL 연결을 암호화할 수 있습니다. 또한 PostgreSQL DB 인스턴스에 대한 모든 연결에서 SSL을 사용하도록 지정할 수도 있습니다.

주제

- [PostgreSQL DB 인스턴스에 SSL 연결 요구 \(p. 1059\)](#)
- [SSL 연결 상태 확인 \(p. 1060\)](#)

PostgreSQL의 모든 AWS 리전에서 SSL 지원 기능을 사용할 수 있습니다. Amazon RDS는 PostgreSQL DB 인스턴스가 생성될 때 인스턴스의 SSL 인증서를 만듭니다. SSL 인증서 확인을 활성화하는 경우에는 SSL 인증서에 스피핑 공격으로부터 보호해주는 SSL 인증서를 위한 일반 이름(CN)으로 DB 인스턴스 앤드포인트가 포함됩니다.

SSL을 통해 PostgreSQL DB 인스턴스에 연결하려면

1. <https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem>에 저장된 인증서를 다운로드 합니다.
2. 다음과 같이 사용 중인 운영 체제로 인증서를 가져옵니다.
3. 연결 문자열에 sslmode=verify-full을 추가하여 SSL을 통해 PostgreSQL DB 인스턴스에 연결합니다. sslmode=verify-full을 사용하면 SSL 연결에서 SSL 인증서의 앤드포인트와 비교하여 DB 인스턴스 앤드포인트를 확인합니다.

`sslrootcert` 파라미터를 사용하여 인증서를 참조합니다(예: `sslrootcert=rds-ssl-ca-cert.pem`).

`psql` 프로그램을 사용하여 PostgreSQL DB 인스턴스에 연결하는 예는 다음과 같습니다.

```
$ psql -h testpg.cdhmuqifdpib.us-east-1.rds.amazonaws.com -p 5432 \
"dbname=testpg user=testuser sslrootcert=rds-ca-2015-root.pem sslmode=verify-full"
```

PostgreSQL DB 인스턴스에 SSL 연결 요구

`rds.force_ssl` 파라미터를 사용하여 PostgreSQL DB 인스턴스에 대한 연결이 SSL을 사용하도록 요구할 수 있습니다. 기본적으로 `rds.force_ssl` 파라미터는 0(해제)으로 설정됩니다. `rds.force_ssl` 파라미터를 1(설정)로 설정하면 해당 DB 인스턴스에 대한 연결에 대해 SSL을 요구합니다. `rds.force_ssl` 파라미터는 0(해제)으로 설정하면 해당 DB 인스턴스에 대한 연결에 대해 SSL을 요구합니다. `rds.force_ssl` 파라미터를 1(설정)로 설정하면 해당 DB 인스턴스에 대한 연결에 대해 SSL을 요구합니다.

터를 업데이트해도 PostgreSQL ssl 파라미터가 1(설정)로 설정되고, DB 인스턴스의 pg_hba.conf 파일이 새로운 SSL 구성을 지원하도록 수정됩니다.

rds.force_ssl 파라미터 값은 DB 인스턴스의 파라미터 그룹을 업데이트하여 설정할 수 있습니다. DB 인스턴스의 파라미터 그룹이 기본 파라미터 그룹이 아니고 rds.force_ssl 파라미터를 1로 설정할 때 ssl 파라미터가 이미 1로 설정되어 있을 경우 DB 인스턴스를 재부팅할 필요가 없습니다. 그렇지 않을 경우 변경 사항을 적용하려면 DB 인스턴스를 재부팅해야 합니다. 파라미터 그룹에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.

DB 인스턴스에 대해 rds.force_ssl 파라미터를 1로 설정하면 연결 시 다음과 같이 SSL이 요구된다는 출력이 표시됩니다.

```
$ psql postgres -h SOMEHOST.amazonaws.com -p 8192 -U someuser
psql (9.3.12, server 9.4.4)
WARNING: psql major version 9.3, server major version 9.4.
Some psql features might not work.
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.

postgres=>
```

SSL 연결 상태 확인

DB 인스턴스에 연결할 때 로그온 배너에 연결의 암호화된 상태가 표시됩니다.

```
Password for user master:
psql (9.3.12)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.

postgres=>
```

또한, sslinfo 확장을 로드한 다음 ssl_is_used() 함수를 호출하여 SSL이 사용 중인지 확인할 수 있습니다. 이 함수는 연결이 SSL을 사용할 경우 t을 반환하고, 그렇지 않으면 f를 반환합니다.

```
postgres=> create extension sslinfo;
CREATE EXTENSION

postgres=> select ssl_is_used();
 ssl_is_used
-----
t
(1 row)
```

select ssl_cipher() 명령을 사용하여 SSL 암호를 확인할 수 있습니다.

```
postgres=> select ssl_cipher();
ssl_cipher
-----
DHE-RSA-AES256-SHA
(1 row)
```

set rds.force_ssl을 활성화하고 인스턴스를 다시 시작하면 SSL이 아닌 연결은 다음 메시지와 함께 거부됩니다.

```
$ 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
$
```

Amazon RDS 관련 제한

이번 주제에서는 Amazon RDS 리소스 제한 값과 각 제약 조건의 명명 방법에 대해 설명합니다.

주제

- [Amazon RDS의 제한 값 \(p. 1062\)](#)
- [Amazon RDS의 제약 조건 명명 \(p. 1063\)](#)
- [Amazon RDS의 파일 크기 제한 \(p. 1065\)](#)

Amazon RDS의 제한 값

각 AWS 계정에는 AWS 리전마다 생성할 수 있는 Amazon RDS 리소스 수에 제한이 있습니다. 리소스가 제한 값에 이르면 해당 리소스 생성을 위한 추가 호출이 예외와 함께 실패합니다.

다음 표는 각 리전의 리소스와 그 제한 값을 나타냅니다.

Resource	기본 제한
클러스터	40
클러스터 파라미터 그룹	50
교차 리전 스냅샷 복사 요청	5
DB 인스턴스	40
이벤트 구독	20
수동 스냅샷 수	100
수동 클러스터 스냅샷 수	100
옵션 그룹 수	20
파라미터 그룹 수	50
마스터당 읽기 전용 복제본	5
예약 인스턴스	40
DB 보안 그룹별 규칙	20
VPC 보안 그룹당 규칙	50 인바운드 50 아웃바운드
DB 보안 그룹	25
VPC 보안 그룹	5
서브넷 그룹 수	50
서브넷 그룹 1개당 서브넷 수	20
리소스당 태그	50
모든 DB 인스턴스의 총 스토리지	100TiB

Note

기본적으로 최대 총 40개의 Amazon RDS DB 인스턴스를 실행할 수 있습니다. '라이선스 포함' 모델에서는 이러한 인스턴스 40개 중 최대 10개가 Oracle 또는 SQL Server DB 인스턴스일 수 있습니다. "BYOL" 모델에서는 40개 모두가 MySQL, MariaDB, PostgreSQL 또는 Oracle일 수 있습니다. 애플리케이션에 더 많은 DB 인스턴스가 필요한 경우 [RDS DB 인스턴스 제한 요청](#)이라는 요청 양식을 통해 추가 DB 인스턴스를 요청할 수 있습니다.

AWS Backup이 관리하는 백업은 수동 스냅샷 제한을 위한 수동 스냅샷으로 간주됩니다. AWS Backup에 대한 자세한 내용은 [AWS 백업 Lambda 개발자 안내서](#)를 참조하십시오.

Amazon RDS의 제약 조건 명명

다음 표는 Amazon RDS의 제약 조건을 명명하는 방법을 나타냅니다.

DB 인스턴스 식별자	<ul style="list-style-type: none">1~63자 의 영숫자 문자 또는 하이픈으로 구성되어야 합니다.첫 번째 문자는 글자이어야 합니다.하이픈으로 끝나거나 하이픈이 2개 연속으로 이어져서는 안 됩니다.각 리전별로 AWS 계정 1개의 모든 DB 인스턴스는 고유해야 합니다.
데이터베이스 이름	<p>데이터베이스 이름 제약 조건은 각 데이터베이스 엔진에 따라 다릅니다.</p> <p>MySQL 및 MariaDB</p> <ul style="list-style-type: none">1~64자 의 영숫자 문자로 구성되어야 합니다.데이터베이스 엔진에 포함된 단어는 사용할 수 없습니다. <p>Oracle</p> <ul style="list-style-type: none">문자 수는 8개를 초과할 수 없습니다. <p>PostgreSQL</p> <ul style="list-style-type: none">1~63자 의 영숫자 문자로 구성되어야 합니다.글자나 밑줄로 시작되어야 합니다. 이후 문자는 글자, 밑줄 또는 숫자(0~9)가 될 수 있습니다.데이터베이스 엔진에 포함된 단어는 사용할 수 없습니다. <p>SQL Server</p> <ul style="list-style-type: none">해당 사항 없음 SQL Server의 경우, DB 인스턴스를 만든 후에 데이터베이스를 만듭니다. 데이터베이스 이름은 일반적인 SQL Server 명명 규칙을 따릅니다.
마스터 사용자 이름	<p>마스터 사용자 이름 제약 조건은 각 데이터베이스 엔진에 따라 다릅니다.</p> <p>MariaDB</p> <ul style="list-style-type: none">1~16자 의 영숫자 문자로 구성되어야 합니다.데이터베이스 엔진에 포함된 단어는 사용할 수 없습니다.

	<p>MySQL</p> <ul style="list-style-type: none">• 1~16자의 영숫자 문자로 구성되어야 합니다.• 첫 번째 문자는 글자이어야 합니다.• 데이터베이스 엔진에 포함된 단어는 사용할 수 없습니다. <p>Oracle</p> <ul style="list-style-type: none">• 1~30자의 영숫자 문자로 구성되어야 합니다.• 첫 번째 문자는 글자이어야 합니다.• 데이터베이스 엔진에 포함된 단어는 사용할 수 없습니다. <p>PostgreSQL</p> <ul style="list-style-type: none">• 1~63자의 영숫자 문자로 구성되어야 합니다.• 첫 번째 문자는 글자이어야 합니다.• 데이터베이스 엔진에 포함된 단어는 사용할 수 없습니다. <p>SQL Server</p> <ul style="list-style-type: none">• 1~64자의 영숫자 문자로 구성되어야 합니다.• 첫 번째 문자는 글자이어야 합니다.• 데이터베이스 엔진에 포함된 단어는 사용할 수 없습니다.
마스터 암호	<p>마스터 데이터베이스 사용자의 암호는 "/" , "##", 또는 "@"를 제외하고 인쇄 가능한 모든 ASCII 문자를 사용할 수 있습니다. 마스터 암호 제약 조건은 각 데이터베이스 엔진에 따라 다릅니다.</p> <p>MySQL 및 MariaDB</p> <ul style="list-style-type: none">• 8~41자의 문자로 구성되어야 합니다. <p>Oracle</p> <ul style="list-style-type: none">• 8~30자의 문자로 구성되어야 합니다. <p>PostgreSQL</p> <ul style="list-style-type: none">• 8~128자의 문자로 구성되어야 합니다. <p>SQL Server</p> <ul style="list-style-type: none">• 8~128자의 문자로 구성되어야 합니다.
DB 파라미터 그룹 이름	<ul style="list-style-type: none">• 1~255자의 영숫자 문자로 구성되어야 합니다.• 첫 번째 문자는 글자이어야 합니다.• 하이픈은 허용되지만 이름은 하이픈으로 끝나거나 하이픈이 2개 연속으로 이어져서는 안 됩니다.
DB 서브넷 그룹 이름	<ul style="list-style-type: none">• 1~255자로 구성되어야 합니다.• 영숫자, 스페이스, 하이픈, 밑줄, 마침표를 사용할 수 있습니다.

Amazon RDS의 파일 크기 제한

파일 크기 제한은 Amazon RDS DB 인스턴스에 적용됩니다.

Amazon RDS의 MySQL 파일 크기 제한

Amazon RDS MySQL DB 인스턴스의 경우 최대 프로비저닝 스토리지 제한으로 인해 InnoDB 테이블당 파일 테이블스페이스를 사용하여 각 테이블의 크기가 최대 16TB로 제한됩니다. 또한 이 제한은 시스템 테이블스페이스를 최대 16TB의 크기로 제한합니다. MySQL DB 인스턴스에서는 테이블이 각각 자체 테이블스페이스에 들어 있는 InnoDB 테이블당 파일 테이블스페이스가 기본적으로 설정됩니다. 자세한 내용은 [Amazon RDS DB 인스턴스 스토리지 \(p. 101\)](#) 단원을 참조하십시오.

Note

일부 기존 DB 인스턴스에는 하한이 있습니다. 예를 들어 2014년 4월 이전에 생성된 MySQL DB 인스턴스는 파일과 테이블 크기 제한이 2TB입니다. 마찬가지로 DB 인스턴스의 생성 시기와 상관없이 2014년 4월 이전에 생성된 DB 스냅샷에서 생성한 DB 인스턴스나 읽기 전용 복제본 역시 파일 크기가 2TB로 제한됩니다.

애플리케이션에 따라 InnoDB 테이블당 파일 테이블스페이스 사용에 대한 장점과 단점은 서로 다릅니다. 애플리케이션에 가장 적합한 접근 방식을 확인하려면 MySQL 문서의 [InnoDB 테이블당 파일 모드](#)을(를) 참조하십시오.

테이블을 최대 파일 크기로 늘리도록 허용하는 것은 권장하지 않습니다. 일반적으로 모범 사례는 성능 및 복구 시간을 향상할 수 있도록 데이터를 더 작은 테이블로 분할하는 것입니다.

라지 테이블을 여러 개의 스몰 테이블로 분할하는 데 사용할 수 있는 한 가지 옵션으로 파티셔닝이 있습니다. 파티셔닝을 수행하면 사용자가 지정하는 규칙에 따라 라지 테이블의 일부가 개별 파일로 배포됩니다. 예를 들어, 트랜잭션을 날짜별로 저장하는 경우 파티셔닝을 사용하여 이전 트랜잭션을 개별 파일로 배포하는 파티셔닝 규칙을 생성할 수 있습니다. 이렇게 하면 애플리케이션에서 즉시 사용할 필요가 없는 이전 트랜잭션 데이터를 주기적으로 보관할 수 있습니다. 자세한 내용은 MySQL 설명서의 [Partitioning](#)을 참조하십시오.

테이블의 파일 크기를 확인하는 방법

다음 SQL 명령을 사용하여 크기가 너무 커서 파티셔닝을 수행해야 하는 테이블이 있는지 확인합니다. 테이블 통계를 업데이트하려면 각 테이블에서 `ANALYZE TABLE` 명령을 실행합니다. 자세한 내용은 MySQL 설명서에서 [ANALYZE TABLE](#)을 참조하십시오.

```
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');
```

InnoDB 테이블당 파일 테이블스페이스를 활성화하는 방법

- InnoDB 테이블당 파일 테이블스페이스를 활성화하려면 DB 인스턴스에 대한 파라미터 그룹에서 `innodb_file_per_table` 파라미터를 1로 설정합니다.

InnoDB 테이블당 파일 테이블스페이스를 비활성화하는 방법

- InnoDB 테이블당 파일 테이블스페이스를 비활성화하려면 DB 인스턴스에 대한 파라미터 그룹에서 `innodb_file_per_table` 파라미터를 0으로 설정합니다.

파라미터 그룹 업데이트에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#)을(를) 참조하십시오.

InnoDB 테이블당 파일 테이블스페이스를 활성화하거나 비활성화하면 ALTER TABLE 명령을 실행할 수 있습니다. 이 명령을 사용하여 다음의 예와 같이 전역 테이블스페이스에서 자체 테이블스페이스로 또는 자체 테이블스페이스에서 전역 테이블스페이스로 테이블을 이동할 수 있습니다.

```
ALTER TABLE table_name ENGINE=InnoDB, ALGORITHM=COPY;
```

Amazon RDS의 MariaDB 파일 크기 제한

Amazon RDS MariaDB DB 인스턴스의 경우, 최대 프로비저닝 스토리지 제한으로 인해 InnoDB 테이블당 파일 테이블스페이스를 사용하여 각 테이블의 크기가 최대 16TB로 제한됩니다. 또한 이 제한은 시스템 테이블스페이스를 최대 16TB의 크기로 제한합니다. Amazon RDS MariaDB DB 인스턴스에서는 테이블이 각각 자체 테이블스페이스에 들어 있는 InnoDB 테이블당 파일 테이블스페이스가 기본적으로 설정됩니다. 자세한 내용은 [Amazon RDS DB 인스턴스 스토리지 \(p. 101\)](#) 단원을 참조하십시오.

애플리케이션에 따라 InnoDB 테이블당 파일 테이블스페이스 사용에 대한 장점과 단점은 서로 다릅니다. 애플리케이션에 가장 적합한 접근 방식을 확인하려면 MySQL 문서의 [InnoDB 테이블당 파일 모드](#)을(를) 참조하십시오.

테이블을 최대 파일 크기로 늘리도록 허용하는 것은 권장하지 않습니다. 일반적으로 모범 사례는 성능 및 복구 시간을 향상할 수 있도록 데이터를 더 작은 테이블로 분할하는 것입니다.

라지 테이블을 여러 개의 스몰 테이블로 분할하는 데 사용할 수 있는 한 가지 옵션으로 파티셔닝이 있습니다. 파티셔닝을 수행하면 사용자가 지정하는 규칙에 따라 라지 테이블의 일부가 개별 파일로 배포됩니다. 예를 들어, 트랜잭션을 날짜별로 저장하는 경우 파티셔닝을 사용하여 이전 트랜잭션을 개별 파일로 배포하는 파티셔닝 규칙을 생성할 수 있습니다. 이렇게 하면 애플리케이션에서 즉시 사용할 필요가 없는 이전 트랜잭션 데이터를 주기적으로 보관할 수 있습니다. 자세한 내용은 MySQL 문서의 <https://dev.mysql.com/doc/refman/5.6/en/partitioning.html>을(를) 참조하십시오.

테이블의 파일 크기를 확인하는 방법

다음 SQL 명령을 사용하여 크기가 너무 커서 파티셔닝을 수행해야 하는 테이블이 있는지 확인합니다. 테이블 통계를 업데이트 하려면 각 테이블에서 ANALYZE TABLE 명령을 실행합니다. 자세한 내용은 MySQL 설명서에서 [ANALYZE TABLE](#)을 참조하십시오.

```
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');
```

InnoDB 테이블당 파일 테이블스페이스를 활성화하는 방법

- InnoDB 테이블당 파일 테이블스페이스를 활성화하려면 DB 인스턴스에 대한 파라미터 그룹에서 innodb_file_per_table 파라미터를 1로 설정합니다.

InnoDB 테이블당 파일 테이블스페이스를 비활성화하는 방법

- InnoDB 테이블당 파일 테이블스페이스를 비활성화하려면 DB 인스턴스에 대한 파라미터 그룹에서 innodb_file_per_table 파라미터를 0으로 설정합니다.

파라미터 그룹 업데이트에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#)을(를) 참조하십시오.

InnoDB 테이블당 파일 테이블스페이스를 활성화하거나 비활성화하면 ALTER TABLE 명령을 실행할 수 있습니다. 이 명령을 사용하여 다음의 예와 같이 전역 테이블스페이스에서 자체 테이블스페이스로 또는 자체 테이블스페이스에서 전역 테이블스페이스로 테이블을 이동할 수 있습니다.

```
ALTER TABLE table_name ENGINE=InnoDB, ALGORITHM=COPY;
```

Amazon RDS 문제 해결

다음 단원을 통해 Amazon RDS 및 Aurora의 DB 인스턴스와 관련해 발생한 문제를 해결할 수 있습니다.

주제

- [Amazon RDS DB 인스턴스에 연결할 수 없음 \(p. 1068\)](#)
- [Amazon RDS 보안 문제 \(p. 1069\)](#)
- [DB 인스턴스 소유자 역할 암호 재설정 \(p. 1070\)](#)
- [Amazon RDS DB 인스턴스 중단 또는 재부팅 \(p. 1070\)](#)
- [Amazon RDS DB 파라미터 변경 사항이 적용 안 됨 \(p. 1071\)](#)
- [Amazon RDS DB 인스턴스 스토리지 부족 \(p. 1071\)](#)
- [Amazon RDS 부족한 DB 인스턴스 용량 \(p. 1072\)](#)
- [Amazon RDS MySQL 및 MariaDB 문제 \(p. 1072\)](#)
- [Amazon RDS Oracle GoldenGate 문제 \(p. 1078\)](#)
- [Amazon RDS SQL Server DB 인스턴스에 연결할 수 없음 \(p. 1078\)](#)
- [Amazon RDS PostgreSQL DB 인스턴스에 연결할 수 없음 \(p. 1079\)](#)
- [백업 보존 기간을 0으로 설정 불가 \(p. 1079\)](#)

Amazon RDS API를 사용해 문제를 디버깅하는 방법에 대한 자세한 내용은 [Amazon RDS에서 애플리케이션 문제 해결 \(p. 1080\)](#) 단원을 참조하십시오.

Amazon RDS DB 인스턴스에 연결할 수 없음

DB 인스턴스에 연결할 수 없을 때는 공통적인 원인은 다음과 같습니다.

- 로컬 방화벽에서 적용되는 액세스 규칙과 인스턴스의 보안 그룹에 있는 DB 인스턴스에 액세스하기 위한 권한을 부여한 수신 IP 주소가 동기화되어 있지 않습니다. 보안 그룹의 수신 규칙에 문제가 있을 가능성이 매우 높습니다. 기본적으로 DB 인스턴스는 액세스를 허용하지 않습니다. 액세스 권한은 보안 그룹을 통해 부여됩니다. 액세스 권한을 부여하려면 상황에 맞는 구체적인 수신 및 송신 규칙으로 자체적인 보안 그룹을 만들어야 합니다. 필요한 경우 DB 인스턴스 안팎의 소스와 관련된 트래픽을 허용하는 VPC와 연결된 보안 그룹에 규칙을 추가하십시오. IP 주소, IP 주소의 범위 또는 다른 VPC 보안 그룹을 지정할 수 있습니다.

보안 그룹 설정에 대한 자세한 정보는 [보안 그룹을 생성하여 VPC 내부의 DB 인스턴스에 대한 액세스를 제공 \(p. 8\)](#) 단원을 참조하십시오.

- 로컬 방화벽 제한 때문에 DB 인스턴스를 만들 때 지정한 포트를 사용하여 통신을 주고받을 수 없습니다. 이 경우에는 네트워크 관리자에게 문의하여 네트워크에서 지정한 포트를 인바운드 및 아웃바운드 통신에 사용할 수 있는지 확인하십시오.
- DB 인스턴스가 여전히 생성 중이므로 아직 사용할 수는 없습니다. DB 인스턴스의 크기에 따라, 인스턴스를 사용할 수 있을 때까지 최장 20분까지 걸릴 수 있습니다.

Amazon RDS DB 인스턴스 연결 테스트

공통 Linux 또는 Windows 도구를 사용하여 DB 인스턴스에 대한 연결을 테스트할 수 있습니다.

Linux 또는 Unix 터미널에서 다음을 입력하여 연결을 테스트할 수 있습니다(<DB-instance-endpoint>를 엔드포인트로 바꾸고 <port>를 DB 인스턴스의 포트로 바꿈).

```
nc -zv <DB-instance-endpoint> <port>
```

예를 들어 다음은 샘플 명령과 반환 값을 나타낸 것입니다.

```
nc -zv postgresql1.c6c8mn7tsdgv0.us-west-2.rds.amazonaws.com 8299
Connection to postgresql1.c6c8mn7tsdgv0.us-west-2.rds.amazonaws.com 8299 port [tcp/vvr-data] succeeded!
```

Windows 사용자는 Telnet을 사용하여 DB 인스턴스에 대한 연결을 테스트할 수 있습니다. Telnet 작업은 연결 테스트 이외의 목적으로는 지원되지 않습니다. 연결에 성공한 경우 이 작업을 수행할 때 아무런 메시지도 반환되지 않습니다. 연결에 실패한 경우 다음과 같은 오류 메시지가 수신됩니다.

```
C:\>telnet sg-postgresql1.c6c8mntzhgv0.us-west-2.rds.amazonaws.com 819
Connecting To sg-postgresql1.c6c8mntzhgv0.us-west-2.rds.amazonaws.com...Could not open
connection to the host, on port 819: Connect failed
```

Telnet 작업으로 성공 메시지가 반환되면 보안 그룹이 올바로 구성된 것입니다.

Note

Amazon RDS는 ping을 포함하여 ICMP(Internet Control Message Protocol) 트래픽을 수락합니다.

연결 인증 문제 해결

DB 인스턴스에 연결할 수 있지만 인증 오류가 발생하는 경우 DB 인스턴스에 대한 마스터 사용자 암호를 재설정하고 싶을 수도 있을 것입니다. RDS 인스턴스를 수정하여 이 작업을 수행할 수 있습니다.

DB 인스턴스 수정에 대한 자세한 정보는 다음 주제 중 하나를 참조하십시오.

- MySQL 데이터베이스 엔진 기반 DB 인스턴스의 변경 (p. 600)
- Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 (p. 742)
- Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 (p. 509)
- PostgreSQL 데이터베이스 엔진을 실행 중인 DB 인스턴스 수정 (p. 956)

Amazon RDS 보안 문제

보안 문제를 피하려면 사용자 계정에 마스터 AWS 사용자 이름과 암호를 절대 사용하지 마십시오. 모범 사례에 따라 마스터 AWS 계정을 사용하여 IAM 사용자를 만들고 이런 사용자를 DB 사용자 계정에 할당하는 것이 좋습니다. 필요한 경우 마스터 계정을 사용하여 다른 사용자 계정을 만들 수도 있습니다.

IAM 사용자를 만드는 방법에 대한 자세한 정보는 [IAM 사용자 생성 \(p. 5\)](#) 단원을 참조하십시오.

오류 메시지 "계정 속성을 불러오지 못했습니다. 일부 콘솔 기능이 손상되었을 수 있습니다."

이 오류가 발생하는 이유는 몇 가지가 있습니다. 계정에 권한이 누락되었거나 계정 설정이 올바르지 않을 수 있습니다. 새 계정인 경우 계정이 준비되기까지 충분한 시간이 지나지 않았을 수도 있습니다. 기존 계정이라

면 DB 인스턴스 생성과 같은 특정 작업을 수행하기 위한 액세스 정책에 권한이 없을 수 있습니다. 이 문제를 해결하려면 IAM 관리자가 필요한 역할을 해당 계정에 제공해야 합니다. 자세한 정보는 IAM 설명서 단원을 참조하십시오.

DB 인스턴스 소유자 역할 암호 재설정

마스터 암호를 재설정하여 DB 인스턴스에 할당된 권한을 재설정할 수 있습니다. 예를 들어 SQL Server 데이터베이스의 db_owner 역할의 암호를 분실했을 경우 DB 인스턴스 마스터 암호를 수정하여 db_owner 역할 암호를 재설정할 수 있습니다. DB 인스턴스 암호를 변경하면 DB 인스턴스에 다시 액세스할 수 있고, 수정된 db_owner 암호를 사용하여 데이터베이스에 액세스할 수 있으며, 실수로 취소되었을 수 있는 db_owner 역할에 대한 권한을 복원할 수 있습니다. Amazon RDS 콘솔, AWS CLI 명령 [modify-db-instance](#), 또는 [ModifyDBInstance](#) 작업을 사용하여 DB 인스턴스 암호를 변경할 수 있습니다.

SQL Server DB 인스턴스 수정에 대한 자세한 정보는 [Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 \(p. 509\)](#) 단원을 참조하십시오.

Amazon RDS DB 인스턴스 중단 또는 재부팅

DB 인스턴스가 재부팅될 때, DB 인스턴스가 DB 인스턴스에 대한 액세스를 차단하는 상태에 놓여 있을 때, 그리고 데이터베이스가 다시 시작될 때 DB 인스턴스 중단이 발생할 수 있습니다. DB 인스턴스를 수동으로 재부팅할 때 또는 DB 인스턴스 설정을 변경하고 이 변경 사항을 적용하기 위해 재부팅해야 할 때 재부팅이 이루어질 수 있습니다.

DB 인스턴스에 대한 설정을 수정할 때 [Apply Immediately] 설정을 사용하여 변경 사항을 적용할 시점을 결정할 수 있습니다.

DB 인스턴스 작업과 Apply Immediately(즉시 적용) 값 설정이 미치는 효과를 보여주는 표를 보려면 [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#) 단원을 참조하십시오.

변경 사항을 적용하려면 재부팅해야 하는 설정을 변경할 때 또는 수동으로 재부팅할 때만 DB 인스턴스가 재부팅됩니다. 설정을 변경하고 변경 사항을 즉시 적용할 것을 요청하는 경우에 재부팅이 수행되거나, DB 인스턴스의 유지 관리 기간 중에 재부팅이 수행될 수 있습니다.

다음 중 한 가지가 발생할 때는 그 즉시 DB 인스턴스가 재부팅됩니다.

- DB 인스턴스에 대한 백업 보존 기간을 0에서 0이 아닌 값으로 변경하거나 0이 아닌 값에서 0으로 변경하고 [Apply Immediately]를 [true]로 설정할 때
- DB 인스턴스 클래스를 변경하고 [Apply Immediately]가 [true]로 설정되어 있을 때
- 스토리지 유형을 Magnetic (Standard)(마그네틱(표준))에서 General Purpose (SSD)(범용(SSD)) 또는 Provisioned IOPS (SSD)(프로비저닝된 IOPS(SSD))로 변경하거나 Provisioned IOPS (SSD)(프로비저닝된 IOPS(SSD)) 또는 General Purpose (SSD)(범용(SSD))에서 Magnetic (Standard)(마그네틱(표준))으로 변경하거나 표준에서 PIOPS로 변경합니다.

유지 관리 기간 중에 다음 중 한 가지가 발생할 때 DB 인스턴스가 재부팅됩니다.

- DB 인스턴스에 대한 백업 보존 기간을 0에서 0이 아닌 값으로 변경하거나 0이 아닌 값에서 0으로 변경하고 [Apply Immediately]가 [false]로 설정되어 있을 때
- DB 인스턴스 클래스를 변경하고 [Apply Immediately]가 [false]로 설정되어 있을 때

DB 파라미터 그룹에서 정적 파라미터를 변경할 때, 파라미터 그룹과 연결된 DB 인스턴스를 재부팅해야 변경 사항이 적용됩니다. 변경하려면 수동으로 재부팅해야 합니다. 유지 관리 기간 중에는 DB 인스턴스가 자동으로 재부팅되지 않기 때문입니다.

Amazon RDS DB 파라미터 변경 사항이 적용 안 됨

DB 파라미터 그룹에서 파라미터를 변경하지만 변경 사항이 적용되지는 않을 경우 DB 파라미터 그룹과 연결된 DB 인스턴스를 재부팅해야 할 것입니다. 동적 파라미터를 변경할 때는 변경 사항이 즉시 적용되며, 정적 파라미터를 변경할 때는 파라미터 그룹과 연결된 DB 인스턴스를 재부팅해야 변경 사항이 적용됩니다.

RDS 콘솔을 사용하거나 RebootDbInstance API 작업을 명시적으로 호출하여 DB 인스턴스를 재부팅할 수 있습니다(DB 인스턴스가 Multi-AZ deployment에 있는 경우 장애 조치 없음). 고정 파라미터 변경 후 연결된 DB 인스턴스를 재부팅하도록 하면 ModifyDBInstance를 호출하여 DB 인스턴스 클래스를 변경하는 경우와 같이 잘못된 파라미터 구성이 API 호출에 영향을 주는 위험을 완화할 수 있습니다. 자세한 정보는 [DB 파라미터 그룹의 파라미터 설정 \(p. 164\)](#) 단원을 참조하십시오.

Amazon RDS DB 인스턴스 스토리지 부족

DB 인스턴스에 스토리지 공간이 부족할 경우 DB 인스턴스를 더 이상 사용하지 못할 수 있습니다. CloudWatch에 게시되는 FreeStorageSpace 메트릭을 계속 모니터링하여 DB 인스턴스에 충분한 스토리지 여유 공간이 있는지 확인해야 합니다.

데이터베이스 인스턴스에 스토리지가 부족할 경우 상태가 storage-full로 바뀝니다. 예를 들어 스토리지를 모두 소진한 DB 인스턴스에 대해 DescribeDBInstances 작업을 호출하면 다음과 같이 출력됩니다.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance

DBINSTANCE mydbinstance 2009-12-22T23:06:11.915Z db.m3.large mysql5.6 50 sa
storage-full mydbinstance.clla4j4jgyp.us-east-1.rds.amazonaws.com 3306
us-east-1b 3
SECGROUP default active
PARAMGRP default.mysql5.6 in-sync
```

이 상황에서 벗어나려면 ModifyDBInstance 작업이나 다음 AWS CLI 명령을 사용하여 인스턴스에 스토리지 공간을 더 추가하십시오.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--allocated-storage 60 \
--apply-immediately
```

Windows의 경우:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--allocated-storage 60 ^
--apply-immediately
```

```
DBINSTANCE mydbinstance 2009-12-22T23:06:11.915Z db.m3.large mysql5.6 50 sa
storage-full mydbinstance.clla4j4jgyp.us-east-1.rds.amazonaws.com 3306
us-east-1b 3 60
SECGROUP default active
PARAMGRP default.mysql5.6 in-sync
```

이제는 DB 인스턴스를 설명할 때 DB 인스턴스의 상태가 modifying이 되어 스토리지가 확장되고 있는 종임을 나타내게 됩니다.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE mydbinstance 2009-12-22T23:06:11.915Z db.m3.large mysql5.6 50 sa
modifying mydbinstance.clla4j4jgyp.us-east-1.rds.amazonaws.com
3306 us-east-1b 3 60
SECGROUP default active
PARAMGRP default.mysql5.6 in-sync
```

스토리지 확장이 완료되면 DB 인스턴스 상태가 available로 바뀝니다.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE mydbinstance 2009-12-22T23:06:11.915Z db.m3.large mysql5.6 60 sa
available mydbinstance.clla4j4jgyp.us-east-1.rds.amazonaws.com 3306
us-east-1b 3
SECGROUP default active
PARAMGRP default.mysql5.6 in-sync
```

DescribeEvents 작업을 사용하여 스토리지 공간이 고갈될 때 알림 메시지를 받을 수 있습니다. 예를 들어 이 시나리오에서는 이런 작업 후에 DescribeEvents 호출을 실행하면 다음과 같이 출력됩니다.

```
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 부족한 DB 인스턴스 용량

DB 인스턴스 클래스를 수정하려고 할 때 InsufficientDBInstanceCapacity 오류가 발생하면 DB 인스턴스가 EC2-Classic 플랫폼에 있기 때문일 수 있습니다. 즉, VPC에 없습니다. 일부 DB 인스턴스 클래스에는 VPC가 필요합니다. 예를 들어 EC2-Classic 플랫폼에 있는데 VPC가 필요한 DB 인스턴스 클래스로 전환하여 용량을 늘리려고 하면 이 오류가 발생합니다. VPC에서만 사용 가능한 Amazon Elastic Compute Cloud 인스턴스 유형에 대한 자세한 정보는 Amazon Elastic Compute Cloud 사용 설명서의 [VPC에서만 사용할 수 있는 인스턴스 유형](#)을 참조하십시오.

이 문제를 해결하려면 DB 인스턴스를 VPC로 이동합니다. 자세한 내용은 [VPC에 있지 않은 DB 인스턴스를 VPC로 이동 \(p. 420\)](#) 단원을 참조하십시오.

DB 인스턴스 수정에 대한 자세한 정보는 [Amazon RDS DB 인스턴스 수정 \(p. 111\)](#) 단원을 참조하십시오. Amazon EC2의 인스턴스 용량 문제 해결에 대한 자세한 정보는 Amazon Elastic Compute Cloud 사용 설명서의 [인스턴스 용량 문제 해결](#)을 참조하십시오.

Amazon RDS MySQL 및 MariaDB 문제

MySQL 및 MariaDB DB 인스턴스의 문제를 진단하고 수정할 수 있습니다.

인덱스 병합 최적화가 잘못된 결과를 반환

이 문제는 MySQL DB 인스턴스에만 적용됩니다.

인덱스 병합 최적화를 사용하는 쿼리는 MySQL 5.5.37에서 도입된 MySQL 쿼리 옵티마이저의 버그로 인해 잘못된 결과를 반환할 수 있습니다. 여러 개의 인덱스가 있는 테이블에 대해 쿼리를 실행하면 옵티마이저가 여러 개의 인덱스를 기반으로 여러 범위의 행을 스캔하지만 결과를 올바르게 함께 병합하지 않습니다. 쿼리 최적화 프로그램 버그에 대한 자세한 내용은 MySQL 버그 데이터베이스의 <http://bugs.mysql.com/bug.php?id=72745> 및 <http://bugs.mysql.com/bug.php?id=68194>을(를) 참조하십시오.

예를 들면, 검색 인수가 인덱싱된 열을 참조하는 2개의 인덱스가 있는 테이블에 대한 쿼리를 고려합니다.

```
SELECT * FROM table1
  WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

이 경우 검색 엔진이 두 인덱스를 모두 검색합니다. 그러나 버그로 인해 병합 결과가 정확하지 않습니다.

이 문제를 해결하려면 다음 중 한 가지 방법을 시도하면 됩니다.

- MySQL DB 인스턴스용 DB 파라미터 그룹에서 `optimizer_switch` 파라미터를 `index_merge=off`로 설정합니다. DB 파라미터 그룹 파라미터 설정에 대한 자세한 내용은 [DB 파라미터 그룹 작업 \(p. 161\)](#) 단원을 참조하십시오.
- MySQL DB 인스턴스를 MySQL 버전 5.6, 5.7 또는 8.0로 업그레이드하십시오. 자세한 내용은 [MySQL DB 스냅샷 업그레이드 \(p. 612\)](#)을(를) 참조하십시오.
- 인스턴스를 업그레이드하거나 `optimizer_switch` 파라미터를 변경할 수 없는 경우, 쿼리에 대한 인덱스를 명시적으로 확인하여 버그를 해결할 수 있습니다. 예:

```
SELECT * FROM table1
  USE INDEX covering_index
  WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

자세한 정보는 [인덱스 병합 최적화](#) 단원을 참조하십시오.

Read Replica 사이의 지연 문제 진단 및 해결

MySQL 또는 MariaDB 읽기 전용 복제본을 만들고 읽기 전용 복제본을 사용할 수 있게 된 후, Amazon RDS는 우선 읽기 전용 복제본 만들기 작업이 시작된 시간부터 원본 DB 인스턴스에서 변경된 내용을 복제합니다. 이 단계 중에 Read Replica에 대한 복제 지연 시간은 0보다 큽니다. Amazon RDS `ReplicaLag` 지표를 보고 Amazon CloudWatch에서 이 지연 시간을 모니터링할 수 있습니다.

`ReplicaLag` 지표는 MySQL 또는 MariaDB `SHOW SLAVE STATUS` 명령에서 `Seconds_Behind_Master` 필드 값을 보고합니다. 자세한 정보는 [SHOW SLAVE STATUS](#) 단원을 참조하십시오. `ReplicaLag` 지표가 0에 도달하면 복제본이 원본 DB 인스턴스를 따라잡은 것입니다. `ReplicaLag` 지표가 -1을 반환하는 경우 복제가 활성 상태가 아닐 수 있습니다. 복제 오류 문제를 해결하는 방법은 [MySQL 또는 MariaDB 읽기 복제 오류 진단 및 해결 \(p. 1074\)](#) 단원을 참조하십시오. `ReplicaLag` 값이 -1인 경우 `Seconds_Behind_Master` 값을 결정할 수 없거나 이 값이 NULL이라는 의미일 수도 있습니다.

네트워크가 중단된 기간 동안이나 유지 관리 기간 중에 패치가 적용될 때 `ReplicaLag` 지표는 -1을 반환합니다. 이 경우에는 네트워크 연결이 복원되거나 유지 관리 기간이 종료되기를 기다린 후 `ReplicaLag` 지표를 다시 확인합니다.

MySQL 및 MariaDB 읽기 복제 기술은 비동기식이기 때문에, 원본 DB 인스턴스에서 `BinLogDiskUsage` 지표와 읽기 전용 복제본에서 `ReplicaLag` 지표의 경우 가끔 증가할 것으로 예상할 수 있습니다. 예를 들어 원본 DB 인스턴스에 대해 대량의 쓰기 작업이 동시에 발생할 수 있는 반면, 읽기 전용 복제본에 대한 쓰기 작업은 단일 I/O 스레드를 사용하여 직렬화됩니다. 이로 인해 원본 인스턴스와 읽기 전용 복제본 사이에 지연이 발생할 수 있습니다. 읽기 전용 복제본과 MySQL에 대한 자세한 정보는 MySQL 문서의 [Replication Implementation Details](#)를 참조하십시오. 읽기 전용 복제본과 MariaDB에 대한 자세한 정보는 MariaDB 설명서에서 [Replication Overview](#)를 참조하십시오.

다음을 수행하여 원본 DB 인스턴스에 대한 업데이트와 읽기 전용 복제본에 대한 후속 업데이트 사이의 지연을 줄일 수 있습니다.

- 원본 DB 인스턴스의 스토리지 크기에 필적하는 스토리지 크기를 가지도록 읽기 전용 복제본의 DB 인스턴스 클래스를 설정합니다.
- 원본 DB 인스턴스와 읽기 전용 복제본에 사용되는 DB 파라미터 그룹의 파라미터 설정이 호환되는지 확인합니다. 자세한 정보와 예는 다음 섹션에서 `max_allowed_packet` 파라미터에 대해 설명한 내용을 참조하십시오.
- 쿼리 캐시를 비활성화합니다. 자주 수정되는 테이블의 경우, 쿼리 캐시를 사용하면 캐시가 자주 잡기고 새로 고쳐지기 때문에 복제 지연이 늘어날 수 있습니다. 이럴 경우 쿼리 캐시를 비활성화하면 복제 지연이 줄어드는 효과를 볼 수도 있습니다. DB 인스턴스에 대한 DB 파라미터 그룹에서 `query_cache_type parameter`를 0으로 설정하여 쿼리 캐시를 비활성화할 수 있습니다. 쿼리 캐시에 대한 자세한 정보는 [Query Cache Configuration](#)을 참조하십시오.
- MySQL용 InnoDB, MariaDB 10.2 이상용 InnoDB, MariaDB 10.1 이하용 XtraDB의 읽기 전용 복제본에서 버퍼풀을 워밍합니다. 자주 업데이트되는 작은 테이블 집합이 있고 InnoDB 또는 XtraDB 테이블 스키마를 사용 중이라면 이런 테이블을 읽기 전용 복제본에 디스크립트합니다. 그러면 데이터베이스 엔진이 디스크에서 해당 테이블의 행을 검사한 다음 버퍼풀에 캐시하므로, 복제 지연을 줄일 수 있습니다. 다음은 그 한 예입니다.

Linux, OS X, Unix의 경우:

```
PROMPT> mysqldump \
-h <endpoint> \
--port=<port> \
-u=<username> \
-p <password> \
database_name table1 table2 > /dev/null
```

Windows의 경우:

```
PROMPT> mysqldump ^
-h <endpoint> ^
--port=<port> ^
-u=<username> ^
-p <password> ^
database_name table1 table2 > /dev/null
```

MySQL 또는 MariaDB 읽기 복제 오류 진단 및 해결

Amazon RDS는 읽기 전용 복제본의 복제 상태를 모니터링하고, 어떤 이유로든 복제가 중지되는 경우 읽기 전용 복제본 인스턴스의 [Replication State] 필드를 [Error]로 업데이트합니다. [Replication Error] 필드를 확인하여 MySQL 또는 MariaDB 엔진에서 발생한 관련 오류의 세부 정보를 검토할 수 있습니다. [RDS-EVENT-0045 \(p. 295\)](#), [RDS-EVENT-0046 \(p. 295\)](#) 및 [RDS-EVENT-0047 \(p. 294\)](#)을 포함하여 읽기 전용 복제본의 상태를 표시하는 이벤트도 생성됩니다. 이벤트와 이벤트 구독에 대한 자세한 정보는 [Amazon RDS 이벤트 알림 서비스 사용 \(p. 291\)](#) 단원을 참조하십시오. MySQL 오류 메시지가 반환되는 경우 [MySQL 오류 메시지 문서](#)에 설명되어 있는 오류를 검토하십시오. MariaDB 오류 메시지가 반환되는 경우 [MariaDB 오류 메시지 문서](#)에 설명되어 있는 오류를 검토하십시오.

복제 오류의 원인이 되는 공통적인 상황은 다음과 같습니다.

- 읽기 전용 복제본에 대한 `max_allowed_packet` 파라미터의 값은 원본 DB 인스턴스에 대한 `max_allowed_packet` 파라미터보다 작습니다.

`max_allowed_packet` 파라미터는 데이터베이스에서 실행할 수 있는 데이터 조작 언어(DML)의 최대 크기 지정에 사용되는 DB 파라미터 그룹에서 설정할 수 있는 사용자 지정 파라미터입니다. 원본 DB 인스턴스에 대한 `max_allowed_packet` 파라미터 값이 읽기 전용 복제본에 대한 `max_allowed_packet` 파

라미터 값보다 작을 경우 복제 프로세스에서 오류가 발생하여 복제가 중지될 수 있습니다. 가장 흔한 오류는 `packet bigger than 'max_allowed_packet' bytes`입니다. 원본 및 읽기 전용 복제본이 같은 `max_allowed_packet` 파라미터 값을 가진 DB 파라미터 그룹을 사용하도록 하여 이 오류를 수정할 수 있습니다.

- 읽기 전용 복제본에 있는 테이블에 쓰기. 읽기 전용 복제본에서 인덱스를 만들 경우 `read_only` 파라미터를 0으로 설정하여 인덱스를 만들어야 합니다. 읽기 전용 복제본에 있는 테이블에 데이터를 쓰면 복제가 중단될 수 있습니다.
- MyISAM 같은 비트랜잭션 스토리지 엔진을 사용할 때. 읽기 전용 복제본에는 트랜잭션 스토리지 엔진이 필요합니다. 복제는 MySQL용 InnoDB, MariaDB 10.2 이상용 InnoDB, MariaDB 10.1 이하용 XtraDB 스토리지 엔진에만 지원됩니다.

다음 명령으로 MyISAM 테이블을 InnoDB로 변환할 수 있습니다.

```
alter table <schema>.<table_name> engine=innodb;
```

- `SYSDATE()`와 같이 안전하지 않은 비결정적 쿼리 사용. 자세한 정보는 [Determination of Safe and Unsafe Statements in Binary Logging](#)을 참조하십시오.

다음 단계를 통해 복제 오류를 해결할 수 있습니다.

- 논리적 오류가 발생했는데 이 오류를 건너뛰어도 안전할 경우에는 [현재 복제 오류 넘어가기](#) (p. 670)에 설명되어 있는 단계를 따르십시오. MySQL 또는 MariaDB DB 인스턴스에서는 `mysql_rds_skip_repl_error` 프로시저를 포함한 버전이 실행 중이어야 합니다. 자세한 정보는 [mysql.rds_skip_repl_error](#) (p. 692) 단원을 참조하십시오.
- Binlog 위치 문제가 발생하는 경우 `mysql_rds_next_master_log` 명령으로 슬레이브 재생 위치를 변경할 수 있습니다. 슬레이브 재생 위치를 변경하려면 MySQL 또는 MariaDB DB 인스턴스에서 `mysql_rds_next_master_log` 명령을 지원하는 버전이 실행 중이어야 합니다. 버전 정보는 [mysql.rds_next_master_log](#) (p. 693)를 참조하십시오.
- 높은 DML 부하 때문에 일시적인 성능 문제가 발생할 경우 읽기 전용 복제본의 DB 파라미터 그룹에서 `innodb_flush_log_at_trx_commit` 파라미터를 2로 설정할 수 있습니다. 그러면 일시적으로 원자성, 일관성, 격리성 및 내구성(ACID)이 감소하지만 읽기 전용 복제본이 변화를 따라잡는데 도움이 될 수 있습니다.
- 읽기 전용 복제본을 삭제하고 앤드포인트가 이전 읽기 전용 복제본의 앤드포인트와 동일하게 유지되도록 같은 DB 인스턴스 식별자를 사용하여 인스턴스를 만들 수 있습니다.

복제 오류가 해결되면 Replication State가 replicating으로 변경됩니다. 자세한 정보는 [MySQL 읽기 전용 복제본의 문제 해결](#) (p. 649) 단원을 참조하십시오.

이진 로깅이 활성화된 상태에서 트리거를 생성하려면 SUPER 권한 필요

RDS MySQL 또는 MariaDB DB 인스턴스에서 트리거 생성을 시도할 때 다음 오류가 발생할 수 있습니다.

```
"You do not have the SUPER privilege and binary logging is enabled"
```

이진 로깅이 활성화된 상태에서 트리거를 사용하려면 RDS MySQL 및 MariaDB DB 인스턴스로 제한된 SUPER 권한이 필요합니다. `log_bin_trust_function_creators` 파라미터를 true로 설정하면 SUPER 권한 없이 이진 로깅이 활성화된 상태에서 트리거를 생성할 수 있습니다. `log_bin_trust_function_creators`를 true로 설정하려면 새 DB 파라미터 그룹을 생성하거나 기존 DB 파라미터 그룹을 수정해야 합니다.

이진 로깅이 활성화된 상태에서 RDS MySQL 또는 MariaDB DB 인스턴스에서 트리거를 생성할 수 있는 새 DB 파라미터 그룹을 생성하려면 다음 CLI 명령을 사용합니다. 기존 파라미터 그룹을 수정하려면 2단계부터 시작합니다.

CLI를 사용하여 이진 로깅이 활성화된 상태에서 트리거를 허용하는 새 파라미터 그룹을 생성하려면

- 새 파라미터 그룹을 생성해야 합니다.

Linux, OS X, Unix의 경우:

```
aws rds create-db-parameter-group \
--db-parameter-group-name allow-triggers \
--db-parameter-group-family mysql5.5 \
--description "parameter group allowing triggers"
```

Windows의 경우:

```
aws rds create-db-parameter-group ^
--db-parameter-group-name allow-triggers ^
--db-parameter-group-family mysql5.5 ^
--description "parameter group allowing triggers"
```

- DB 파라미터 그룹이 트리거를 허용하도록 수정합니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name allow-triggers \
--parameters "name=log_bin_trust_function_creators,value=true, method=pending-reboot"
```

Windows의 경우:

```
aws rds modify-db-parameter-group ^
--db-parameter-group-name allow-triggers ^
--parameters "name=log_bin_trust_function_creators,value=true, method=pending-reboot"
```

- DB 인스턴스가 새 DB 파라미터 그룹을 사용하도록 수정합니다.

Linux, OS X, Unix의 경우:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--db-parameter-group-name allow-triggers \
--apply-immediately
```

Windows의 경우:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--db-parameter-group-name allow-triggers ^
--apply-immediately
```

- 변경 사항을 적용하려면 DB 인스턴스를 수동으로 재부팅합니다.

```
aws rds reboot-db-instance mydbinstance
```

특정 시점으로 복원 오류 진단 및 해결

임시 테이블을 포함한 DB 인스턴스 복원

MySQL 또는 MariaDB DB 인스턴스의 특정 시점으로 복원(PITR)을 시도할 때 다음 오류가 발생할 수 있습니다.

```
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.
```

PITR은 DB 인스턴스를 특정 시점으로 복원하기 위해 MySQL 또는 MariaDB의 백업 스냅샷과 binlog에 모두 의존합니다. Binlog에서는 임시 테이블 정보를 신뢰할 수 없어 PITR 오류가 발생할 수 있습니다. MySQL 또는 MariaDB DB 인스턴스에서 임시 테이블을 사용하면 백업을 더 자주 수행하여 PITR 오류 발생 가능성을 최소화할 수 있습니다. PITR 오류는 임시 테이블 생성과 다음 백업 스냅샷 생성 사이의 시간에 발생할 가능성이 가장 높습니다.

인 메모리 테이블을 포함한 DB 인스턴스 복원

인 메모리 테이블이 있는 데이터베이스를 복원할 때 문제가 발생할 수 있습니다. 인 메모리 테이블은 다시 시작하는 동안 제거됩니다. 따라서 재부팅 후 인 메모리 테이블이 비어 있을 수 있습니다. 인 메모리 테이블을 사용할 때는 다시 시작할 경우에 대비하여 빈 테이블을 처리하기 위한 솔루션을 설계하는 것이 좋습니다. 복제된 DB 인스턴스와 함께 인 메모리 테이블을 사용 중이라면, 읽기 전용 복제본이 재부팅하여 빈 인 메모리 테이블에서 데이터를 복원할 수 없는 경우 다시 시작한 후에 읽기 전용 복제본을 다시 만들어야 할 수도 있습니다.

백업과 PITR에 대한 자세한 정보는 [백업 작업 \(p. 196\)](#) 및 [DB 인스턴스를 지정된 시간으로 복원 \(p. 221\)](#) 단원을 참조하십시오.

Slave Down 또는 Disabled 오류

`mysql.rds_skip_repl_error` 명령을 호출하면 다음 오류 메시지가 표시될 수 있습니다. `Slave is down or disabled.`

이 오류 메시지는 복제가 중지되었고 재시작할 수 없기 때문에 표시됩니다.

많은 수의 오류를 건너뛰어야 하는 경우, 복제 지연이 이진 로그 파일의 기본 보관 기간 이상으로 늘어날 수 있습니다. 이 경우, 이진 로그 파일이 복제본에서 재실행되기 전에 지워지기 때문에 치명적 오류가 발생할 수 있습니다. 이 제거는 복제를 중지시키며, 복제 오류를 건너뛰기 위해 더 이상 `mysql.rds_skip_repl_error` 명령을 호출할 수 없습니다.

이 문제는 복제 마스터에서 이진 로그 파일이 보관되는 시간을 늘림으로써 완화할 수 있습니다. binlog 보관 시간을 늘린 후에 복제를 재시작하고 필요에 따라 `mysql.rds_skip_repl_error` 명령을 호출할 수 있습니다.

binlog 보관 기간을 설정하려면 [mysql.rds_set_configuration \(p. 695\)](#) 절차를 사용하여 'binlog 보관 시간' 구성 파라미터와 DB 클러스터에 binlog 파일을 보관할 시간(최대 720시간(30일))을 함께 지정합니다. 다음 예제에서는 binlog 파일의 보관 기간을 48시간으로 설정합니다.

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

읽기 전용 복제본 만들기 실패 또는 치명적 오류 1236 으로 복제 중단

MySQL 또는 MariaDB의 DB 인스턴스에 대한 기본 파라미터 값을 변경한 후 다음과 같은 문제가 발생할 수 있습니다.

- DB 인스턴스의 읽기 전용 복제본을 만들 수 없습니다.
- `fatal error 1236`로 인해 복제가 실패합니다.

MySQL 또는 MariaDB의 DB 인스턴스에 대한 기본 파라미터 값 중 일부는 트랜잭션이 커밋되기 전에 이전 로그에 트랜잭션을 기록하여 각각의 커밋이 완전히 동기화되도록 함으로써 데이터베이스의 ACID를 준수하고 읽기 전용 복제본의 충돌을 방지하는 데 도움이 됩니다. 성능 향상을 위해 이러한 파라미터를 기본값에서 다른 값으로 변경하면, 이진 로그에 기록되지 않은 트랜잭션의 경우 복제에 실패할 수 있습니다.

이 문제를 해결하려면 다음 파라미터 값을 설정하십시오.

- `sync-binlog = 1`
- `innodb_support_xa = 1`
- `innodb_flush_log_at_trx_commit = 1`

Amazon RDS Oracle GoldenGate 문제

충분한 시간 동안 로그 보존

원본 데이터베이스는 아카이빙된 다시 실행 로그를 보존해야 합니다. 로고 보존 기간은 시간 단위로 지정됩니다. Oracle GoldenGate가 필요에 따라 원본 인스턴스에서 로그를 복구할 수 있도록, 이 기간은 원본 인스턴스의 잠재적 가동 중지 시간이나 통신 또는 네트워킹 문제가 발생할 가능성이 있는 기간보다 길어야 합니다. 로그 보존 기간으로 필요한 절대 최소값은 1시간입니다. 로그 보존을 활성화하지 않거나 보존 값이 너무 작으면 다음 메시지가 표시됩니다.

```
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.
```

Amazon RDS SQL Server DB 인스턴스에 연결할 수 없음

SQL Server Management Studio를 사용하여 DB 인스턴스에 연결하는 데 문제가 있을 때, 몇 가지 공통된 원인은 다음과 같습니다.

- 로컬 방화벽에서 적용되는 액세스 규칙과 인스턴스의 보안 그룹에 있는 DB 인스턴스에 액세스하기 위한 권한을 부여한 IP 주소가 등기화되어 있지 않습니다. Microsoft SQL Server Management Studio와 함께 DB 인스턴스의 엔드포인트와 포트를 사용하는데 연결할 수 없다면 방화벽의 송신 또는 수신 규칙에 문제가 있을 가능성이 큽니다. 액세스 권한을 부여하려면 상황에 맞는 구체적인 수신 및 송신 규칙으로 자체적인 보안 그룹을 만들어야 합니다. 보안 그룹에 대한 자세한 정보는 [보안 그룹을 통한 액세스 제어 \(p. 390\)](#) 단원을 참조하십시오.

- 로컬 방화벽 제한 때문에 DB 인스턴스를 만들 때 지정한 포트를 사용하여 통신을 주고받을 수 없습니다. 이 경우에는 네트워크 관리자에게 문의하여 네트워크에서 지정한 포트를 인바운드 및 아웃바운드 통신에 사용할 수 있는지 확인하십시오.
- DB 인스턴스가 여전히 생성 중이므로 아직 사용할 수는 없습니다. DB 인스턴스의 크기에 따라, 인스턴스를 사용할 수 있을 때까지 최장 20분까지 걸릴 수 있습니다.

지정한 포트를 통해 통신을 송수신할 있는 경우 다음 SQL Server 오류가 있는지 확인하십시오.

- SQL Server에 대한 연결을 열 수 없음 - Microsoft SQL Server, 오류: 53 – Microsoft SQL Server Management Studio를 사용할 때 서버 이름을 지정할 경우 포트 번호를 포함해야 합니다. 예를 들어 DB 인스턴스의 서버 이름(포트 번호 포함)은 `sqlsvr-pdz.c6c8mdfntzgv0.region.rds.amazonaws.com,1433`일 수 있습니다.
- 대상 컴퓨터에서 연결을 거부했으므로 연결하지 못함 - Microsoft SQL Server, 오류: 10061 – 이 경우에는 DB 인스턴스에 연결했지만 연결이 거부되었습니다. 이 오류는 주로 사용자 이름이나 암호가 잘못되었을 때 발생합니다.

Amazon RDS PostgreSQL DB 인스턴스에 연결할 수 없음

PostgreSQL DB 인스턴스에 연결하려 할 때 가장 일반적으로 발생하는 문제는 DB 인스턴스에 할당된 보안 그룹의 액세스 규칙이 잘못된 경우입니다. 기본적으로 DB 인스턴스는 액세스를 허용하지 않습니다. 액세스 권한은 보안 그룹을 통해 부여됩니다. 액세스 권한을 부여하려면 상황에 맞는 구체적인 수신 및 송신 규칙으로 자체적인 보안 그룹을 만들어야 합니다. DB 인스턴스를 위한 보안 그룹을 만드는 자세한 방법은 [보안 그룹을 생성하여 VPC 내부의 DB 인스턴스에 대한 액세스를 제공 \(p. 8\)](#) 단원을 참조하십시오.

가장 흔한 오류는 `could not connect to server: Connection timed out`입니다. 이 오류가 발생할 경우 호스트 이름이 DB 인스턴스 엔드포인트이고 포트 번호가 올바른지 확인하십시오. DB 인스턴스에 할당된 보안 그룹에 로컬 방화벽을 통한 액세스를 허용하는 데 필요한 규칙이 있는지 확인하십시오.

백업 보존 기간을 0으로 설정 불가

몇 가지 이유로 백업 보존 기간을 0으로 설정해야 할 수 있습니다. 예를 들어 보존 기간을 0으로 설정하면 자동 백업을 즉시 비활성화할 수 있습니다. 이 값을 0으로 설정하고 보존 기간은 1에서 35 사이여야 한다는 메시지를 수신하는 경우, 인스턴스에 대해 읽기 전용 복제본을 설정하지 않았는지 확인하십시오. 읽기 전용 복제본에는 읽기 전용 복제본 로그를 관리하기 위한 백업이 필요하므로 보존 기간을 0으로 설정할 수 없습니다.

Amazon RDS 애플리케이션 프로그래밍 인터페이스(API) 레퍼런스

Amazon Relational Database Service(Amazon RDS)는 AWS Management 콘솔과 AWS Command Line Interface(AWS CLI) 외에 애플리케이션 프로그래밍 인터페이스(API)도 제공합니다. API를 사용하여 의 DB 인스턴스 및 기타 객체 관리 작업을 자동화할 수 있습니다.

- API 작업의 알파벳 순 목록은 [API 작업](#)을 참조하십시오.
- 데이터 형식에 대한 알파벳 순 목록은 [데이터 형식](#)을 참조하십시오.
- 공통 쿼리 파라미터 목록은 [공통 파라미터](#)를 참조하십시오.
- 오류 코드에 대한 설명은 [공통 오류](#)를 참조하십시오.

AWS CLI에 대한 자세한 정보는 [Amazon RDS에 대한 AWS Command Line Interface 레퍼런스](#)를 참조하십시오.

주제

- [Query API 사용 \(p. 1080\)](#)
- [Amazon RDS에서 애플리케이션 문제 해결 \(p. 1080\)](#)

Query API 사용

다음 섹션에서는 Query API와 함께 사용되는 파라미터 및 요청 인증에 대해 설명합니다.

쿼리 파라미터

HTTP 쿼리 기반 요청은 GET 또는 POST와 같은 HTTP 동사와 Action 쿼리 매개 변수를 사용하는 HTTP 요청입니다.

각 쿼리 요청은 인증 및 작업을 처리할 수 있도록 일부 공통 파라미터를 포함해야 합니다.

일부 작업은 파라미터의 목록을 허용합니다. 이러한 목록은 `param.n` 표기법을 사용하여 지정됩니다. `n`의 값은 1부터 시작하는 정수입니다.

Amazon RDS 리전과 엔드포인트에 대한 자세한 정보는 Amazon Web Services 일반 참조의 리전 및 엔드포인트 단원에서 [Amazon Relational Database Service\(RDS\)](#)를 참조하십시오.

쿼리 요청 인증

HTTPS를 통해서만 쿼리 요청을 보낼 수 있으며 모든 쿼리 요청에는 서명이 포함되어야 합니다. AWS 서명 버전 4 또는 서명 버전 2를 사용해야 합니다. 자세한 정보는 [서명 버전 4 서명 프로세스](#) 및 [서명 버전 2 서명 프로세스](#) 단원을 참조하십시오.

Amazon RDS에서 애플리케이션 문제 해결

Amazon RDS는 Amazon RDS API와 상호 작용하는 동안 발생하는 문제를 해결할 때 도움이 되도록 구체적이고 서술적인 오류를 제공합니다.

주제

- [오류 검색 \(p. 1081\)](#)
- [문제 해결 팁 \(p. 1081\)](#)

Amazon RDS DB 인스턴스 문제 해결에 대한 자세한 내용은 [Amazon RDS 문제 해결 \(p. 1068\)](#) 단원을 참조 하십시오.

오류 검색

일반적으로 사용자는 시간을 소비하여 결과를 처리하기 전에 애플리케이션이 먼저 해당 요청으로 오류가 발생되는지 여부를 확인하려고 합니다. 오류 발생 여부를 확인하는 가장 쉬운 방법은 Amazon RDS API의 응답에서 `Error` 노드를 찾는 것입니다.

XPath 구문은 `Error` 노드의 발생뿐만 아니라 오류 코드 및 메시지를 쉽게 검색할 수 있는 간단한 방법을 제공합니다. 다음 코드 조각에서는 요청 중에 오류가 발생했는지 여부를 파악하기 위해 Perl 및 XML::XPath 모듈을 사용합니다. 오류가 발생되면 코드는 응답에 첫 번째 오류 코드와 메시지를 인쇄합니다.

```
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"; }
```

문제 해결 팁

다음 절차를 통해 Amazon RDS API의 문제를 진단하고 해결하는 것이 좋습니다.

- <http://statusaws.amazon.com>에서 Amazon RDS가 해당 AWS 리전에서 일반적으로 작동하는지 확인합니다.
- **요청 구조 확인**

각 Amazon RDS 작업에 대한 참조 페이지는 Amazon RDS API 참조에 있습니다. 파라미터를 올바르게 사용하고 있는지 여부를 다시 확인합니다. 어떤 문제가 발생할 수 있을 지에 대해 미리 알아보려면 샘플 요청이나 사용자 시나리오를 살펴보고 이러한 샘플이 유사한 작업을 하고 있는지 확인하십시오.

- **포럼 확인**

Amazon RDS와 관련하여 다른 사람들이 경험한 문제에 대한 해결책을 검색할 수 있는 개발 커뮤니티 포럼이 있습니다. 포럼을 보려면

<https://forums.aws.amazon.com/>

문서 이력

- 최종 설명서 업데이트: 2019년 5월 22일
- 현재 API 버전: 2014-10-31

다음 표에서는 2018년 5월 이후 Amazon RDS 사용 설명서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

update-history-change	update-history-description	update-history-date
Amazon RDS는 Microsoft SQL Server Audit에 대한 지원을 추가 합니다. (p. 1082)	Microsoft SQL Server에 Amazon RDS를 사용하면 SQL Server Audit을 사용하여 서버 및 데이터베이스 수준 이벤트를 감사하고 DB 인스턴스에 대한 결과를 보거나 감사 로그 파일을 Amazon S3에 직접 보낼 수 있습니다. 자세한 내용은 SQL Server Audit 을 참조하십시오.	May 23, 2019
Amazon RDS 권장 사항 개선 (p. 1082)	Amazon RDS에서 데이터베이스 리소스에 대한 자동 권장 사항을 개선했습니다. 예를 들어 Amazon RDS는 이제 데이터베이스 파라미터에 대한 권장 사항을 제공합니다. 자세한 내용은 Amazon RDS 권장 사항 사용 을 참조하십시오.	May 22, 2019
Amazon RDS SQL Server용 DB 인스턴스 하나당 더 많은 데이터베이스 지원 (p. 1082)	Microsoft SQL Server를 실행하는 각 DB 인스턴스에서 최대 30개의 데이터베이스를 만들 수 있습니다. 자세한 내용은 Microsoft SQL Server DB 인스턴스 제한 을 참조하십시오.	May 21, 2019
MariaDB, MySQL 및 PostgreSQL 용 Amazon RDS에 64TB 및 80K IOPS의 스토리지 지원 (p. 1082)	이제 최대 64TB의 스토리지 및 프로비저닝된 최대 80,000 IOPS를 사용하여 MariaDB, MySQL 및 PostgreSQL용 Amazon RDS DB 인스턴스를 만들 수 있습니다. 자세한 내용은 DB 인스턴스 스토리지 를 참조하십시오.	May 20, 2019
MySQL용 Amazon RDS는 업그레이드 사전 점검을 지원 (p. 1082)	DB 인스턴스를 MySQL 5.7에서 MySQL 8.0으로 업그레이드할 때 Amazon RDS가 비호환성 여부를 사전 점검합니다. 자세한 정보는 MySQL 5.7에서 8.0로 업그레이드하기 위한 사전 점검 을 참조하십시오.	May 17, 2019
Oracle용 Amazon RDS 2019년 4월 RU, RUR 및 PSU (p. 1082)	Oracle용 Amazon RDS는 2019년 4월 릴리스 업데이트(RU) 및 릴리스 업데이트 개정(RUR)을 지원하	May 16, 2019

기 위해 데이터베이스 엔진 버전
12.2.0.1.ru-2019-04.rur-2019-04.r1
을 릴리스했습니다. Oracle용
Amazon RDS는 2019년 4월
Oracle Database Patch Set
Update(PSU)를 지원하기 위
해 데이터베이스 엔진 버전
12.1.0.2.v16 및 11.2.0.4.v20도
릴리스했습니다. 자세한 내용은
[Oracle 데이터베이스 엔진 출시
정보 단원](#)을 참조하십시오.

MySQL 암호 확인 플러그인 지원 (p. 1082)	이제 MySQL DB 인스턴스의 Amazon RDS 보안 향상을 위해 MySQL validate_password 플러그인을 사용할 수 있습니다. 자세한 내용은 암호 확인 플러그 인 사용 을 참조하십시오.	May 16, 2019
Oracle용 Amazon RDS의 성능 개선 도우미 카운터 (p. 1082)	이제 Oracle DB 인스턴스용 성능 개선 도우미 차트에 성능 카운터 를 추가할 수 있습니다. 자세한 내 용은 Amazon RDS for Oracle의 성능 개선 도우미 카운터 를 참조 하십시오.	May 8, 2019
PostgreSQL용 Amazon RDS 버전 11.2, 10.7, 9.6.12, 9.5.16 및 9.4.21 (p. 1082)	PostgreSQL용 Amazon RDS는 마이너 버전 11.2, 10.7, 9.6.12, 9.5.16 및 9.4.21을 지원합니 다. 자세한 내용은 지원되는 PostgreSQL 데이터베이스 버전 을 참조하십시오.	May 1, 2019
초 단위 결제 지원 (p. 1082)	Amazon RDS는 이제 온디맨드 인스턴스의 경우 AWS GovCloud (US)를 제외한 모든 AWS 리전에 서 1초 증분 단위로 청구됩니다. 자세한 내용은 Amazon RDS에 대 한 DB 인스턴스 청구 단원을 참조 하십시오.	April 25, 2019
Amazon S3에서 PostgreSQL용 Amazon RDS에 필요한 데이터 가져오기 지원 (p. 1082)	이제 Amazon S3 파일의 데이터를 RDS PostgreSQL DB 인스턴스의 테이블로 가져올 수 있습니다. 자 세한 내용은 Amazon S3 데이터를 RDS PostgreSQL DB 인스턴스로 가져오기 를 참조하십시오.	April 24, 2019
Amazon S3에서 5.7 백업 복원 지원 (p. 1082)	이제 MySQL 버전 5.7 데이터베 이스의 백업을 생성하여 Amazon S3에 저장한 다음 MySQL을 실행 하는 새로운 Amazon RDS DB 인 스턴스에 백업 파일을 복원할 수 있습니다. 자세한 내용은 Amazon RDS MySQL DB 인스턴스로 백업 복원 을 참조하십시오.	April 17, 2019

PostgreSQL용 Amazon RDS에 필요한 여러 메이저 버전 업그레이드 지원 (p. 1082)	이제 PostgreSQL용 Amazon RDS를 사용하여 DB 엔진을 업그레이드할 때 여러 메이저 버전 중에서 선택할 수 있습니다. 이 기능을 사용하면 PostgreSQL 엔진 버전을 업그레이드할 때 새로운 메이저 버전으로 건너뛸 수 있습니다. 자세한 내용은 PostgreSQL DB 엔진 업그레이드 를 참조하십시오.	April 16, 2019
Oracle용 Amazon RDS에 64TiB의 스토리지 지원 (p. 1082)	이제 최대 64TB의 스토리지 및 프로비저닝된 최대 80,000 IOPS를 사용하여 Oracle용 Amazon RDS DB 인스턴스를 만들 수 있습니다. 자세한 내용은 DB 인스턴스 스토리지 를 참조하십시오.	April 4, 2019
MySQL 8.0.15 지원 (p. 1082)	이제 MySQL 버전 8.0.15를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 MySQL 버전 을 참조하십시오.	April 3, 2019
MariaDB 10.3.13 지원 (p. 1082)	이제 MariaDB 버전 10.3.13을 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 MariaDB 버전 을 참조하십시오.	April 3, 2019
Amazon RDS에서 사용되지 않는 Microsoft SQL Server 2008 R2 버전 (p. 1082)	2019년 7월 9일부터 이 버전에 대한 확장 지원이 종료될 예정이므로, Microsoft SQL Server 2008 R2 지원은 더 이상 제공되지 않습니다. 기존 Microsoft SQL Server 2008 R2 스냅샷은 2019년 6월 1일부터 Microsoft SQL Server 2012의 최신 마이너 버전으로자동 업그레이드될 예정입니다. 자세한 내용은 Amazon RDS에서의 Microsoft SQL Server 2008 R2 지원 을 참조하십시오.	April 2, 2019
Microsoft SQL Server 2017에 지원되는 상시 가동 가용성 그룹 (p. 1082)	이제 SQL Server 2017 Enterprise Edition 14.00.3049.1 이상에서 항상 가용성 그룹을 사용할 수 있습니다. 자세한 내용은 Microsoft SQL Server의 다중 AZ 배포 를 참조하십시오.	March 29, 2019
볼륨 지표 보기 (p. 1082)	이제 데이터베이스 및 로그 저장에 사용되는 물리적 디바이스인 Amazon Elastic Block Store(Amazon EBS) 볼륨의 지표를 볼 수 있습니다. 자세한 내용은 확장 모니터링 보기 를 참조하십시오.	March 20, 2019

MySQL 5.7.25 지원 (p. 1082)	이제 MySQL 버전 5.7.25를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 MySQL 버전을 참조하십시오.	March 19, 2019
Oracle용 Amazon RDS에서 RMAN DBA 작업 지원 (p. 1082)	Oracle용 Amazon RDS는 이제 RMAN 백업을 포함하여 Oracle Recovery Manager(RMAN) DBA 작업을 지원합니다. 자세한 내용은 Oracle DB 인스턴스에 대한 공통 RMA(Recovery Manager) 작업을 참조하십시오.	March 14, 2019
PostgreSQL용 Amazon RDS에서 버전 11.1 지원 (p. 1082)	이제 PostgreSQL 버전 11.1을 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 PostgreSQL 버전 11.1을 참조하십시오.	March 12, 2019
SQL Server용 Amazon RDS에서 다중 파일 복원을 사용할 수 있습니다. (p. 1082)	이제 SQL Server용 Amazon RDS를 사용하여 여러 파일에서 복원할 수 있습니다. 자세한 내용은 데 이터베이스 복원을 참조하십시오.	March 11, 2019
PostgreSQL용 Amazon RDS에 Amazon RDS 성능 개선 도우미 사용 가능 (p. 1082)	이제 PostgreSQL용 Amazon RDS에 Amazon RDS 성능 개선 도우미를 사용할 수 있습니다. 자세한 내용은 Amazon RDS 성능 개선 도우미 사용 단원을 참조하십시오.	March 11, 2019
Oracle용 Amazon RDS에서 읽기 전용 복제본 지원 (p. 1082)	Oracle용 Amazon RDS는 활성 데이터 보호를 사용한 읽기 전용 복제본을 지원합니다. 자세한 내용은 읽기 전용 복제본 작업 및 Oracle 읽기 전용 복제본 작업을 참조하십시오.	March 11, 2019
MariaDB 10.2.21 (p. 1082)	이제 MariaDB 버전 10.2.21을 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 MariaDB 버전을 참조하십시오.	March 11, 2019
MySQL 8.0.13 및 5.7.24 (p. 1082)	이제 MySQL 버전 8.0.13 및 5.7.24를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 MySQL 버전을 참조하십시오.	March 8, 2019
SQL Server용 Amazon RDS에 Amazon RDS 성능 개선 도우미 사용 가능 (p. 1082)	이제 SQL Server용 Amazon RDS에 Amazon RDS 성능 개선 도우미를 사용할 수 있습니다. 자세한 내용은 Amazon RDS 성능 개선 도우미 사용 단원을 참조하십시오.	March 4, 2019

Oracle용 Amazon RDS에서 Amazon S3 통합 지원 (p. 1082)	이제 Oracle용 Amazon RDS DB 인스턴스와 버킷 사이에서 파일을 전송할 수 있습니다. 자세한 내용은 Oracle용 Amazon RDS와 Amazon S3 통합을 참조하십시오.	February 26, 2019
MySQL용 Amazon RDS 및 MariaDB용 Amazon RDS에서 db.t3 DB 인스턴스 클래스 지원 (p. 1082)	이제 db.t3 DB 인스턴스 클래스를 사용하는 MySQL 또는 MariaDB를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 DB 인스턴스 클래스 단원 을 참조하십시오.	February 20, 2019
MySQL용 Amazon RDS 및 MariaDB용 Amazon RDS에서 db.r5 DB 인스턴스 클래스 지원 (p. 1082)	이제 db.r5 DB 인스턴스 클래스를 사용하는 MySQL 또는 MariaDB를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 DB 인스턴스 클래스 단원 을 참조하십시오.	February 20, 2019
Amazon RDS MySQL 및 PostgreSQL용 성능 개선 도우미 카운터 (p. 1082)	이제 Amazon RDS MySQL 및 PostgreSQL DB 인스턴스용 성능 개선 도우미 차트에 성능 카운터를 추가할 수 있습니다. 자세한 내용은 성능 개선 도우미 대시보드 구성 요소 를 참조하십시오.	February 19, 2019
이제 PostgreSQL용 Amazon RDS에서 적응형 autovacuum 파라미터 튜닝 지원 (p. 1082)	PostgreSQL용 Amazon RDS를 사용하는 적응형 autovacuum 파라미터 튜닝은 autovacuum 파라미터 값을 자동으로 조정하여 트랜잭션 ID 랙어라운드를 방지합니다. 자세한 내용은 트랜잭션 ID 랙 어라운드 가능성 감소 를 참조하십시오.	February 12, 2019
Oracle용 Amazon RDS에서 Oracle APEX 버전 18.1.v1 및 18.2.v1 지원 (p. 1082)	이제 Oracle용 Amazon RDS에서 Oracle Application Express(APEX) 버전 18.1.v1 및 18.2.v1을 지원합니다. 자세한 내용은 Oracle Application Express 단원을 참조하십시오.	February 11, 2019
Oracle용 Amazon RDS 2019년 1월 RU, RUR 및 PSU (p. 1082)	Oracle용 Amazon RDS는 2019년 1월 릴리스 업데이트(RU) 및 릴리스 업데이트 개정(RUR)을 지원하기 위해 데이터베이스 엔진 버전 12.2.0.1.ru-2019-01.rur-2019-01.r1을 릴리스했습니다. Oracle용 Amazon RDS는 2019년 1월 Oracle Database Patch Set Update(PSU)를 지원하기 위해 데이터베이스 엔진 버전 12.1.0.2.v15 및 11.2.0.4.v19도 릴리스했습니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 단원을 참조하십시오.	February 8, 2019

Amazon RDS 성능 개선 도우미에서 Amazon RDS MySQL에 더 많은 SQL 텍스트 보기기를 지원 (p. 1082)	Amazon RDS 성능 개선 도우미는 이제 Amazon RDS MySQL DB 인스턴스에 대해 성능 개선 도우미 대시보드에서 더 많은 SQL 텍스트를 볼 수 있도록 지원합니다. 자세한 내용은 성능 개선 도우미 대시보드에서 더 많은 SQL 텍스트 보기 단원을 참조하십시오.	February 6, 2019
PostgreSQL용 Amazon RDS가 db.t3 DB 인스턴스 클래스 지원 (p. 1082)	이제 db.t3 DB 인스턴스 클래스를 사용하는 PostgreSQL 실행 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 DB 인스턴스 클래스 단원을 참조하십시오.	January 25, 2019
Oracle용 Amazon RDS에서 db.t3 DB 인스턴스 클래스 지원 (p. 1082)	이제 Oracle 실행 Amazon RDS DB 인스턴스를 생성하여 db.t3 DB 인스턴스 클래스를 사용할 수 있습니다. 자세한 내용은 DB 인스턴스 클래스 단원을 참조하십시오.	January 25, 2019
Amazon RDS 성능 개선 도우미에서 Amazon RDS PostgreSQL에 더 많은 SQL 텍스트 보기기를 지원 (p. 1082)	Amazon RDS 성능 개선 도우미는 이제 Amazon RDS PostgreSQL DB 인스턴스에 대해 성능 개선 도우미 대시보드에서 더 많은 SQL 텍스트를 볼 수 있도록 지원합니다. 자세한 내용은 성능 개선 도우미 대시보드에서 더 많은 SQL 텍스트 보기 단원을 참조하십시오.	January 24, 2019
Oracle용 Amazon RDS에서 SQLT 새 버전 지원 (p. 1082)	이제 Oracle용 Amazon RDS는 SQLT 버전 12.2.180725를 지원합니다. 자세한 내용은 Oracle SQLT 단원을 참조하십시오.	January 22, 2019
PostgreSQL용 Amazon RDS의 새 마이너 버전 지원 (p. 1082)	이제 PostgreSQL용 Amazon RDS는 10.6, 9.6.11, 9.5.15, 9.4.20 및 9.3.25와 같은 새로운 마이너 버전을 지원합니다. 자세한 내용은 PostgreSQL용 Amazon RDS 버전 및 확장 을 참조하십시오.	December 19, 2018
PostgreSQL용 Amazon RDS가 이제 제한된 암호 관리 지원 (p. 1082)	PostgreSQL용 Amazon RDS에서 <code>rds.restrict_password_commands</code> 파라미터와 <code>rds_password</code> 역할을 사용하여 사용자 암호 및 암호 만료 변경을 관리할 수 있는 사람을 제한할 수 있습니다. 자세한 내용은 Restricting Password Management 단원을 참조하십시오.	December 19, 2018

PostgreSQL용 Amazon RDS 가 db.r5 DB 인스턴스 클래스 지원 (p. 1082)	이제 db.r5 DB 인스턴스 클래스를 사용하는 PostgreSQL 실행 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 DB 인스턴스 클래스 단원을 참조하십시오.	December 19, 2018
PostgreSQL용 Amazon RDS 가 Amazon CloudWatch Logs에 데이터베이스 로그 업로드 지원 (p. 1082)	PostgreSQL용 Amazon RDS가 CloudWatch Logs에 데이터베이스 로그를 업로드하는 것을 지원합니다. 자세한 내용은 Publishing PostgreSQL Logs to CloudWatch Logs 단원을 참조하십시오.	December 10, 2018
Oracle용 Amazon RDS가 db.r5 DB 인스턴스 클래스 지원 (p. 1082)	이제 db.r5 DB 인스턴스 클래스를 사용하는 Oracle 실행 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 DB 인스턴스 클래스 단원을 참조하십시오.	November 20, 2018
PostgreSQL용 Amazon RDS가 db.m5 DB 인스턴스 클래스 지원 (p. 1082)	이제 db.m5 DB 인스턴스 클래스를 사용하는 PostgreSQL 실행 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 DB 인스턴스 클래스 단원을 참조하십시오.	November 15, 2018
DB 인스턴스를 삭제할 때 백업 보존 (p. 1082)	Amazon RDS가 DB 인스턴스를 삭제할 때 자동 백업 보존을 지원합니다. 자세한 내용은 백업 작업 을 참조하십시오.	November 15, 2018
Oracle용 Amazon RDS에서 새 미이저 버전 지원 (p. 1082)	이제 Oracle 버전 12.2를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Oracle 12c Version 12.2.0.1 with Amazon RDS 단원을 참조하십시오.	November 13, 2018
Oracle용 Amazon RDS 2018년 10월 PSU (p. 1082)	Oracle용 Amazon RDS는 2018년 10월 Oracle Database Patch Set Update(PSU)를 지원하기 위해 데이터베이스 엔진 버전 12.1.0.2.v14 및 11.2.0.4.v18을 릴리스했습니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 단원을 참조하십시오.	November 13, 2018
SQL Server용 Amazon RDS가 상시 작동 지원 (p. 1082)	SQL Server용 Amazon RDS는 상시 가동 가용성 그룹을 지원합니다. 자세한 내용은 Microsoft SQL Server의 다중 AZ 배포 를 참조하십시오.	November 8, 2018

PostgreSQL용 Amazon RDS가 사용자 지정 DNS 서버를 사용하여 아웃바운드 네트워크 액세스 지원 (p. 1082)	PostgreSQL용 Amazon RDS가 사용자 지정 DNS 서버를 사용하여 아웃바운드 네트워크 액세스를 지원합니다. 자세한 내용은 아웃바운드 네트워크 액세스에 사용자 지정 DNS 서버 사용 을 참조하십시오.	November 8, 2018
MariaDB, MySQL, PostgreSQL용 Amazon RDS가 32TiB 스토리지 지원 (p. 1082)	이제 MySQL, MariaDB, PostgreSQL에 대한 최대 32TiB 스토리지의 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 DB 인스턴스 스토리지를 참조하십시오.	November 7, 2018
Oracle용 Amazon RDS는 확장 데이터 유형을 지원합니다. (p. 1082)	이제 Oracle을 실행하는 Amazon RDS DB 인스턴스에 대해 확장 데이터 유형을 활성화할 수 있습니다. 확장 데이터 유형의 경우 VARCHAR2, NVARCHAR2 및 RAW 데이터 유형의 최대 크기는 32,767바이트입니다. 자세한 내용은 확장 데이터 유형 사용 단원을 참조하십시오.	November 6, 2018
Oracle용 Amazon RDS는 db.m5 DB 인스턴스 클래스를 지원합니다 (p. 1082)	이제 Oracle 실행 Amazon RDS DB 인스턴스를 생성하여 db.m5 DB 인스턴스 클래스를 사용할 수 있습니다. 자세한 내용은 DB 인스턴스 클래스 단원을 참조하십시오.	November 2, 2018
SE, SE1 또는 SE2에서 EE로 Oracle용 Amazon RDS 마이그레이션 (p. 1082)	이제 Oracle 데이터베이스 스탠드 에디션(SE, SE1 또는 SE2)에서 Oracle 데이터베이스 엔터프라이즈 에디션(EE)으로 마이그레이션할 수 있습니다. 자세한 내용은 Oracle 에디션 간 마이그레이션 단원을 참조하십시오.	October 31, 2018
Amazon RDS가 이제 다중 AZ 인스턴스를 중지할 수 있음 (p. 1082)	Amazon RDS가 이제 다중 AZ 배포의 일부인 DB 인스턴스를 중지할 수 있습니다. 이전에는 중지 인스턴스에 다중 AZ 인스턴스에 대한 제한 기능이 있었습니다. 자세한 내용은 Amazon RDS DB 인스턴스 임시 중지 단원을 참조하십시오.	October 29, 2018
Amazon RDS Oracle에 Amazon RDS 성능 개선 도우미를 사용 가능합니다 (p. 1082)	이제 Amazon RDS Oracle에 Amazon RDS 성능 개선 도우미를 사용 가능합니다. 자세한 내용은 Amazon RDS 성능 개선 도우미 사용 단원을 참조하십시오.	October 29, 2018

PostgreSQL용 Amazon RDS의 데이터베이스 미리 보기 환경에서 PostgreSQL 버전 11 지원 (p. 1082)	이제 PostgreSQL용 Amazon RDS가 데이터베이스 미리 보기 환경에서 PostgreSQL 버전 11 지원합니다. 자세한 내용은 데이터베이스 미리 보기 환경에서 Amazon RDS에 대한 PostgreSQL 버전 11 단원을 참조 하십시오.	October 25, 2018
MySQL에서 새로운 메이저 버전 지원 (p. 1082)	이제 MySQL 버전 8.0을 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 MySQL 버전 을 참조하십시오.	October 23, 2018
MariaDB에서 새로운 메이저 버전 지원 (p. 1082)	이제 MariaDB 버전 10.3을 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 MariaDB 버전 을 참조하십시오.	October 23, 2018
Oracle용 Amazon RDS에서 Oracle JVM 지원 (p. 1082)	이제 Oracle용 Amazon RDS는 Oracle Java 가상 머신(JVM) 옵션을 지원합니다. 자세한 내용은 Oracle Java 가상 머신 항목 참조하십시오.	October 16, 2018
복원 및 특정 시점으로 복구를 위한 사용자 지정 파라미터 그룹 (p. 1082)	이제 스냅샷을 복원하거나 특정 시점으로 복구 작업을 수행할 때 사용자 지정 파라미터 그룹을 지정할 수 있습니다. 자세한 내용은 DB 스냅샷으로 복원 및 DB 인스턴스를 지정된 시간으로 복원 단원 을 참조하십시오.	October 15, 2018
Oracle용 Amazon RDS에서 32TiB 스토리지 지원 (p. 1082)	이제 최대 32TiB 스토리지의 Oracle RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 DB 인스턴스 스토리지 를 참조하십시오.	October 15, 2018
MySQL용 Amazon RDS에서 GTID 지원 (p. 1082)	이제 MySQL용 Amazon RDS는 모든 DB 인스턴스 및 복제 구성에서 고유한 전역 트랜잭션 식별자(GTID)를 지원합니다. 자세한 내용은 Amazon RDS MySQL에 대한 GTID 기반 복제 사용 단원을 참조하십시오.	October 10, 2018
MySQL 5.7.23, 5.6.41 및 5.5.61 (p. 1082)	이제 MySQL 버전 5.7.23, 5.6.41 및 5.5.61을 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 MySQL 버전 을 참조하십시오.	October 8, 2018

PostgreSQL용 Amazon RDS의 새 마이너 버전 지원 (p. 1082)	이제 PostgreSQL용 Amazon RDS는 10.5, 9.6.10, 9.5.14, 9.4.19 및 9.3.24와 같은 새로운 마이너 버전을 지원합니다. 자세한 내용은 PostgreSQL용 Amazon RDS 버전 및 확장 을 참조하십시오.	October 4, 2018
Oracle용 Amazon RDS에서 SQLT 새 버전 지원 (p. 1082)	이제 Oracle용 Amazon RDS는 SQLT 버전 12.2.180331을 지원합니다. 자세한 내용은 Oracle SQLT 단원을 참조하십시오.	October 4, 2018
Oracle용 Amazon RDS 2018년 7월 PSU (p. 1082)	Oracle용 Amazon RDS는 2018년 7월 Oracle Database Patch Set Update(PSU)를 지원하기 위해 데이터베이스 엔진 버전 12.1.0.2.v13 및 11.2.0.4.v17을 릴리스했습니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 단원을 참조하십시오.	October 3, 2018
이제 PostgreSQL용 Amazon RDS에서 IAM 인증 지원 (p. 1082)	이제 PostgreSQL용 Amazon RDS는 IAM 인증을 지원합니다. 자세한 사항은 MySQL 및 PostgreSQL에 대한 IAM 데이터베이스 인증 단원을 참조하십시오.	September 27, 2018
Amazon RDS DB 인스턴스에 대한 삭제 방지를 활성화할 수 있습니다. (p. 1082)	DB 인스턴스에 대해 삭제 방지를 활성화하면 모든 사용자가 데이터베이스를 삭제할 수 없습니다. 자세한 내용은 DB 인스턴스 삭제 단원을 참조하십시오.	September 26, 2018
MySQL용 Amazon RDS 및 MariaDB용 Amazon RDS는 db.m5 DB 인스턴스 클래스를 지원합니다. (p. 1082)	이제 db.m5 DB 인스턴스 클래스를 사용하는 MySQL 또는 MariaDB를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 DB 인스턴스 클래스 단원을 참조하십시오.	September 18, 2018
Amazon RDS는 이제 SQL Server 2017에 대한 업그레이드를 지원합니다. (p. 1082)	SQL Server 2008을 제외한 어떤 버전에서든 SQL Server 2017로 기존 DB 인스턴스를 업그레이드 할 수 있습니다. SQL Server 2008에서 업그레이드 하려면 먼저 다음 버전 중 하나로 업그레이드 하십시오. 자세한 사항은 Microsoft SQL 서버 DB 엔진 업그레이드 단원을 참조하십시오.	September 11, 2018

Amazon RDS PostgreSQL, 데이터베이스 미리 보기 환경에서 PostgreSQL 버전 11 베타 3 지원 (p. 1082)	이 릴리스에서는 Write-Ahead Log(WAL) 세그먼트 크기 (<code>wal_segment_size</code>)가 이제 64MB로 설정됩니다. PostgreSQL 버전 11 Beta 3에 대한 자세한 내용은 PostgreSQL 11 Beta 3 릴리스 를 참조하십시오. 데이터베이스 미리 보기 환경에 대한 자세한 내용은 데이터베이스 미리 보기 환경 작업 을 참조하십시오.	September 7, 2018
Amazon Aurora 사용 설명서 (p. 1082)	Amazon Aurora 사용 설명서는 모든 Amazon Aurora 개념을 설명하고 콘솔과 명령줄 인터페이스로 다양한 기능을 사용하는 방법에 대한 지침을 제공합니다. Amazon RDS 사용 설명서는 이제 Aurora 이외의 다른 데이터베이스 엔진에 적용됩니다.	August 31, 2018
Amazon RDS MySQL에 Amazon RDS 성능 개선 도우미 사용 가능 (p. 1082)	이제 Amazon RDS MySQL에 Amazon RDS 성능 개선 도우미를 사용할 수 있습니다. 자세한 내용은 Amazon RDS 성능 개선 도우미 사용 단원을 참조하십시오.	August 28, 2018
이제 PostgreSQL과 호환되는 Aurora는 Aurora Auto Scaling을 지원합니다. (p. 1082)	이제 PostgreSQL과 호환되는 Aurora에 Aurora 복제본의 Auto Scaling을 사용할 수 있습니다. 자세한 내용은 Aurora 복제본에 Amazon Aurora Auto Scaling 사용 을 참조하십시오.	August 16, 2018
Aurora MySQL용 Aurora Serverless (p. 1082)	Aurora Serverless는 Amazon Aurora에 대한 온디맨드 방식의 Auto Scaling 구성입니다. 자세한 내용은 Amazon Aurora Serverless 사용 을 참조하십시오.	August 9, 2018
MySQL용 Amazon RDS에서 지역 복제 지원 (p. 1082)	이제 MySQL용 Amazon RDS에서 지역 복제를 통해 복구를 위한 전략으로 지원합니다. 자세한 내용은 MySQL을 사용한 지역 복제 구성 을 참조하십시오.	August 6, 2018
MySQL 5.7.22 및 5.6.40 (p. 1082)	이제 MySQL 버전 5.7.22 및 5.6.40를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 MySQL 버전 을 참조하십시오.	August 6, 2018
Aurora는 이제 중국(닝샤) 지역에서 사용할 수 있습니다 (p. 1082)	이제 Aurora MySQL 및 Aurora PostgreSQL을 중국(닝샤) 리전에서 사용할 수 있습니다. 자세한 내용은 Amazon Aurora MySQL 가용성 및 Amazon Aurora PostgreSQL 가용성을 참조하십시오.	August 6, 2018

Aurora MySQL에 Amazon RDS 성능 개선 도우미 사용 가능 (p. 1082)	이제 Aurora MySQL에 Amazon RDS 성능 개선 도우미를 사용 할 수 있습니다. 자세한 내용은 Amazon RDS 성능 개선 도우미 사용 을 참조하십시오.	August 6, 2018
Amazon CloudWatch와 Amazon RDS 성능 개선 도우미 통합 (p. 1082)	Amazon RDS 성능 개선 도우미는 지표를 Amazon CloudWatch에 자동으로 게시합니다. 자세한 내용은 CloudWatch에 게시되는 성능 개선 도우미 지표 단원을 참조하십시오.	August 6, 2018
PostgreSQL용 Amazon RDS의 새 마이너 버전 지원 (p. 1082)	PostgreSQL용 Amazon RDS 은 10.4, 9.6.9, 9.5.13, 9.4.18 및 9.3.23과 같은 새로운 마이너 버전을 지원합니다. 자세한 내용은 PostgreSQL용 Amazon RDS 버전 및 확장 을 참조하십시오.	July 25, 2018
Amazon RDS 권장 사항 (p. 1082)	Amazon RDS에서 이제 데이터베이스 리소스에 대한 자동 권장 사항을 제공합니다. 자세한 내용은 Amazon RDS 권장 사항 사용 을 참조하십시오.	July 25, 2018
AWS 리전에 대한 충분 스냅샷 복사본 (p. 1082)	Amazon RDS에서 비암호화 및 암호화된 인스턴스 모두에 대해 AWS 리전에 대한 충분 스냅샷 복사본을 지원합니다. 자세한 내용은 AWS 리전 간 스냅샷 복사 단원을 참조하십시오.	July 24, 2018
PostgreSQL용 Amazon RDS에 Amazon RDS 성능 개선 도우미 사용 가능 (p. 1082)	이제 PostgreSQL용 Amazon RDS에 Amazon RDS 성능 개선 도우미를 사용할 수 있습니다. 자세한 내용은 Amazon RDS 성능 개선 도우미 사용 단원을 참조하십시오.	July 18, 2018
Amazon RDS for Oracle에서 Oracle APEX 버전 5.1.4.v1 지원 (p. 1082)	Amazon RDS for Oracle에서 이제 Oracle Application Express(APEX) 버전 5.1.4.v1을 지원합니다. 자세한 내용은 Oracle Application Express 단원을 참조하십시오.	July 10, 2018
Oracle용 Amazon RDS, Amazon CloudWatch Logs에 로그 게시 지원 (p. 1082)	이제 Oracle용 Amazon RDS는 CloudWatch Logs의 로그 그룹에 알림, 감사, 추적 및 리스너 로그 데이터 게시를 지원합니다. 자세한 내용은 Amazon CloudWatch Logs에 Oracle 로그 게시 단원을 참조하십시오.	July 9, 2018

MariaDB 10.2.15, 10.1.34 및 10.0.35 (p. 1082)	이제 MariaDB 버전 10.2.15, 10.1.34 및 10.0.35를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 MariaDB 버전 을 참조하십시오.	July 5, 2018
Aurora PostgreSQL 1.2를 사용할 수 있으며 PostgreSQL 9.6.8과 호환됨 (p. 1082)	이제 Aurora PostgreSQL 1.2를 사용할 수 있으며 PostgreSQL 9.6.8과 호환됩니다. 자세한 내용은 버전 1.2 단원 을 참조하십시오.	June 27, 2018
Amazon RDS PostgreSQL용 읽기 전용 복제본에서 다중 AZ 배포 지원 (p. 1082)	Amazon RDS PostgreSQL의 RDS 읽기 전용 복제본에서 이제 다중 가용 영역이 지원됩니다. 자세한 내용은 PostgreSQL 읽기 전용 복제본 작업 을 참조하십시오.	June 25, 2018
Aurora PostgreSQL에 성능 개선 도우미 사용 가능 (p. 1082)	성능 개선 도우미를 Aurora PostgreSQL에 일반적으로 사용할 수 있으며, 성능 데이터의 보존 연장이 지원됩니다. 자세한 내용은 Amazon RDS 성능 개선 도우미 사용 을 참조하십시오.	June 21, 2018
미국 서부(캘리포니아 북부) 리전에서 Aurora PostgreSQL 사용 가능 (p. 1082)	이제 미국 서부(캘리포니아 북부) 리전에서 Aurora PostgreSQL을 사용할 수 있습니다. 자세한 내용은 Amazon Aurora PostgreSQL의 가용성을 참조하십시오.	June 11, 2018
Amazon RDS for Oracle에서 이제 CPU 구성 지원 (p. 1082)	Amazon RDS for Oracle은 DB 인스턴스 클래스의 프로세스에 대한 CPU 코어 수 및 각 코어의 스레드 수 구성을 지원합니다. 자세한 내용은 DB 인스턴스 클래스의 프로세스 구성 을 참조하십시오.	June 5, 2018
Oracle용 Amazon RDS 2018년 4월 PSU (p. 1082)	Oracle용 Amazon RDS가 2018년 4월 Oracle Database Patch Set Update(PSU)를 지원하기 위해 데이터베이스 엔진 버전 12.1.0.2.v12 및 11.2.0.4.v16를 릴리스했습니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 단원을 참조하십시오.	June 1, 2018

이전 업데이트

다음 표에서는 2018년 6월 이전 Amazon RDS 사용 설명서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다.

변경 사항	설명	변경 날짜
Amazon RDS PostgreSQL, 데이터베이스 미리 보기 환경에서 PostgreSQL 버전 11 베타 1 지원	<p>PostgreSQL 버전 11 베타 1에는 여러 개선 사항이 포함되어 있으며 관련 설명은 PostgreSQL 11 베타 1 출시!에 나와 있습니다.</p> <p>데이터베이스 미리 보기 환경에 대한 자세한 내용은 데이터베이스 미리 보기 환경 작업 (p. 1011) 단원을 참조하십시오.</p>	2018년 5월 31일
Oracle용 Amazon RDS 이제 TLS 버전 1.0 및 1.2 지원	Oracle용 Amazon RDS는 TLS(전송 계층 보안) 버전 1.0 및 1.2를 지원합니다. 자세한 내용은 Oracle SSL 옵션에 대한 TLS 버전 (p. 810) 단원을 참조하십시오.	2018년 5월 30일
Aurora MySQL, Amazon CloudWatch Logs에 로그 게시 지원	이제 Aurora MySQL은 CloudWatch Logs의 로그 그룹에 일반, 느린, 감사, 오류 로그 데이터 게시를 지원합니다. 자세한 내용은 CloudWatch Logs에 Aurora MySQL 게시 단원을 참조하십시오.	2018년 5월 23일
Amazon RDS PostgreSQL의 데이터베이스 미리 보기 환경	이제 Amazon RDS PostgreSQL의 새 인스턴스를 미리 보기 모드로 시작할 수 있습니다. 데이터베이스 미리 보기 환경에 대한 자세한 내용은 데이터베이스 미리 보기 환경 작업 (p. 1011) 단원을 참조하십시오.	2018년 5월 22일
Amazon RDS Oracle DB 인스턴스, 새로운 DB 인스턴스 클래스 지원	Oracle DB 인스턴스가 이제 db.x1e 및 db.x1 DB 인스턴스 클래스를 지원합니다. 자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 및 Oracle을 위한 DB 인스턴스 클래스 지원 (p. 705) 단원을 참조하십시오.	2018년 5월 22일
이제 Amazon RDS PostgreSQL이 읽기 전용 복제본에서 postgres_fdw를 지원합니다.	postgres_fdw를 사용하여 읽기 전용 복제본에서 원격 서버로 연결할 수 있습니다. 자세한 내용은 postgres_fdw 확장으로 외부 데이터 액세스 (p. 1009) 단원을 참조하십시오.	2018년 5월 17일
오라클용 Amazon RDS 이제 sqlnet.ora 파라미터 설정 지원	이제 오라클용 Amazon RDS로 sqlnet.ora 파라미터를 설정할 수 있습니다. 자세한 내용은 Oracle sqlnet.ora 파라미터 수정 (p. 750) 단원을 참조하십시오.	2018년 5월 10일
아시아 태평양(서울) 리전에서 Aurora PostgreSQL을 이용할 수 있습니다.	Aurora PostgreSQL은 이제 아시아 태평양(서울) 리전에서 사용할 수 있습니다. 자세한 내용은 Amazon Aurora PostgreSQL 가용성 을 참조하십시오.	2018년 5월 9일
Aurora MySQL은 역추적을 지원	이제 Aurora MySQL은 백업에서 데이터를 복구하지 않고도 특정 시간으로 DB 클러스터 "되감기"를 지원합니다. 자세한 내용은 Amazon Aurora DB 클러스터 역추적 을 참조하십시오.	2018년 5월 9일
Aurora MySQL, 외부 MySQL에서 암호화된 마이그레이션 및 복제 지원	이제 Aurora MySQL은 외부 MySQL 데이터베이스에서 암호화된 마이그레이션 및 복제를 지원합니다. 자세한 내용은 외부 MySQL 데이터베이스에서 Amazon Aurora MySQL DB 클러스터로 데이터 마이그레이션 및 Aurora 및 MySQL 간 또는 Aurora 및 다른 Aurora DB 클러스터 간 복제 단원을 참조하십시오.	2018년 5월 4일
Copy-on-Write 프로토콜에 대한	이제 Aurora PostgreSQL 데이터베이스 클러스터의 데이터베이스를 복제할 수 있습니다. 자세한 내용은 Aurora DB 클러스터에서 데이터베이스 복제 단원을 참조하십시오.	2018년 10월 4일

변경 사항	설명	변경 날짜
PostgreSQL과 호환되는 Aurora 지원		
MariaDB 10.2.12, 10.1.31 및 10.0.34	이제 MariaDB 버전 10.2.12, 10.1.31 및 10.0.34를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS MariaDB 버전 (p. 428) 단원을 참조하십시오.	2018년 3월 21일
새 리전을 위한 Aurora PostgreSQL 지원	이제 EU(런던) 및 아시아 태평양(싱가포르) 리전에서 Aurora PostgreSQL을 사용할 수 있습니다. 자세한 내용은 Amazon Aurora PostgreSQL 가용성을 참조하십시오.	2018년 3월 13일
MySQL 5.7.21, 5.6.39 및 5.5.59	이제 MySQL 버전 5.7.21, 5.6.39 및 5.5.59를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 MySQL 버전 (p. 578) 단원을 참조하십시오.	2018년 3월 9일
Oracle용 Amazon RDS는 이제 Oracle REST Data Services 지원	Oracle용 Amazon RDS는 APEX 옵션의 일부로 Oracle REST Data Services를 지원합니다. 자세한 내용은 Oracle Application Express (p. 783) 단원을 참조하십시오.	2018년 3월 9일
새로운 AWS 리전에서 MySQL과 호환되는 Amazon Aurora 사용 가능	이제 아시아 태평양(싱가포르) 리전에서 Aurora MySQL을 사용할 수 있습니다. Aurora MySQL을 사용할 수 있는 AWS 리전의 전체 목록은 Amazon Aurora MySQL 가용성을 참조하십시오.	2018년 3월 6일
PostgreSQL 10.1 지원	이제 Amazon RDS는 PostgreSQL 버전 10.1을 지원합니다. 자세한 내용은 Amazon RDS의 PostgreSQL 버전 10.1 (p. 1019) 단원을 참조하십시오.	2018년 2월 27일
Oracle 2018년 1월 PSU	Oracle용 Amazon RDS는 2018년 1월 Oracle Database Patch Set Update(PSU)를 지원하기 위해 데이터베이스 엔진 버전 12.1.0.2.v11 및 11.2.0.4.v15를 릴리스했습니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 (p. 902) 단원을 참조하십시오.	2018년 2월 22일
Microsoft SQL Server를 실행하는 Amazon RDS DB 인스턴스는 CDC(변경 데이터 캡처)를 지원	이제 Microsoft SQL Server용 Amazon RDS를 실행하는 DB 인스턴스는 CDC(변경 데이터 캡처)를 지원합니다. 자세한 내용은 Microsoft SQL Server DB 인스턴스에 대한 변경 데이터 캡처 지원 (p. 488) 단원을 참조하십시오.	2018년 2월 6일
Aurora MySQL, 새로운 메이저 버전 지원	이제 MySQL 버전 5.7을 실행하는 Aurora MySQL DB 클러스터를 생성할 수 있습니다. 자세한 내용은 Amazon Aurora MySQL 데이터베이스 엔진 업데이트 2018-02-06 을 참조하십시오.	2018년 2월 6일
PostgreSQL 9.6.6 지원	이제 Amazon RDS PostgreSQL은 버전 9.6.6을 지원합니다. 이 릴리스에는 접두사 및 orafce 확장에 대한 지원도 포함됩니다. 자세한 내용은 Amazon RDS의 PostgreSQL 버전 9.6.6 (p. 1021) 단원을 참조하십시오.	2018년 1월 19일
Amazon CloudWatch Logs에 MySQL 및 MariaDB 로그 게시	이제 CloudWatch Logs에 MySQL 및 MariaDB 로그 데이터를 게시할 수 있습니다. 자세한 내용은 CloudWatch Logs에 MySQL 로그 게시 (p. 320) 및 Amazon CloudWatch Logs에 MariaDB 로그 게시 (p. 311) 단원을 참조하십시오.	2018년 1월 17일

변경 사항	설명	변경 날짜
읽기 전용 복제본에 대한 다중 AZ 지원	이제 읽기 전용 복제본을 다중 AZ DB 인스턴스로 생성할 수 있습니다. Amazon RDS는 복제본에 대한 장애 조치 지원을 위해 다른 가용 영역에 복제본의 대기를 생성합니다. 읽기 전용 복제본을 다중 AZ DB 인스턴스로 생성하는 작업은 원본 데이터베이스가 다중 AZ DB 인스턴스인지 여부와는 독립적입니다. 자세한 내용은 읽기 전용 복제본 작업 (p. 135) 단원을 참조하십시오.	2018년 1월 11일
MariaDB용 Amazon RDS는 새로운 메이저 버전 지원	이제 MariaDB 버전 10.2를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS에 서의 MariaDB 10.2 지원 (p. 429) 단원을 참조하십시오.	2018년 1월 3일
새로운 AWS 리전에서 PostgreSQL과 호환되는 Amazon Aurora 사용 가능	이제 EU(파리) 리전에서 Aurora PostgreSQL을 사용할 수 있습니다. Aurora PostgreSQL을 사용할 수 있는 AWS 리전의 전체 목록은 Amazon Aurora PostgreSQL 가용성 을 참조하십시오.	2017년 12월 22일
Aurora PostgreSQL은 이제 새 인스턴스 유형을 지원	이제 Aurora PostgreSQL은 새 인스턴스 유형을 지원합니다. 전체 인스턴스 유형 목록은 DB 인스턴스 클래스 선택 을 참조하십시오.	2017년 12월 20일
Oracle 2017년 10월 PSU	Oracle용 Amazon RDS는 2017년 10월 Oracle Database Patch Set Update(PSU)를 지원하기 위해 데이터베이스 엔진 버전 12.1.0.2.v10 및 11.2.0.4.v14를 릴리스했습니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 (p. 902) 단원을 참조하십시오.	2017년 12월 19일
새로운 AWS 리전에서 MySQL과 호환되는 Amazon Aurora 사용 가능	이제 EU(파리) 리전에서 Aurora MySQL을 사용할 수 있습니다. Aurora MySQL을 사용할 수 있는 AWS 리전의 전체 목록은 Amazon Aurora MySQL 가용성 을 참조하십시오.	2017년 12월 18일
Aurora MySQL은 해시 조인을 지원	이 기능은 동등 조인을 사용하여 많은 양의 데이터를 조인해야 하는 경우 쿼리 성능을 향상 시킬 수 있습니다. 자세한 내용은 Aurora MySQL에서 해시 조인 작업 을 참조하십시오.	2017년 12월 11일
Aurora MySQL은 AWS Lambda 함수를 호출하는 네이티브 함수를 지원합니다.	Aurora MySQL을 사용할 때 <code>lambda_sync</code> 및 <code>lambda_async</code> 네이티브 함수를 호출할 수 있습니다. 자세한 내용은 Amazon Aurora MySQL DB 클러스터에서 Lambda 함수 호출 을 참조하십시오.	2017년 12월 11일
Aurora PostgreSQL HIPAA 규정 준수 추가	Aurora PostgreSQL은 HIPAA 준수 애플리케이션 구축을 지원합니다. 자세한 내용은 Amazon Aurora PostgreSQL 작업 을 참조하십시오.	2017년 12월 6일
추가 AWS 리전에서 PostgreSQL 호환성을 갖춘 Amazon Aurora 사용 가능	이제 4개의 새로운 AWS 리전에서 PostgreSQL 호환성을 갖춘 Amazon Aurora를 사용할 수 있습니다. 자세한 내용은 Amazon Aurora PostgreSQL 가용성 을 참조하십시오.	2017년 11월 22일
Microsoft SQL Server를 실행하는 Amazon RDS DB 인스턴스의 스토리지 수정	이제 SQL Server를 실행하는 Amazon RDS DB 인스턴스의 스토리지를 수정할 수 있습니다. 자세한 내용은 Microsoft SQL Server 데이터베이스 엔진 기반 DB 인스턴스의 수정 (p. 509) 단원을 참조하십시오.	2017년 11월 21일

변경 사항	설명	변경 날짜
Amazon RDS는 Linux 기반 엔진의 16TiB 스토리지 지원	이제 MySQL, MariaDB, PostgreSQL 및 Oracle RDS DB 인스턴스는 스토리지에서 최대 16TiB까지 생성할 수 있습니다. 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.	2017년 11월 21일
Amazon RDS는 신속한 스토리지 확장을 지원	이제 몇 분 안에 MySQL, MariaDB, PostgreSQL 및 Oracle RDS DB 인스턴스에 스토리지를 추가할 수 있습니다. 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.	2017년 11월 21일
Amazon RDS는 MariaDB 버전 10.1.26 및 10.0.32를 지원	이제 MariaDB 버전 10.1.26 및 10.0.32를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS MariaDB 버전 (p. 428) 단원을 참조하십시오.	2017년 11월 20일
이제 Microsoft SQL Server용 Amazon RDS는 새로운 DB 인스턴스 클래스를 지원	이제 db.r4 및 db.m4.16xlarge DB 인스턴스 클래스를 사용하는 SQL Server를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Microsoft SQL Server를 위한 DB 인스턴스 클래스 지원 (p. 481) 단원을 참조하십시오.	2017년 11월 20일
이제 MySQL 및 MariaDB용 Amazon RDS는 새로운 DB 인스턴스 클래스를 지원	이제 db.r4, db.m4.16xlarge, db.t2.xlarge 및 db.t2.2xlarge DB 인스턴스 클래스를 사용하는 MySQL 및 MariaDB를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 단원을 참조하십시오.	2017년 11월 20일
SQL Server 2017	이제 Microsoft SQL Server 2017을 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. SQL Server 2016 SP1 CU5를 실행하여 DB 인스턴스를 생성할 수도 있습니다. 자세한 내용은 Amazon RDS의 Microsoft SQL Server (p. 477) 단원을 참조하십시오.	2017년 11월 17일
Amazon S3에서 MySQL 백업 복원	이제 온프레미스 데이터베이스의 백업을 생성하여 Amazon S3에 저장한 다음 MySQL을 실행하는 새로운 Amazon RDS DB 인스턴스에 백업 파일을 복원할 수 있습니다. 자세한 내용은 Amazon RDS MySQL DB 인스턴스로 백업 복원 (p. 619) 단원을 참조하십시오.	2017년 11월 17일
Aurora 복제본을 통한 Auto Scaling	이제 Amazon Aurora MySQL은 Aurora Auto Scaling을 지원합니다. Aurora Auto Scaling은 연결이나 워크로드 증가 또는 감소에 따라 Aurora 복제본의 수를 동적으로 조정합니다. 자세한 내용은 Aurora 복제본에 Amazon Aurora Auto Scaling 사용 을 참조하십시오.	2017년 11월 17일
Oracle 기본 에디션 지원	이제 Oracle DB 인스턴스용 Amazon RDS는 DB 인스턴스의 기본 에디션 설정을 지원합니다. 자세한 내용은 DB 인스턴스의 기본 에디션 설정 (p. 850) 단원을 참조하십시오.	2017년 11월 3일
Oracle DB 인스턴스 파일 확인	이제 Oracle DB 인스턴스용 Amazon RDS는 Oracle Recovery Manager(RMAN)의 논리적 확인 유ти리티를 사용하여 DB 인스턴스 파일을 확인하는 작업을 지원합니다. 자세한 내용은 DB 인스턴스 파일 확인 (p. 862) 단원을 참조하십시오.	2017년 11월 3일

변경 사항	설명	변경 날짜
Oracle 2017년 7월 PSU	Oracle용 Amazon RDS는 2017년 7월 Oracle Database Patch Set Update(PSU)를 지원하기 위해 데이터베이스 엔진 버전 12.1.0.2.v9 및 11.2.0.4.v13을 릴리스했습니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 (p. 902) 단원을 참조하십시오.	2017년 11월 3일
OEM 13c용 Management Agent	이제 Amazon RDS Oracle DB 인스턴스는 Oracle Enterprise Manager(OEM) Cloud Control 13c용 Management Agent를 지원합니다. 자세한 내용은 Oracle Management Agent for Enterprise Manager Cloud Control (p. 793) 단원을 참조하십시오.	2017년 11월 1일
PostgreSQL 9.6.5, 9.5.9, 9.4.14 및 9.3.19	이제 PostgreSQL 버전 9.6.5., 9.5.9, 9.4.14 및 9.3.19를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 지원되는 PostgreSQL 데이터베이스 버전 (p. 1014) 단원을 참조하십시오.	2017년 11월 1일
Microsoft SQL Server 스냅샷의 스토리지 재구성	이제 Microsoft SQL Server를 실행하는 Amazon RDS DB 인스턴스로 스냅샷을 복원할 때 스토리지를 재구성할 수 있습니다. 자세한 내용은 DB 스냅샷에서 복원 (p. 205) 단원을 참조하십시오.	2017년 10월 26일
MySQL과 호환되는 Aurora의 비동기식 키 미리 가져오기	비동기식 키 미리 가져오기(AKP)는 필요하기 전에 메모리 키를 미리 가져와서 캐싱되지 않은 인덱스 조인 성능을 향상 시킵니다. 자세한 내용은 Amazon Aurora에서 비동기식 키 미리 가져오기 작업 을 참조하십시오.	2017년 10월 26일
MySQL 5.7.19, 5.6.37 및 5.5.57	이제 MySQL 버전 5.7.19, 5.6.37 및 5.5.57를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS의 MySQL 버전 (p. 578) 단원을 참조하십시오.	2017년 10월 25일
Amazon Aurora PostgreSQL 호환 에디션 정식 출시	Amazon Aurora PostgreSQL 호환 에디션은 새로 배포하는 PostgreSQL이든, 혹은 기존에 배포한 PostgreSQL이든 상관없이 설치, 조작 및 조정이 간편하고 비용 효율적이기 때문에 비즈니스와 애플리케이션에 더욱 많은 시간을 투자할 수 있습니다. 자세한 내용은 Amazon Aurora PostgreSQL 작업 을 참조하십시오.	2017년 10월 24일
Amazon RDS Oracle DB 인스턴스, 새로운 DB 인스턴스 클래스 지원	이제 Amazon RDS Oracle DB 인스턴스는 메모리에 최적화된 차세대(db.r4) 인스턴스 클래스를 지원합니다. 이제 Amazon RDS Oracle DB 인스턴스는 db.m4.16xlarge, db.t2.xlarge, db.t2.2xlarge 등 새로운 현재 세대의 인스턴스 클래스도 지원합니다. 자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 및 Oracle을 위한 DB 인스턴스 클래스 지원 (p. 705) 단원을 참조하십시오.	2017년 10월 23일

변경 사항	설명	변경 날짜
새로운 기능	<p>새로운 예약 인스턴스와 기존 예약 인스턴스 모두 이제 동일한 DB 인스턴스 클래스에서 다양한 크기를 지원할 수 있게 되었습니다. 동일한 AWS 리전, 데이터베이스 엔진 및 인스턴스 패밀리를 비롯해 AZ 구성 간에서도 유연한 크기의 예약 인스턴스를 DB 인스턴스에 사용할 수 있습니다. 이렇게 유연한 크기의 예약 인스턴스를 지원하는 데이터베이스 엔진으로는 Amazon Aurora, MariaDB, MySQL, Oracle(Bring Your Own License), PostgreSQL이 있습니다. 자세한 내용은 유연한 크기의 예약 DB 인스턴스 (p. 185) 단원을 참조하십시오.</p>	2017년 10월 11일
새로운 기능	<p>이제 최적의 성능을 얻기 위해 Oracle SQLT 옵션을 사용하여 SQL 문을 튜닝할 수 있습니다. 자세한 내용은 Oracle SQLT (p. 817) 단원을 참조하십시오.</p>	2017년 9월 22일
새로운 기능	<p>Amazon RDS Oracle DB 인스턴스의 기존 수동 DB 스냅샷이 있는 경우 이제 Oracle 데이터베이스 엔진의 최신 버전으로 스냅샷을 업그레이드 할 수 있습니다. 자세한 내용은 Oracle DB 스냅샷 업그레이드 (p. 756) 단원을 참조하십시오.</p>	2017년 9월 20일
새로운 기능	<p>이제 Oracle Spatial을 사용하여 Oracle을 실행하는 Amazon RDS DB 인스턴스의 공간 데이터를 저장, 검색, 업데이트 및 쿼리할 수 있습니다. 자세한 내용은 Oracle Spatial (p. 815) 단원을 참조하십시오.</p>	2017년 9월 15일
새로운 기능	<p>이제 Oracle Locator를 사용하여 Oracle을 실행하는 Amazon RDS DB 인스턴스와 함께 인터넷 및 무선 서비스 기반 애플리케이션과 파트너 기반 GIS 솔루션을 지원할 수 있습니다. 자세한 내용은 Oracle Locator (p. 803) 단원을 참조하십시오.</p>	2017년 9월 15일
새로운 기능	<p>이제 Oracle Multimedia를 사용하여 Oracle을 실행하는 Amazon RDS DB 인스턴스의 이미지, 오디오, 동영상 및 기타 이종 미디어 데이터를 저장하고 관리하며 검색할 수 있습니다. 자세한 내용은 Oracle Multimedia (p. 805) 단원을 참조하십시오.</p>	2017년 9월 15일
새로운 기능	<p>이제 감사 로그를 Amazon Aurora MySQL DB 클러스터에서 Amazon CloudWatch Logs로 내보낼 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs에 Aurora MySQL 게시 단원을 참조하십시오.</p>	2017년 9월 14일
새로운 기능	<p>이제 Amazon RDS는 Oracle을 실행하는 DB 인스턴스에 대한 Oracle Application Express(APEX)의 여러 버전을 지원합니다. 자세한 내용은 Oracle Application Express (p. 783) 단원을 참조하십시오.</p>	2017년 9월 13일
새로운 기능	<p>이제 Amazon Aurora를 사용하여 암호화되지 않거나 암호화된 DB 스냅샷 또는 Amazon RDS MySQL DB 인스턴스를 암호화된 Aurora MySQL DB 클러스터로 마이그레이션 할 수 있습니다. 자세한 내용은 Aurora로 RDS MySQL 스냅샷 마이그레이션 및 Aurora 읽기 전용 복제본을 사용하여 MySQL DB 인스턴스에서 Amazon Aurora MySQL DB 클러스터로 데이터 마이그레이션 단원을 참조하십시오.</p>	2017년 9월 5일

변경 사항	설명	변경 날짜
새로운 기능	Microsoft SQL Server 데이터베이스용 Amazon RDS를 사용하여 HIPAA 인증 애플리케이션을 개발할 수 있습니다. 자세한 내용은 Microsoft SQL Server DB 인스턴스에 대한 규정 준수 프로그램 지원 (p. 482) 단원을 참조하십시오.	2017년 8월 31일
새로운 기능	이제 MariaDB 데이터베이스용 Amazon RDS를 사용하여 HIPAA 인증 애플리케이션을 개발할 수 있습니다. 자세한 내용은 Amazon RDS MariaDB (p. 426) 단원을 참조하십시오.	2017년 8월 31일
새로운 기능	이제 할당된 스토리지를 최대 16TiB까지 설정하고 프로비저닝된 IOPS를 1:1-50:1의 스토리지 범위로 설정하여 Microsoft SQL Server를 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS DB 인스턴스 스토리지 (p. 101) 단원을 참조하십시오.	2017년 8월 22일
새로운 기능	이제는 EU(프랑크푸르트) 리전에서 Microsoft SQL Server 기반 DB 인스턴스에 다중 AZ 배포를 사용할 수 있습니다. 자세한 내용은 Microsoft SQL Server의 다중 AZ 배포 (p. 537) 단원을 참조하십시오.	2017년 8월 3일
새로운 기능	이제 MariaDB 버전 10.1.23 및 10.0.31을 실행하는 Amazon RDS DB 인스턴스를 생성할 수 있습니다. 자세한 내용은 Amazon RDS MariaDB 버전 (p. 428) 단원을 참조하십시오.	2017년 7월 17일
새로운 기능	이제 Amazon RDS는 모든 AWS 리전의 라이선스 포함 서비스 모델에서 Microsoft SQL Server Enterprise Edition을 지원합니다. 자세한 내용은 Amazon RDS의 Microsoft SQL Server 라이선싱 (p. 493) 단원을 참조하십시오.	2017년 7월 13일
새로운 기능	이제 Oracle용 Amazon RDS는 데이터베이스 확장성을 향상하기 위해 Linux 커널 방대한 페이지를 지원합니다. 방대한 페이지를 사용하면 페이지 표가 작아지고 메모리 관리에 사용되는 CPU 시간이 줄어 대용량 데이터베이스 인스턴스의 성능이 높아집니다. Oracle 버전 12.1.0.2 및 11.2.0.4의 모든 에디션을 실행하는 Amazon RDS DB 인스턴스에서 방대한 페이지를 사용할 수 있습니다. 자세한 내용은 Oracle DB 인스턴스에 방대한 페이지 사용 (p. 721) 단원을 참조하십시오.	2017년 7월 7일
새로운 기능	Aurora가 아닌 모든 DB 엔진에서 db.t2.small 및 db.t2.medium DB 인스턴스 클래스에 유휴 시 암호화 (EAR)를 지원하도록 업데이트되었습니다. 자세한 내용은 Amazon RDS 암호화 가용성 (p. 387) 단원을 참조하십시오.	2017년 27월 6일
새로운 기능	EU(프랑크푸르트) 리전에서 Amazon Aurora를 지원하도록 업데이트되었습니다. 자세한 내용은 Amazon Aurora MySQL 가용성 을 참조하십시오.	2017년 6월 16일
새로운 기능	이제 AWS 리전 간에 DB 스냅샷을 복사할 때 옵션 그룹을 지정할 수 있습니다. 자세한 내용은 옵션 그룹 고려 사항 (p. 209) 단원을 참조하십시오.	2017년 6월 12일

변경 사항	설명	변경 날짜
새로운 기능	이제 특수 DB 인스턴스에서 만들어진 DB 스냅샷을 AWS 리전 간에 복사할 수 있습니다. Oracle TDE, Microsoft SQL Server TDE 및 미러링을 포함한 Microsoft SQL Server 다중 AZ를 사용하는 DB 인스턴스에서 스냅샷을 복사할 수 있습니다. 자세한 내용은 DB 스냅샷 복사 (p. 209) 단원을 참조하십시오.	2017년 6월 12일
새로운 기능	이제 Amazon Aurora를 사용하여 Amazon Aurora DB 클러스터의 모든 데이터베이스를 빠르고 비용 효과적으로 복사할 수 있습니다. 자세한 내용은 Aurora DB 클러스터에서 데이터베이스 복제 단원을 참조하십시오.	2017년 6월 12일
새로운 기능	이제 Amazon RDS는 Microsoft SQL Server 2016 SP1 CU2를 지원합니다. 자세한 내용은 Amazon RDS의 Microsoft SQL Server (p. 477) 단원을 참조하십시오.	2017년 6월 7일
새로운 기능	Oracle용 Amazon RDS는 2017년 4월 Oracle Database Patch Set Update(PSU)를 지원하기 위해 데이터베이스 엔진 버전 12.1.0.2.v8 및 11.2.0.4.v12를 릴리스했습니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 (p. 902) 단원을 참조하십시오.	2017년 5월 23일
새로운 기능	이제 Amazon RDS는 PostgreSQL 버전 9.6.2, 9.5.6, 9.4.11 및 9.3.16를 지원합니다. 자세한 내용은 지원되는 PostgreSQL 데이터베이스 버전 (p. 1014) 단원을 참조하십시오.	2017년 5월 3일
평가판	Amazon Aurora의 PostgreSQL 호환성 미리 보기. 자세한 내용은 Amazon Aurora PostgreSQL 작업 을 참조하십시오.	2017년 4월 19일
새로운 기능	이제 Amazon Aurora를 사용하여 ALTER TABLE tbl_name ADD COLUMN col_name column_definition 작업을 거의 동시에 실행할 수 있습니다. 이 작업은 테이블을 복사하거나 다른 DML 명령문에 영향을 거의 주지 않고 완료됩니다. 자세한 내용은 빠른 DDL을 사용하여 Amazon Aurora에서 테이블 변경 을 참조하십시오.	2017년 5월 4일
새로운 기능	새 모니터링 명령 SHOW VOLUME STATUS를 추가하여 볼륨의 노드 및 디스크 수를 표시합니다. 자세한 내용은 Aurora DB 클러스터에서 볼륨 상태 표시 단원을 참조하십시오.	2017년 5월 4일
새로운 기능	Oracle용 Amazon RDS에는 이제 2017년 1월 Oracle Database PSU(Patch Set Update)가 포함됩니다. 이로써 데이터베이스 엔진 버전 12.1.0.2.v7 및 11.2.0.4.v11도 지원하게 되었습니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 (p. 902) 단원을 참조하십시오.	2017년 3월 21일
새로운 기능	이제 Amazon RDS에서 Oracle용 사용자 지정 암호 확인 기능으로 자체 사용자 지정 로직을 사용할 수 있습니다. 자세한 내용은 사용자 지정 암호 확인 함수 생성 (p. 841) 단원을 참조하십시오.	2017년 3월 21일
새로운 기능	이제 Amazon RDS에서 Oracle DB 인스턴스의 온라인 및 아카이브 다시 실행 로그 파일에 액세스할 수 있습니다. 자세한 내용은 트랜잭션 로그 액세스 (p. 857) 단원을 참조하십시오.	2017년 3월 21일

변경 사항	설명	변경 날짜
새로운 기능	이제 동일한 리전의 계정 간에 암호화된 DB 클러스터 스냅샷과 암호화되지 않은 DB 클러스터 스냅샷을 모두 복사할 수 있습니다. 자세한 내용은 계정 간 DB 클러스터 스냅샷 복사 단원을 참조하십시오.	2017년 3월 7일
새로운 기능	이제 동일한 리전의 계정 간에 암호화된 DB 클러스터 스냅샷을 공유할 수 있습니다. 자세한 내용은 DB 클러스터 스냅샷 공유 단원을 참조하십시오.	2017년 3월 7일
새로운 기능	이제 암호화된 Amazon Aurora MySQL DB 클러스터를 복제하여 리전 간 Aurora 복제본을 생성할 수 있습니다. 자세한 내용은 AWS 리전 간 Aurora MySQL DB 클러스터 복제 단원을 참조하십시오.	2017년 3월 7일
새로운 기능	이제 Microsoft SQL Server를 실행하는 DB 인스턴스에 대한 모든 연결이 SSL(Secure Sockets Layer)을 사용하도록 요구할 수 있습니다. 자세한 내용은 Microsoft SQL Server DB 인스턴스와 함께 SSL 사용 (p. 541) 단원을 참조하십시오.	2017년 2월 27일
새로운 기능	이제 현지 시간대를 15가지 추가 시간대 중 하나로 설정할 수 있습니다. 자세한 내용은 지원되는 시간대 (p. 490) 단원을 참조하십시오.	2017년 2월 27일
새로운 기능	이제 Amazon RDS 프로시저 <code>msdb.dbo.rds_shrink_tempdbfile</code> 을 사용하여 Microsoft SQL Server를 실행하는 DB 인스턴스의 tempdb 데이터베이스를 축소할 수 있습니다. 자세한 내용은 tempdb 데이터베이스 축소 (p. 556) 단원을 참조하십시오.	2017년 2월 17일
새로운 기능	이제 Enterprise와 Standard Edition Microsoft SQL Server 데이터베이스를 Amazon RDS DB 인스턴스에서 Amazon S3로 내보낼 때 백업 파일을 압축할 수 있습니다. 자세한 내용은 백업 파일 압축 (p. 527) 단원을 참조하십시오.	2017년 2월 17일
새로운 기능	Amazon RDS는 이제 Oracle을 실행하는 DB 인스턴스에서 아웃바운드 네트워크 액세스에 사용되는 DNS 이름을 확인하는 사용자 지정 DNS 서버를 지원합니다. 자세한 내용은 사용자 지정 DNS 서버 설정 (p. 844) 단원을 참조하십시오.	2017년 1월 26일
새로운 기능	Amazon RDS는 이제 다른 리전에서의 암호화된 읽기 전용 복제본 생성을 지원합니다. 자세한 내용은 다른 AWS 리전에서 읽기 전용 복제본 만들기 (p. 141) 및 CreateDBInstanceReadReplica 단원을 참조하십시오.	2017년 1월 23일
새로운 기능	이제 Amazon RDS는 MySQL 5.1에서 MySQL 5.5로 MySQL DB 스냅샷 업그레이드를 지원합니다. 자세한 내용은 MySQL DB 스냅샷 업그레이드 (p. 612) 및 ModifyDBSnapshot 단원을 참조하십시오.	2017년 1월 20일
새로운 기능	Amazon RDS는 이제 MariaDB, MySQL, Oracle, PostgreSQL, Microsoft SQL Server 데이터베이스 엔진의 경우, 암호화된 DB 스냅샷을 다른 리전으로 복사할 수 있도록 지원합니다. 자세한 내용은 DB 스냅샷 복사 (p. 209) 및 CopyDBSnapshot 단원을 참조하십시오.	2016년 20월 12일

변경 사항	설명	변경 날짜
새로운 기능	이제 Amazon RDS는 MariaDB 10.1을 실행하는 새 DB 인스턴스로 Amazon RDS MySQL 5.6 DB 스냅샷을 마이그레이션할 수 있도록 지원합니다. 자세한 내용은 MySQL DB 스냅샷에서 MariaDB DB 인스턴스로 데이터 마이그레이션 (p. 456) 단원을 참조하십시오.	2016년 20월 12일
새로운 기능	이제 Amazon Aurora MySQL은 공간 인덱싱을 지원합니다. 공간 인덱싱은 공간 데이터를 사용하는 쿼리를 위한 대규모 데이터 세트에서의 쿼리 성능을 향상시킵니다. 자세한 내용은 Amazon Aurora MySQL 및 공간 데이터 단원을 참조하십시오.	2016년 14월 12일
새로운 기능	Oracle용 Amazon RDS에는 이제 2016년 10월 Oracle Database PSU(Patch Set Update)가 포함되어 있습니다. 이로써 Oracle 데이터베이스 엔진 버전 12.1.0.2.v6 및 11.2.0.4.v10도 지원하게 되었습니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 (p. 902) 단원을 참조하십시오.	2016년 12월 12일
새로운 기능	Amazon RDS는 이제 Oracle을 실행하는 DB 인스턴스에서 아웃바운드 네트워크 액세스를 지원합니다. utl_http, utl_tcp 및 utl_smtp를 사용하여 DB 인스턴스에서 네트워크에 연결할 수 있습니다. 자세한 내용은 Oracle DB 인스턴스에 utl_http, utl_tcp 및 utl_smtp 사용 (p. 724) 단원을 참조하십시오.	2016년 5월 12일
새로운 기능	Amazon RDS는 MySQL 버전 5.1에 대한 지원을 만료했습니다. 하지만 기존 MySQL 5.1 스냅샷을 MySQL 5.5 인스턴스로 복원할 수 있습니다. 자세한 내용은 Amazon RDS MySQL에 대해 지원되는 스토리지 엔진 (p. 580) 단원을 참조하십시오.	2016년 11월 15일
새로운 기능	이제 Amazon RDS는 PostgreSQL 버전 9.6.1을 지원합니다. 자세한 내용은 Amazon RDS의 PostgreSQL 버전 9.6.1 (p. 1022) 단원을 참조하십시오.	2016년 11월 11일
새로운 기능	이제 Amazon RDS는 Microsoft SQL Server 2016 RTM CU2를 지원합니다. 자세한 내용은 Amazon RDS의 Microsoft SQL Server (p. 477) 단원을 참조하십시오.	2016년 11월 4일
새로운 기능	Amazon RDS는 이제 Oracle을 실행하는 DB 인스턴스의 마지막 버전 업그레이드를 지원합니다. 이제 Oracle DB 인스턴스를 11g에서 12c로 업그레이드할 수 있습니다. 자세한 내용은 Oracle DB 엔진 업그레이드 (p. 752) 단원을 참조하십시오.	2016년 11월 2일
새로운 기능	이제 Microsoft SQL Server 2014 Enterprise Edition을 실행하는 DB 인스턴스를 생성할 수 있습니다. 이제 Amazon RDS는 모든 에디션과 모든 리전에 SQL Server 2014 SP2를 지원합니다. 자세한 내용은 Amazon RDS의 Microsoft SQL Server (p. 477) 단원을 참조하십시오.	2016년 10월 25일

변경 사항	설명	변경 날짜
새로운 기능	이제 Amazon Aurora MySQL가 다른 AWS 서비스와 통합되었습니다. Amazon S3 버킷의 테이블에 텍스트 또는 XML 데이터를 로드하거나 데이터베이스 코드에서 AWS Lambda 함수를 호출할 수 있습니다. 자세한 내용은 기타 AWS Services와 Aurora MySQL 통합 을 참조하십시오.	2016년 10월 18일
새로운 기능	이제 Microsoft SQL Server를 실행하는 Amazon RDS DB 인스턴스의 tempdb 데이터베이스에 액세스할 수 있습니다. Microsoft SQL Server Management Studio(SSMS)를 통한 Transact-SQL 또는 기타 표준 SQL 클라이언트 애플리케이션을 사용하여 tempdb 데이터베이스에 액세스할 수 있습니다. 자세한 내용은 Amazon RDS에서 실행되는 Microsoft SQL Server DB 인스턴스의 tempdb 데이터베이스에 액세스 (p. 555) 단원을 참조하십시오.	2016년 9월 29일
새로운 기능	이제 Oracle을 실행 중인 Amazon RDS DB 인스턴스와 함께 UTL_MAIL 패키지를 사용할 수 있습니다. 자세한 내용은 Oracle UTL_MAIL (p. 828) 단원을 참조하십시오.	2016년 9월 20일
새로운 기능	Oracle용 Amazon RDS에는 이제 2016년 7월 Oracle Database PSU(Patch Set Update)가 포함됩니다. 여기서는 Oracle 데이터베이스 엔진 버전 12.1.0.2.v5, 12.1.0.1.v6 및 11.2.0.4.v9에 대한 지원을 추가합니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 (p. 902) 단원을 참조하십시오.	2016년 9월 20일
새로운 기능	이제 새 Microsoft SQL Server DB 인스턴스의 시간대를 애플리케이션의 시간대와 일치하도록 현지 시간대로 설정할 수 있습니다. 자세한 내용은 Microsoft SQL Server DB 인스턴스의 현지 시간대 (p. 489) 단원을 참조하십시오.	2016년 9월 19일
새로운 기능	새로운 PostgreSQL 버전 9.5.4, 9.4.9 및 9.3.14에 대한 지원을 추가했습니다. 또한 PostgreSQL 논리적 복제, PostgreSQL 이벤트 트리거 및 PostgreSQL stats_temp_directory용 RAM 디스크에 대한 지원도 추가했습니다. 자세한 내용은 지원되는 PostgreSQL 데이터베이스 버전 (p. 1014), Amazon RDS의 PostgreSQL에 대한 논리적 복제 (p. 1054), Amazon RDS의 PostgreSQL에 대한 이벤트 트리거 (p. 1056) 및 stats_temp_directory에 대한 RAM 디스크 (p. 1057) 단원을 참조하십시오.	2016년 9월 14일
새로운 기능	이제 Oracle Label Security 옵션을 사용하여 Oracle 12c 기반 Amazon RDS DB 인스턴스의 각 테이블 행에 대한 액세스를 제어할 수 있습니다. Oracle Label Security에서는 정책 기반 관리 모델을 통해 규제 준수를 이행할 뿐만 아니라 민감한 데이터에 대한 액세스를 적절한 권한을 가진 사용자로 제한할 수 있습니다. 자세한 내용은 Oracle 레이블 보안 (p. 801) 단원을 참조하십시오.	2016년 9월 8일

변경 사항	설명	변경 날짜
새로운 기능	<p>이제 DB 클러스터에서 사용 가능한 Aurora 복제본 간에 연결을 로드 밸런싱하는 리더 엔드포인트를 사용하여 Amazon Aurora DB 클러스터에 연결할 수 있습니다. 클라이언트가 리더 엔드포인트에 대한 새로운 연결을 요청하면 Aurora는 DB 클러스터 내의 Aurora 복제본 간에 연결 요청을 분배합니다. 이 기능은 DB 클러스터 내의 여러 Aurora 복제본 간에 읽기 워크로드의 균형을 유지하는 데 도움이 됩니다. 자세한 내용은 Amazon Aurora 엔드포인트 단원을 참조하십시오.</p>	2016년 9월 8일
새로운 기능	<p>이제 Oracle을 실행 중인 Amazon RDS DB 인스턴스에서 Oracle Enterprise Manager Cloud Control을 지원할 수 있습니다. DB 인스턴스에서 Management Agent를 활성화하고 Oracle Management Service(OMS)와 데이터를 공유할 수 있습니다. 자세한 내용은 Oracle Management Agent for Enterprise Manager Cloud Control (p. 793) 단원을 참조하십시오.</p>	2016년 9월 1일
새로운 기능	<p>이 릴리스는 리소스의 ARN을 가져오는 작업을 추가로 지원합니다. 자세한 내용은 기존 ARN 가져오기 (p. 176) 단원을 참조하십시오.</p>	2016년 8월 23일
새로운 기능	<p>이제 리소스 관리 및 비용 추적을 위해 각 Amazon RDS 리소스에 최대 50개의 태그를 지정할 수 있습니다. 자세한 내용은 Amazon RDS 리소스에 태그 지정 (p. 131) 단원을 참조하십시오.</p>	2016년 8월 19일
새로운 기능	<p>Amazon RDS는 현재 Oracle Standard Edition Two에 대해 License Included 모델을 지원합니다. 자세한 내용은 Oracle 데이터베이스 엔진을 실행하는 DB 인스턴스 생성 (p. 727) 단원을 참조하십시오.</p> <p>이제 Microsoft SQL Server와 Oracle을 실행하는 Amazon RDS DB 인스턴스의 라이선스 모델을 변경할 수 있습니다. 자세한 내용은 Amazon RDS의 Microsoft SQL Server 라이선싱 (p. 493) 및 Oracle 라이선싱 (p. 704) 단원을 참조하십시오.</p>	2016년 8월 5일
새로운 기능	<p>이제 AWS Management 콘솔을 사용하여 손쉽게 DB 인스턴스를 다른 VPC로 이동하거나 같은 VPC에 있는 다른 서브넷 그룹으로 이동 할 수 있습니다. 자세한 내용은 DB 인스턴스에 대한 VPC 업데이트 (p. 420) 단원을 참조하십시오.</p> <p>DB 인스턴스가 VPC 내에 있지 않을 경우 이제는 AWS Management 콘솔을 사용하여 DB 인스턴스를 VPC 내로 손쉽게 이동할 수 있습니다. 자세한 내용은 VPC에 있지 않은 DB 인스턴스를 VPC로 이동 (p. 420) 단원을 참조하십시오.</p>	2016년 8월 4일
새로운 기능	<p>이제 Amazon RDS에서 전체 백업 파일(.bak 파일)을 사용하여 Microsoft SQL Server 데이터베이스에 대한 기본 백업 및 복원을 할 수 있습니다. 이제 SQL Server 데이터베이스를 Amazon RDS로 손쉽게 마이그레이션하고, 이동하기 쉬운 파일 하나로 데이터베이스를 가져오거나 내보내고, 스토리지에 Amazon S3를 사용하고 암호화에 AWS KMS를 사용할 수 있습니다. 자세한 내용은 SQL Server 데이터베이스 가져오기 및 내보내기 (p. 520) 단원을 참조하십시오.</p>	2016년 7월 27일

변경 사항	설명	변경 날짜
새로운 기능	이제 MySQL 데이터베이스에서 Amazon Simple Storage Service(Amazon S3) 버킷으로 소스 파일을 복사한 다음 해당 파일에서 Amazon Aurora DB 클러스터를 복원할 수 있습니다. <code>mysqldump</code> 를 사용하여 데이터를 마이그레이션하는 것보다 이 방법이 훨씬 더 빠를 것입니다. 자세한 내용은 외부 MySQL 데이터베이스에서 Aurora MySQL DB 클러스터로 데이터 마이그레이션 을 참조하십시오.	2016년 7월 20일
새로운 기능	이제 복원 작업 중 AWS Key Management Service(AWS KMS) 암호화 키를 포함 시킴으로써 암호화되지 않은 Amazon Aurora DB 클러스터 스냅샷을 복원하여 암호화된 Amazon Aurora DB 클러스터를 만들 수 있습니다. 자세한 내용은 Amazon RDS 리소스 암호화 단원을 참조하십시오.	2016년 6월 30일
새로운 기능	Oracle용 Amazon RDS에는 이제 2016년 4월 Oracle Database PSU(Patch Set Update)가 포함됩니다. 이 PSU는 Oracle Database 엔진 버전 12.1.0.2.v4, 12.1.0.1.v5 및 11.2.0.4.v8에 대한 지원을 추가합니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 (p. 902) 단원을 참조하십시오.	2016년 6월 17일
새로운 기능	Oracle RCU(Repository Creation Utility)를 사용하여 Oracle 용 Amazon RDS에 리포지토리를 만들 수 있습니다. 자세한 내용은 Oracle용 Amazon RDS에서 Oracle Repository Creation Utility 사용 (p. 895) 단원을 참조하십시오.	2016년 6월 17일
새로운 기능	PostgreSQL 교차 리전 읽기 전용 복제본에 대한 지원이 추가됩니다. 자세한 내용은 다른 AWS 리전에서 읽기 전용 복제본 만들기 (p. 141) 단원을 참조하십시오.	2016년 6월 16일
새로운 기능	이제 AWS Management 콘솔을 사용하여 Microsoft SQL Server DB 인스턴스에 미러링을 통한 다중 AZ를 손쉽게 추가할 수 있습니다. 자세한 내용은 다중 AZ를 Microsoft SQL Server DB 인스턴스에 추가 (p. 538) 단원을 참조하십시오.	2016년 6월 9일
새로운 기능	이제 아시아 태평양(시드니), 아시아 태평양(도쿄) 및 남아메리카(상파울루)와 같은 추가 리전에서 SQL Server 미러링을 통한 다중 AZ 배포를 사용할 수 있습니다. 자세한 내용은 Microsoft SQL Server의 다중 AZ 배포 (p. 537) 단원을 참조하십시오.	2016년 6월 9일
새로운 기능	MariaDB 버전 10.1을 지원하도록 업데이트했습니다. 자세한 내용은 Amazon RDS MariaDB (p. 426) 단원을 참조하십시오.	2016년 6월 1일
새로운 기능	읽기 전용 복제본인 Amazon Aurora 리전 간 DB 클러스터를 지원하도록 업데이트되었습니다. 자세한 내용은 AWS 리전 간 Aurora MySQL DB 클러스터 복제 단원을 참조하십시오.	2016년 6월 1일
새로운 기능	현재 Enhanced Monitoring을 Oracle DB 인스턴스에 사용할 수 있습니다. 자세한 내용은 확장 모니터링 (p. 238) 및 Oracle 데이터베이스 엔진 기반 DB 인스턴스의 변경 (p. 742) 단원을 참조하십시오.	2016년 5월 27일
새로운 기능	Amazon Aurora DB 클러스터 스냅샷에 대한 수동 스냅샷 공유를 지원하도록 업데이트되었습니다. 자세한 내용은 DB 클러스터 스냅샷 공유 단원을 참조하십시오.	2016년 5월 18일

변경 사항	설명	변경 날짜
새로운 기능	이제 MariaDB 갑사 플러그인을 사용하여 MariaDB 및 MySQL 데이터베이스 인스턴스에서의 데이터베이스 활동을 로깅할 수 있습니다. 자세한 내용은 MariaDB 데이터베이스 엔진을 위한 옵션 (p. 467) 및 MySQL DB 인스턴스 옵션 (p. 663) 단원을 참조하십시오.	2016년 4월 27일
새로운 기능	이제 MySQL 버전 5.6을 버전 5.7로 업그레이드하는 데 인플레이스 메이저 버전을 사용할 수 있습니다. 자세한 내용은 MySQL DB 엔진 업그레이드 (p. 607) 단원을 참조하십시오.	2016년 4월 26일
새로운 기능	현재 Enhanced Monitoring을 Microsoft SQL Server DB 인스턴스에 사용할 수 있습니다. 자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.	2016년 4월 22일
새로운 기능	PostgreSQL 버전 9.5.2, 9.4.7 및 9.3.12에 대한 지원을 추가했습니다. 자세한 내용은 지원되는 PostgreSQL 데이터베이스 버전 (p. 1014) 단원을 참조하십시오.	2016년 4월 8일
새로운 기능	2016년 1월 Oracle PSU(Patch Set Updates)에서 Oracle 데이터베이스 버전 11.2.0.4.v7, 12.1.0.1.v4 및 12.1.0.2.v3를 지원하도록 업데이트되었습니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 (p. 902) 단원을 참조하십시오.	2016년 4월 1일
새로운 기능	Amazon RDS 콘솔에서 Amazon Aurora Clusters(클러스터) 보기를 제공하도록 업데이트했습니다. 자세한 내용은 Aurora DB 클러스터 보기 단원을 참조하십시오.	2016년 4월 1일
새로운 기능	아시아 태평양(서울) 리전에서 미러링을 통해 SQL Server 다중 AZ를 지원하도록 업데이트했습니다. 자세한 내용은 Microsoft SQL Server의 다중 AZ 배포 (p. 537) 단원을 참조하십시오.	2016년 3월 31일
새로운 기능	아시아 태평양(서울) 리전에서 미러링을 통해 Amazon Aurora 다중 AZ를 지원하도록 업데이트했습니다. 자세한 내용은 Amazon Aurora MySQL 가용성 을 참조하십시오.	2016년 3월 31일
새로운 기능	PostgreSQL DB 인스턴스에는 SSL을 사용하기 위해 연결해야 하는 기능이 있습니다. 자세한 내용은 PostgreSQL DB 인스턴스와 함께 SSL 사용 (p. 1059) 단원을 참조하십시오.	2016년 3월 25일
새로운 기능	현재 Enhanced Monitoring을 PostgreSQL DB 인스턴스에 사용할 수 있습니다. 자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.	2016년 3월 25일
새로운 기능	이제 Microsoft SQL Server DB 인스턴스에서 Windows 인증을 사용하여 사용자를 인증할 수 있습니다. 자세한 내용은 Microsoft SQL Server DB 인스턴스를 통한 Windows 인증 사용 (p. 568) 단원을 참조하십시오.	2016년 3월 23일
새로운 기능	현재 아시아 태평양(서울) 리전에서 Enhanced Monitoring을 사용할 수 있습니다. 자세한 내용은 확장 모니터링 (p. 238) 단원을 참조하십시오.	2016년 3월 16일
새로운 기능	이제 장애 조치 시 기본 인스턴스로 승격할 Aurora Replicas 순서를 사용자 지정할 수 있습니다. 자세한 내용은 Aurora DB 클러스터의 내결함성 을 참조하십시오	2016년 3월 14일

변경 사항	설명	변경 날짜
새로운 기능	Aurora DB 클러스터로 마이그레이션할 때 암호화를 지원하도록 업데이트되었습니다. 자세한 내용은 Aurora DB 클러스터로 데이터 마이그레이션 을 참조하십시오.	2016년 3월 2일
새로운 기능	Aurora DB 클러스터의 현지 시간대를 지원하도록 업데이트되었습니다. 자세한 내용은 Aurora DB 클러스터의 로컬 시간대 를 참조하십시오.	2016년 3월 1일
새로운 기능	MySQL 버전 5.7에 대한 최신 Amazon RDS DB 인스턴스 클래스 지원이 추가되도록 업데이트되었습니다.	2016년 2월 22일
새로운 기능	AWS GovCloud (US-West) 리전에서 db.r3 및 db.t2 DB 인스턴스 클래스를 지원하도록 업데이트되었습니다.	2016년 2월 11일
새로운 기능	DB 스냅샷의 사본을 암호화하는 작업과 암호화된 DB 스냅샷을 공유하는 작업을 지원하도록 업데이트되었습니다. 자세한 내용은 스냅샷 복사 (p. 207) 및 DB 스냅샷 공유 (p. 215) 단원을 참조하십시오.	2016년 2월 11일
새로운 기능	아시아 태평양(시드니) 리전에서 Amazon Aurora를 지원하도록 업데이트되었습니다. 자세한 내용은 Amazon Aurora MySQL 가용성을 참조하십시오.	2016년 2월 11일
새로운 기능	Oracle DB 인스턴스를 위한 SSL을 지원하도록 업데이트되었습니다. 자세한 내용은 Oracle DB 인스턴스에 SSL 사용 (p. 708) 단원을 참조하십시오.	2016년 2월 9일
새로운 기능	MySQL 및 MariaDB DB 인스턴스의 현지 시간대를 지원하도록 업데이트되었습니다. 자세한 내용은 MySQL DB 인스턴스의 현지 시간대 (p. 585) 및 MariaDB DB 인스턴스의 현지 시간대 (p. 435) 단원을 참조하십시오.	2015년 12월 21일
새로운 기능	MySQL 및 MariaDB 인스턴스와 Aurora DB 클러스터에 대한 OS 지표의 Enhanced Monitoring을 지원하도록 업데이트되었습니다. 자세한 내용은 DB 인스턴스 측정치 보기 (p. 236) 단원을 참조하십시오.	2015년 12월 18일
새로운 기능	BYOL(Bring-Your-Own-License) 라이선싱이 포함된 Oracle Standard Edition 2를 지원하도록 업데이트되었습니다. Oracle 버전 11.2.0.4.v5, 12.1.0.1.v3 및 12.1.0.2.v2에 대한 지원도 추가되었습니다. 자세한 내용은 Oracle 데이터베이스 엔진 출시 정보 (p. 902) 단원을 참조하십시오.	2015년 12월 14일
새로운 기능	MySQL 버전 5.5에 대한 db.t2, db.r3 및 db.m4 DB 인스턴스 클래스를 지원하도록 업데이트했습니다. 자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 단원을 참조하십시오.	2015년 12월 4일
새로운 기능	기존 DB 인스턴스에 대한 데이터베이스 포트를 수정할 수 있도록 업데이트되었습니다.	2015년 12월 3일
새로운 기능	PostgreSQL 버전 9.3.10 및 9.4.5 DB 인스턴스에 대한 새로운 세 가지 확장명을 지원하도록 업데이트되었습니다. 자세한 내용은 지원되는 PostgreSQL 데이터베이스 버전 (p. 1014) 단원을 참조하십시오.	2015년 12월 1일

변경 사항	설명	변경 날짜
새로운 기능	PostgreSQL 버전 9.3.10 및 9.4.5 DB 인스턴스를 지원하도록 업데이트되었습니다. 자세한 내용은 지원되는 PostgreSQL 데이터베이스 버전 (p. 1014) 단원을 참조하십시오.	2015년 27월 11일
새로운 기능	PostgreSQL 인스턴스에 대한 데이터베이스 엔진의 메이저 버전 업그레이드를 지원하도록 업데이트되었습니다. 자세한 내용은 Amazon RDS용 PostgreSQL DB 엔진 업그레이드 (p. 963) 단원을 참조하십시오.	2015년 11월 19일
새로운 기능	기존 DB 인스턴스에 대한 퍼블릭 액세스를 수정할 수 있도록 업데이트되었습니다. db.m4 표준 DB 인스턴스 클래스를 지원하도록 업데이트되었습니다.	2015년 11월 11일
새로운 기능	수동 DB 스냅샷 공유를 지원하도록 업데이트되었습니다. 자세한 내용은 DB 스냅샷 공유 (p. 215) 단원을 참조하십시오.	2015년 10월 28일
새로운 기능	Microsoft SQL Server 2014 Web, Express 및 Standard 버전을 지원하도록 업데이트되었습니다.	2015년 10월 26일
새로운 기능	MySQL 기반 MariaDB 데이터베이스 엔진을 지원하도록 업데이트되었습니다. 자세한 내용은 Amazon RDS MariaDB (p. 426) 단원을 참조하십시오.	2015년 10월 7일
새로운 기능	아시아 태평양(도쿄) 리전에서 Amazon Aurora를 지원하도록 업데이트되었습니다. 자세한 내용은 Amazon Aurora MySQL 가용성을 참조하십시오.	2015년 10월 7일
새로운 기능	모든 DB 엔진에 대해 db.t2 확장 가능 DB 인스턴스 클래스를 지원하고 db.t2.large DB 인스턴스 클래스를 추가할 수 있도록 업데이트되었습니다. 자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 단원을 참조하십시오.	2015년 9월 25일
새로운 기능	R3 및 T2 DB 인스턴스 클래스에서 Oracle DB 인스턴스를 지원하도록 업데이트되었습니다. 자세한 내용은 DB 인스턴스 클래스 선택 (p. 77) 단원을 참조하십시오.	2015년 8월 5일
새로운 기능	PostgreSQL 버전 9.4.4 및 9.3.9를 지원하도록 업데이트했습니다. 자세한 내용은 지원되는 PostgreSQL 데이터베이스 버전 (p. 1014) 단원을 참조하십시오.	2015년 7월 30일
새로운 기능	이제 라이선스 포함 서비스 모델에서 Microsoft SQL Server Enterprise Edition을 사용할 수 있습니다. 자세한 내용은 Amazon RDS의 Microsoft SQL Server 라이선싱 (p. 493) 단원을 참조하십시오.	2015년 7월 29일
새로운 기능	Amazon Aurora가 공식 출시되었습니다. Amazon Aurora DB 엔진은 DB 클러스터에서 여러 개의 DB 인스턴스를 지원합니다. 자세한 내용은 Amazon Aurora란 무엇입니까? 를 참조하십시오.	2015년 7월 27일
새로운 기능	DB 스냅샷으로 태그 복사를 지원하도록 업데이트되었습니다.	2015년 7월 20일

변경 사항	설명	변경 날짜
새로운 기능	인 메모리 옵션, Oracle 11g April PSU 패치, AWS CloudHSM과의 통합 개선 등, Oracle 12c 데이터베이스 버전 "12.1.0.2"를 지원하도록 업데이트되었습니다.	2015년 7월 20일
새로운 기능	모든 DB 엔진에 대한 스토리지 크기 증가와 SQL Server용 프로비저닝된 IOPS의 증가를 지원하도록 업데이트되었습니다.	2015년 6월 18일
새로운 기능	예약 DB 인스턴스를 위한 옵션들을 업데이트했습니다.	2015년 6월 15일
새로운 기능	Oracle 버전 12c를 지원하도록 업데이트되었습니다.	2015년 4월 2일
새로운 기능	PostgreSQL 버전 9.3.6 및 9.4.1을 지원하도록 업데이트되었습니다.	2015년 3월 18일
새로운 기능	TDE를 지원하는 Oracle DB 인스턴스와 함께 Amazon CloudHSM의 사용을 지원하도록 업데이트되었습니다.	2015년 1월 8일
새로운 기능	휴면 중인 데이터 암호화와 새로운 API 버전 2014-10-31을 지원하도록 업데이트되었습니다.	2015년 1월 6일
새로운 기능	2014년 10월 PSU가 포함된 Oracle 버전 11.2.0.4.v3을 지원하도록 업데이트되었습니다.	2014년 11월 20일
새로운 기능	새로운 Amazon DB 엔진인 Aurora를 추가하도록 업데이트되었습니다. Amazon Aurora DB 엔진은 한 DB 클러스터에서 여러 개의 DB 인스턴스를 지원합니다. Amazon Aurora는 현재 미리 보기 출시 중이며 변경될 수 있습니다. 자세한 내용은 Amazon Aurora란 무엇입니까? 를 참조하십시오.	2014년 11월 12일
새로운 기능	PostgreSQL 읽기 전용 복제본을 지원하도록 업데이트되었습니다.	2014년 11월 10일
새로운 기능	Oracle 11.2.0.4v2를 지원하도록 업데이트되었습니다.	2014년 10월 16일
새로운 API와 기능	GP2 스토리지 유형과 새로운 API 버전 2014-09-01을 지원하도록 업데이트되었습니다. 기존 옵션 또는 파라미터 그룹을 복사하여 새로운 옵션 또는 파라미터 그룹을 생성할 수 있는 기능을 지원하도록 업데이트되었습니다.	2014년 10월 7일
새로운 기능	MySQL 5.6.19 이후 버전을 기반으로 하는 DB 인스턴스에 InnoDB 캐시 워밍을 지원하도록 업데이트되었습니다.	2014년 9월 3일
새로운 기능	MySQL 5.6, SQL Server 및 PostgreSQL 데이터베이스 엔진에 연결할 때 SSL 인증서 검증을 지원하도록 업데이트되었습니다.	2014년 8월 5일
새로운 기능	버스트가 가능한 db.t2 DB 인스턴스 클래스를 지원하도록 업데이트되었습니다.	2014년 8월 4일
새로운 기능	MySQL 5.6, SQL Server 및 PostgreSQL 데이터베이스 엔진에 메모리에 최적화된 db.r3 DB 인스턴스 클래스 사용을 지원하도록 업데이트되었습니다.	2014년 5월 28일
새로운 기능	SQL Server 미러링을 사용하여 SQL Server 다중 AZ 배포를 지원하도록 업데이트되었습니다.	2014년 5월 19일

변경 사항	설명	변경 날짜
새로운 기능	MySQL 5.5에서 5.6으로 버전 업그레이드를 지원하도록 업데이트되었습니다.	2014년 4월 23일
새로운 기능	Oracle 11.2.0.4를 지원하도록 업데이트되었습니다.	2014년 4월 23일
새로운 기능	Oracle GoldenGate를 지원하도록 업데이트되었습니다.	2014년 4월 3일
새로운 기능	M3 DB 인스턴스 클래스를 지원하도록 업데이트되었습니다.	2014년 2월 20일
새로운 기능	Oracle Timezone 옵션을 지원하도록 업데이트되었습니다.	2014년 1월 13일
새로운 기능	리전이 다른 Amazon RDS MySQL DB 인스턴스 간에도 복제를 지원하도록 업데이트되었습니다.	2013년 11월 26일
새로운 기능	PostgreSQL DB 엔진을 지원하도록 업데이트되었습니다.	2013년 11월 14일
새로운 기능	SQL Server Transparent Data Encryption(TDE)을 지원하도록 업데이트되었습니다.	2013년 11월 7일
새로운 API와 기능	리전 간 DB 스냅샷 복사와 새로운 API 버전 2013-09-09를 지원하도록 업데이트되었습니다.	2013년 10월 31일
새로운 기능	Oracle Statspack을 지원하도록 업데이트되었습니다.	2013년 9월 26일
새로운 기능	Amazon RDS 기반 MySQL 인스턴스와 온프레미스 또는 Amazon EC2 기반 MySQL 인스턴스 사이에 복제를 이용해 데이터를 가져오거나 내보낼 수 있는 기능을 지원하도록 업데이트되었습니다.	2013년 9월 5일
새로운 기능	MySQL 5.6에 db.cr1.8xlarge DB 인스턴스 클래스를 지원하도록 업데이트되었습니다.	2013년 9월 4일
새로운 기능	읽기 전용 복제본의 복제를 지원하도록 업데이트되었습니다.	2013년 8월 28일
새로운 기능	병렬 방식의 읽기 전용 복제본 생성을 지원하도록 업데이트되었습니다.	2013년 7월 22일
새로운 기능	모든 Amazon RDS 리소스에 대한 사용 권한 세분화와 태그 지정을 지원하도록 업데이트되었습니다.	2013년 7월 8일
새로운 기능	MySQL 5.6 memcached 인터페이스와 바이너리 로그 액세스 지원을 포함하여 새로운 인스턴스에 MySQL 5.6을 지원하도록 업데이트되었습니다.	2013년 7월 1일
새로운 기능	MySQL 5.1에서 5.5로 메이저 버전 업그레이드를 지원하도록 업데이트되었습니다.	2013년 6월 20일
새로운 기능	DB 파라미터 그룹에서 연산식을 파라미터 값으로 사용할 수 있도록 업데이트되었습니다.	2013년 6월 20일
새로운 API와 기능	읽기 전용 복제본 상태와 새로운 API 버전 2013-05-15를 지원하도록 업데이트되었습니다.	2013년 5월 23일
새로운 기능	기본 네트워크 암호화를 위한 Oracle Advanced Security 기능과 Oracle Transparent Data Encryption을 지원하도록 업데이트되었습니다.	2013년 4월 18일

변경 사항	설명	변경 날짜
새로운 기능	SQL Server의 메이저 버전 업그레이드와 프로비저닝된 IOPS의 추가 기능을 지원하도록 업데이트되었습니다.	2013년 3월 13일
새로운 기능	RDS에서 기본 VPC를 지원하도록 업데이트되었습니다.	2013년 3월 11일
새로운 API와 기능	로그 액세스와 새로운 API 버전 2013-02-12를 지원하도록 업데이트되었습니다.	2013년 3월 4일
새로운 기능	RDS 이벤트 알림 구독을 지원하도록 업데이트되었습니다.	2013년 2월 4일
새로운 API와 기능	DB 인스턴스의 이름 바꾸기를 비롯해 VPC의 DB 보안 그룹 구성 요소를 VPC 보안 그룹으로 마이그레이션할 수 있는 기능을 지원하도록 업데이트되었습니다.	2013년 1월 14일
새로운 기능	AWS GovCloud (US-West)를 지원하도록 업데이트되었습니다.	2012년 12월 17일
새로운 기능	m1.medium 및 m1.xlarge DB 인스턴스 클래스를 지원하도록 업데이트되었습니다.	2012년 11월 6일
새로운 기능	읽기 전용 복제본 승격을 지원하도록 업데이트되었습니다.	2012년 11월 10일
새로운 기능	Microsoft SQL Server DB 인스턴스에서 SSL을 지원하도록 업데이트되었습니다.	2012년 10월 10일
새로운 기능	Oracle micro DB 인스턴스를 지원하도록 업데이트되었습니다.	2012년 9월 27일
새로운 기능	SQL Server 2012를 지원하도록 업데이트되었습니다.	2012년 9월 26일
새로운 API와 기능	프로비저닝된 IOPS를 지원하도록 업데이트되었습니다. API 버전 2012-09-17.	2012년 9월 25일
새로운 기능	SQL Server가 VPC의 DB 인스턴스를 지원하도록, 그리고 Oracle이 Data Pump를 지원하도록 업데이트되었습니다.	2012년 9월 13일
새로운 기능	SQL Server 에이전트를 지원하도록 업데이트되었습니다.	2012년 8월 22일
새로운 기능	DB 인스턴스의 태그 지정을 지원하도록 업데이트되었습니다.	2012년 8월 21일
새로운 기능	VPC에서 Oracle APEX 및 XML DB, Oracle Timezones, 그리고 Oracle DB 인스턴스를 지원하도록 업데이트되었습니다.	2012년 8월 16일
새로운 기능	VPC에서 SQL Server 데이터베이스 엔진 튜닝 어드바이저 와 Oracle DB 인스턴스를 지원하도록 업데이트되었습니다.	2012년 7월 18일
새로운 기능	옵션 그룹과 첫 번째 옵션인 Oracle Enterprise Manager Database Control을 지원하도록 업데이트되었습니다.	2012년 5월 29일
새로운 기능	Amazon Virtual Private Cloud에서 읽기 전용 복제본을 지원하도록 업데이트되었습니다.	2012년 5월 17일
새로운 기능	Microsoft SQL Server를 지원하도록 업데이트되었습니다.	2012년 5월 8일
새로운 기능	강제 장애 조치, Oracle DB 인스턴스의 다중 AZ 배포, 그리고 Oracle DB 인스턴스에서 기본으로 제공되는 문자 외에 문자 집합을 지원하도록 업데이트되었습니다.	2012년 5월 2일

Amazon Relational Database Service 사용 설명서
이전 업데이트

변경 사항	설명	변경 날짜
새로운 기능	Amazon Virtual Private Cloud(VPC)을 지원하도록 업데이트되었습니다.	2012년 13월 2일
업데이트 내용	새로운 예약 인스턴스 유형을 지원하도록 업데이트되었습니다.	2011년 12월 19일
새로운 기능	Oracle 엔진을 지원하도록 업데이트되었습니다.	2011년 5월 23일
업데이트 내용	콘솔이 업데이트되었습니다.	2011년 5월 13일
업데이트 내용	백업 및 유지 관리 기간을 단축하도록 내용이 편집되었습니다.	2011년 2월 28일
새로운 기능	MySQL 5.5 지원이 추가되었습니다.	2011년 1월 31일
새로운 기능	읽기 전용 복제본 지원이 추가되었습니다.	2010년 10월 4일
새로운 기능	AWS Identity and Access Management(IAM)에 대한 지원을 추가했습니다.	2010년 9월 2일
새로운 기능	DB 엔진 버전 관리 기능이 추가되었습니다.	2010년 8월 16일
새로운 기능	예약 DB 인스턴스가 추가되었습니다.	2010년 8월 16일
새로운 기능	Amazon RDS는 현재 DB 인스턴스에 대한 SSL 연결을 지원합니다.	2010년 6월 28일
새 설명서	이 설명서는 Amazon RDS 사용 설명서의 최초 릴리스입니다.	2010년 6월 7일