



관리 안내서

Amazon Redshift



Amazon Redshift: 관리 안내서

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

.....	xvi
Amazon Redshift란 무엇입니까?	1
Amazon Redshift를 처음 사용하시나요?	1
Amazon Redshift Serverless 기능 개요	2
Amazon Redshift 프로비저닝 클러스터 개요	4
클러스터 관리	5
클러스터 액세스 및 보안	5
클러스터 모니터링	7
데이터베이스 수	8
Amazon Redshift Serverless를 Amazon Redshift 프로비저닝된 데이터 웨어하우스와 비교	8
프로비저닝된 클러스터에 Amazon Redshift 관리 인터페이스 사용	33
AWS SDK 작업	34
HTTP 요청 서명	35
Amazon Redshift CLI 설정	40
Amazon Redshift Serverless	42
Amazon Redshift 서버리스란 무엇인가요?	42
Amazon Redshift Serverless 콘솔	43
Amazon Redshift Serverless 사용 시 고려 사항	46
Amazon Redshift Serverless의 컴퓨팅 용량	50
서버리스 엔드포인트 용량에 대한 고려 사항 및 제한 사항	52
AI 기반 크기 조정 및 최적화	54
Amazon Redshift Serverless에 대한 청구	59
컴퓨팅 용량에 대한 결제	59
온디맨드 컴퓨팅 용량에 대한 결제	59
서버리스 예약에 대한 결제	64
스토리지에 대한 결제	70
Amazon Redshift Serverless 무료 평가판 사용	71
결제 사용량 메모	71
연결 풀링 사용 시 Amazon Redshift Serverless 청구	73
제로 ETL을 통한 Amazon Redshift Serverless 비용 최적화	74
Amazon Redshift 서버리스에 연결	74
Amazon Redshift 서버리스에 연결	74
JDBC 드라이버를 통해 Amazon Redshift 서버리스에 연결	75
Data API를 사용하여 Amazon Redshift Serverless에 연결	76

SSL을 사용하여 Amazon Redshift Serverless에 연결	77
Amazon Redshift 관리형 VPC 엔드포인트에서 Amazon Redshift Serverless에 연결	79
인터페이스 VPC 엔드포인트에서 Amazon Redshift Serverless에 연결(AWS PrivateLink)	79
다른 계정의 Redshift VPC 엔드포인트에서 Amazon Redshift Serverless에 연결	80
추가 리소스	84
페더레이션 사용자를 위한 데이터베이스 역할 정의	85
데이터베이스 역할 정의	86
페더레이션 사용자에게 부여할 데이터베이스 역할을 정의하는 사용 사례	86
추가 리소스	88
Amazon Redshift Serverless의 Identity and Access Management	88
권한 부여	88
IAM 자격 증명 시작하기	90
데이터베이스 역할 권한으로 데이터베이스 객체에 액세스	91
프로비저닝된 클러스터를 Amazon Redshift Serverless로 마이그레이션	92
프로비저닝된 클러스터의 스냅샷 생성	93
드라이버를 사용하여 Amazon Redshift Serverless에 연결	94
Amazon Redshift Serverless SDK 사용	100
작업 그룹 및 네임스페이스	100
콘솔을 사용하는 작업 그룹 및 네임스페이스	100
AWS Command Line Interface 및 Amazon Redshift Serverless API를 사용하는 작업 그룹 및 네임스페이스	101
작업 그룹	102
네임스페이스	107
쿼리 및 워크로드 모니터링	111
쿼리 모니터링 정책 추가	112
사용자에 대한 쿼리 모니터링 권한 부여	114
역할에 대한 쿼리 모니터링 권한 부여	114
사용 제한 설정	115
쿼리 제한 설정	116
쿼리 대기열 설정	116
대시보드를 사용하여 요약 데이터 확인	121
감사 로깅	121
CloudWatch의 로그 이벤트	122
CloudWatch 지표	122
스냅샷 및 복구 시점	129
AWS Backup 통합	131

스냅샷 생성	131
최종 스냅샷 생성	132
스냅샷 공유 또는 스냅샷 권한 제거	132
스냅샷 일정 예약	133
스냅샷 보존 기간 업데이트	136
스냅샷 삭제	136
스냅샷 복원	136
복구 시점 변환	138
복구 시점 복원	138
다른 AWS 리전에 백업 복사	138
테이블 복원	140
데이터 공유	141
고려 사항	141
데이터 공유를 볼 수 있는 액세스 권한 부여	142
AWS Glue Data Catalog에 네임스페이스 등록	142
리소스에 태그 지정	143
Amazon Redshift 프로비저닝 클러스터	145
클러스터 및 노드	145
노드 유형 세부 정보	147
노드 수 결정	150
EC2를 사용하여 클러스터 생성	150
Amazon Virtual Private Cloud(Amazon VPC)	151
기본 디스크 공간 경보	151
클러스터 상태	152
프로비저닝된 클러스터 사용 시 고려 사항	154
리전 및 가용 영역 고려 사항	154
클러스터 유지 관리	154
클러스터 작업	164
클러스터 생성	165
디스크 공간 경보 생성	168
클러스터 보기	169
클러스터 수정	169
클러스터 크기 조정	170
클러스터 이름 변경	186
클러스터의 릴리스 버전 업그레이드	187
클러스터 일시 중지 및 다시 시작	188

클러스터 재부팅	190
클러스터 재부팅	190
사용량 한도 설정	194
클러스터 종료 및 삭제	195
스냅샷 및 백업	196
AWS Glue Data Catalog에 클러스터 등록	220
다중 AZ 배포	220
다중 AZ 배포 설정	221
새 클러스터 생성 시 다중 AZ 설정	223
스냅샷에서 복원된 데이터 웨어하우스에 대한 다중 AZ 설정	226
단일 AZ 데이터 웨어하우스를 다중 AZ 데이터 웨어하우스로 변환	227
다중 AZ 데이터 웨어하우스를 단일 AZ 데이터 웨어하우스로 변환	229
다중 AZ 데이터 웨어하우스 크기 조정	230
다중 AZ 배포를 통한 장애 조치	231
다중 AZ 데이터 웨어하우스의 쿼리 및 로드 보기	232
다중 AZ 배포에서 쿼리 모니터링	233
클러스터에 대한 쿼리 종료	234
클러스터 성능 모니터링	234
성능 데이터	235
성능 데이터 보기	251
쿼리 실행 분석	274
경보 생성	276
실행 쿼리 종료	277
CloudWatch 콘솔의 성능 지표	277
쿼리 프로파일러	279
쿼리 및 데이터베이스 모니터링	288
Sys 뷰 기반 쿼리 및 데이터베이스 모니터링	295
Tracks	298
트랙 간 전환	299
트랙 및 복원	299
서버리스 작업 그룹 업데이트	300
버전 관리	300
작업 그룹 또는 클러스터 버전 확인	301
제로 ETL 통합	302
고려 사항	304
대상에서 기록 모드 사용 시 고려 사항	306

제로 ETL 통합 소스가 Aurora 또는 Amazon RDS인 경우 고려 사항	307
제로 ETL 통합 소스가 DynamoDB인 경우 고려 사항	308
제로 ETL 통합 소스가 Salesforce, SAP, ServiceNow 및 Zendesk와 같은 애플리케이션일 경우 고려 사항	309
제로 ETL 통합 시작하기	309
대상 Amazon Redshift 데이터 웨어하우스 생성 및 구성	310
대/소문자 기능 사용 설정	310
Amazon Redshift에서 권한 부여 구성	312
제로 ETL 통합 생성	317
대상 데이터베이스 생성	328
복제된 데이터 쿼리	331
구체화된 뷰로 복제된 데이터 쿼리	331
DynamoDB에서 복제된 데이터 쿼리	333
제로 ETL 통합 보기	334
기록 모드	337
데이터 공유	340
제로 ETL 통합 모니터링	340
Amazon Redshift 시스템 뷰로 제로 ETL 통합 모니터링	340
Amazon EventBridge로 제로 ETL 통합 모니터링	341
제로 ETL 통합의 지표	341
DynamoDB에 대한 제로 ETL 통합 수정	343
DynamoDB에 대한 제로 ETL 통합 삭제	344
지원되는 리전	345
Aurora MySQL	345
Aurora PostgreSQL	347
Amazon DynamoDB	349
Amazon RDS for MySQL	351
엔터프라이즈 애플리케이션	353
제로 ETL 통합 문제 해결	355
데이터베이스 쿼리	370
Amazon Redshift에 연결	370
Amazon Redshift 쿼리 편집기 v2를 사용하여 데이터베이스 쿼리	371
AWS 계정 구성	372
쿼리 편집기 v2 열기	379
Amazon Redshift 데이터베이스에 연결	384
Amazon Redshift 데이터베이스 찾아보기	385

데이터베이스 객체 생성	387
쿼리 및 탭 기록 보기	394
Amazon Q 생성형 SQL과의 상호 작용	395
데이터를 데이터베이스로 로드	407
쿼리 작성	416
Notebooks	422
AWS Glue Data Catalog 쿼리	425
데이터 레이크 쿼리	428
데이터 공유	430
예약된 쿼리	433
결과 시각화	443
팀으로 협업 및 공유	449
Query Editor V1을 사용하여 데이터베이스 쿼리	450
고려 사항	451
SQL 클라이언트 도구를 사용하여 데이터 웨어하우스에 연결	452
클라이언트 도구와의 연결을 위한 권장 사항	452
Amazon Redshift에서 연결 구성	453
연결을 위한 보안 옵션 구성	636
클라이언트 도구 및 코드에서 연결	645
인증 프로파일을 사용하여 Amazon Redshift에 연결	692
Amazon Redshift 연결 문제 해결	695
데이터 API 사용	703
Data API 작업	703
Data API 호출 시 고려할 사항	704
데이터베이스 인증 보안 인증 정보 선택	706
JDBC 데이터 형식 매핑	707
파라미터로 SQL 문 실행	708
Amazon Redshift Data API를 호출할 때 맥동성 토큰으로 SQL 문 실행	709
세션 재사용으로 SQL 문 실행	711
결과 가져오기	712
액세스 권한 부여	715
신뢰할 수 있는 ID 전파	724
데이터 API 호출	727
데이터 API 문제 해결	755
Amazon EventBridge로 Data API 작업 예약	756
Data API 모니터링	760

데이터 API와 함께 AWS KMS 사용	762
Amazon SageMaker Unified Studio 사용	765
파라미터 그룹	767
기본 파라미터 값	768
워크로드 관리	769
WLM 동적 및 정적 속성	770
WLM 구성 파라미터의 속성	770
AWS CLI를 사용하여 WLM 파라미터 구성	777
파라미터 그룹 생성	785
파라미터 그룹 수정	785
쿼리 모니터링 규칙 생성	789
파라미터 그룹 삭제	791
AWS 파트너와 통합	792
AWS 파트너를 통한 데이터 로드	793
예약 노드	795
예약 노드	795
예약 노드 간 요금 비교	796
예약 노드의 작동 방식	797
예약 노드 및 통합 결제	798
예약 노드 예제	799
예 1	799
예제 2	799
예 3	799
예 4	799
예 5	800
예 6	800
예약 노드 구입	800
보안	803
데이터 보호	804
데이터 암호화	805
데이터 토큰화	822
인터넷워크 트래픽 라우팅	823
ID 및 액세스 관리	824
ID를 사용한 인증	824
교차 서비스 혼동된 대리인 방지	825
액세스 관리	826

액세스 관리 개요	827
ID 기반 정책(IAM 정책) 사용	833
네이티브 ID 제공업체(idP) 페더레이션	892
서비스 연결 역할 사용	923
IAM 인증을 이용한 데이터베이스 사용자 자격 증명 생성	931
AWS 서비스에 대한 액세스 권한 부여	986
관리자 암호 관리	1020
AWS Secrets Manager 통합에 필요한 권한	1021
관리자 보안 암호 교체	1022
Amazon Redshift와 함께 AWS Secrets Manager 사용 시 고려 사항	1022
ARN 보안 암호 검색	1023
데이터베이스 연결 자격 증명을 위한 보안 암호 생성	1023
로그 및 모니터링	1026
데이터베이스 감사 로깅	1027
CloudTrail을 사용하여 로깅	1043
규정 준수 확인	1054
복원성	1056
인프라 보안	1056
네트워크 격리	823
보안 그룹	1058
인터페이스 VPC 엔드포인트	1058
구성 및 취약성 분석	1066
Identity Center 인증	1067
사전 조건	1067
Identity Center 인증 작동 방식	1068
GetIdentityCenterAuthToken API 작업	1068
프로비저닝된 클러스터에 대한 GetIdentityCenterAuthToken	1068
서버리스 작업 그룹을 위한 GetIdentityCenterAuthToken	1069
응답 구문	1070
드라이버 통합	1071
직접 토큰 사용	1071
Java 코드 예제	1072
IAM 정책 요구 사항	1072
API 권한	1073
예제 IAM 정책	1073
리전별 가용성	1074

문제 해결	1075
API 오류	1075
연결 오류	1075
네트워킹 작업	1077
클라이언트 연결의 사용자 지정 도메인 이름	1077
도메인 이름 등록	1078
도메인 이름에 대한 인증서 요청	1079
사용자 지정 도메인 구성	1080
프로비저닝된 클러스터 또는 작업 그룹에 연결	1081
사용자 지정 도메인이 할당된 클러스터 이름 변경	1082
사용자 지정 도메인 연결 설명	1083
사용자 지정 도메인을 다른 인증서와 연결	1084
사용자 지정 도메인 삭제	1085
Redshift 관리형 VPC 엔드포인트	1085
고려 사항	1086
VPC에 대한 액세스 권한 부여	1087
Redshift 관리형 VPC 엔드포인트 생성	1088
VPC의 Redshift 리소스	1089
VPC에서 클러스터 또는 작업 그룹 생성	1091
VPC 보안 그룹	1093
클러스터 또는 작업 그룹의 보안 설정 구성	1094
Redshift 리소스의 서브넷	1097
VPC 및 서브넷에 대한 퍼블릭 액세스 차단	1100
Enhanced VPC Routing으로 네트워크 트래픽 제어	1101
VPC 엔드포인트를 사용하여 데이터베이스 트래픽 제어	1102
향상된 VPC 라우팅 켜기	1104
Redshift Spectrum으로 Amazon S3 버킷에 액세스	1105
이벤트	1110
클러스터 이벤트 알림 구독	1110
이벤트 알림 구독 생성	1112
프로비저닝된 클러스터 이벤트 알림	1113
Amazon Redshift Serverless 이벤트 알림	1134
제로 ETL 통합 이벤트 알림	1145
할당량 및 제한	1153
Amazon Redshift 객체에 대한 할당량	1153
Amazon Redshift Serverless 객체 할당량	1159

Amazon Redshift 데이터 API의 할당량	1162
쿼리 편집기 v2 객체에 대한 할당량	1164
Amazon Redshift Spectrum 객체 할당량 및 제한	1165
명명 제약 조건	1166
리소스 태깅	1169
태그 지정 요구 사항	1170
리소스 태그 관리	1170
AWS Backup 통합	1172
고려 사항	1173
제한 사항	1174
Amazon Redshift에서 AWS Backup 사용	1174
클러스터 버전	1175
패치 199	1176
새로운 기능	1176
패치 198	1178
새로운 기능	1178
패치 197	1180
새로운 기능	1180
패치 196	1182
새로운 기능	1183
패치 195	1183
새로운 기능	1183
패치 194	1184
새로운 기능	1184
패치 193	1185
새로운 기능	1185
패치 192	1186
새로운 기능	1186
패치 191	1186
새로운 기능	1187
패치 190	1187
새로운 기능	1187
패치 189	1188
새로운 기능	1188
패치 188	1189
새로운 기능	1189

패치 187	1190
새로운 기능	1190
패치 186	1191
새로운 기능	1193
패치 185	1193
새로운 기능	1194
패치 184	1195
새로운 기능	1195
패치 183	1196
새로운 기능	1197
패치 182	1198
새로운 기능	1198
패치 181	1200
새로운 기능	1200
패치 180	1201
새로운 기능	1202
패치 179	1203
새로운 기능	1204
패치 178	1205
새로운 기능	1205
패치 177	1207
새로운 기능	1208
패치 176	1209
새로운 기능	1210
패치 175	1211
새로운 기능	1211
패치 174	1212
이 버전의 새로운 기능	1212
이 버전의 새로운 기능	1212
이 버전의 새로운 기능	1212
이 버전의 새로운 기능	1212
이 버전의 새로운 기능	1212
이 버전의 새로운 기능	1212
이 버전의 새로운 기능	1212
패치 173	1213
이 버전의 새로운 기능	1213

이 버전의 새로운 기능	1213
이 버전의 새로운 기능	1213
이 버전의 새로운 기능	1213
이 버전의 새로운 기능	1213
이 버전의 새로운 기능	1213
이 버전의 새로운 기능	1213
이 버전의 새로운 기능	1213
이 버전의 새로운 기능	1213
이 버전의 새로운 기능	1213
패치 172	1215
새로운 기능	1215
패치 171	1216
새로운 기능	1216
패치 170	1216
새로운 기능	1216
패치 169	1216
새로운 기능	1217
패치 168	1217
새로운 기능	1217
동작 변경 사항	1219
예정된 동작 변경 사항	1219
2026년 6월 30일 이후 Scalar Python UDF의 지원 종료	1219
2026년 2월 27일 이후 구체화된 뷰(MV) 자동 새로 고침 동작 변경	1220
Amazon Redshift는 2026년 2월 16일 이후 데이터 공유를 통해 소비자 정보에 액세스하는 합 수를 지원하지 않습니다.	1220
2026년 1월 31일부터 적용되는 Transport Layer Security(TLS) 버전 변경 사항	1220
Amazon Redshift는 2025년 10월 30일 이후 새 스칼라 Python UDF 생성을 지원하지 않음 .	1221
최근 동작 변경 사항	1222
Amazon Redshift는 2025년 8월 26일 이후 최신 IANA 시간대 데이터베이스를 사용합니다.	1222
2025년 8월 15일 이후 적용되는 Amazon Redshift Serverless RPU 변경 사항	1223
2025년 8월 10일 이후 적용되는 데이터베이스 감사 로깅 변경 사항	1223
2025년 6월 27일 이후 적용되는 서버리스 작업 그룹에 대한 가상 프라이빗 클라우드 엔드포 인트 변경 사항	1223
2025년 5월 2일 이후에 적용되는 쿼리 모니터링 변경 사항	1224
2025년 1월 10일 후에 적용되는 보안 변경 사항	1224
코드 예제	1227

기본 사항	1228
Hello Amazon Redshift	1228
기본 사항 알아보기	1233
작업	1299
시나리오	1357
Amazon Redshift 데이터를 추적하는 웹 애플리케이션 만들기	1357
문서 기록	1359

Amazon Redshift는 패치 198부터 새 Python UDF 생성을 더 이상 지원하지 않습니다. 기존 Python UDF는 2026년 6월 30일까지 계속 작동합니다. 자세한 내용은 [블로그 게시물](#)을 참조하세요.

Amazon Redshift란 무엇입니까?

Amazon Redshift 관리 가이드입니다. Amazon Redshift는 클라우드에서 완전히 관리되는 페타바이트 급 데이터 웨어하우스 서비스입니다. Amazon Redshift Serverless를 사용하면 프로비저닝된 데이터 웨어하우스를 구성하지 않아도 데이터를 액세스하고 분석할 수 있습니다. 리소스가 자동으로 프로비저닝하고 데이터 웨어하우스 용량이 지능적으로 크기 조정되어 가장 까다롭고 예측할 수 없는 워크로드에도 빠른 성능을 제공합니다. 데이터 웨어하우스가 유휴 상태일 때는 요금이 발생하지 않으므로 사용량에 대한 요금만 지불합니다. Amazon Redshift 쿼리 에디터 v2 또는 자주 사용하는 비즈니스 인텔리전스(BI) 도구에서 바로 데이터를 로드하고 쿼리를 시작할 수 있습니다. 사용하기 쉽고 관리가 필요 없는 환경에서 최고의 가격 대비 성능과 친숙한 SQL 기능을 활용하세요.

Amazon Redshift는 데이터 집합의 크기와 상관없이 오늘날 사용되는 것과 동일한 SQL 기반 도구 및 비즈니스 인텔리전스 애플리케이션을 사용하여 쿼리 성능을 가속화합니다.

Amazon Redshift를 처음 사용하시나요?

Amazon Redshift를 처음 사용할 경우 먼저 다음 섹션을 읽을 것을 권장합니다.

- [서비스 하이라이트 및 요금](#) – 이 제품 세부 정보 페이지에서는 Amazon Redshift 가치 제안, 서비스 하이라이트 및 요금에 대해 알아봅니다.
- [Amazon Redshift Serverless 데이터 웨어하우스 시작하기](#) – 이 주제에서는 서버리스 데이터 웨어하우스를 설정하고, 리소스를 생성하고, 샘플 데이터를 쿼리하는 프로세스를 안내합니다.
- [Amazon Redshift 데이터베이스 개발자 안내서](#) - 데이터베이스 개발자를 위한 이 안내서에서는 데이터 웨어하우스를 구성하는 데이터베이스의 설계부터 빌드, 쿼리 및 유지 관리에 이르는 방법에 대해서 설명합니다.

Amazon Redshift 리소스를 수동으로 관리하려는 경우 데이터 쿼리 요구 사항에 맞게 프로비저닝된 클러스터를 생성할 수 있습니다. 자세한 내용은 [Amazon Redshift 클러스터](#) 섹션을 참조하세요.

애플리케이션 개발자라면 Amazon Redshift API 또는 AWS 소프트웨어 개발 키트(SDK) 라이브러리를 사용하여 프로그래밍 방식으로 클러스터를 관리할 수 있습니다. Amazon Redshift API를 사용하는 경우에는 서명을 통해 API에 대한 모든 HTTP 또는 HTTPS 요청을 인증해야 합니다. MySQL 옵션에 대한 자세한 내용은 [HTTP 요청 서명](#) 섹션을 참조하세요.

API, CLI 및 SDK에 대한 자세한 내용은 다음 링크에서 확인할 수 있습니다.

- [Amazon Redshift Serverless API 참조](#)

- [Amazon Redshift API 참조](#)
- [Amazon Redshift 데이터 API 참조](#)
- [AWS CLI 명령 참조](#)
- [Amazon Web Services용 도구](#)의 SDK 참조 자료

Amazon Redshift Serverless 기능 개요

Amazon Redshift 프로비저닝된 데이터 웨어하우스에서 지원하는 대부분의 기능은 Amazon Redshift Serverless에서도 지원됩니다. 다음은 몇 가지 주요 기능입니다.

기능	설명
스냅샷	Amazon Redshift Serverless의 스냅샷이나 프로비저닝된 데이터 웨어하우스를 Amazon Redshift Serverless로 복원할 수 있습니다. 자세한 내용은 스냅샷 및 복구 시점 섹션을 참조하세요.
복구 시점	Amazon Redshift Serverless는 30분마다 자동으로 복구 시점을 생성합니다. 이러한 복구 시점은 24시간 동안 보관됩니다. 실수로 작성하거나 삭제한 후 복구 시점을 사용하여 복원할 수 있습니다. 복구 시점에서 복원하면 Amazon Redshift Serverless 데이터베이스에 있는 모든 데이터가 이전 시점으로 복원됩니다. 더 오랜 기간 동안 복구 시점을 유지해야 하는 경우 복구 시점에서 스냅샷을 생성할 수도 있습니다. 자세한 내용은 스냅샷 및 복구 시점 섹션을 참조하세요.
기본 RPU 용량	Redshift 처리 장치(RPU)로 기본 용량을 설정할 수 있습니다. 1RPU는 16GB의 메모리를 제공합니다. 이 설정을 사용하면 사용 중인 리소스와 워크로드 비용 간의 균형을 조절할 수 있습니다. 이 값을 늘려 사용 가능한 리소스를 늘리고 쿼리 성능을 개선하거나 이 값을 낮추어 지출을 제한할 수 있습니다. 기본값은 128RPU입니다. 하루에 사용되는 RPU와 같은 사용량 한도를 설정하여 비용을 관리할 수도 있습니다. 자세한 내용은 Amazon Redshift Serverless에 대한 청구 섹션을 참조하세요.
데이터 공유의 사용량 한도	콘솔 또는 API를 사용하여 생산자 리전에서 소비자 리전으로 전송되는 데이터의 양을 제한할 수 있습니다. 이 데이터 전송 비용은 AWS 리전에 따라 다르며 테라바이트 단위로 측정됩니다. 데이터 공유에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 콘솔을 사용하여 데이터 공유 시작하기 를 참조하세요.

기능	설명
사용자 정의 함수(UDF)	Amazon Redshift Serverless에서 사용자 정의 함수(UDF)를 실행할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 사용자 정의 함수 생성 을 참조하세요.
저장 프로시저	Amazon Redshift Serverless에서 저장 프로시저를 실행할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 저장 프로시저 생성 을 참조하세요.
구체화된 보기	Amazon Redshift Serverless에서 구체화된 보기를 생성할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 구체화된 보기 생성 을 참조하세요.
공간 함수	Amazon Redshift Serverless에서 공간 함수를 실행할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 공간 데이터 쿼리 를 참조하세요.
페더레이션 쿼리	Amazon Redshift Serverless에서 쿼리를 실행하여 Aurora DB 클러스터 및 Amazon RDS 데이터베이스와 데이터를 조인할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 연합 쿼리를 사용하여 데이터 쿼리 를 참조하세요.
데이터 레이크 쿼리	Amazon Redshift Serverless에서 쿼리를 실행하여 Amazon S3 데이터 레이크와 데이터를 조인할 수 있습니다. 자세한 정보는 Amazon Redshift 관리 가이드의 데이터 레이크 쿼리하기 섹션을 참조하세요.
HyperLogLog	Amazon Redshift Serverless에서 HyperLogLog 함수를 실행할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 HyperLogLog 스케치 사용 을 참조하세요.
데이터베이스 간 데이터 쿼리	Amazon Redshift Serverless에서 데이터베이스 전반에 걸쳐 데이터를 쿼리할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 데이터베이스 간 데이터 쿼리 를 참조하세요.
데이터 공유	Amazon Redshift Serverless에서 프로비저닝된 클러스터의 데이터 공유에 액세스할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 클러스터 간 데이터 공유 를 참조하세요.

기능	설명
반정형 데이터 쿼리	Amazon Redshift Serverless에서 SUPER 데이터 형식으로 반정형 데이터를 수집하고 저장할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 반정형 데이터 수집 및 쿼리 를 참조하세요.
리소스에 태그 지정	AWS CLI 또는 Amazon Redshift Serverless API를 통해 리소스와 관련된 메타데이터를 사용하여 리소스에 태그를 지정할 수 있습니다. 자세한 내용은 리소스에 태그 지정 섹션을 참조하세요.
기계 학습	Amazon Redshift Serverless에서 Amazon Redshift 기계 학습을 사용할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 기계 학습 사용 을 참조하세요.
SQL 명령 및 함수	몇 가지 예외(예: REBOOT_CLUSTER)를 제외하고는 Amazon Redshift Serverless에서 Amazon Redshift SQL 명령 및 함수를 사용할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 SQL 참조 를 참조하세요.
CloudFormation 리소스	CloudFormation 템플릿을 사용하여 Amazon Redshift Serverless 리소스를 배포하고 업데이트할 수 있습니다. 이 통합을 통해 리소스 관리에 소요되는 시간을 줄이고 애플리케이션에 집중할 수 있습니다. Amazon Redshift Serverless의 CloudFormation 리소스에 대한 자세한 내용은 Amazon Redshift Serverless resource type reference (Amazon Redshift Serverless 리소스 유형 참조)를 참조하세요.
CloudTrail 리소스	Amazon Redshift Serverless는 Amazon Redshift Serverless에서 수행한 작업의 기록을 제공하기 위해 AWS CloudTrail과 통합됩니다. CloudTrail은 Amazon Redshift Serverless에 대한 모든 API 호출을 이벤트로 캡처합니다. 자세한 내용은 Amazon Redshift Serverless용 CloudTrail 을 참조하세요.

Amazon Redshift 프로비저닝 클러스터 개요

Amazon Redshift 서비스는 데이터 웨어하우스의 설정, 운영 및 조정 작업을 모두 관리합니다. 이 태스크 중에는 용량 프로비저닝과 클러스터 모니터링 및 백업, 그리고 Amazon Redshift 엔진의 패치 및 업그레이드도 포함됩니다.

다음 동영상은 Amazon Redshift 쿼리 에디터 v2를 사용하여 클러스터를 생성하고 데이터를 쿼리 하는 방법을 보여줍니다.

클러스터 관리

Amazon Redshift 클러스터는 리더 노드 1개와 컴퓨팅 노드 1개 이상으로 구성된 노드 집합입니다. 필요한 컴퓨팅 노드 유형 및 수는 데이터의 크기와 실행할 쿼리 수 그리고 필요한 쿼리 런타임 성능에 따라 결정됩니다.

클러스터 생성 및 관리

데이터 웨어하우징 요건에 따라 처음에는 소규모의 단일 노드 클러스터로 시작하여 요건이 바뀌면서 대규모의 다중 노드 클러스터로 쉽게 확장할 수 있습니다. 또한 서비스 중단 없이 컴퓨팅 노드를 클러스터에 추가하거나 제거할 수도 있습니다. 자세한 내용은 [Amazon Redshift 프로비저닝 클러스터](#) 섹션을 참조하세요.

컴퓨팅 노드 예약

클러스터를 1년 이상 계속해서 실행할 계획이라면 컴퓨팅 노드를 1년 또는 3년 단위로 예약하여 비용을 줄일 수 있습니다. 컴퓨팅 노드를 예약하면 온디맨드 방식으로 프로비저닝하여 시간 단위로 요금을 지불하는 것과 비교하여 비용 절감 효과가 매우 큼니다. 자세한 내용은 [예약 노드](#) 섹션을 참조하세요.

클러스터 스냅샷 생성

스냅샷은 클러스터의 특정 시점 백업입니다. 자동 및 수동의 2가지 유형의 스냅샷이 있습니다. Amazon Redshift는 암호화된 보안 소켓 계층(SSL) 연결을 사용하여 이러한 스냅샷을 Amazon Simple Storage Service(Amazon S3)에 내부적으로 저장합니다. 스냅샷에서 복원해야 하는 경우에는 Amazon Redshift가 새로운 클러스터를 생성하고 지정한 스냅샷에서 데이터를 가져옵니다. 스냅샷 복사에 대한 자세한 내용은 [Amazon Redshift 스냅샷 및 백업](#) 섹션을 참조하세요.

클러스터 액세스 및 보안

Amazon Redshift에는 클러스터 액세스 및 보안과 관련된 기능이 몇 가지 있습니다. 이러한 기능들은 클러스터에 대한 액세스를 제어하고, 연결 규칙을 정의하고, 데이터 및 연결을 암호화하는 데 효과적입니다. 또한 Amazon Redshift의 데이터베이스 액세스 및 보안 관련 기능만으로 그치지 않습니다. 데이터베이스 보안에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [데이터베이스 보안 관리](#) 섹션을 참조하세요.

AWS 계정 및 IAM 자격 증명

기본적으로 Amazon Redshift 클러스터는 클러스터를 생성한 AWS 계정에서만 액세스가 가능합니다. 그 밖의 다른 계정에서는 액세스할 수 없도록 클러스터가 잠깁니다. AWS 계정에서는 AWS Identity and Access Management(IAM) 서비스를 사용하여 사용자 계정을 생성하거나, 혹은 클러스터 작업

에 대한 계정의 제어 권한을 관리할 수 있습니다. 자세한 내용은 [Amazon Redshift의 보안](#) 섹션을 참조하세요. IAM 역할에 대한 지침 및 모범 사례를 비롯한 IAM ID 관리에 대한 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#) 섹션을 참조하세요.

보안 그룹

기본적으로 생성하는 모든 클러스터는 모든 사용자에게 닫힙니다. IAM 자격 증명은 Amazon Redshift 콘솔, 명령줄 인터페이스(CLI), API 및 SDK와 같은 Amazon Redshift API 관련 리소스에 대한 액세스만 제어합니다. SQL 클라이언트 도구에서 JDBC 또는 ODBC를 통해 클러스터에 액세스하기 위해서는 다음과 같이 보안 그룹을 사용해야 합니다.

- Amazon Redshift 클러스터에 EC2-VPC 플랫폼을 사용하는 경우에는 VPC 보안 그룹을 사용해야 합니다. EC2-VPC 플랫폼에서 클러스터를 시작하는 것이 좋습니다.

EC2-Classic을 사용하여 클러스터를 시작한 후 클러스터를 VPC로 이동할 수 없습니다. 그러나 Amazon Redshift 콘솔을 사용하여 EC2-Classic 스냅샷을 EC2-VPC 클러스터로 복원할 수 있습니다. 자세한 내용은 [스냅샷에서 클러스터 복원](#) 섹션을 참조하세요.

- Amazon Redshift 클러스터에 EC2-Classic 플랫폼을 사용하는 경우에는 Amazon Redshift 보안 그룹을 사용해야 합니다.

위의 두 경우 모두 특정 범위의 CIDR/IP 주소에 명시적인 인바운드 액세스 권한을 부여하는 규칙을 보안 그룹에 또는 Amazon EC2 인스턴스에서 SQL 클라이언트를 실행하는 경우에는 Amazon Elastic Compute Cloud(Amazon EC2) 보안 그룹에 추가합니다. 자세한 내용은 [Amazon Redshift 보안 그룹](#) 섹션을 참조하세요.

인바운드 액세스 규칙 외에도 데이터베이스 사용자를 생성하여 클러스터 자체의 데이터베이스 인증을 위한 자격 증명을 제공합니다. 자세한 내용은 이번 주제에서 전반부 [데이터베이스 수](#) 섹션을 참조하세요.

암호화

클러스터를 프로비저닝할 때 옵션으로 클러스터 암호화를 선택하여 보안을 강화할 수 있습니다. 암호화를 사용하면 Amazon Redshift가 모든 데이터를 암호화하여 사용자 생성 테이블에 저장합니다. AWS Key Management Service(AWS KMS)를 사용하여 Amazon Redshift 암호화 키를 관리할 수 있습니다.

암호화는 변경 불가능한 클러스터 속성입니다. 암호화된 클러스터를 암호화되지 않은 클러스터로 전환하려면 데이터를 언로드한 후 새로운 클러스터에 다시 로드하는 방법이 유일합니다. 암호화는 클러스터와 모든 백업에 적용됩니다. 암호화된 스냅샷에서 클러스터를 복원하면 복원된 클러스터 역시 암호화됩니다.

암호화, 키 및 하드웨어 보안 모듈에 대한 자세한 내용은 [Amazon Redshift 데이터베이스 암호화](#) 섹션을 참조하세요.

SSL 연결

SSL(Secure Sockets Layer) 암호화를 사용하여 SQL 클라이언트와 클러스터 사이의 연결을 암호화할 수 있습니다. 자세한 내용은 [연결을 위한 보안 옵션 구성](#) 섹션을 참조하세요.

클러스터 모니터링

Amazon Redshift에는 모니터링과 관련된 기능이 몇 가지 있습니다. 데이터베이스 감사 로깅을 사용하여 활동 로그를 생성하고 이벤트 및 알림 구독을 구성하여 관심 있는 정보를 추적할 수 있습니다. 클러스터와 데이터베이스의 상태 및 성능에 대해 알아보려면 Amazon Redshift 및 Amazon CloudWatch의 지표를 사용합니다.

데이터베이스 감사 로깅

데이터베이스 감사 로깅 기능을 사용하여 인증 시도 횟수, 연결, 차단, 데이터베이스 사용자 정의 변경, 데이터베이스에서 실행된 쿼리 등에 대한 정보를 추적할 수 있습니다. 이러한 정보는 Amazon Redshift에서 보안 및 문제 해결 용도로 사용하는 데 유용합니다. 로그는 Amazon S3 버킷에 저장됩니다. 자세한 내용은 [데이터베이스 감사 로깅](#) 섹션을 참조하세요.

이벤트 및 알림

Amazon Redshift는 이벤트를 추적하여 이벤트 관련 정보를 AWS 계정에 몇 주간 저장합니다. 또한 이벤트가 있을 때마다 Amazon Redshift는 이벤트 발생 날짜, 설명, 이벤트 소스(클러스터, 파라미터 그룹 또는 스냅샷), 소스 ID 등의 정보를 보고합니다. 이벤트 필터 집합을 지정하여 Amazon Redshift 이벤트 알림 구독을 생성할 수 있습니다. 이후 필터 기준과 일치하는 이벤트가 발생하면 Amazon Redshift가 Amazon Simple Notification Service를 사용하여 이벤트 발생 사실을 알립니다. 이벤트에 대한 자세한 내용은 [Amazon Redshift 이벤트](#) 섹션을 참조하세요.

성능

Amazon Redshift는 클러스터와 데이터베이스의 상태 및 성능을 추적할 수 있도록 성능 지표와 데이터를 제공합니다. Amazon Redshift는 Amazon CloudWatch 지표를 사용하여 CPU 사용량, 지연 시간, 처리량 등 클러스터의 물리적 요소를 모니터링합니다. Amazon Redshift는 또한 클러스터의 데이터베이스 활동을 모니터링하는 데 도움이 되는 쿼리 및 로드 성능 데이터를 제공합니다. 성능 지표 및 모니터링에 대한 자세한 내용은 [Amazon Redshift 클러스터 성능 모니터링](#) 섹션을 참조하세요.

데이터베이스 수

Amazon Redshift는 클러스터를 프로비저닝할 때 데이터베이스를 1개 생성합니다. 이때 생성된 데이터베이스가 데이터를 로드하거나, 데이터에 대한 쿼리를 실행할 때 사용됩니다. 그 밖에 데이터베이스가 추가로 필요하다면 SQL 명령을 실행하여 생성할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [1단계: 데이터베이스 생성](#)을 참조하세요.

클러스터를 프로비저닝할 때는 클러스터에 생성되는 모든 데이터베이스에 액세스할 수 있는 관리자를 지정합니다. 이 관리자는 처음부터 데이터베이스 액세스 권한을 갖는 유일한 사용자이자 슈퍼 사용자입니다. 단, 관리자는 슈퍼 사용자와 사용자를 추가로 생성할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [슈퍼 사용자](#)와 [사용자](#)를 참조하세요.

Amazon Redshift는 파라미터 그룹을 사용하여 데이터 표현 방식이나 부동 소수점 정밀도 같이 클러스터에 속한 모든 데이터베이스의 동작을 정의합니다. 클러스터를 프로비저닝하면서 파라미터 그룹을 지정하지 않으면 Amazon Redshift가 기본 파라미터 그룹을 클러스터와 연결합니다. 자세한 내용은 [Amazon Redshift 파라미터 그룹](#) 섹션을 참조하세요.

Amazon Redshift의 데이터베이스에 대한 자세한 내용은 [Amazon Redshift 데이터베이스 개발자 안내서](#)를 참조하세요.

Amazon Redshift Serverless를 Amazon Redshift 프로비저닝된 데이터 웨어하우스와 비교

Amazon Redshift Serverless는 일부 개념 및 기능이 Amazon Redshift Redshift 프로비저닝된 데이터 웨어하우스와 다릅니다. 예를 들어, Amazon Redshift Serverless에는 클러스터나 노드의 개념이 없습니다. 다음 테이블에서는 Amazon Redshift Serverless의 기능과 동작을 설명하고 프로비저닝된 클러스터의 해당 기능과 비교하여 차이점을 설명합니다.

기능	설명	서버리스	프로비저닝됨
작업 그룹 및 네임스페이스	Amazon Redshift Serverless 내에서 워크로드를 격리하고 다양한 리	네임스페이스는 데이터베이스 객체와 사용자의 모임입니다. 작업 그	프로비저닝된 클러스터는 사용자가 직접 관리하는 컴퓨팅 노드와 리더 노드의 모임입니다. 자세한 내용은 Amazon Redshift 프로비저닝 클러스터 섹션을 참조하세요.

기능	설명	서버리스	프로비저닝됨
	소스를 관리하려면 네임스페이스와 작업 그룹을 생성하고 스토리지 리소스와 컴퓨팅 리소스를 별도로 관리하면 됩니다.	그룹은 컴퓨팅 리소스의 모음입니다. 자세한 내용을 알아보고 Amazon Redshift Serverless의 설계를 이해하려면 Amazon Redshift Serverless 를 참조하세요.	

기능	설명	서버리스	프로비저닝됨
노드 유형	Amazon Redshift Serverless를 사용하는 경우 프로비저닝된 Amazon Redshift 클러스터를 사용할 때와 마찬가지로 노드 유형을 선택하거나 노드 수를 지정하지 않습니다.	Amazon Redshift Serverless에서 자동으로 용량을 프로비저닝하고 관리합니다. 선택적으로 기본 데이터 웨어하우스 용량을 지정하여 워크로드에 적합한 가격과 성능의 균형을 선택할 수 있습니다. 또한 최대 RPU 시간을 지정하여 비용을 예측할 수 있도록 비용 제한을 설정할 수 있습니다. 자세한 내용은 Amazon Redshift Serverless	비용 및 성능 사양을 충족하는 노드 유형으로 클러스터를 구축합니다. 자세한 내용은 Amazon Redshift 프로비저닝 클러스터 섹션을 참조하세요.

기능	설명	서버리스	프로비저닝됨
		의 컴퓨팅 용량 섹션을 참조하세요.	
워크로드 관리 및 동시성 확장	Amazon Redshift는 부하가 높은 기간에도 확장할 수 있습니다. 또한 Amazon Redshift Serverless는 간헐적인 고부하 기간에 맞춰 확장할 수 있습니다.	Amazon Redshift Serverless는 자동으로 리소스를 효율적으로 관리하며 비용 제어 임계값 내에서 워크로드에 따라 확장합니다. 자세한 내용은 컴퓨팅 용량에 대한 결제 섹션을 참조하세요.	프로비저닝된 데이터 웨어하우스를 사용하면 클러스터에서 동시성 확장을 활성화하여 고부하 기간을 처리할 수 있습니다. 자세한 내용은 동시성 확장 을 참조하세요.

기능	설명	서버리스	프로비저닝됨
포트	연결하는 데 사용하는 포트 번호입니다.	Amazon Redshift Serverless를 사용하면 포트 범위 5431~5455 또는 8191~8215에서 다른 포트로 변경할 수 있습니다. 자세한 내용은 Amazon Redshift 서버리스에 연결 섹션을 참조하세요.	프로비저닝된 클러스터에서는 연결할 포트를 선택할 수 있습니다.

기능	설명	서버리스	프로비저닝됨
크기 조정	워크로드에 적합한 성능을 낼 수 있도록 컴퓨팅 리소스를 추가 또는 제거합니다.	Amazon Redshift Serverless에는 크기 조정이 적용되지 않습니다. 그러나 가격 및 성능 요구 사항에 따라 기본 데이터 웨어하우스 RPU 용량을 변경할 수 있습니다. 자세한 내용은 Amazon Redshift Serverless의 컴퓨팅 용량 섹션을 참조하세요.	프로비저닝된 클러스터에서는 클러스터 크기 조정을 수행하여 노드를 추가하거나 제거할 수 있습니다. 자세한 내용은 Amazon Redshift에서 클러스터 관리 개요 를 참조하세요.

기능	설명	서버리스	프로비저닝됨
일시 중지 및 다시 시작	실행할 워크로드가 없을 때는 프로비저닝된 클러스터를 일시 중지하여 비용을 절감할 수 있습니다.	Amazon Redshift Serverless를 사용하는 경우 쿼리 실행에 대해서만 비용을 결제하므로 일시 중지하거나 다시 시작할 필요가 없습니다. 자세한 내용은 컴퓨팅 용량에 대한 결제 섹션을 참조하세요.	다양한 시간에서의 워크로드 평가를 기반으로 클러스터를 수동으로 일시 중지하고 다시 시작할 수 있습니다. 자세한 내용은 Amazon Redshift에서 클러스터 관리 개요 를 참조하세요.

기능	설명	서버리스	프로비저닝됨
스펙트럼 쿼리를 사용하여 외부 데이터 쿼리	Amazon S3 버킷의 데이터를 JSON과 같은 다양한 형식으로 쿼리할 수 있습니다.	컴퓨팅 리소스가 워크로드를 처리할 때는 요금이 발생합니다. 다른 트랜잭션과 마찬가지로 Redshift Spectrum 데이터가 쿼리될 때 요금이 발생합니다. 자세한 내용은 컴퓨팅 용량에 대한 결제 섹션을 참조하세요.	프로비저닝된 데이터 웨어하우스에서는 Amazon Redshift Spectrum 용량이 Amazon Redshift 클러스터에서 쿼리되는 별도의 서버에 존재합니다. 자세한 내용은 Amazon Redshift Spectrum을 사용하여 외부 데이터 쿼리 를 참조하세요.

기능	설명	서버리스	프로비저닝됨
컴퓨팅 리소스 요금	Amazon Redshift와 Amazon Redshift Serverless의 요금 청구 금액 비교.	Amazon Redshift에서는 실행하는 워크로드에 대해 초 단위의 RPU 시간 당 비용이 결제되며, 최소 요금 단위는 60 초입니다. 여기에는 Amazon S3에서 열린 파일 형식의 데이터에 액세스하는 쿼리가 포함됩니다. 자세한 내용은 컴퓨팅 용량에 대한 결제 섹션을 참조하세요.	프로비저닝된 클러스터에서는 클러스터가 일시 중지되지 않을 때 초당 요금이 발생합니다.

기능	설명	서버리스	프로비저닝됨
유지 관리 윈도우	서버 유지 관리 작동 방식.	Amazon Redshift Serverless 에는 유지 관리 기간 이 없습니 다. 업데이 트는 원활 하게 처리 됩니다. 자 세한 내용 은 Amazon Redshift Serverless 란 무엇인 가요? 를 참 조하세요.	프로비저닝된 클러스터에서는 패치가 발생할 때 유지 관리 기간을 지정합니다. 일반적으로 사용량이 적을 때 반복 시간을 선택합니다.
암호화	데이터베이 스 암호화 를 활성화 할 수 있습 니다.	Amazon Redshift Serverless는 항 상 AWS KMS(AWS 관리형 또 는 고객 관 리형 키 사 용)를 사용 하여 암호 화됩니다.	프로비저닝된 데이터 웨어하우스의 데이터는 AWS KMS(AWS 관리형 또는 고객 관리형 키 사용)를 사용하 여 암호화될 수도 있고 암호화되지 않을 수도 있습니다. Amazon Redshift 데이터베이스 암호화 을(를) 참조하세 요.

기능	설명	서버리스	프로비저닝됨
스토리지 요금	스토리지 요금 산정 방식.	Amazon Redshift Serverless. 요금은 월별 용량 (GB)에 따 라 계산됩 니다. 컬 퓨팅 용량 에 대한 결 제율(률) 참 조 하세요.	RA3 노드가 있는 프로비저닝된 클러스터에서 스토리지는 컴퓨팅 리소스와 별도 요금이 발생합니다.

기능	설명	서버리스	프로비저닝됨
사용자 관리	사용자 관리 방법.	Amazon Redshift Serverless에서 사용자는 IAM 또는 Redshift 사용자입니다. 자세한 내용은 Amazon Redshift Serverless의 Identity and Access Management 섹션을 참조하세요. IAM 역할에 대한 모범 사례를 비롯한 IAM ID 관리에 대한 자세한 내용은 Amazon Redshift의 Identity and Access Management 섹션을 참조하세요.	프로비저닝된 데이터 웨어하우스에서 사용자는 IAM 또는 Redshift 사용자입니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 데이터베이스 보안 관리를 참조하세요. IAM 역할에 대한 모범 사례를 비롯한 IAM ID 관리에 대한 자세한 내용은 Amazon Redshift의 Identity and Access Management 섹션을 참조하세요.

기능	설명	서버리스	프로비저닝됨
		t 섹션을 참조하세요.	

기능	설명	서버리스	프로비저닝됨
JDBC/ ODBC 도 구 및 호환 성	클라이언트 연결 작동 방식.	Amazon Redshift Serverless는 JDBC 또는 ODBC 호 환 도구 또 는 클라이 언트 애플 리케이션 과 호환됩 니다. 드 라이버에 대한 자세 한 내용은 Amazon Redshift 관리 가 이드의 연 결 구성 섹 션을 참조 하세요. Amazon Redshift Serverless 에 연결하 는 방법에 대한 자세 한 내용은 Redshift 서 버리스에 연결 을 참 조하세요.	프로비저닝된 Amazon Redshift는 JDBC 또는 ODBC 호환 도구 또는 클라이언트 애플리케이션과 호환됩니 다. 드라이버에 대한 자세한 내용은 Amazon Redshift 관리 가이드의 연결 구성 섹션을 참조하세요. 클러스터 연결에 관한 자세한 내용은 SQL 클라이언트 도구를 사 용하여 Amazon Redshift 데이터 웨어하우스에 연결 을 참조하세요.

기능	설명	서버리스	프로비저닝됨
로그인 시 보안 인증 요구	보안 인증 정보 처리 방법.	Amazon Redshift Serverless의 경우 모든 인스턴스에서 보안 인증 정보를 입력할 필요가 없습니다. 자세한 내용은 Amazon Redshift 서버리스에 연결 섹션을 참조하세요.	Amazon Redshift에 액세스하려면 IAM 역할과 연결된 사용자의 로그인 자격 증명이 필요합니다. IAM 역할에는 프로비저닝된 데이터 웨어하우스에 대한 특권한이 부여됩니다. 인증된 사용자는 데이터베이스, Redshift 콘솔 및 쿼리 편집기 v2에 직접 연결할 수 있습니다.

기능	설명	서버리스	프로비저닝됨
데이터 API	웹 서비스 및 기타 애플리케이션에서 데이터에 액세스할 수 있습니다.	Amazon Redshift Serverless는 Amazon Redshift Data API를 지원합니다. Amazon Redshift Serverless에서는 cluster-identity 파라미터 대신 workgroup-name 파라미터를 사용합니다. 데이터 API 호출에 대한 자세한 내용은 Amazon Redshift Data API 사용 섹션을 참조하세요.	프로비저닝된 Amazon Redshift는 Amazon Redshift Data API를 지원합니다. Amazon Redshift 클러스터에서는 workgroup-name 파라미터 대신 cluster-identity 파라미터를 사용합니다. 데이터 API 호출에 대한 자세한 내용은 Amazon Redshift Data API 사용 섹션을 참조하세요.

기능	설명	서버리스	프로비저닝됨
스냅샷	특정 시점으로 복구 기능을 제공합니다.	Amazon Redshift Serverless에서는 스냅샷 및 복구 시점을 지원합니다. 네임스페이스의 스냅샷 및 복구 시점에 대한 자세한 내용은 스냅샷 및 복구 시점 섹션을 참조하세요.	프로비저닝된 클러스터는 스냅샷을 지원합니다. 자세한 내용은 콘솔을 사용한 클러스터 관리 를 참조하세요.

기능	설명	서버리스	프로비저닝됨
데이터 공유	동일한 계정 또는 다른 계정의 데이터베이스 간에 데이터를 공유하는 기능을 제공합니다.	Amazon Redshift Serverless에서는 프로비저닝된 데이터 웨어하우스가 수행하는 모든 데이터 공유 기능을 지원합니다. Amazon Redshift Serverless와 프로비저닝된 데이터 웨어하우스, 도구 또는 클라이언트 애플리케이션 간의 데이터 공유도 지원합니다.	프로비저닝된 클러스터에서는 교차 데이터베이스, 교차 계정, 교차 리전 및 AWS Data Exchange 데이터 공유를 지원합니다. 자세한 내용은 Amazon Redshift에서 클러스터 간 데이터 공유 를 참조하세요.

기능	설명	서버리스	프로비저닝됨
트랙	소프트웨어 업데이트 일정을 제공합니다.	Amazon Redshift Serverless에는 트랙에 대한 개념이 없습니다. 버전 및 업데이트는 서비스가 처리합니다. Amazon Redshift Serverless의 설계에 대한 자세한 내용은 스냅샷 및 복구 시점을 참조 하세요.	프로비저닝된 클러스터에서는 현재 트랙과 후행 트랙 간 전환을 지원합니다.

기능	설명	서버리스	프로비저닝됨
시스템 테이블 및 뷰	리소스 및 시스템 메타데이터를 모니터링하는 방법을 제공합니다.	Amazon Redshift Serverless에서는 시스템 테이블 및 뷰를 지원합니다. 시스템 테이블에 대한 자세한 내용은 Amazon Redshift Serverless로 쿼리 및 워크로드 모니터링 섹션을 참조하세요. 이전에 프로비저닝된 시스템 테이블 및 뷰를 사용하던 쿼리를 새 뷰를 사용하도록 마이그레이션하는 방법에 대한 자세한 내용은 SYS 모니터링 뷰로 마이	프로비저닝된 데이터 웨어하우스는 모니터링 및 시스템 메타데이터가 필요한 기타 작업에 대한 기존 시스템 테이블 및 뷰 세트를 지원합니다.

기능	설명	서버리스	프로비저닝됨
		그레이션 을 참조하세요.	
파라미터 그룹	이것은 클러스터에서 생성되는 모든 데이터베이스에 적용되는 파라미터 그룹입니다. 이러한 파라미터가 쿼리 제한 시간이나 날짜 스타일 같은 데이터베이스 설정을 구성합니다.	Amazon Redshift Serverless에는 파라미터 그룹의 개념이 없습니다.	프로비저닝된 데이터 웨어하우스는 파라미터 그룹을 지원합니다. 프로비저닝된 클러스터의 파라미터 그룹에 대한 자세한 내용은 Amazon Redshift 파라미터 그룹 섹션을 참조하세요.

기능	설명	서버리스	프로비저닝됨
쿼리 모니터링	실행된 쿼리에 대한 시간 기반 보기를 제공합니다.	Amazon Redshift Serverless에 대한 쿼리 모니터링을 수행하려면 사용자가 데이터베이스에 연결하여 시스템 테이블을 사용해야 합니다. 따라서 쿼리 모니터링 및 시스템 테이블이 동기화됩니다. Amazon Redshift Serverless에 대한 시스템 테이블 쿼리는 쿼리 모니터링을 사용하기 위해 IAM 사용자에게 매핑된 데이터베이스 사용자	프로비저닝된 클러스터의 쿼리 모니터링은 시스템 테이블의 모든 데이터를 표시하지는 않습니다.

기능	설명	서버리스	프로비저닝됨
		를 사용합니다. 쿼리 모니터링에 대한 자세한 내용은 Amazon Redshift Serverless로 쿼리 및 워크로드 모니터링을 참조 하십시오.	

기능	설명	서버리스	프로비저닝됨
감사 로깅	연결 및 사용자 작업에 대한 정보를 데이터베이스에 기록합니다.	Amazon Redshift Serverless의 경우 CloudWatch가 감사 로그의 대상입니다. Amazon S3 기반의 감사 로그 전달은 Amazon Redshift Serverless에서 지원되지 않습니다. 자세한 내용은 Amazon Redshift Serverless용 감사 로깅을 참조 하세요.	프로비저닝된 클러스터의 경우 Amazon S3 기반의 감사 로그 전달이 표준이었습니다. 이제 CloudWatch로의 감사 로그 전달이 프로비저닝된 데이터 웨어하우스까지 확대 적용됩니다.

기능	설명	서버리스	프로비저닝됨
이벤트 알림	Amazon EventBridge는 애플리케이션을 다양한 소스의 이벤트 데이터와 연결하는 데 사용할 수 있는 서버리스 이벤트 버스 서비스입니다.	Amazon Redshift Serverless는 Amazon EventBridge를 통해 이벤트 알림을 관리하여 데이터 웨어하우스의 변경 사항에 대한 최신 정보를 제공합니다. 자세한 내용은 Amazon EventBridge로 Amazon Redshift Serverless 이벤트 알림 섹션을 참조하세요.	프로비저닝된 클러스터의 경우, Amazon Redshift 콘솔을 사용하여 이벤트 알림을 관리하고 이벤트 구독을 생성할 수 있습니다. 자세한 내용은 이벤트 알림 구독 생성 섹션을 참조하세요.

기능	설명	서버리스	프로비저닝됨
커서 제약 조건	Amazon Redshift는 모든 커서 결과 세트의 크기에 제약을 적용합니다.	Amazon Redshift Serverless의 커서 최대 총 결과 세트 크기는 150,000MB입니다.	프로비저닝된 클러스터의 경우 커서 총 결과 세트의 최대 크기는 클러스터 유형에 따라 다릅니다. 자세한 내용은 커서 제약 조건 을 참조하세요.

프로비저닝된 클러스터에 Amazon Redshift 관리 인터페이스 사용

Note

이 주제에서는 프로비저닝된 클러스터에 대한 Amazon Redshift 관리 인터페이스에 중점을 둡니다. Amazon Redshift Serverless와 Amazon Redshift Data API는 관리 인터페이스가 유사합니다.

Amazon Redshift는 AWS SDK, AWS Command Line Interface(AWS CLI), Amazon Redshift 관리 API 등 Amazon Redshift 클러스터를 생성, 관리 및 삭제하는 데 사용할 수 있는 여러 관리 인터페이스를 지원합니다.

Amazon Redshift API – 요청을 제출하여 이 Amazon Redshift 관리 API를 호출할 수 있습니다. 요청은 이름이 GET인 파라미터와 함께 HTTP 동사 POST 또는 Action를 사용하는 HTTP 또는 HTTPS 요청입니다. Amazon Redshift API 호출은 Amazon Redshift 서비스에 액세스할 수 있는 가장 직접적인 방법입니다. 그러나 이 방법을 사용하려면 애플리케이션에서 오류 처리 및 요청에 서명할 해시 생성 등과 같은 하위 수준의 세부 정보를 처리해야 합니다.

- Amazon Redshift API 요청 작성 및 서명에 대한 자세한 내용은 [HTTP 요청 서명](#) 섹션을 참조하세요.
- Amazon Redshift API 작업 및 데이터 형식에 대한 자세한 내용은 [Amazon Redshift API Reference](#)를 참조하세요.

AWS SDK – AWS SDK를 사용하여 Amazon Redshift 클러스터 관련 작업을 수행할 수 있습니다. 일부 SDK 라이브러리에는 기본적인 Amazon Redshift API가 래핑되어 있습니다. 이러한 라이브러리들은 API 기능을 특정 프로그래밍 언어로 통합하여 서명 계산, 요청 재시도 횟수 처리, 오류 처리 등 비교적 중요하지 않은 세부 정보를 다수 처리합니다. SDK 라이브러리에서 래퍼(wrapper) 함수를 호출하면 Amazon Redshift 클러스터 관리용 애플리케이션의 개발 프로세스가 크게 간소화됩니다.

- Amazon Redshift는 Java, .NET, PHP, Python, Ruby 및 Node.js용 AWS SDK에서 지원됩니다. Amazon Redshift에서 사용되는 래퍼 함수는 각 SDK의 참조 설명서에 기록되어 있습니다. AWS SDK 목록과 각 설명서 링크는 [Amazon Web Services용 도구](#)에서 확인할 수 있습니다.
- 본 안내서는 Java SDK를 사용한 Amazon Redshift 작업 예를 제공하고 있습니다. 보다 일반적인 AWS SDK 코드 예시는 [AWS SDK를 사용한 Amazon Redshift용 코드 예제](#) 섹션을 참조하세요.

AWS CLI – CLI는 Windows, Mac 및 Linux 컴퓨터에서 AWS 서비스를 관리하는 데 사용할 수 있는 일련의 명령줄 도구를 제공합니다. AWS CLI에는 Amazon Redshift API 작업에 따른 명령이 포함되어 있습니다.

- Amazon Redshift CLI 설치 및 설정에 대한 자세한 내용은 [Amazon Redshift CLI 설정](#) 섹션을 참조하세요.
- Amazon Redshift CLI 명령에 대한 참조 자료는 AWS CLI Reference의 [Amazon Redshift](#)를 참조하세요.

AWS SDK와 함께 이 서비스 사용

다양한 프로그래밍 언어에 대해 AWS 소프트웨어 개발 키트(SDK)를 사용할 수 있습니다. 각 SDK는 개발자가 선호하는 언어로 애플리케이션을 쉽게 구축할 수 있도록 하는 API, 코드 예제 및 설명서를 제공합니다.

SDK 설명서	코드 예제
AWS SDK for C++	AWS SDK for C++ 코드 예제
AWS CLI	AWS CLI 코드 예제
AWS SDK for Go	AWS SDK for Go 코드 예제
AWS SDK for Java	AWS SDK for Java 코드 예제

SDK 설명서	코드 예제
AWS SDK for JavaScript	AWS SDK for JavaScript 코드 예제
AWS SDK for Kotlin	AWS SDK for Kotlin 코드 예제
AWS SDK for .NET	AWS SDK for .NET 코드 예제
AWS SDK for PHP	AWS SDK for PHP 코드 예제
AWS Tools for PowerShell	AWS Tools for PowerShell 코드 예제
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) 코드 예제
AWS SDK for Ruby	AWS SDK for Ruby 코드 예제
AWS SDK for Rust	AWS SDK for Rust 코드 예제
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP 코드 예제
AWS SDK for Swift	AWS SDK for Swift 코드 예제

예제 사용 가능 여부

필요한 예제를 찾을 수 없습니까? 이 페이지 하단의 피드백 제공 링크를 사용하여 코드 예제를 요청합니다.

HTTP 요청 서명

Amazon Redshift는 관리 API에 보내는 요청마다 서명을 통해 인증을 받도록 합니다. 이번 주제에서는 요청의 서명 방법에 대해서 설명합니다.

AWS 소프트웨어 개발 키트(SDK) 중 하나 또는 AWS Command Line Interface를 사용하는 경우에는 요청 서명이 자동으로 처리되기 때문에 이번 섹션을 건너뛰어도 좋습니다. AWS SDK 사용에 대한 자세한 내용은 [프로비저닝된 클러스터에 Amazon Redshift 관리 인터페이스 사용](#) 섹션을 참조하세요. Amazon Redshift 명령줄 인터페이스 사용에 대한 자세한 내용은 [Amazon Redshift 명령줄 레퍼런스](#) 섹션을 참조하세요.

요청에 서명하려면 암호화 해시 함수를 이용해 디지털 서명을 계산해야 합니다. 암호화 해시는 입력을 근거로 하여 고유 해시 값을 반환하는 함수입니다. 해시 함수에 대한 입력에는 요청 텍스트와 임시 보안 인증 정보를 통해 얻은 비밀 액세스 키가 포함됩니다. 해시 함수는 요청에 서명으로 포함하는 해시 값을 반환합니다. 서명은 요청에서 Authorization 헤더의 일부입니다.

Note

사용자가 AWS Management Console 외부에서 AWS와 상호 작용하려면 프로그래밍 방식의 액세스 권한이 필요합니다. 프로그래밍 방식의 액세스 권한을 부여하는 방법은 AWS에 액세스 하는 사용자 유형에 따라 다릅니다.

사용자에게 프로그래밍 방식의 액세스 권한을 부여하려면 다음 옵션 중 하나를 선택합니다.

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
IAM	(권장됨) 콘솔 자격 증명을 임시 자격 증명으로 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. - AWS CLI의 경우 AWS Command Line Interface 사용 설명서의 AWS 로컬 개발을 위한 로그인을 참조하세요. - AWS SDK의 경우 AWS SDK 및 도구 참조 안내서의 AWS 로컬 개발을 위한 로그인을 참조하세요.
작업 인력 ID (IAM Identity Center에서 관리되는 사용자)	임시 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. AWS CLI에 대해서는 AWS Command Line Interface 사용자 안내서에서 AWS IAM Identity Center을 사용하도록 AWS CLI 구성을 참조하세요.

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
		AWS SDK, 도구, AWS API에 대해서는 AWS SDK 및 도구 참조 가이드에서 IAM Identity Center 인증을 참조하세요.
IAM	임시 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	IAM 사용자 설명서의 AWS 리소스와 함께 임시 자격 증명 사용 에 나와 있는 지침을 따르세요.
IAM	(권장되지 않음) 장기 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. - AWS CLI에 대해서는 AWS Command Line Interface 사용자 안내서에서 IAM 사용자 자격 증명을 사용한 인증을 참조하세요. - AWS SDK와 도구는 AWS SDK 및 도구 참조 가이드에서 장기 자격 증명을 사용한 인증을 참조하세요. - AWS API는 IAM 사용자 설명서에서 IAM 사용자의 액세스 키 관리를 참조하세요.

Amazon Redshift는 요청을 수신한 후 본인이 요청에 서명할 때 사용한 것과 동일한 해시 함수 및 입력을 사용하여 서명을 재계산합니다. 그 결과 얻은 서명이 요청의 서명과 일치할 경우 Amazon Redshift가 요청을 처리합니다. 그렇지 않으면 요청이 거부됩니다.

Amazon Redshift는 [AWS 서명 버전 4](#)를 이용한 인증을 지원합니다. 서명을 계산하기 위한 프로세스는 다음 세 작업으로 구성됩니다. 이 작업들은 아래 예에도 설명되어 있습니다.

- [작업 1: 표준 요청 생성](#)

HTTP 요청을 정규 형식으로 재배열합니다. 정식 양식을 사용해야 하는 이유는 Amazon Redshift가 서명을 계산하여 본인이 전송한 서명과 비교할 때 동일한 정식 양식을 사용하기 때문입니다.

- [작업 2: 서명할 문자열 생성](#)

암호화 해시 함수에 대한 입력 값 중 하나로 사용할 문자열을 만듭니다. 서명할 문자열이라는 문자열은 해시 알고리즘의 이름, 요청 날짜, 자격 증명 범위 문자열, 이전 작업에서 정규화된 요청을 연결한 것입니다. 자격 증명 범위 문자열 자체는 날짜, 리전 및 서비스 정보를 연결한 것입니다.

- [작업 3: 서명 계산](#)

서명할 문자열과 파생된 키라는 두 개의 입력 문자열을 허용하는 암호화 해시 함수를 사용하여 요청에 대한 서명을 계산합니다. 파생된 키는 보안 액세스 키로 시작해서 자격 증명 범위 문자열을 사용하여 일련의 해시 기반 메시지 인증 코드(HMAC)를 만들어 계산됩니다.

서명 계산 예시

다음 예시에서는 [CreateCluster](#)에 대해 서명을 생성하는 세부 과정을 안내합니다. 이 예는 서명 계산 방법을 점검하기 위한 참조로 사용할 수 있습니다. 기타 참조 계산은 IAM 사용 설명서의 [서명 요청 예제 섹션](#)에 포함되어 있습니다.

GET 또는 POST 요청을 사용하여 Amazon Redshift에 요청을 전송할 수 있습니다. 두 가지의 차이는, GET 요청의 경우 파라미터가 쿼리 문자열 파라미터로 전송되는 반면 POST 요청의 경우에는 요청 본문에 포함된다는 데 있습니다. 아래 예는 POST 요청을 나타낸 것입니다.

이 예시에서는 다음과 같이 가정합니다.

- 요청 타임스탬프는 Fri, 07 Dec 2012 00:00:00 GMT입니다.
- 엔드포인트는 미국 동부(버지니아 북부) 리전인 us-east-1입니다.

일반적인 요청 구문은 다음과 같습니다.

```
https://redshift.us-east-1.amazonaws.com/
?Action=CreateCluster
&ClusterIdentifier=examplecluster
&MasterUsername=masteruser
&MasterUserPassword=12345678Aa
&NumberOfNode=2
&NodeType=dc2.large
&Version=2012-12-01
&x-amz-algorithm=AWS4-HMAC-SHA256
&x-amz-credential=AKIAIOSFODNN7EXAMPLE/20121207/us-east-1/redshift/aws4_request
&x-amz-date=20121207T000000Z
&x-amz-signedheaders=content-type;host;x-amz-date
```

[작업 1: 표준 요청 생성](#)에서 계산되는 요청의 정규 형식은 다음과 같습니다.

```
POST
/

content-type:application/x-www-form-urlencoded; charset=utf-8
host:redshift.us-east-1.amazonaws.com
x-amz-date:20121207T000000Z

content-type;host;x-amz-date
55141b5d2aff6042ccd9d2af808fdf95ac78255e25b823d2dbd720226de1625d
```

표준 요청의 마지막 줄은 요청 본문의 해시입니다. 정식 요청에서 세 번째 줄은 이 API에 대한 쿼리 파라미터가 없기 때문에 비어있습니다.

[작업 2: 서명할 문자열 생성](#)에서 서명할 문자열은 다음과 같습니다.

```
AWS4-HMAC-SHA256
20121207T000000Z
20121207/us-east-1/redshift/aws4_request
06b6bef4f4f060a5558b60c627cc6c5b5a959b9902b5ac2187be80cbac0714
```

서명할 문자열의 첫째 줄은 알고리즘, 둘째 줄은 타임스탬프, 셋째 줄은 자격 증명 범위, 마지막 줄은 [작업 1: 표준 요청 생성](#)의 표준 요청에 대한 해시입니다. 자격 증명 범위에서 사용하는 서비스 이름은 redshift입니다.

[작업 3: 서명 계산](#)에서 파생된 키는 다음과 같이 표현할 수 있습니다.

```
derived key = HMAC(HMAC(HMAC(HMAC("AWS4" + YourSecretAccessKey, "20121207"), "us-east-1"), "redshift"), "aws4_request")
```

파생된 키는 연속된 해시 함수로 계산됩니다. 위 공식의 내부 HMAC 문부터 시작하여 구문 **AWS4**를 비밀 액세스 키와 병합한 후 이것을 데이터 "us-east-1"을 해시 처리하는 키로 사용합니다. 이러한 해시 결과가 다음 해시 함수의 키가 됩니다.

파생된 키에 대한 계산을 마친 후에는 계산된 결과를 두 입력 문자열인 서명할 문자열과 파생된 키를 허용하는 해시 함수에 사용합니다. 예를 들어 앞선 내용을 고려하여 보안 액세스 키 wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY와 서명할 문자열을 사용하는 경우 계산된 서명은 다음과 같습니다.

```
9a6b557aa9f38dea83d9215d8f0eae54100877f3e0735d38498d7ae489117920
```

마지막 단계는 Authorization 헤더를 생성하는 것입니다. 데모용 액세스 키 AKIAIOSFODNN7EXAMPLE의 경우 헤더는 다음과 같습니다(가독성을 높이기 위해 줄 바꿈을 추가함).

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20121207/us-east-1/redshift/aws4_request,
SignedHeaders=content-type;host;x-amz-date,
Signature=9a6b557aa9f38dea83d9215d8f0eae54100877f3e0735d38498d7ae489117920
```

Amazon Redshift CLI 설정

이번 섹션에서는 Amazon Redshift 관리에 사용할 목적으로 AWS CLI 명령줄 도구를 설정하고 실행하는 방법에 대해 설명합니다. Amazon Redshift 명령줄 도구는 AWS Command Line Interface(AWS CLI)에서 실행되므로 Python(<https://www.python.org/>)을 사용합니다. AWS CLI는 Python만 지원한다면 어떤 운영 체제에서든지 실행할 수 있습니다.

AWS Command Line Interface 설치

Amazon Redshift 명령줄 도구를 사용하려면 먼저 AWS CLI를 설정한 다음 Amazon Redshift CLI 옵션을 정의하는 구성 파일을 추가합니다.

이미 다른 AWS CLI 서비스에서 AWS를 설치 및 구성했다면 이번 단계를 건너뛰어도 좋습니다.

AWS Command Line Interface를 설치하려면

1. [최신 버전의 AWS CLI 설치 또는 업데이트](#)로 이동한 다음, AWS CLI 설치 지침을 따르세요.

CLI 액세스를 위해서는 액세스 키 ID 및 비밀 액세스 키가 필요합니다. 가능하다면 장기 액세스 키 대신 임시 보안 인증 정보를 사용하세요. 임시 보안 인증도 액세스 키 ID와 비밀 액세스 키로 구성되지만 보안 인증이 만료되는 시간을 나타내는 보안 토큰이 포함되어 있습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 리소스에서 임시 자격 증명 사용](#)을 참조하세요.

2. 액세스 키, 기본 리전, 명령 출력 형식 같은 구성 정보가 저장된 파일을 생성합니다. 그런 다음 해당 파일을 참조할 수 있는 `AWS_CONFIG_FILE` 환경 변수를 설정합니다. 자세한 지침은 AWS Command Line Interface 사용 설명서의 [AWS 명령줄 인터페이스 구성](#)을 참조하세요.
3. 테스트 명령을 실행하여 AWS CLI 인터페이스가 올바르게 실행되는지 확인합니다. 예를 들어 다음 명령을 실행하면 AWS CLI의 도움말 정보가 표시되어야 합니다.

```
aws help
```

다음 명령을 실행하면 Amazon Redshift의 도움말 정보가 표시되어야 합니다.

```
aws redshift help
```

Amazon Redshift CLI 명령에 대한 참조 자료는 AWS CLI Reference의 [Amazon Redshift](#)를 참조하세요.

Amazon Redshift Serverless

Amazon Redshift Serverless를 사용하면 온프레미스 데이터 웨어하우스를 프로비저닝 및 관리할 필요 없이 손쉽게 분석을 실행하고 규모를 조정할 수 있습니다. 이제 Amazon Redshift Serverless를 통해 데이터 분석가, 개발자 및 데이터 과학자가 Amazon Redshift를 사용하여 데이터를 클라우드의 데이터 웨어하우스에 로드하고 레코드를 쿼리하여 몇 초 만에 데이터에서 인사이트를 얻을 수 있습니다. Amazon Redshift는 데이터 웨어하우스 용량을 자동으로 프로비저닝하고 크기 조정하여 까다롭고 예측할 수 없는 워크로드의 성능을 향상합니다. 사용하는 용량에 대한 비용만 지불합니다. 기존 분석 및 비즈니스 인텔리전스 애플리케이션을 변경할 필요 없이 이 단순성의 이점을 얻을 수 있습니다.

Amazon Redshift Serverless 서비스 수준 계약(SLA)에 대한 정보는 [Amazon Redshift 서비스 수준 계약](#)을 참조하세요.

Amazon Redshift 서버리스란 무엇인가요?

Amazon Redshift Serverless는 데이터 웨어하우스 용량을 자동으로 프로비저닝하고 기본 리소스를 지능적으로 확장합니다. Amazon Redshift Serverless는 용량을 몇 초 만에 조정하여 가장 까다롭고 변동성이 큰 워크로드에 대해서도 일관되게 높은 성능과 단순화된 운영을 제공합니다.

Amazon Redshift Serverless를 사용하면 다음과 같은 기능을 활용할 수 있습니다.

- Amazon Redshift 프로비저닝된 클러스터를 설정, 조정 및 관리할 필요 없이 데이터에 액세스하고 분석합니다.
- 뛰어난 Amazon Redshift SQL 기능, 업계 최고의 성능 및 데이터 레이크 통합 기능을 사용하여 데이터 웨어하우스, 데이터 레이크 및 운영 데이터 소스 전반에 걸쳐 원활하게 쿼리합니다.
- 지능적인 자동 크기 조정을 통해 가장 까다롭고 변동성이 큰 워크로드에 대해서도 일관되게 고성능과 단순화된 운영을 제공합니다.
- 작업 그룹 및 네임스페이스를 사용하여 세분화된 비용 제어를 통해 컴퓨팅 리소스와 데이터를 구성할 수 있습니다.
- 데이터 웨어하우스를 사용할 때만 비용을 지불합니다.

Amazon Redshift Serverless를 사용하면 콘솔 인터페이스로 서버리스 데이터 웨어하우스 또는 API에 연결하여 애플리케이션을 구축할 수 있습니다. 데이터 웨어하우스를 통해 Amazon Redshift 관리형 스토리지와 Amazon S3 데이터 레이크에 액세스할 수 있습니다.

이 동영상에서는 데이터 웨어하우스 인프라를 관리할 필요 없이 손쉽게 분석을 실행하고 크기 조정할 수 있는 Amazon Redshift Serverless를 소개합니다.

Amazon Redshift Serverless 콘솔

Amazon Redshift Serverless 콘솔 사용을 시작하려면 [Amazon Redshift Serverless 시작하기](#) 동영상을 시청하세요.

서버리스 대시보드

서버리스 대시보드(Serverless dashboard) 페이지에서 리소스 요약과 사용량 그래프를 볼 수 있습니다.

- 네임스페이스 개요(Namespace overview) - 이 섹션에는 네임스페이스 내의 스냅샷 및 데이터 공유 양이 표시됩니다.
- 작업 그룹(Workgroups) - 이 섹션에는 Amazon Redshift Serverless 내의 모든 작업 그룹이 표시됩니다.
- 쿼리 지표(Queries metrics) - 이 섹션에는 지난 1시간 동안의 쿼리 활동이 표시됩니다.
- 사용된 RPU 용량(RPU capacity used) - 이 섹션에는 지난 1시간 동안 사용된 용량이 표시됩니다.
- 무료 평가판(Free trial) - 이 섹션에는 AWS 계정에 남아 있는 무료 평가판 크레딧이 표시됩니다. 여기에는 동일한 계정에서 스냅샷, 스토리지, 작업 그룹 등을 비롯한 Amazon Redshift Serverless 리소스 및 작업의 모든 사용량이 포함됩니다.
- 경고(Alarms) - 이 섹션에는 Amazon Redshift Serverless에서 구성한 경고가 표시됩니다.

데이터 백업

데이터 백업(Data backup) 탭에서 다음 작업을 수행할 수 있습니다.

- 스냅샷(Snapshots) - Amazon Redshift Serverless 데이터의 스냅샷을 생성, 삭제 및 관리할 수 있습니다. 기본 보존 기간은 indefinitely이지만 보존 기간을 1일에서 3653일 사이의 값으로 구성할 수 있습니다. 스냅샷에서 네임스페이스를 복원하도록 AWS 계정에 권한을 부여할 수 있습니다.
- 복구 시점(Recovery points) - 지난 24시간 내에 실수로 작성하거나 삭제한 경우 복구할 수 있도록 자동으로 생성된 복구 시점을 표시합니다. 데이터를 복구하기 위해 복구 시점을 사용 가능한 모든 네임스페이스로 복원할 수 있습니다. 더 오랜 기간 동안 복구 시점을 유지하려면 복구 시점에서 스냅샷을 생성할 수 있습니다. 기본 보존 기간은 indefinitely이지만 보존 기간을 1일에서 3653일 사이의 값으로 구성할 수 있습니다.

데이터 액세스

데이터 액세스(Data access) 탭에서 다음 작업을 수행할 수 있습니다.

- 네트워크 및 보안 설정 - VPC 관련 값, AWS KMS 암호화 값 및 감사 로깅 값을 볼 수 있습니다. 감사 로깅만 업데이트할 수 있습니다.
- AWS KMS key - AWS KMS key는 Amazon Redshift Serverless에서 리소스를 암호화하는 데 사용됩니다.
- 권한(Permissions) - Amazon Redshift Serverless가 사용자를 대신하여 리소스를 사용하기 위해 수임할 수 있는 IAM 역할을 관리할 수 있습니다. 자세한 내용은 [Amazon Redshift Serverless의 Identity and Access Management](#) 섹션을 참조하세요.
- Redshift 관리형 VPC 엔드포인트(Redshift-managed VPC endpoints) - 다른 VPC 또는 서브넷에서 Amazon Redshift Serverless 인스턴스에 액세스할 수 있습니다. 자세한 내용은 [다른 VPC 엔드포인트에서 Amazon Redshift Serverless에 연결](#) 섹션을 참조하세요.

한도

한도(Limits) 탭에서 다음 작업을 수행할 수 있습니다.

- Redshift 처리 장치(RPU)의 기본 용량(Base capacity in Redshift processing units (RPU)) 설정 - 워크로드를 처리하는 데 사용되는 기본 용량을 설정할 수 있습니다. 쿼리 성능을 향상시키려면 RPU 값을 늘리세요.
- 사용량 한도(Usage limits) - 작업이 시작되기 전 일정 기간 동안 Amazon Redshift Serverless 인스턴스에서 사용할 수 있는 최대 컴퓨팅 리소스입니다. Amazon Redshift Serverless에서 워크로드를 실행하는 데 사용하는 리소스 양을 제한합니다. 사용량은 Redshift 처리 장치(RPU) 시간으로 측정됩니다. RPU 시간은 한 시간 내에 사용되는 RPU 수입니다. 다음과 같이 설정한 한도에 도달할 때 수행할 작업을 결정합니다.
 - 알림을 보냅니다.
 - 시스템 테이블에 항목을 로그합니다.
 - 사용자 쿼리를 해제합니다.

최대 4개까지 한도를 설정할 수 있습니다.

- Query limits(쿼리 한도) - 성능 및 제한을 모니터링하기 위해 제한을 추가할 수 있습니다. 쿼리 모니터링 제한 사항에 대한 자세한 내용은 [WLM 쿼리 모니터링 규칙](#)을 참조하세요.

자세한 내용은 [Amazon Redshift Serverless의 컴퓨팅 용량](#) 섹션을 참조하세요.

데이터 공유

Datashare(Datashares) 탭에서 다음 작업을 수행할 수 있습니다.

- 내 네임스페이스에서 생성된 datashare(Datashares created in my namespace) 설정 - datashare를 생성하여 다른 네임스페이스 및 AWS 계정과 공유할 수 있습니다.
- 다른 네임스페이스 및 AWS 계정의 Datashare - 다른 네임스페이스 및 AWS 계정의 Datashare에서 데이터베이스를 생성할 수 있습니다.

데이터 공유에 대한 자세한 내용은 [Amazon Redshift Serverless에서 데이터 공유](#) 섹션을 참조하세요.

쿼리 및 데이터베이스 모니터링

쿼리 및 데이터베이스 모니터링(Query and database monitoring) 페이지에서 쿼리 기록(Query history) 및 데이터베이스 성능(Database performance) 그래프를 볼 수 있습니다.

쿼리 기록(Query history) 탭에서 다음 그래프를 볼 수 있습니다(쿼리 목록(Query list)과 리소스 지표(Resource metrics) 중에서 선택할 수 있음).

- 쿼리 런타임(Query runtime) - 이 그래프는 동일한 기간에 실행 중인 쿼리를 보여줍니다. 그래프에서 막대를 선택하여 더 많은 쿼리 실행 세부 정보를 확인합니다.
- 쿼리 및 로드(Queries and loads) - 이 섹션에는 쿼리 ID(Query ID)별로 쿼리와 로드가 나열됩니다.
- 사용된 RPU 용량(RPU capacity used) - 이 그래프는 Redshift 처리 장치(RPU)의 전체 용량을 보여줍니다.
- 데이터베이스 연결(Database connections) - 이 그래프는 활성 데이터베이스 연결 수를 보여줍니다.

데이터베이스 성능

데이터베이스 성능(Database performance) 탭에 다음 그래프가 표시됩니다.

- 초당 완료된 쿼리(Queries completed per second) - 이 그래프는 초당 완료된 평균 쿼리 수를 보여줍니다.
- 쿼리 기간(Query duration) - 이 그래프는 쿼리를 완료하는 데 걸리는 평균 시간을 보여줍니다.
- 데이터베이스 연결(Database connections) - 이 그래프는 활성 데이터베이스 연결 수를 보여줍니다.
- 실행 중인 쿼리(Running queries) - 이 그래프는 지정된 시간에 실행 중인 총 쿼리 수를 보여줍니다.
- 대기열에 있는 쿼리(Queued queries) - 이 그래프는 지정된 시간에 대기열에 있는 총 쿼리 수를 보여줍니다.
- 쿼리 실행 시간 분석(Query run time breakdown) - 이 그래프는 쿼리 유형별로 쿼리를 실행하는 데 소요된 총 시간을 보여줍니다.

리소스 모니터링

리소스 모니터링(Resource monitoring) 페이지에서 사용된 리소스의 그래프를 볼 수 있습니다. 여러 패킷을 기준으로 데이터를 필터링할 수 있습니다.

- 지표 필터(Metric filter) - 지표 필터를 사용하여 특정 작업 그룹에 대한 필터를 선택하고 시간 범위와 시간 간격을 선택할 수 있습니다.
- 사용된 RPU 용량(RPU capacity used) - 이 그래프는 Redshift 처리 장치(RPU)의 전체 용량을 보여줍니다.
- 컴퓨팅 사용량 - 이 그래프는 선택한 기간 동안 RPU 시간 사용량을 기간별로 보여줍니다. 범위가 6시간 미만인 경우 RPU 시간이 정확한 시간으로 표시됩니다. 범위가 6시간 이상인 경우 RPU 시간이 평균으로 표시됩니다.
- 자동 최적화를 위한 추가 컴퓨팅 요금 초 수 - 이 그래프는 선택한 시간 범위의 자동 데이터베이스 최적화에 대해 청구된 RPU-초 수를 보여줍니다. Amazon Redshift에서 추가 컴퓨팅 리소스를 사용하여 최적화를 실행하는 경우 자동 최적화에 대한 요금이 부과됩니다. 자세한 내용은 [자동 데이터베이스 최적화를 위한 추가 컴퓨팅 리소스 할당](#)을 참조하세요.

데이터 공유

Datashare(Datashares) 페이지에서 내 계정(In my account) 및 다른 계정(From other accounts)의 datashare를 관리할 수 있습니다. 데이터 공유에 대한 자세한 내용은 [Amazon Redshift Serverless에서 데이터 공유](#) 섹션을 참조하세요.

AWS Glue Data Catalog

AWS Glue Data Catalog 탭에서 AWS Glue Data Catalog에 대한 네임스페이스의 등록 상태를 볼 수 있습니다. 이 탭은 등록 프로세스를 시작한 후에만 나타납니다. 네임스페이스를 AWS Glue Data Catalog에 등록하는 방법에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Apache Iceberg compatibility for Amazon Redshift](#)를 참조하세요.

Amazon Redshift Serverless 사용 시 고려 사항

Amazon Redshift Serverless를 사용할 수 있는 AWS 리전 목록은 Amazon Web Services 일반 참조의 [Amazon Redshift Serverless API](#)에 나열된 엔드포인트를 참조하세요.

Amazon Redshift Serverless에서 사용하는 일부 리소스에는 할당량이 적용됩니다. 자세한 내용은 [Amazon Redshift Serverless 객체 할당량](#) 섹션을 참조하세요.

커서를 선언할 때 Amazon Redshift Serverless에 대한 결과 세트 크기 사양은 [DECLARE](#)에 지정됩니다. Amazon Redshift Serverless의 커서 최대 총 결과 세트 크기는 150,000MB입니다.

온라인 패치 - Amazon Redshift Serverless는 기존 유지 관리 기간 없이 자동 소프트웨어 업데이트를 제공합니다. 새 업데이트를 사용할 수 있게 되면 유휴 기간 동안 릴리스 후 14일 이내에 해당 업데이트가 적용됩니다. 업데이트 프로세스는 일반적으로 최대 15분이 걸립니다. 14일 이내에 15분의 유휴 기간이 발생하지 않으면 Serverless 엔드포인트를 잠시 사용할 수 없게 될 수 있습니다. 이 시간 동안 엔드포인트에 대한 애플리케이션 연결이 실패할 수 있습니다. 'Amazon Redshift용 클러스터 버전' 설명서에서 Redshift 패치 릴리스를 모니터링할 수 있습니다. Amazon Redshift Serverless 서비스 수준 계약(SLA)에 대한 정보는 [Amazon Redshift 서비스 수준 계약](#)을 참조하세요.

트랙 - Amazon Redshift가 새 작업 그룹 버전을 릴리스하면 작업 그룹이 자동으로 업데이트됩니다. 작업 그룹을 최신 릴리스로 업데이트할지 이전 릴리스로 업데이트할지 여부를 제어할 수 있습니다. 트랙에 대한 자세한 내용은 [Amazon Redshift 프로비저닝된 클러스터 및 서버리스 작업 그룹에 대한 트랙](#) 섹션을 참조하세요.

가용 영역 ID - Amazon Redshift Serverless 인스턴스를 구성할 때 추가 고려 사항을 열고 서브넷에 제공된 서브넷 ID에 지원되는 가용 영역 ID가 2개 이상 포함되어 있는지 확인합니다.

- 향상된 VPC 라우팅(EVR)을 사용하지 않는 작업 그룹에는 가용 영역(AZ)이 두 개 필요합니다.
- EVR을 사용하는 작업 그룹에는 3개의 AZ가 필요합니다.

서브넷과 가용 영역 ID 매핑을 보려면 VPC 콘솔로 이동하고 서브넷(Subnets)을 선택하여 가용 영역 ID와 함께 서브넷 ID 목록을 확인합니다. 서브넷이 지원되는 가용 영역 ID에 매핑되었는지 확인합니다. 서브넷을 생성하려면 Amazon VPC 사용 설명서의 [VPC에 서브넷 생성](#)을 참조하세요.

2개의 서브넷(EVR 사용 안 함) - 서브넷은 2개 이상 있어야 하며 2개의 가용 영역에 걸쳐 있어야 합니다.

3개의 서브넷(EVR 사용) - EVR을 사용하는 경우 서브넷은 3개 이상 있어야 하며 3개 이상의 가용 영역에 걸쳐 있어야 합니다.

무료 IP 주소 요구 사항 - 향상된 VPC 라우팅(EVR)을 활성화하지 않고 Redshift Serverless를 사용하는 경우 각 서브넷에서 사용 가능한 무료 IP 주소가 3개 이상 있어야 합니다. 이는 서비스가 제대로 작동하기 위한 필수 사항입니다.

Redshift Serverless 배포를 위해 RPU를 업데이트할 때 서비스의 운영 요구 사항을 수용하기 위해 각 서브넷에서 최소 3개의 무료 IP 주소를 사용할 수 있어야 합니다.

Amazon VPC의 IP 주소 할당 및 IP 주소 지정에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [VPC 및 서브넷의 IP 주소 지정](#)을 참조하세요.

Without EVR

향상된 VPC 라우팅을 사용하지 않는 경우, 기본 RPU 크기(4~1024 RPU)나 작업 그룹의 RPU 사용량과 관계없이 각 서브넷마다 최소 3개의 사용 가능한 IP 주소가 있어야 합니다. 3개의 IP 주소가 필요한 것은 AI 기반 규모 조정 및 최적화 기능이 활성화된 작업 그룹에도 적용됩니다.

With Enhanced VPC Routing (EVR)

Redshift Serverless에서 향상된 VPC 라우팅을 사용하는 경우 작업 그룹을 생성할 때 필요한 최소 IP 주소 수는 다음과 같습니다.

Redshift 처리 단위(RPU)	무료 IP 주소 필요	최소 CIDR 크기
4	9	/27
8	9	/27
16	13	/27
32	13	/27
64	21	/27
128	37	/26
256	69	/25
512	133	/24
1024	261	/23

또한 EVR을 사용하는 경우 더 많은 RPU를 사용하기 위해 작업 그룹을 업데이트할 때 무료 IP 주소가 필요합니다. 작업 그룹에 대한 서브넷을 업데이트할 때 필요한 무료 IP 주소 수는 다음과 같습니다.

Redshift 처리 장치(RPU)	업데이트된 Redshift 처리 장치(RPU)	무료 IP 주소 필요
4	8	10
8	16	10
16	32	13
32	64	16
64	128	28
128	256	52
256	512	100
512	1024	197

 Note

최대 기본 RPU 용량인 1024는 다음 AWS 리전에만 사용할 수 있습니다.

- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(오리곤)
- 유럽(아일랜드)
- 유럽(런던)

IP 주소 할당에 대한 자세한 내용은 [Amazon VPC 사용 설명서](#)의 IP 주소 지정을 참조하세요.

마이그레이션 이후 스토리지 공간 - 소규모 Amazon Redshift 프로비저닝된 클러스터를 Amazon Redshift Serverless로 마이그레이션할 때 마이그레이션 후 스토리지 공간 할당이 증가할 수 있습니다. 이는 스토리지 공간 할당을 최적화한 결과로 스토리지 공간이 미리 할당되었기 때문입니다. 이 공간은 Amazon Redshift Serverless에서 데이터가 증가함에 따라 일정 기간 동안 사용됩니다.

Amazon Redshift Serverless와 Amazon Redshift 프로비저닝된 클러스터 간의 데이터 공유 - Amazon Redshift Serverless가 생산자이고 프로비저닝된 클러스터가 소비자인 데이터 공유에서 프로비저닝된 클러스터의 클러스터 버전은 1.0.38214보다 이후 버전이어야 합니다. 이보다 이전의 클러스터 버전을 사용하면 쿼리를 실행할 때 오류가 발생합니다. 유지 관리(Maintenance) 탭에 있는 Amazon Redshift 콘솔에서 클러스터 버전을 볼 수 있습니다. 또한 `SELECT version();`을 실행할 수도 있습니다.

최대 쿼리 실행 시간(Max query execution time) - 경과된 쿼리 실행 시간(초)입니다. 실행 시간에는 대기열에서 대기하는 데 소모한 시간은 포함되지 않습니다. 쿼리가 설정된 실행 시간을 초과하면 Amazon Redshift Serverless가 쿼리를 중단합니다. 유효한 값은 0~86,399입니다.

인터리브 정렬 키가 있는 테이블의 마이그레이션 - Amazon Redshift에서 프로비저닝한 클러스터를 Amazon Redshift Serverless로 마이그레이션하는 경우, Redshift는 인터리브된 정렬 키와 DISTSTYLE KEY가 있는 테이블을 복합 정렬 키로 변환합니다. DISTSTYLE은 변경되지 않습니다. 분산 스타일에 대한 자세한 내용은 Amazon Redshift 개발자 안내서의 [데이터 분산 스타일 작업](#)을 참조하세요. 정렬 키에 대한 자세한 내용은 [정렬 키 작업](#)을 참조하세요.

VPC 공유 - 공유 VPC에서 Amazon Redshift Serverless 작업 그룹을 생성할 수 있습니다. 이 경우 리소스 공유를 삭제하면 작업 그룹을 사용할 수 없게 될 수 있으므로 리소스 공유를 삭제하지 않는 것이 좋습니다.

IPv6 지원:

Amazon Redshift Serverless는 AWS 가상 프라이빗 클라우드(VPC) 내에서 IPv4 및 IPv6 주소(듀얼 스택) 또는 IPv4 전용 구성을 모두 사용하여 Amazon Redshift 작업 그룹 구성을 지원합니다. 새 Amazon Redshift Serverless 작업 그룹을 생성할 때 IPv6 지원을 활성화하거나 IPv6 주소 지정을 지원하도록 기존 작업 그룹을 수정할 수 있습니다. 이 기능을 사용하면 IPv6 지원 VPC 서브넷에 Amazon Redshift Serverless 웨어하우스를 배포하고 애플리케이션의 확장 주소 공간 요구 사항을 지원하도록 네트워크 설정을 구성할 수 있습니다. 이제 애플리케이션은 IPv4 또는 IPv6 프로토콜을 사용해 Amazon Redshift Serverless 웨어하우스와 통신하여 기존 및 향후 네트워크 아키텍처와의 호환성을 보장할 수 있습니다.

Amazon Redshift Serverless의 컴퓨팅 용량

Amazon Redshift Serverless를 사용하면 워크로드 요구 사항에 맞게 컴퓨팅 용량이 자동으로 스케일 업 및 스케일 다운됩니다. 컴퓨팅 용량은 Amazon Redshift Serverless 워크로드에 할당된 처리 능력과 메모리를 말합니다. 일반적인 사용 사례에는 트래픽이 가장 많은 기간을 처리하거나, 복잡한 분석을 실행하거나, 대용량 데이터를 효율적으로 처리하는 작업이 포함됩니다. 다음 용어는 Amazon Redshift가 컴퓨팅 용량을 관리하는 방법에 대한 세부 정보를 제공합니다.

RPU

Amazon Redshift Serverless는 데이터 웨어하우스 용량을 Redshift 처리 장치(RPU)로 측정합니다. RPU는 워크로드를 처리하는 데 사용되는 리소스입니다. 1RPU는 16GB의 메모리를 제공합니다.

기본 용량

이 설정은 Amazon Redshift가 쿼리를 처리하는 데 사용하는 기본 데이터 웨어하우스 용량을 지정합니다. 기본 용량은 RPU로 지정됩니다. Redshift 처리 장치(RPU)로 기본 용량을 설정할 수 있습니다. 기본 용량을 높게 설정하면 특히 많은 리소스가 필요한 데이터 처리 작업에서 쿼리 성능이 향상됩니다. Amazon Redshift Serverless의 기본 용량 기본값은 128개의 RPU입니다. 기본 용량 설정을 4RPU에서 512RPU까지 조정할 수 있습니다. 이 값을 4RPU로 설정하거나, 8RPU부터 8개 단위(8,16,24...512)로 설정할 수 있습니다. AWS Console, UpdateWorkgroup API 작업 또는 AWS CLI의 `update-workgroup` 작업을 사용하여 이 값을 설정할 수 있습니다.

최소 4RPU의 기본 용량으로 데이터 웨어하우스 비용 및 용량 요구 사항에 따라 간단한 워크로드부터 복잡한 워크로드까지 유연하게 실행할 수 있습니다. 4개의 기본 RPU 용량은 32TB 미만의 데이터가 포함된 웨어하우스를 대상으로 하며, 8, 16 및 24개의 기본 RPU 용량은 128TB 미만의 데이터가 필요한 워크로드를 대상으로 합니다. 데이터 요구 사항이 128TB보다 큰 경우 최소 32개의 기본 RPU를 사용해야 합니다. 테이블에 많은 열이 있고 동시성이 높은 워크로드의 경우 32개 이상의 기본 RPU를 사용하는 것이 좋습니다.

사용 가능한 최대 기본 RPU인 1024는 워크로드에 최고 수준의 컴퓨팅 리소스를 추가합니다. 따라서 매우 복잡한 워크로드를 지원할 수 있는 유연성이 향상되고 데이터 로드 및 쿼리 속도가 빨라집니다.

Note

확장된 최대 기본 RPU 용량인 1024는 다음 AWS 리전에서 사용할 수 있습니다. 다른 리전에서는 최대 기본 용량이 512RPU입니다.

- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(오리곤)
- 유럽(아일랜드)
- 유럽(프랑크푸르트)

기본 용량을 512~1024 사이로 설정할 때 32단위로 RPU를 늘리거나 줄일 수 있습니다.

더 크고 복잡한 워크로드를 관리하는 경우 Redshift Serverless 데이터 웨어하우스의 크기를 늘리는 것을 고려해 보세요. 대형 웨어하우스는 더 많은 컴퓨팅 리소스에 액세스할 수 있으므로, 쿼리를 더 효율적으로 처리할 수 있습니다.

기본 용량이 더 큰 것이 유리한 몇 가지 예는 다음과 같습니다.

- 실행하는 데 오랜 시간이 걸리는 복잡한 쿼리가 있는 경우
- 테이블에 많은 수의 열이 있는 경우
- 쿼리의 JOIN 수가 많은 경우
- 쿼리는 데이터 레이크와 같은 외부 소스에서 대량의 데이터를 수집하거나 스캔합니다.

Amazon Redshift Serverless 할당량 및 제한에 대한 자세한 내용을 알아보려면 [Amazon Redshift Serverless 객체 할당량](#) 섹션으로 이동하세요.

Amazon Redshift Serverless에 대한 고려 사항 및 제한 사항

다음은 Amazon Redshift Serverless 용량에 대한 고려 사항 및 제한 사항입니다. 일반적인 Redshift Serverless 고려 사항은 [Amazon Redshift Serverless 사용 시 고려 사항](#) 섹션을 참조하세요.

- 4개의 기본 RPU 구성은 최대 32TB의 관리형 스토리지 용량을 지원합니다. 32TB가 넘는 관리형 스토리지를 사용하는 경우 기본 RPU를 8RPU 미만으로 설정할 수 없습니다.
- 8개 또는 16개의 기본 RPU 구성은 최대 128TB의 Redshift 관리형 스토리지 용량을 지원합니다. 128TB가 넘는 관리형 스토리지를 사용하는 경우 기본 RPU를 32RPU 미만으로 설정할 수 없습니다.
- 작업 그룹의 기본 용량을 편집하면 작업 그룹에서 실행 중인 일부 쿼리가 취소될 수 있습니다.
- Redshift Serverless는 다음 증분을 사용하여 데이터 웨어하우스의 RPU 규모를 조정합니다.
 - 4~8RPU: 4RPU씩 증가합니다.
 - 8~512RPU: 8RPU씩 증가합니다.
 - 512~1024RPU: 32RPU씩 증가합니다.
- vacuum boost는 8RPU 이상에서만 지원됩니다. 8RPU 이하의 경우 다음 명령을 대신 사용하세요.

```
VACUUM [FULL | SORT ONLY | DELETE ONLY | REINDEX | RECLUSTER] [table_name] [TO threshold PERCENT]
```


4개의 Redshift 처리 장치(RPU) 용량이 있는 Redshift Serverless

4개의 기본 RPU 용량이 있는 Redshift Serverless는 소규모 워크로드나 덜 까다로운 워크로드에 적합합니다. 엔트리 레벨 구성으로, 유연하고 비용 효과적인 솔루션을 제공합니다. 이 엔트리 레벨 구성은 최대 다음 리소스가 있는 데이터 웨어하우스를 지원합니다.

- 최대 32TB의 Redshift 관리형 스토리지
- 테이블당 최대 100개의 열
- 64GB 메모리

이러한 제한을 초과해야 하는 경우 오토 스케일링에 의존하지 않고 기본 용량을 수동으로 늘려야 합니다. 데이터 웨어하우스를 4RPU 이상으로 확장하면 데이터 웨어하우스는 더 많은 RPU를 계속 사용하게 되며 Amazon Redshift가 데이터 웨어하우스를 4RPU로 다시 축소하지 않습니다.

Note

4개의 기본 RPU를 사용할 때 100개가 넘는 열이 있는 테이블을 만들 수 있지만, 테이블의 열 개수를 100개로 제한하는 것이 좋습니다. 이 제한을 초과하면 쿼리 실행 중에 데이터 웨어하우스의 메모리가 소진되어 성능이 저하될 수 있습니다.

다음 AWS 리전에서 4RPU를 사용하는 데이터 웨어하우스를 생성할 수 있습니다.

- 미국 동부(오하이오)
- 미국 동부(버지니아 북부)
- 미국 서부(캘리포니아 북부)
- 미국 서부(오리건)
- 아시아 태평양(뭄바이)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(도쿄)
- 유럽(아일랜드)
- 유럽(스톡홀름)

AI 기반 크기 조정 및 최적화

AI 기반 규모 조정 및 최적화 기능은 Amazon Redshift Serverless를 사용할 수 있는 모든 AWS 리전에 사용 가능합니다.

Amazon Redshift Serverless는 다양한 워크로드 요구 사항을 충족하는 고급 AI 기반 크기 조정 및 최적화 기능을 제공합니다. 데이터 웨어하우스에는 다음과 같은 프로비저닝 문제가 있을 수 있습니다.

- 리소스 집약적인 쿼리의 성능을 개선하기 위해 데이터 웨어하우스가 과도하게 프로비저닝될 수 있습니다.
- 비용 절감을 위해 데이터 웨어하우스가 과소 프로비저닝될 수 있습니다.

데이터 웨어하우스 워크로드의 성능과 비용 간에 적절한 균형을 맞추는 것은 특히 임시 쿼리와 데이터 볼륨 증가로 인해 어렵습니다. 리소스 사용량이 적은 쿼리와 많은 쿼리로 구성된 혼합 워크로드를 실행할 때는 지능형 크기 조정이 필요합니다. AI 기반 크기 조정 및 최적화 기능은 데이터 증가에 따라 서버리스 컴퓨팅 또는 RPU 크기를 자동으로 조정합니다. 또한 이 기능은 지정한 가격 대비 성능 목표 내에서 쿼리 성능을 유지하는 데 도움이 됩니다. AI 기반 크기 조정 및 최적화는 데이터 볼륨이 증가함에 따라 컴퓨팅 리소스를 동적으로 할당하여 쿼리가 성능 목표를 계속 충족할 수 있도록 합니다. AI 기반 크기 조정 및 최적화를 통해 서비스는 수동 개입이나 복잡한 용량 계획 없이 변화하는 워크로드 요구 사항에 원활하게 적응할 수 있습니다.

Amazon Redshift Serverless는 쿼리 복잡성 및 데이터 볼륨과 같은 요인을 기반으로 보다 포괄적이고 응답성이 뛰어난 크기 조정 솔루션을 제공합니다. 이 기능을 사용하면 다양한 워크로드와 증가하는 데이터세트를 효율적으로 처리할 수 있는 유연성을 유지하면서 워크로드 가격 대비 성능을 최적화할 수 있습니다. Amazon Redshift Serverless는 Serverless 작업 그룹에 지정된 가격 대비 성능 목표를 충족하기 위해 Amazon Redshift Serverless 엔드포인트에 AI 기반 최적화를 자동으로 수행할 수 있습니다. 이 자동 가격 대비 성능 최적화는 워크로드에 기본 용량을 얼마로 설정해야 할지 모르거나 워크로드의 일부에 리소스를 더 많이 할당하면 이점이 있는 경우에 특히 유용합니다.

예제

조직에서 일반적으로 32RPU만 필요한 워크로드를 실행하다가 갑자기 더 복잡한 쿼리를 사용하는 경우 적절한 기본 용량을 알지 못할 수 있습니다. 기본 용량을 더 높게 설정하면 성능이 향상되지만 비용도 증가하므로 비용이 예상과 일치하지 않을 수 있습니다. Amazon Redshift Serverless는 AI 기반 크기 조정 및 리소스 최적화를 사용하여 조직에 맞게 비용을 최적화하는 동시에 가격 대비 성능 목표를 충족하도록 RPU를 자동으로 조정합니다. 이 자동 최적화는 워크로드 크기와 상관없이 유용합니다. 자동 최적화는 복잡한 쿼리가 많은 경우 조직의 가격 대비 성능 목표를 달성하는 데 도움이 될 수 있습니다.

Note

가격 대비 성능 목표는 작업 그룹별 설정입니다. 작업 그룹마다 가격 대비 성능 목표가 다를 수 있습니다.

비용을 예측할 수 있도록 Amazon Redshift Serverless가 워크로드에 할당할 수 있는 최대 용량 한도를 설정하세요.

가격 대비 성능 목표를 구성하려면 AWS 콘솔을 사용하세요. Serverless 작업 그룹을 만들 때 가격 대비 성능 목표를 명시적으로 활성화해야 합니다. Serverless 작업 그룹을 만든 후에 가격 대비 성능 목표를 수정할 수도 있습니다. 가격 대비 성능 목표를 활성화하면 기본적으로 밸런싱됨으로 설정됩니다.

작업 그룹의 가격 대비 성능 목표를 편집하는 방법

1. Amazon Redshift Serverless 콘솔에서 작업 그룹 구성을 선택합니다.
2. 가격 대비 성능 목표를 편집할 작업 그룹을 선택합니다. 권한 탭을 선택한 다음 편집을 선택합니다.
3. 가격 대비 성능 목표를 선택하고 슬라이더를 원하는 설정으로 조정합니다.
4. 변경 사항 저장을 선택합니다.
5. Amazon Redshift Serverless가 워크로드에 할당할 수 있는 최대 RPU의 양을 업데이트하려면 작업 그룹 구성 섹션의 한도 탭을 선택하세요.

가격 대비 성능 목표 슬라이더를 사용하여 비용과 성능 간에 원하는 균형을 설정할 수 있습니다. 슬라이더를 이동하여 다음 옵션 중 하나를 선택할 수 있습니다.

- 비용에 최적화 - 이 설정은 비용 절감을 우선시합니다. Amazon Redshift Serverless는 추가 요금이 발생하지 않는 선에서 컴퓨팅 용량을 자동으로 스케일 업하려고 합니다. 또한 Amazon Redshift Serverless는 비용을 절감하기 위해 컴퓨팅 리소스를 스케일 다운하려고 시도하므로 쿼리 런타임이 늘어날 수 있습니다.
- 밸런싱됨 - 이 설정은 성능과 비용 간의 균형을 만듭니다. Amazon Redshift Serverless는 성능을 위해 크기를 조정하며 적당한 수준의 비용 증가 또는 감소를 초래할 수 있습니다. 이는 대부분의 Amazon Redshift Serverless 데이터 웨어하우스에 권장되는 설정입니다.
- 성능에 최적화 - 이 설정은 성능을 우선시합니다. Amazon Redshift가 고성능을 위해 공격적으로 확장하므로 비용이 더 많이 발생할 수 있습니다.

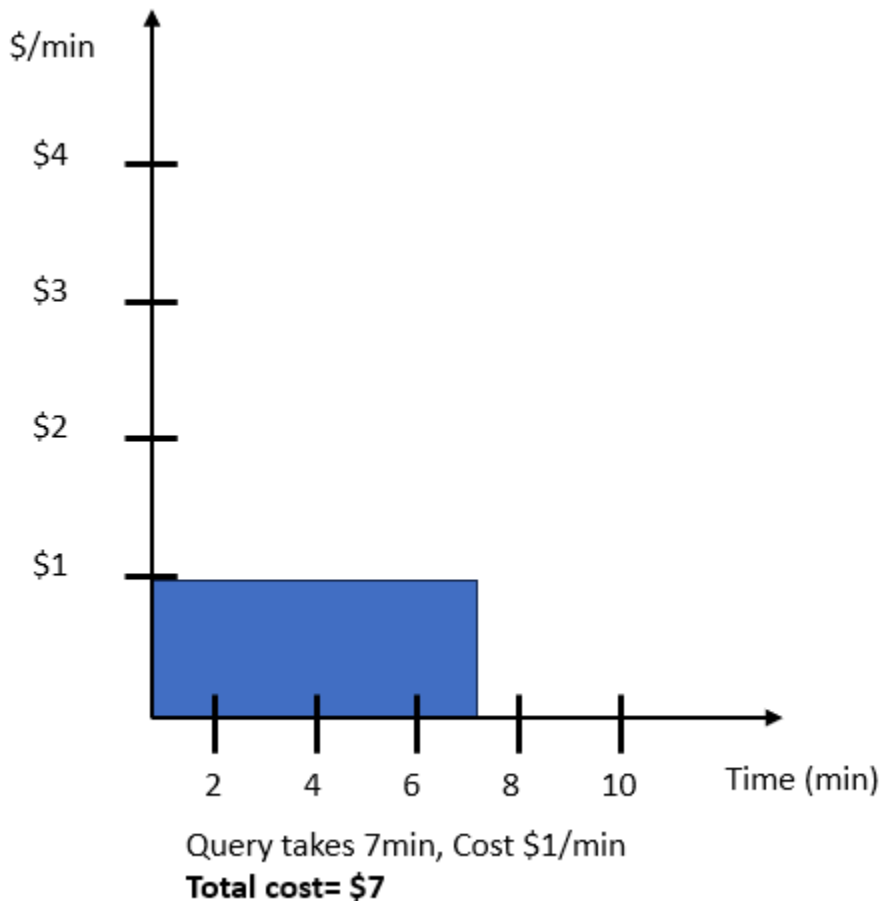
- 중간 위치: 슬라이더를 밸런싱됨과 비용에 최적화 또는 성능에 최적화 사이의 두 중간 위치 중 하나로 설정할 수도 있습니다. 비용 또는 성능에 대한 전체 최적화가 너무 극단적일 경우 이러한 설정을 사용합니다.

가격 대비 성능 목표 선택 시 고려 사항

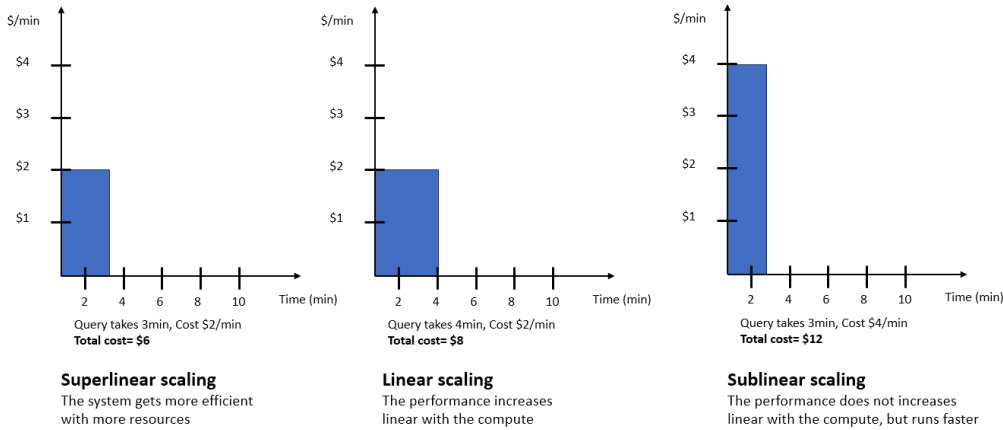
가격 대비 성능 슬라이더를 사용하여 워크로드에 원하는 가격 대비 성능 목표를 선택할 수 있습니다. AI 기반 크기 조정 및 최적화 알고리즘은 시간 경과에 따라 워크로드 기록에서 학습하며 예측 및 결정 정확도를 개선합니다.

예제

이 예시에서는 7분이 걸리고 비용이 7 USD인 쿼리를 가정합니다. 다음 그림은 크기 조정을 하지 않을 경우 쿼리 런타임 및 비용을 보여줍니다.



주어진 쿼리는 아래와 같이 몇 가지 다른 방식으로 스케일 인될 수 있습니다. 선택한 가격 대비 성능 목표에 따라 AI 기반 크기 조정은 쿼리가 성능과 비용을 어떻게 상쇄하는지 예측하고 그에 따라 크기를 조정합니다. 다른 슬라이더 옵션을 선택하면 다음과 같은 결과가 나타납니다.



- **비용에 최적화** - 비용에 최적화 옵션을 사용하면 데이터 웨어하우스가 비용을 낮추는 선택을 우선시하여 크기를 조정합니다. 이전 예시에서 초선형 크기 조정 접근 방식이 이러한 동작을 보여줍니다. 크기 조정은 크기 조정 모델 예측에 따라 비용 효율적인 방식으로 수행할 수 있는 경우에만 발생합니다. 크기 조정 모델에서 지정된 워크로드에 대해 비용에 최적화된 크기 조정이 불가능하다고 예측하는 경우 데이터 웨어하우스는 크기가 조정되지 않습니다.
- **밸런싱됨** - 밸런싱됨 옵션을 사용하면 시스템이 비용 및 성능 고려 사항의 균형을 유지하면서 크기를 조정하고 비용이 조금 증가할 수 있습니다. 밸런싱됨 옵션은 초선형, 선형 및 아선형 워크로드 크기 조정을 수행합니다.
- **성능에 최적화** - 성능에 최적화 옵션을 사용하면 시스템이 성능 개선을 위한 이전의 방법을 사용하는 것 외에도, 비용이 더 높고 런타임 개선에 비례하지 않더라도 크기를 조정합니다. 성능에 최적화를 사용하면 가능한 경우 시스템에서 초선형 크기 조정, 선형 크기 조정 및 아선형 크기 조정을 수행합니다. 슬라이더 위치가 성능에 최적화 위치에 가까울수록 Amazon Redshift Serverless가 아선형 크기 조정을 더 많이 허용합니다.

가격 대비 성능 슬라이더를 설정할 때 다음 사항에 유의하세요.

- 언제든지 가격 대비 성능 설정을 변경할 수 있지만 워크로드 크기 조정은 즉시 변경되지 않습니다. 크기 조정은 시스템이 현재 워크로드에 대해 학습함에 따라 시간이 지나면서 변경됩니다. 새 설정의 영향을 확인하기 위해 1~3일 동안 Serverless 작업 그룹을 모니터링하는 것이 좋습니다.
- 가격 대비 성능 슬라이더 옵션 최대 용량과 최대 RPU 시간은 함께 작동합니다. 최대 용량과 최대 RPU 시간은 Amazon Redshift Serverless가 데이터 웨어하우스의 크기 조정을 허용하는 최대 RPU와 Amazon Redshift Serverless가 데이터 웨어하우스의 소비를 허용하는 최대 RPU 시간을 제한하

는 제어 기능입니다. Amazon Redshift Serverless는 가격 대비 성능 목표 설정과 관계없이 이러한 설정을 항상 준수하고 적용합니다.

리소스 오토 스케일링 모니터링

다음과 같은 방법으로 AI 기반 RPU 크기 조정을 모니터링할 수 있습니다.

- Amazon Redshift 콘솔에서 사용된 RPU 용량 그래프를 검토합니다.
- CloudWatch에서 AWS/Redshift-Serverless 및 Workgroup 아래의 ComputeCapacity 지표를 모니터링합니다.
- [SYS_QUERY_HISTORY](#) 뷰를 쿼리합니다. 특정 쿼리 ID 또는 쿼리 텍스트를 제공하여 기간을 지정합니다. 이 기간을 사용하여 [SYS_SERVERLESS_USAGE](#) 시스템 뷰를 쿼리하여 compute_capacity 값을 찾습니다. compute_capacity 필드에는 쿼리 런타임 중에 크기 조정된 RPU가 표시됩니다.

다음 예시를 사용하여 SYS_QUERY_HISTORY 뷰를 쿼리합니다. 샘플 값을 쿼리 텍스트로 바꿉니다.

```
select query_id,query_text,start_time,end_time, elapsed_time/1000000.0
duration_in_seconds
from sys_query_history
where query_text like '<query_text>'
and query_text not like '%sys_query_history%'
order by start_time desc
```

다음 쿼리를 실행하여 start_time에서 end_time까지의 기간 동안 compute_capacity의 크기 조정이 어떻게 이루어졌는지 확인합니다. 다음 쿼리의 start_time 및 end_time을 이전 쿼리의 출력으로 바꿉니다.

```
select * from sys_serverless_usage
where end_time >= 'start_time'
and end_time <= DATEADD(minute,1,'end_time')
order by end_time asc
```

이러한 기능 사용에 대한 단계별 지침은 [Configure monitoring, limits, and alarms in Amazon Redshift Serverless to keep costs predictable](#)을 참조하세요.

AI 기반 크기 조정 및 최적화 사용 시 고려 사항

AI 기반 크기 조정 및 최적화를 사용할 때는 다음 사항을 고려하세요.

- 32~512개의 기본 RPU가 필요한 Amazon Redshift Serverless의 기존 워크로드는 최적의 결과를 위해 Amazon Redshift Serverless AI 기반 크기 조정 및 최적화를 사용하는 것이 좋습니다. 기본 RPU가 32개 미만이거나 512개 이상인 경우 이 기능을 사용하지 않는 것이 좋습니다.
- 결과는 다를 수 있지만 가격 대비 성능 목표는 워크로드를 자동으로 최적화합니다. 시스템이 대표 워크로드를 실행하여 특정 패턴을 학습할 수 있도록 이 기능을 장기간 사용하는 것이 좋습니다.
- AI 기반 크기 조정 및 최적화는 Amazon Redshift Serverless 인스턴스에서 실행되는 워크로드에 따라 Serverless 작업 그룹에 최적화를 적용하는 데 최적의 시간을 사용합니다.

AI 기반 최적화와 리소스 크기 조정에 대해 자세히 알아보려면 다음 동영상을 시청하세요.

Amazon Redshift Serverless에 대한 청구

컴퓨팅 용량에 대한 결제

다음 두 가지 방법으로 Amazon Redshift Serverless의 용량을 구매할 수 있습니다.

- 온디맨드 용량 구매 - 온디맨드 컴퓨팅 용량을 선택하면 종량제로 리소스 비용을 지불합니다. Amazon Redshift Serverless를 이제 막 사용하기 시작했거나 아직 일정한 사용 패턴을 잘 모르는 경우에 가장 적합합니다. 온디맨드는 가장 유연한 방법입니다. 자세한 내용은 [온디맨드 컴퓨팅 용량에 대한 결제](#) 섹션을 참조하세요.
- 예약 구매 - 예약을 통해 특정 기간(예: 1년) 동안 사전 설정된 양의 컴퓨팅 리소스를 구매하면 할인이 제공됩니다. 일정량의 용량을 꾸준히 사용하려는 경우 좋은 방법입니다. 일부 용량 요구량을 예측할 수 있을 때 비용을 절감하는 데 유용합니다. 자세한 내용은 [서버리스 예약에 대한 결제](#) 섹션을 참조하세요.

예약과 온디맨드 리소스를 함께 사용할 수 있습니다. 둘 중 하나만 사용해야 하는 것은 아닙니다.

요금에 대한 자세한 내용은 [Amazon Redshift 요금](#)을 참조하세요.

온디맨드 컴퓨팅 용량에 대한 결제

기본 용량 및 결제에 미치는 영향

쿼리가 실행되면 지정된 기간 동안 사용된 용량에 따라 초 단위 RPU 시간으로 요금이 청구됩니다. 실행 중인 쿼리가 없으면 컴퓨팅 용량에 대한 요금이 청구되지 않습니다. 또한 저장된 데이터 양에 따라 Redshift 관리형 스토리지(RMS)에 요금이 청구됩니다.

작업 그룹을 만들 때 컴퓨팅의 기본 용량을 설정할 수 있는 옵션이 있습니다. 작업 그룹 수준에서 워크로드의 가격/성능 요구 사항을 충족하려면 기존 작업 그룹의 기본 용량을 더 높이거나 낮게 조정하세요. 작업 그룹 구성에서 작업 그룹을 선택하고 제한 탭을 선택하여 콘솔을 사용하여 기본 용량을 변경합니다.

쿼리 수가 증가하면 Amazon Redshift Serverless가 일관된 성능을 제공하기 위해 자동으로 확장합니다.

최대 RPU 시간 사용량 한도

Amazon Redshift Serverless에 대한 비용을 예측 가능한 상태로 유지하도록, 일별, 주별 또는 월별 사용량인 최대 RPU 시간(Maximum RPU hours)을 설정할 수 있습니다. 콘솔 또는 API를 사용하여 설정할 수 있습니다. 한도에 도달하면 로그 항목이 시스템 테이블에 기록되도록 지정하거나 알림을 받거나 사용자 쿼리를 끌 수 있습니다. 최대 RPU 시간을 설정하면 비용을 제어하는 데 도움이 됩니다. 최대 RPU 시간 설정은 데이터 웨어하우스의 데이터에 액세스하는 쿼리와 외부 데이터(예: Amazon S3의 외부 테이블)에 액세스하는 쿼리를 실행할 때 모두 작업 그룹에 적용됩니다.

다음은 예제입니다.

매주 100시간으로 한도를 설정한다고 가정해 보겠습니다. 콘솔에서 설정하려면 다음을 수행합니다.

1. 작업 그룹을 선택한 다음 한도 탭에서 사용량 한도 관리를 선택합니다.
2. 사용량 한도를 추가하고 빈도는 주별, 기간은 100시간을 선택한 다음 사용자 쿼리 끄기로 동작을 설정합니다.

이 예시에서는 일주일당 100RPU 시간 한도에 도달하면 쿼리가 꺼집니다.

작업 그룹의 최대 RPU 시간을 설정해도 작업 그룹의 성능이나 컴퓨팅 리소스가 제한되지는 않습니다. 쿼리 처리에 영향을 주지 않고 언제든지 설정을 조정할 수 있습니다. 최대 RPU 시간은 가격 및 성능 요구 사항을 충족하는 데 도움이 되도록 설정하는 것이 목표입니다. 서버리스 결제에 대한 자세한 내용은 [Amazon Redshift 요금](#)을 참조하세요.

Amazon Redshift Serverless의 비용을 예측 가능한 상태로 유지하는 또 다른 방법은 AWS [Cost Anomaly Detection](#)을 사용하여 예상치 못하게 비용이 청구될 가능성을 줄이고 더 높은 제어력을 확보하는 것입니다.

Note

[Amazon Redshift 요금 계산기](#)는 가격을 추정하는 데 유용합니다. 필요한 컴퓨팅 리소스를 입력하면 비용을 미리 볼 수 있습니다.

최대 용량을 설정하여 컴퓨팅 리소스 비용 통제

최대 용량 설정은 Amazon Redshift Serverless가 스케일 업할 수 있는 RPU 상한 역할을 합니다. 컴퓨팅 리소스 비용을 통제하는 데 도움이 됩니다. 기본 용량이 사용 가능한 컴퓨팅 리소스의 최소량을 설정하는 것과 유사하게 최대 용량은 RPU 사용량의 상한을 설정합니다. 이렇게 하면 계획에 맞게 지출하는 데 도움이 됩니다. 최대 용량은 구체적으로 각 작업 그룹에 적용되며 이로 인해 항상 컴퓨팅 사용량이 제한됩니다.

최대 용량과 RPU 시간 사용량 한도의 차이

최대 RPU 시간 한도와 최대 용량 설정의 목적은 모두 비용을 통제하는 것입니다. 하지만 두 설정은 서로 다른 방법을 통해 비용 통제를 수행합니다. 차이점은 다음과 같습니다.

- **최대 용량** - 이 설정은 Amazon Redshift Serverless가 크기 조정 목적으로 사용하는 최대 RPU 수를 설정합니다. 자동 컴퓨팅 크기 조정이 필요한 경우 최대 용량 값을 높이면 쿼리 처리량을 높일 수 있습니다. 최대 용량 한도에 도달하면 작업 그룹은 리소스를 더 이상 스케일 업하지 않습니다.
- **최대 RPU 시간 사용량 한도** - 최대 용량과 달리 이 설정은 용량 상한을 설정하지 않습니다. 하지만 비용을 제한하는 데 도움이 되는 다른 작업을 수행합니다. 여기에는 로그에 항목 추가, 알림, 원하는 경우 쿼리 실행 중지 등이 포함됩니다.

최대 용량을 배타적으로 사용할 수도 있고, 최대 RPU 시간 사용량 한도에 따른 조치로 보완할 수도 있습니다.

최대 용량 사용 사례

작업 그룹마다 최대 용량을 다르게 설정할 수 있습니다. 이는 예산 요구 사항을 적용하는 데 도움이 됩니다. 작동 방식을 설명하기 위해 다음과 같은 상황을 가정하겠습니다.

- 기본 용량이 256RPU로 설정된 작업 그룹이 있습니다. 매월 대부분 256RPU를 약간 넘는 안정적인 워크로드를 유지합니다.
- 최대 용량은 512RPU로 설정되어 있습니다.

임시 통계 보고서를 작성하기 위해 3일 동안 사용량이 예상치 못하게 많아졌다고 가정해 보겠습니다. 이 경우 컴퓨팅 비용이 512RPU에 대한 비용을 초과하지 않도록 최대 용량이 설정되어 있습니다. 이렇게 하면 컴퓨팅 용량이 이 상한을 초과하지 않도록 할 수 있습니다.

최대 용량에 대한 사용 참고 사항

다음 참고 사항은 최대 용량을 적절하게 설정하는 데 도움이 될 수 있습니다.

- Amazon Redshift Serverless 작업 그룹마다 최대 용량을 다르게 설정할 수 있습니다.
- 일정 기간 동안 리소스를 매우 많이 사용하고 최대 용량을 낮은 RPU 수준으로 설정하면 워크로드 처리가 지연되어 사용자 경험이 최적화되지 않을 수 있습니다.
- 최대 용량 설정을 구성하더라도 RPU 사용량이 많은 시간에도 쿼리 실행에 방해가 되지 않습니다. 쿼리 실행이 중단될 수 있는 사용량 한도와는 작동 방식이 다릅니다. 이 설정은 작업 그룹에서 사용할 수 있는 컴퓨팅 리소스만 제한합니다. Amazon Redshift Serverless 대시보드에서 일정 기간 동안 사용된 용량을 볼 수 있습니다. 요약 데이터를 보는 방법에 대한 자세한 내용은 [대시보드에서 Amazon Redshift Serverless 요약 데이터 확인](#)을 참조하세요.
- 최대 용량 설정은 5632RPU입니다.

최대 용량 설정 방법

콘솔에서 최대 용량을 설정할 수 있습니다. 기존 작업 그룹의 경우 작업 그룹 구성에서 설정을 변경할 수 있습니다. CLI에서 다음 샘플과 같은 명령을 사용하여 설정할 수도 있습니다.

```
aws redshift-serverless update-workgroup --workgroup-name myworkgroup --max-capacity 512
```

이렇게 하면 주어진 이름의 작업 그룹에 최대 용량이 설정됩니다. 설정 후 콘솔에서 값을 확인하여 확인할 수 있습니다. CLI를 사용하여 `get-workgroup` 명령을 실행하여 값을 확인할 수도 있습니다.

다음과 같이 최대 용량 설정을 -1로 설정하여 설정을 끌 수 있습니다.

```
aws redshift-serverless update-workgroup --workgroup-name myworkgroup --max-capacity -1
```

Amazon Redshift Serverless 사용량 및 비용 모니터링

Amazon Redshift Serverless의 사용량 및 결제를 예측할 수 있는 방법에는 여러 가지가 있습니다. 시스템 뷰는 쿼리 및 사용 데이터를 포함한 시스템 메타데이터가 적시에 이루어지며 사용자가 이를 쿼리하기 위해 따로 설정하지 않아도 되므로 유용할 수 있습니다. CloudWatch도 Amazon Redshift

Serverless 인스턴스의 사용량을 모니터링하는 데 유용할 수 있으며 인사이트를 제공하고 작업을 설정할 수 있는 추가 기능을 제공합니다.

시스템 뷰를 쿼리하여 사용량 시각화

SYS_SERVERLESS_USAGE 시스템 테이블을 쿼리하여 사용량을 추적하고 쿼리 요금을 가져옵니다.

```
select trunc(start_time) "Day",
(sum(charged_seconds)/3600::double
precision) * <Price for 1 RPU> as cost_incurred
from sys_serverless_usage
group by 1
order by 1
```

이 쿼리는 사용량을 기준으로 Amazon Redshift Serverless에 대해 발생하는 일일 비용을 제공합니다.

사용량 및 비용 결정을 위한 사용 참고 사항

- 실행하는 워크로드에 대해 초 단위의 RPU 시간 당 비용이 결제되며, 최소 요금 단위는 60초입니다.
- sys_serverless_usage 시스템 테이블의 레코드는 1분 간격으로 발생한 비용을 보여줍니다. 다음 열을 이해하는 것이 중요합니다.

charged_seconds 열:

- 시간 간격 동안 청구된 계산 단위(RPU) 초를 제공합니다. 결과에는 Amazon Redshift Serverless의 최소 요금이 포함됩니다.
- 트랜잭션 완료 후 컴퓨팅 리소스 사용에 대한 정보가 있습니다. 따라서 트랜잭션이 완료되지 않은 경우 이 열 값은 0이 될 수 있습니다.

compute_seconds 열:

- 실시간 컴퓨팅 사용 정보를 제공합니다. 여기에는 Amazon Redshift Serverless의 최소 요금이 포함되지 않습니다. 따라서 간격 동안 청구되는 청구된 초와 어느 정도 다를 수 있습니다.
- 트랜잭션이 종료되지 않은 경우에도 각 트랜잭션 동안 사용 정보를 표시하므로 제공되는 데이터는 실시간입니다.
- compute_seconds가 0이지만 charged_seconds가 0보다 크거나 그 반대의 경우가 있습니다. 이는 데이터가 시스템 뷰에 기록되는 방식에 따른 정상적인 동작입니다. 서버리스 사용량 세부 정보를 보다 정확하게 표시하려면 SYS_SERVERLESS_USAGE에서 데이터를 집계하는 것이 좋습니다.

쿼리 모니터링에 대한 자세한 내용은 [Amazon Redshift Serverless로 쿼리 및 워크로드 모니터링](#)을 참조하세요.

CloudWatch로 사용량 시각화

CloudWatch에서 사용할 수 있는 지표를 사용하여 사용량을 추적할 수 있습니다. CloudWatch용으로 생성된 지표는 현재 시간(분)에 사용된 총 RPU(초)를 나타내는 ComputeSeconds 및 해당 시간(분)의 총 컴퓨팅 용량을 나타내는 ComputeCapacity입니다. 사용량 지표는 Redshift Serverless 대시보드의 Redshift 콘솔에서도 찾을 수 있습니다. CloudWatch에 대한 자세한 내용은 [Amazon CloudWatch란 무엇인가요?](#)를 참조하세요.

서버리스 예약에 대한 결제

Amazon Redshift Serverless를 사용하면 종량제 요금 모델을 통해 클러스터를 프로비저닝 및 관리하지 않고도 분석을 실행하고 규모를 조정할 수 있습니다. 이제 서버리스 예약을 통해 Redshift Serverless에서 컴퓨팅 비용을 추가로 최적화하고 기존 및 새 워크로드의 비용 예측 가능성을 개선할 수 있습니다.

Amazon Redshift는 AWS 지급인 계정 수준에서 서버리스 예약을 관리하고 여러 AWS 계정 간에 예약을 공유할 수 있으므로 AWS 계정의 모든 Redshift Serverless 워크로드에서 컴퓨팅 비용을 최대 24% 절감할 수 있습니다. Amazon Redshift는 서버리스 예약에 시간 단위로 요금을 청구하고 초 단위로 예약을 측정하여 Redshift Serverless에서 제공하는 유연성을 유지하면서 하루 24시간, 주 7일 일관된 청구 모델을 제공합니다. Amazon Redshift는 지정된 RPU 수준을 초과하는 모든 사용량에 대해 표준 온디맨드 요금을 부과합니다.

Note

온디맨드 사용량을 제한하려면 최대 용량 설정을 사용하여 작업 그룹에 대한 리소스 사용 제한을 설정할 수 있습니다. 자세한 내용은 [Amazon Redshift Serverless에 대한 청구](#) 섹션을 참조하세요.

서버리스 예약의 이점

서버리스 예약은 Amazon Redshift Serverless의 할인된 요금 옵션입니다. 서버리스 예약을 사용하면 선결제 없이 온디맨드(OD) 요금보다 할인된 가격으로 1년 동안 지정된 수의 Redshift 처리 장치(RPU)를 약정할 수 있습니다. 선결제할 경우 더 많은 할인을 받을 수 있습니다. 서버리스 예약을 통해 Serverless에서 컴퓨팅 비용을 최적화하고 기존 및 새 워크로드의 비용 예측 가능성을 개선할 수 있습니다.

각 서버리스 예약은 AWS 계정 수준에서 구매되며 동일한 지급인 계정의 여러 Amazon Redshift Serverless 작업 그룹 간에 공유할 수 있습니다. 이렇게 하면 할인이 적용되는 방식을 유연하게 조정할 수 있습니다. 워크로드 패턴이 다른 여러 작업 그룹이 예약을 공유할 수 있습니다.

서버리스 예약 작동 방식

RPU 예약은 완료하는 데 몇 분 밖에 걸리지 않는 간단한 프로세스입니다. 여기에는 예약할 RPU 수준과 결제 유형 지정이 포함됩니다. Amazon Redshift Serverless는 필요한 예약 수준을 결정하고 사용량을 지속적으로 모니터링하는 데 도움이 되는 표준 AWS 결제 및 비용 관리 도구를 사용합니다. 서버리스 예약은 AWS 지급인 계정 수준에서 관리하고 동일한 지급인 계정에서 공유할 수 있으므로 AWS 계정의 모든 Redshift Serverless 워크로드에서 컴퓨팅 비용을 최대 24% 절감할 수 있습니다. 서버리스 예약은 시간 단위로 요금이 청구되고 초 단위로 측정되므로 Redshift Serverless에서 제공하는 유연성을 유지하면서 하루 24시간, 주 7일 일관된 청구 모델을 제공합니다. 지정된 RPU 수준을 초과하는 모든 사용량에는 표준 Redshift Serverless 온디맨드 요금이 부과됩니다.

동일한 AWS 계정 내에서 여러 서버리스 예약을 구매할 수 있습니다. 추가 서버리스 예약을 구매하면 예약들이 서로 겹쳐집니다. 예를 들어 두 개의 예약을 구매하고 각 예약에 100RPU를 선택하면 할인된 요금으로 총 200RPU가 제공됩니다.

Note

온디맨드 사용량 한도를 설정하려면 Amazon Redshift Serverless 콘솔에서 한도 탭을 선택한 다음 사용량 한도 관리를 선택하여 작업 그룹에 대한 최대 RPU를 설정합니다.

서버리스 예약을 구매하면 해당 예약이 즉시 적용되고 Redshift 콘솔의 Serverless 예약 대시보드에 표시됩니다.

Redshift 처리 장치(RPU) 사용을 분석하여 필요한 예약 수준 결정

Redshift Serverless 예약을 사용하면 1년 동안 특정 수의 Redshift 처리 장치(RPU)를 약정함으로써 예측 가능하고 낮은 컴퓨팅 비용을 적용받아 온디맨드 요금에 비해 할인 혜택을 누릴 수 있습니다. 이러한 할인은 선불 없이 결제 시 최대 20%, 전액 선불 결제 시 최대 24%가 될 수 있습니다. AWS 지급인 계정 수준에서 Redshift Serverless 예약을 구매하면 전체 AWS 연결 계정의 모든 Redshift Serverless 작업 그룹에 자동으로 할인이 적용되므로 여러 팀을 지원하면서 예산을 중앙에서 관리할 수 있습니다. Redshift Serverless는 초 단위로 사용량을 측정하고 매 시간 평균을 계산하여 시간당 요금을 청구하므로 사용한 용량에 대해서만 비용을 지불할 수 있습니다. Redshift Serverless 예약은 계정 전반에 걸친 유연한 적용과 기간 기반 절감을 결합하여 Redshift Serverless의 민첩성을 저하하지 않으면서 예측 가능한 분석 요금을 제공합니다.

예약의 RPU 사용 분석

RPU 사용량 수준은 두 가지 방법 중 하나로 확인할 수 있습니다. Redshift Serverless 대시보드의 7일 뷰를 사용하거나 Cost Explorer를 사용하여 장기 분석을 수행하는 방법입니다. 다음 절차에서는 RPU 사용을 분석하는 방법을 보여줍니다.

방법 1: Redshift Serverless 대시보드(7일 뷰)

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. Serverless 대시보드를 엽니다.
3. 작업 그룹을 선택합니다.
4. 지난 1시간부터 최대 1주까지 기간의 RPU 용량 사용량을 봅니다.

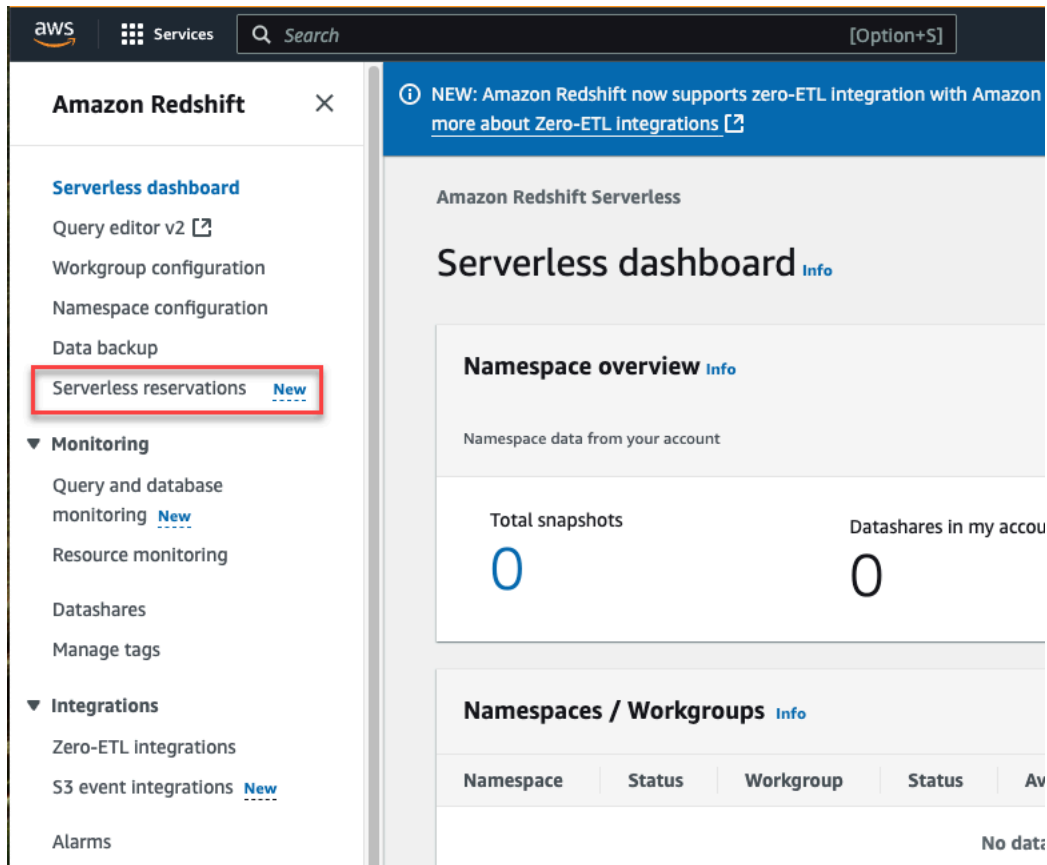
방법 2: AWS Cost Explorer(장기 분석)

1. AWS Management Console에 로그인하고 <https://console.aws.amazon.com/costmanagement/>에서 Cost Explorer 콘솔을 엽니다.
2. 세분성을 시간별로 설정합니다.
3. 사용량 유형으로 그룹화합니다.
4. 다음 필터를 적용합니다.
 - 서비스: Redshift
 - 리전: 로컬 리전
 - 사용량 유형: Redshift:ServerlessUsage용 필터
5. 선택한 리전의 시간당 서버리스 사용량에 대한 비용 및 사용량 그래프를 검토합니다.

콘솔을 사용하여 서버리스 예약 구매

예약을 구매할 때 할인될 RPU 수준을 선택합니다. RPU 수준을 선택하기 전에 기본 용량과 시간 경과에 따라 사용하는 온디맨드 용량을 파악하는 것이 좋습니다. 이 섹션에서는 용량을 확인하고 서버리스 예약을 예약하는 방법을 보여줍니다.

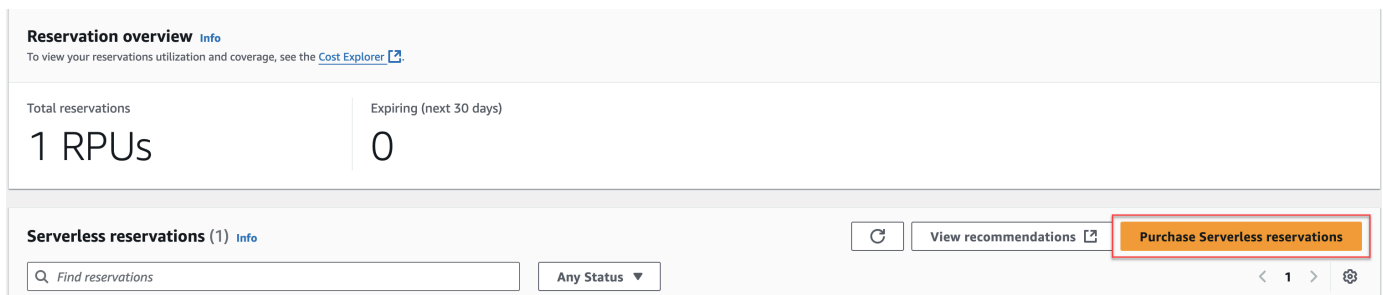
시작하려면 Redshift 콘솔에서 서버리스를 선택한 다음 메뉴에서 서버리스 예약을 선택합니다.



이 기능에 대한 설명과 기존 예약 목록이 콘솔에 나옵니다. 여기에서 예약을 구매하거나 사용 가능한 보고서 및 모니터링 도구를 사용하여 현재 사용량을 확인할 수 있습니다. 이를 통해 기존 RPU 수준과 예약할 적정 RPU 수를 확인할 수 있습니다.

예약을 구매하려면 다음 단계를 완료하세요.

1. 서버리스 예약 구매를 선택합니다.



2. 일련의 선택 항목이 있는 안내가 나타납니다. 예약할 서버리스 예약 RPU 수준을 입력합니다. 어느 수준이 적합한지 확실하지 않은 경우 이 섹션에서 자세히 설명하는 도구를 사용할 수 있습니다.

Serverless reservation

Enter the reserved RPU capacity to purchase for this AWS account.

RPU

The value must range from 1 to any number.

- 결제 유형을 설정합니다. 예약된 RPU에 대해 선결제를 선택하거나 매월 요금을 지불할 수 있습니다. 선결제를 선택하면 더 큰 할인을 받을 수 있습니다.

Payment type

The upfront cost is paid once, when the reservation is purchased. The monthly cost is for comparison only and represents the total hourly cost for 30 days.

☒ All Upfront

Results in a 24% discount

Full upfront payment for the duration of the reservation.

☐ No Upfront

Results in a 20% discount

Monthly installments for the duration of the reservation.

i The discount is applied to the price you currently pay for on-demand RPUs, which is \$0.36 per RPU hour. The comparison table shows the hourly savings for each payment type. For more information, see the [Cost Explorer](#).

- 선택을 마쳤으면 서버리스 예약 구매를 선택한 다음 확인을 선택합니다.

예약을 확인한 후에는 해당 예약이 예약 목록에 표시됩니다.

Serverless reservations (1) Info								
Find reservations			Any Status ▾	View recommendations Purchase Serverless reservations				
Reservation ID	Status ▾	Expiration date ▲	Reservation (RPUs)	Payment type ▾	Up front ▾	Monthly ▾	Effective hourly	
09753c4b-b75a-4a3f-915c-ca97e1b92a7b	Payment-pending	April 22, 2026, 15:48 (UTC-04:00)	1	All Upfront	\$ 2400.00	\$ 0.00	\$ 0.27	

사용 노트

- 예약은 변경하거나 삭제할 수 없습니다. 하지만 추가 예약을 생성하여 더 많은 담당률을 확보할 수 있습니다.
- Redshift Serverless는 온디맨드 RPU를 사용하기 전에 워크로드에 예약된 RPU를 사용하여 비용을 절감합니다. 예약한 RPU 수를 초과하는 경우, 이 추가적인 RPU에 대해서는 Redshift Serverless 온디맨드 요금이 발생하기 시작합니다.
- Amazon Redshift Serverless에 대한 무료 크레딧은 서버리스 예약에 적용되지 않으며 온디맨드로 청구되는 RPU에만 적용됩니다.

서버리스 예약 예시

이 시나리오에서 AWS 지급인/연결된 계정에는 두 개의 Amazon Redshift 작업 그룹이 있습니다.

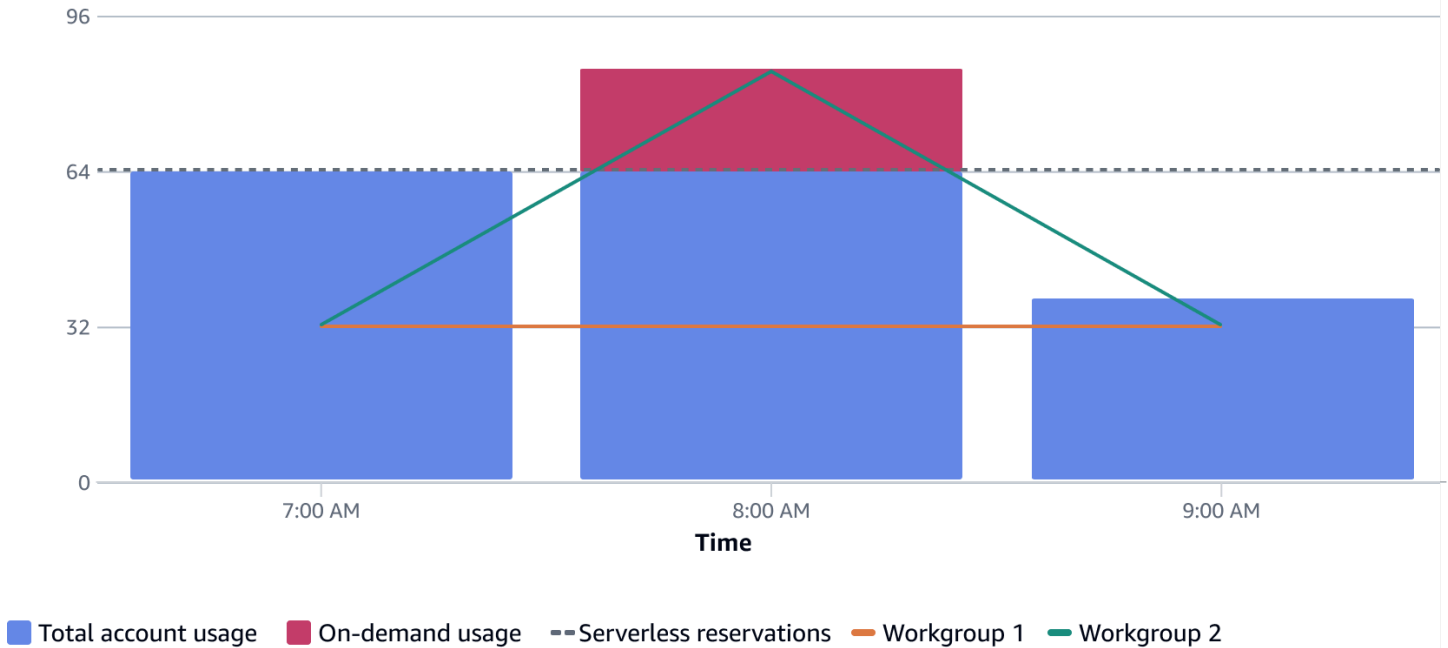
- 작업 그룹 1은 비즈니스 인텔리전스 팀의 경우와 같이 안정적인 상태로 사용됩니다.
- 작업 그룹 2에는 ETL 작업의 경우와 같이 사용량이 급증하는 예측할 수 없는 워크로드가 있습니다.

이러한 작업 그룹의 비용을 최적화하여 1년 서버리스 예약을 구매하려고 합니다. 과거 데이터를 기반으로 두 작업 그룹이 안정적인 상태에서 64RPU를 사용한다는 것을 확인했습니다. 그러나 작업 그룹 2는 가끔 32RPU에서 48RPU로 증가하고 단기간 동안 24RPU로 떨어집니다. 과거 추세에 맞춰 예약의 RPU 수준을 64RPU로 설정했습니다. 시간당 청구 세부 정보는 다음과 같습니다.

- 처음 한 시간 동안은 과거 사용량 추세와 마찬가지로 두 작업 그룹 모두 32RPU를 사용하여 총 계정 사용량이 64RPU였습니다. 이 시간 동안 모든 RPU에는 서버리스 예약 할인 요금이 부과됩니다. 이는 64RPU의 사용 수준이 64RPU 서버리스 예약과 동일하기 때문입니다.
- 두 번째 시간 동안 작업 그룹 1은 계속해서 32RPU를 사용합니다. 하지만 작업 그룹 2는 48RPU로 급증하여 계정 전체 사용량이 80RPU가 됩니다. 이 시간 동안 서버리스 예약 할인 요금으로 64RPU가 청구되고 Redshift Serverless 온디맨드 요금으로 16RPU가 청구됩니다.
- 세 번째 시간 동안 작업 그룹 1은 계속해서 32RPU를 사용하고 작업 그룹 2는 8RPU로 감소합니다. 이 시간에 계정 합계가 40RPU인 경우에도 계정에는 64RPU 서버리스 예약 요금이 부과됩니다.

작업 그룹 사용량 변화와 온디맨드 및 서버리스 예약 요금 청구 세부 정보는 다음 다이어그램을 참조하세요.

Usage (RPU)



AWS CLI 또는 Amazon Redshift API를 사용하여 서버리스 예약 구매

create-reservation을 사용하여 RPU 예약을 생성합니다. 다음은 명령을 보여줍니다.

```
create-reservation
--capacity
--offering-id
```

capacity를 예약할 RPU 수로 설정합니다.

스토리지에 대한 결제

기본 스토리지 용량은 Redshift 관리형 스토리지(RMS)로 결제됩니다. 스토리지는 GB/월별로 결제됩니다. 스토리지 결제는 컴퓨팅 용량에 대한 결제와 별개입니다. 사용자 스냅샷에 사용되는 스토리지는 사용량 티어에 따라 표준 백업 결제 요금으로 결제됩니다.

데이터 전송 비용 및 기계 학습(ML) 비용은 프로비저닝된 클러스터와 마찬가지로 별도로 적용됩니다. AWS 리전 간 스냅샷 복제 및 데이터 공유는 요금 페이지에 설명된 전송 요금으로 결제됩니다. 자세한 내용은 [Amazon Redshift 요금](#)을 참조하세요.

CloudWatch로 결제 사용량 시각화

스냅샷 스토리지 사용량을 추적하는 지표 SnapshotStorage가 생성되어 CloudWatch로 전송됩니다. CloudWatch에 대한 자세한 내용은 [Amazon CloudWatch란 무엇인가요?](#)를 참조하세요.

Amazon Redshift Serverless 무료 평가판 사용

Amazon Redshift Serverless는 무료 평가판을 제공합니다. 무료 평가판을 이용하는 경우 Redshift 콘솔에서 무료 평가판 크레딧 잔액을 확인하고 [SYS_SERVERLESS_USAGE](#) 시스템 보기에서 무료 평가판 사용량을 확인할 수 있습니다. 무료 평가판 사용에 대한 청구 세부 정보는 빌링 콘솔에 표시되지 않으므로 유의하십시오. 무료 평가판 사용이 종료된 후에만 빌링 콘솔에서 사용량을 볼 수 있습니다. Amazon Redshift Serverless 무료 평가판에 대한 자세한 내용은 [Amazon Redshift Serverless 무료 평가판](#)을 참조하세요.

결제 사용량 메모

- 사용량 기록 - 쿼리 또는 트랜잭션은 트랜잭션이 완료, 롤백 또는 중지된 후에만 측정되고 기록됩니다. 예를 들어 트랜잭션이 2일 동안 실행되는 경우 RPU 사용량은 해당 트랜잭션이 완료된 후 기록됩니다. `sys_serverless_usage`를 쿼리하여 진행 중인 사용을 실시간으로 모니터링할 수 있습니다. 트랜잭션 기록은 RPU 사용량 변동 사항으로 반영되고 특정 시간 및 일일 사용 비용에 영향을 줄 수 있습니다.
- 명시적 트랜잭션 작성 - 가장 좋은 것은 트랜잭션을 끝내는 것입니다. 오픈 트랜잭션을 종료하거나 롤백하지 않으면 Amazon Redshift Serverless에서 계속해서 RPU를 사용합니다. 예를 들어 명시적 `BEGIN TRAN`을 작성하는 경우 해당 `COMMIT` 및 `ROLLBACK` 문이 반드시 있어야 합니다.
- 취소된 쿼리 - 쿼리를 실행하고 완료되기 전에 취소해도 쿼리가 실행된 시간에 대해선 결제해야 합니다.
- 확장 - Amazon Redshift Serverless 인스턴스는 일관된 성능을 유지하기 위해 부하가 높은 기간을 처리하도록 확장을 시작할 수 있습니다. Amazon Redshift Serverless 결제에는 기본 컴퓨팅 용량과 확장된 용량 모두가 동일한 RPU 요금으로 포함됩니다.
- 축소 - Amazon Redshift Serverless는 기본 RPU 용량에서 확장하여 부하가 높은 기간을 처리합니다. 경우에 따라 쿼리 로드가 감소한 후 일정 기간 동안 RPU 용량이 높은 설정으로 유지될 수 있습니다. 예상치 못한 비용을 방지하기 위해 콘솔에서 최대 RPU 시간을 설정하는 것이 좋습니다.
- 시스템 테이블 - 시스템 테이블을 쿼리할 때 쿼리 시간에 대해 결제해야 합니다.
- Redshift Spectrum - Amazon Redshift Serverless가 있고 쿼리를 실행하는 경우 데이터 레이크 쿼리에 대해 별도의 요금이 부과되지 않습니다. Amazon S3에 저장된 데이터를 쿼리하는 경우 트랜잭션 시간별 요금은 로컬 데이터에 대한 쿼리와 동일합니다.

- 연합 쿼리 - 연합 쿼리는 데이터 웨어하우스 또는 데이터 레이크의 쿼리와 동일한 방식으로 특정 시간 간격 동안 사용된 RPU 수를 기준으로 요금이 부과됩니다.
- 스토리지 - 스토리지의 경우 GB/월별로 별도로 결제해야 합니다.
- 최소 요금 - 리소스 사용의 최소 요금은 60초이며, 요금은 초 단위로 측정됩니다.
- 스냅샷 결제 - 스냅샷 결제는 변경되지 않습니다. 스토리지에 따라 요금이 부과되며 GB/월 요금으로 결제해야 합니다. 데이터 웨어하우스를 지난 24시간 내 특정 시점으로 30분 단위로 무료 복원할 수 있습니다. 자세한 내용은 [Amazon Redshift 요금](#)을 참조하세요.
- 자동 최적화는 추가 컴퓨팅 리소스를 사용하여 실행 - Amazon Redshift Serverless는 일반적으로 사용자 쿼리와 함께 자동 최적화 작업을 실행합니다. 이러한 작업은 Autonomics라고 하며 이에 대한 요금은 부과되지 않습니다.

추가 컴퓨팅 리소스 할당을 활성화하면 Amazon Redshift는 사용자 활동이 많은 기간에도 필요한 경우 Autonomics를 실행합니다. 이러한 경우 Autonomics를 실행하는 데 소요된 시간에 대한 요금이 청구될 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [자동 데이터베이스 최적화를 위한 추가 컴퓨팅 리소스 할당](#)을 참조하세요.

결제를 예측 가능하게 유지하기 위한 Amazon Redshift Serverless 모범 사례

다음은 일관된 결제를 유지하는 데 도움이 되는 모범 사례 및 기본 제공 설정입니다.

- 각 트랜잭션을 종료해야 합니다. BEGIN을 사용하여 트랜잭션을 시작하는 경우 END를 사용하는 것도 중요합니다.
- 오류 처리에 대한 모범 사례를 활용하여 오류에 적절하게 대응하고 각 트랜잭션을 종료할 수 있습니다. 진행 중인 트랜잭션을 최소화하면 불필요한 RPU 사용을 방지할 수 있습니다.
- SESSION TIMEOUT은 진행 중인 트랜잭션과 유휴 세션을 종료하여 이에 도움이 됩니다. 이로 인해 3,600초(1시간) 이상 유휴 또는 비활성 상태로 유지된 세션이 시간 초과됩니다. 이로 인해 21,600초(6시간) 이상 진행 중이고 비활성 상태로 유지된 모든 트랜잭션이 시간 초과됩니다. 이 시간 초과 설정은 장기 실행 쿼리를 위해 세션을 열어 두려는 경우 등을 위해 필요할 때 특정 사용자에게 대해 명시적으로 변경할 수 있습니다. [CREATE USER](#) 주제에서는 사용자의 SESSION TIMEOUT을 조정하는 방법을 보여줍니다.
- 대부분의 경우 특별히 필요한 사용 사례가 없는 한 SESSION TIMEOUT 값을 연장하지 않는 것이 좋습니다. 진행 중인 트랜잭션이 있는 채로 세션이 유휴 상태로 유지될 경우 세션이 닫힐 때까지 RPU가 사용될 수 있습니다. 이로 인해 불필요한 비용이 발생합니다.

- Amazon Redshift Serverless의 최대 쿼리 실행 시간은 86,399초(24시간)입니다. 개방형 트랜잭션의 최대 비활성 기간은 Amazon Redshift Serverless가 트랜잭션과 연결된 세션을 종료하기 전 6시간입니다. 자세한 내용은 [Amazon Redshift Serverless 객체 할당량](#) 섹션을 참조하세요.

연결 풀링 사용 시 Amazon Redshift Serverless 청구

Amazon Redshift Serverless는 연결 풀에서 전송한 경량 상태 확인 쿼리를 포함하여 모든 수신 쿼리를 청구 가능한 사용자 활동으로 취급합니다. 이 동작은 쿼리가 애플리케이션, JDBC/ODBC 드라이버, 연결 풀링 프레임워크 중 어디에서 시작되었든 관계없이 적용됩니다. 각 상태 확인 쿼리는 컴퓨팅 사용량을 트리거하며 쿼리 용도 또는 오리진에 관계없이 요금이 발생합니다. 따라서 실제 사용자 워크로드가 실행되지 않는 경우에도 열린 연결 풀을 유지하면 비용이 발생할 수 있습니다.

연결 풀링은 애플리케이션과 Amazon Redshift Serverless 엔드포인트 간에 지속적 연결 풀을 유지합니다. 이러한 연결을 정상적이고 사용 가능한 상태로 유지하기 위해 풀링 메커니즘은 종종 경량 쿼리 또는 빈 쿼리(예: `SELECT 1`)를 정기적으로 전송합니다. 이러한 자동 쿼리는 연결 상태를 확인합니다.

연결 풀링을 사용하는 경우 의도하지 않은 요금을 최소화하기 위해 다음 모범 사례를 고려하세요.

- 연결 풀링 구성에서 상태 확인 또는 연결 유지 쿼리의 빈도를 검토하고 최적화하여 상태 확인 빈도를 조정합니다.
- 시스템 유휴 시간 동안 불필요한 연결 이탈 또는 백그라운드 쿼리 활동을 최소화하도록 연결 풀링을 구성하여 유휴 시스템 설정을 최적화합니다.
- 오버헤드를 줄일 수 있다면 애플리케이션 수준 풀링을 구현하거나 연결 수명 주기 관리를 개선합니다.
- 연결 풀링 구성에서 허용하는 경우 하트비트 또는 검증 쿼리를 비활성화합니다. 특정 연결 문자열 파라미터 또는 구성 파일을 확인하여 이러한 설정을 조정합니다.
- TCP 연결 유지 설정 미세 조정: 드라이버의 내부 하트비트 메커니즘을 비활성화할 수 없는 경우 운영 체제 또는 애플리케이션 수준에서 전송 제어 프로토콜(TCP) 연결 유지 설정을 조정하여 연결 시간 초과 문제를 해결합니다. 자세한 내용은 운영 체제, JDBC/ODBC 드라이버 또는 연결 풀 설명서를 참조하세요.
- 데이터베이스 연결 풀링 최적화: 연결을 관리하고 연결 오버헤드를 최소화하도록 연결 풀(HikariCP, Apache 데이터베이스 연결 풀)을 구성합니다. 최대 연결 수, 유휴 제한 시간, 검증 쿼리(필요한 경우)와 같은 파라미터에 주의를 기울입니다. 이 최적화는 Amazon Redshift Serverless 컴퓨팅 사용량을 실제 워크로드 수요에 맞게 조정하여 비용을 잠재적으로 절감하는 데 도움이 됩니다.

제로 ETL을 통한 Amazon Redshift Serverless 비용 최적화

Amazon Redshift Serverless에서 제로 ETL 통합을 실행하면서 비용을 최적화하려면, 워크로드 요구 사항에 따라 환경 규모를 조정하고 새로 고침 설정을 변경할 수 있습니다. 다음과 같이 조정하는 것이 좋습니다.

- 가능한 경우 워크로드에 더 낮은 기본 RPU 용량인 8 RPU를 사용합니다.
- 대상 Redshift 인스턴스의 REFRESH_INTERVAL을 설정하여 최신성과 비용 간 균형을 맞춥니다. 짧은 간격은 거의 실시간 업데이트를 보장하지만 컴퓨팅 비용을 증가시킵니다. 간격이 길수록(5분 이상) 즉각적인 최신성이 중요하지 않은 워크로드(예: 보고 또는 과거 분석)의 요금이 절감됩니다. Redshift 대상 REFRESH_INTERVAL을 편집하려면 [ALTER DATABASE](#) 설명의 새로 고침 간격 절을 참조하세요.
- Amazon Redshift Serverless 환경의 활용도를 극대화하려면 ETL 없이 데이터를 수집하는 동안 분석 워크로드를 동시에 실행합니다. 이렇게 하면 컴퓨팅 용량이 여러 비즈니스 목적에 적극적으로 활용됩니다.

Amazon Redshift 서버리스에 연결

Amazon Redshift Serverless 인스턴스를 설정한 후에는 아래에 설명된 다양한 방법으로 인스턴스에 연결할 수 있습니다. 여러 팀이나 프로젝트가 있고 비용을 별도로 관리하고 싶다면 별도의 AWS 계정을 사용할 수 있습니다.

Amazon Redshift Serverless를 사용할 수 있는 AWS 리전 목록은 Amazon Web Services 일반 참조의 [Amazon Redshift Serverless API](#)에 나열된 엔드포인트를 참조하세요.

Amazon Redshift Serverless는 현재 AWS 리전의 AWS 계정에 있는 서버리스 환경에 연결됩니다. Amazon Redshift Serverless는 포트 범위 5431~5455 및 8191~8215 내의 VPC에서 실행됩니다. 기본 값은 5439입니다. 현재는 API 작업 UpdateWorkgroup 및 AWS CLI 작업 update-workgroup을 통해서만 포트를 변경할 수 있습니다.

Amazon Redshift 서버리스에 연결

다음 구문을 사용하여 Amazon Redshift Serverless의 데이터베이스(이름: dev)에 연결할 수 있습니다.

```
workgroup-name.account-number.aws-region.redshift-serverless.amazonaws.com:port/dev
```

예를 들어 다음 연결 문자열은 us-east-1 리전을 지정합니다.

```
default.123456789012.us-east-1.redshift-serverless.amazonaws.com:5439/dev
```

JDBC 드라이버를 통해 Amazon Redshift 서버리스에 연결

다음 방법 중 하나를 통해 Amazon Redshift에서 제공하는 JDBC 드라이버 버전 2.x 드라이버를 사용하여 원하는 SQL 클라이언트로 Amazon Redshift Serverless에 연결할 수 있습니다.

JDBC 드라이버 버전 2.x를 사용하여 데이터베이스 인증을 위한 로그인 자격 증명과 연결하려면 다음 구문을 사용합니다. 포트 번호는 선택 사항입니다. 포함되지 않은 경우 Amazon Redshift Serverless는 기본적으로 포트 번호 5439로 설정됩니다. 포트 범위 5431~5455 또는 8191~8215에서 다른 포트로 변경할 수 있습니다. 서버리스 엔드포인트의 기본 포트를 변경하려면 AWS CLI 및 Amazon Redshift API를 사용하세요.

```
jdbc:redshift://workgroup-name.account-number.aws-region.redshift-serverless.amazonaws.com:5439/dev
```

예를 들어 다음 연결 문자열은 작업 그룹 기본값, 계정 ID 123456789012 및 리전 us-east-2를 지정합니다.

```
jdbc:redshift://default.123456789012.us-east-2.redshift-serverless.amazonaws.com:5439/dev
```

JDBC 드라이버 버전 2.x를 사용하여 IAM에 연결하려면 다음 구문을 사용합니다. 포트 번호는 선택 사항입니다. 포함되지 않은 경우 Amazon Redshift Serverless는 기본적으로 포트 번호 5439로 설정됩니다. 포트 범위 5431~5455 또는 8191~8215에서 다른 포트로 변경할 수 있습니다. 서버리스 엔드포인트의 기본 포트를 변경하려면 AWS CLI 및 Amazon Redshift API를 사용하세요.

```
jdbc:redshift:iam://workgroup-name.account-number.aws-region.redshift-serverless.amazonaws.com:5439/dev
```

예를 들어 다음 연결 문자열은 작업 그룹 기본값, 계정 ID 123456789012 및 리전 us-east-2를 지정합니다.

```
jdbc:redshift:iam://default.123456789012.us-east-2.redshift-serverless.amazonaws.com:5439/dev
```

ODBC의 경우 다음 구문을 사용합니다.

```
Driver={Amazon Redshift (x64)}; Server=workgroup-name.account-number.aws-region.redshift-serverless.amazonaws.com; Database=dev
```

2.1.0.9 이전 버전의 JDBC 드라이버를 사용 중이며 IAM에 연결하는 경우 다음 구문을 사용해야 합니다.

```
jdbc:redshift:iam://redshift-serverless-<name>:aws-region/database-name
```

예를 들어 다음 연결 문자열은 작업 그룹 기본값 및 AWS 리전 us-east-1을 지정합니다.

```
jdbc:redshift:iam://redshift-serverless-default:us-east-1/dev
```

드라이버에 대한 자세한 내용은 [Amazon Redshift에서 연결 구성](#) 섹션을 참조하세요.

JDBC 및 ODBC 연결 문자열 찾기

SQL 클라이언트 도구를 사용하여 작업 그룹에 연결하려면 JDBC 또는 ODBC 연결 문자열이 있어야 합니다. Amazon Redshift Serverless 콘솔의 작업 그룹 세부 정보 페이지에서 연결 문자열을 확인할 수 있습니다.

작업 그룹의 연결 문자열 찾기

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 Redshift Serverless를 선택합니다.
3. 탐색 메뉴에서 작업 그룹 구성을 선택한 후 목록에서 작업 그룹 이름을 선택하여 세부 정보를 엽니다.
4. JDBC URL 및 ODBC URL 연결 문자열은 일반 정보(General information) 섹션에서 추가 세부 정보와 함께 사용할 수 있습니다. 각 문자열은 작업 그룹이 실행되는 AWS 리전을 기반으로 합니다. 해당 연결 문자열 옆에 있는 아이콘을 선택하여 연결 문자열을 복사합니다.

Data API를 사용하여 Amazon Redshift Serverless에 연결

Amazon Redshift Data API를 사용하여 Amazon Redshift Serverless에 연결할 수도 있습니다. AWS CLI 호출에서 `cluster-identifier` 파라미터 대신 `workgroup-name` 파라미터를 사용합니다.

Data API에 대한 자세한 내용은 [Amazon Redshift Data API 사용](#) 섹션을 참조하세요. Python에서 데이터 API를 호출하는 예제 코드 및 기타 예제는 [Redshift 데이터 API 시작하기](#)를 참조하고 GitHub의 quick-start 및 use-cases 폴더를 살펴보세요.

SSL을 사용하여 Amazon Redshift Serverless에 연결

Amazon Redshift Serverless에 대한 보안 연결 구성

SSL 연결을 지원하기 위해 Redshift Serverless는 [AWS Certificate Manager\(ACM\)](#)에서 발급한 SSL 인증서를 생성해 각 작업 그룹에 설치합니다. ACM 인증서는 대부분의 운영 체제, 웹 브라우저 및 클라이언트에서 공개적으로 신뢰합니다. SQL 클라이언트 또는 애플리케이션의 sslmode 연결 옵션이 require, verify-ca 또는 verify-full로 설정된 상태에서 SSL을 사용하여 Redshift Serverless에 연결하는 경우 인증서 번들을 다운로드해야 합니다. 클라이언트에 인증서가 필요한 경우 Redshift Serverless는 다음과 같이 번들 인증서를 제공합니다.

- <https://s3.amazonaws.com/redshift-downloads/amazon-trust-ca-bundle.crt>에서 번들을 다운로드합니다.
 - 예상 MD5 체크섬 번호는 418dea9b6d5d5de7a8f1ac42e164cdcf입니다.
 - sha256 체크섬 번호는 36dba8e4b8041cd14b9d60158893963301bcbb92e1c456847784de2acb5bd550입니다.

<https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt>에 있던 이전 인증서 번들을 사용하지 마세요.

- 중국 AWS 리전에서는 <https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/amazon-trust-ca-bundle.crt>에서 번들을 다운로드합니다.
 - 예상 MD5 체크섬 번호는 418dea9b6d5d5de7a8f1ac42e164cdcf입니다.
 - sha256 체크섬 번호는 36dba8e4b8041cd14b9d60158893963301bcbb92e1c456847784de2acb5bd550입니다.

<https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ca-bundle.crt> 및 <https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ssl-ca-cert.pem>에 있던 이전 인증서 번들을 사용하지 마세요.

⚠ Important

Redshift Serverless는 SSL 인증서 관리 방법을 바꿨습니다. 계속해서 SSL을 사용하여 작업 그룹에 연결하려면 현재 신뢰하는 루트 CA 인증서를 업데이트해야 할 수 있습니다. SSL 연결용 ACM 인증서에 대한 자세한 내용은 [SSL 연결을 위해 ACM 인증서로 이전](#) 섹션을 참조하세요.

기본적으로 작업 그룹 데이터베이스는 SSL의 사용 여부와 상관없이 연결을 허용합니다.

SSL 연결만 허용하는 새 작업 그룹을 만들려면 `create-workgroup` 명령을 사용하고 `require_ssl` 파라미터를 `true`로 설정합니다. 다음 예제를 사용하려면 네임스페이스 이름을 *yourNamespaceName*으로 바꾸고 *WorkgroupName*을 작업 그룹 이름으로 바꿉니다.

```
aws redshift-serverless create-workgroup \  
--namespace-name yourNamespaceName \  
--workgroup-name yourWorkgroupName \  
--config-parameters parameterKey=require_ssl,parameterValue=true
```

SSL 연결만 허용하는 기존 작업 그룹을 업데이트하려면 `update-workgroup` 명령을 사용하고 `require_ssl` 파라미터를 `true`로 설정합니다. 참고로 Redshift Serverless는 `require_ssl` 파라미터가 업데이트되면 작업 그룹을 다시 시작합니다. 다음 예제에서 *yourWorkgroupName*을 작업 그룹의 이름으로 바꿉니다.

```
aws redshift-serverless update-workgroup \  
--workgroup-name yourWorkgroupName \  
--config-parameters parameterKey=require_ssl,parameterValue=true
```

Amazon Redshift는 ECDHE(Elliptic Curve Diffie—Hellman Ephemeral) 키 계약 프로토콜을 지원합니다. ECDHE 프로토콜에서는 클라이언트와 서버가 비보안 채널을 통해 공유 비밀을 구성하는 데 사용하는 타원 곡선 공개-비공개 키 페어를 갖습니다. ECDHE를 사용하기 위해 Amazon Redshift에서 아무 것도 구성할 필요가 없습니다. ECDHE를 통해 클라이언트와 서버 간 통신을 암호화하는 SQL 클라이언트 도구에서 연결하는 경우 Amazon Redshift가 제공된 암호 목록을 사용해 적절히 연결합니다. 자세한 내용은 Wikipedia의 [Elliptic curve diffie—hellman](#) 및 OpenSSL 웹 사이트의 [Ciphers](#) 섹션을 참조하세요.

Amazon Redshift Serverless에 대한 FIPS 준수 SSL 연결 구성

FIPS 준수 SSL 연결을 사용하는 새 작업 그룹을 생성하려면 `create-workgroup` 명령을 사용하고 `use_fips_ssl` 및 `require_ssl` 파라미터를 `true`로 설정합니다. 다음 예제를 사용하려면 네임스페이스 이름을 *yourNamespaceName*으로 바꾸고 *WorkgroupName*을 작업 그룹 이름으로 바꿉니다.

```
aws redshift-serverless create-workgroup \  
--namespace-name yourNamespaceName \  
--workgroup-name yourWorkgroupName \  
--config-parameters '[{"parameterKey": "require_ssl", "parameterValue": "true"},  
{"parameterKey": "use_fips_ssl", "parameterValue": "true"}]'
```

FIPS 준수 SSL 연결을 사용하도록 기존 작업 그룹을 업데이트하려면 `update-workgroup` 명령을 사용하고 `use_fips_ssl` 및 `require_ssl` 파라미터를 `true`로 설정합니다. 참고로 Redshift Serverless는 `use_fips_ssl` 파라미터가 업데이트되면 작업 그룹을 다시 시작합니다. 다음 예제에서 *yourWorkgroupName*을 작업 그룹의 이름으로 바꿉니다.

```
aws redshift-serverless update-workgroup \  
--workgroup-name yourWorkgroupName \  
--config-parameters '[{"parameterKey": "require_ssl", "parameterValue": "true"},  
{"parameterKey": "use_fips_ssl", "parameterValue": "true"}]'
```

FIPS 준수 연결을 사용하도록 Redshift Serverless를 구성하는 방법에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [use_fips_ssl](#)을 참조하세요.

Amazon Redshift 관리형 VPC 엔드포인트에서 Amazon Redshift Serverless에 연결

다른 VPC 엔드포인트에서 Amazon Redshift Serverless에 연결

Amazon Redshift Serverless 작업 그룹을 위한 관리형 VPC 엔드포인트를 설정 또는 구성하는 방법에 대한 자세한 내용은 [Redshift 관리형 VPC 엔드포인트 작업](#)을 참조하세요.

인터페이스 VPC 엔드포인트에서 Amazon Redshift Serverless에 연결(AWS PrivateLink)

인터페이스 VPC 엔드포인트(AWS PrivateLink)에서 Amazon Redshift Serverless에 연결하는 방법에 대한 자세한 내용은 [인터페이스 VPC 엔드포인트](#) 섹션을 참조하세요.

다른 계정의 Redshift VPC 엔드포인트에서 Amazon Redshift Serverless에 연결

교차 VPC 엔드포인트에서 Amazon Redshift Serverless에 연결

Amazon Redshift Serverless는 VPC에서 프로비저닝됩니다. 다른 계정의 VPC에 액세스 권한을 부여하여 사용자 계정의 Amazon Redshift Serverless에 액세스하도록 할 수 있습니다. 이는 관리형 VPC 엔드포인트의 연결과 비슷하지만, 이 경우 연결은 예를 들어 다른 계정의 데이터베이스 클라이언트에서 시작됩니다. 수행할 수 있는 몇 가지 작업이 있습니다.

- 데이터베이스 소유자는 Amazon Redshift Serverless가 포함된 VPC에 대한 액세스 권한을 동일한 리전의 다른 계정에 부여할 수 있습니다.
- 데이터베이스 소유자는 Amazon Redshift Serverless 액세스를 취소할 수 있습니다.

크로스 계정 액세스의 주요 이점은 데이터베이스 협업이 더 쉬워진다는 것입니다. 데이터베이스가 포함된 계정에서 사용자를 프로비저닝하지 않아도 데이터베이스에 액세스할 수 있으므로 구성 단계가 줄어들고 시간이 절약됩니다.

다른 계정의 VPC에 대한 액세스 권한을 부여하는 데 필요한 권한

액세스를 허용하거나 허용된 액세스를 변경하려면 권한 부여자에게 다음 권한이 포함된 할당된 권한 정책이 필요합니다.

- redshift-serverless:PutResourcePolicy
- redshift-serverless:GetResourcePolicy
- redshift-serverless>DeleteResourcePolicy
- ec2:CreateVpcEndpoint
- ec2:ModifyVpcEndpoint

AWS 관리형 정책 AmazonRedshiftFullAccess에 지정된 다른 권한이 필요할 수 있습니다. 자세한 내용은 [Amazon Redshift Serverless에 권한 부여](#)를 참조하세요.

피부여자에게는 다음 권한이 포함된 할당된 권한 정책이 필요합니다.

- redshift-serverless:ListWorkgroups
- redshift-serverless>CreateEndpointAccess
- redshift-serverless:UpdateEndpointAccess

- redshift-serverless:GetEndpointAccess
- redshift-serverless:ListEndpointAccess
- redshift-serverless>DeleteEndpointAccess

가장 좋은 방법은 권한 정책을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에 할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.

다음은 VPC 간 액세스를 구성하는 데 사용되는 샘플 리소스 정책입니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CrossAccountCrossVPCAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "123456789012",
          "234567890123"
        ]
      },
      "Action": [
        "redshift-serverless:CreateEndpointAccess",
        "redshift-serverless:UpdateEndpointAccess",
        "redshift-serverless>DeleteEndpointAccess",
        "redshift-serverless:GetEndpointAccess"
      ],
      "Resource": "arn:aws:redshift-serverless:*:*:*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-east-1"
        },
        "StringLike": {
          "aws:SourceVpc": [
            "vpc-11223344556677889",
            "vpc-12345678"
          ]
        }
      }
    }
  ]
}
```

```
}
]
}
```

이 섹션의 다음 절차에서는 이러한 작업을 수행하는 사용자가 예를 들어 해당 권한이 나열된 할당된 IAM 역할을 통해 적절한 권한을 할당받은 것으로 가정합니다. 또한 작업 그룹에 적절한 리소스 권한이 부여된 IAM 역할이 연결되어 있다고 가정합니다.

콘솔을 사용하여 다른 계정에 VPC 액세스 권한 부여

이 절차는 데이터베이스 소유자가 데이터베이스 액세스 권한을 부여하려는 경우 데이터베이스 액세스를 구성하는 단계를 보여줍니다.

소유자 계정으로부터 액세스 권한 부여

1. Amazon Redshift Serverless 작업 그룹의 속성에는 데이터 액세스 탭에 부여된 계정이라는 목록이 있습니다. 작업 그룹에 대한 액세스 권한이 부여된 계정 및 VPC가 표시됩니다. 목록을 찾아 액세스 권한 부여를 선택하여 목록에 계정을 추가합니다.
2. 피부여자 정보를 추가할 수 있는 창이 나타납니다. AWS 계정 ID를 입력합니다. 계정 ID는 액세스 권한을 부여할 계정의 12자리 ID입니다.
3. 피부여자에게 모든 VPC에 대한 액세스 권한을 부여하거나 특정 VPC에 대한 액세스 권한을 부여합니다. 특정 VPC에만 액세스 권한을 부여하는 경우 각 ID를 입력하고 VPC 추가를 선택하여 해당 VPC에 ID를 추가할 수 있습니다.
4. 완료한 후 변경 사항 저장을 선택합니다.

변경 사항을 저장하면 해당 계정이 부여된 계정 목록에 나타납니다. 항목에는 계정 ID와 액세스 권한이 부여된 VPC 목록이 표시됩니다.

데이터베이스 소유자는 계정에 대한 액세스 권한을 취소할 수도 있습니다. 소유자는 언제든지 액세스 권한을 취소할 수 있습니다.

계정 액세스 권한 취소

1. 부여된 계정 목록에서 시작할 수 있습니다. 먼저 계정을 하나 이상 선택합니다.
2. 액세스 취소를 선택합니다.

액세스 권한이 부여되면 피부여자의 데이터베이스 관리자는 콘솔을 확인하여 액세스 권한이 있는지 확인할 수 있습니다.

콘솔을 사용하여 다른 계정에 액세스할 수 있는 액세스 권한이 부여되었는지 확인

1. Amazon Redshift Serverless 작업 그룹 속성의 데이터 액세스 탭에는 권한이 부여된 계정이라는 목록이 있습니다. 이 작업 그룹에서 액세스할 수 있는 계정이 표시됩니다. 피부여자는 작업 그룹의 엔드포인트 URL을 사용하여 작업 그룹에 직접 액세스할 수 없습니다. 작업 그룹에 액세스하려면 피부여자는 엔드포인트 섹션으로 이동하여 엔드포인트 생성을 선택합니다.
2. 그런 다음 피부여자는 작업 그룹에 액세스하기 위해 엔드포인트 이름과 VPC를 제공합니다.
3. 엔드포인트가 성공적으로 생성되면 엔드포인트 섹션에 엔드포인트가 나타나고 해당 엔드포인트에 대한 엔드포인트 URL이 표시됩니다. 이 엔드포인트 URL을 사용하여 작업 그룹에 액세스할 수 있습니다.

콘솔을 사용하여 다른 계정에 액세스 권한 부여

액세스 권한을 부여하는 계정은 먼저 `put-resource-policy`를 사용하여 연결할 다른 계정에 액세스 권한을 부여해야 합니다. 데이터베이스 소유자는 `put-resource-policy`를 호출하여 다른 계정이 작업 그룹에 연결을 생성하도록 권한을 부여할 수 있습니다. 그러면 피부여자 계정은 `create-endpoint-authorization`을 사용하여 허용된 VPC를 통해 작업 그룹에 대한 연결을 생성할 수 있습니다.

다음은 특정 계정 및 VPC에 대한 액세스를 허용하기 위해 호출할 수 있는 `put-resource-policy` 속성을 보여줍니다.

```
aws redshift-serverless put-resource-policy
--resource-arn <value>
--policy <value>
```

명령을 호출한 후 `get-resource-policy`를 호출하여 `resource-arn`을 지정함으로써 리소스에 액세스할 수 있는 계정과 VPC를 확인할 수 있습니다.

피부여자는 다음과 같은 호출을 할 수 있습니다. 부여된 액세스 권한에 대한 정보가 표시됩니다. 특히 액세스 권한이 부여된 VPC가 포함된 목록을 반환합니다.

```
aws redshift-serverless list-workgroups
--owner-account <value>
```

이는 피부여자가 권한 부여 계정으로부터 엔드포인트 권한 부여에 대한 정보를 가져오기 위한 것입니다. owner-account는 공유하는 계정입니다. 실행하면 각 작업 그룹에 대해 허용된 VPC 목록인 CrossAccountVpcs가 반환됩니다. 참고로 다음은 작업 그룹에 사용할 수 있는 모든 속성을 보여줍니다.

```
Output: workgroup (Object)
workgroupId String,
workgroupArn String,
workgroupName String,
status: String,
namespaceName: String,
baseCapacity: Integer, (Not-applicable)
enhancedVpcRouting: Boolean,
configParameters: List,
securityGroupIds: List,
subnetIds: List,
endpoint: String,
publiclyAccessible: Boolean,
creationDate: Timestamp,
port: Integer,
CrossAccountVpcs: List
```

Note

[클러스터 재배치](#)는 추가 Redshift 네트워킹 기능을 구성하기 위한 사전 조건은 아니라는 점을 기억하시기 바랍니다. 또한 다음을 활성화하기 위해 기능을 켜 필요도 없습니다.

- 계정 간 또는 리전 간 VPC에서 Redshift로 연결 - 이 섹션에 설명된 대로 하나의 AWS Virtual Private Cloud(VPC)에서 Redshift 데이터베이스를 포함하는 다른 VPC로 연결할 수 있습니다.
- 사용자 지정 도메인 이름 설정 - Amazon Redshift 클러스터 또는 Amazon Redshift Serverless 작업 그룹에 대해 사용자 지정 도메인(사용자 지정 URL이라고도 함)을 만들어 엔드포인트 이름을 더 기억하기 쉽고 간단하게 만들 수 있습니다. 자세한 내용은 [클라이언트 연결에 사용자 지정 도메인 이름 사용](#)을 참조하세요.

추가 리소스

네트워크 트래픽 설정을 지정하는 방법은 [기본 또는 사용자 지정 보안 그룹 구성을 통한 퍼블릭 액세스 가능성](#)에서 확인할 수 있습니다. 클러스터에 퍼블릭 액세스가 가능한 사용 사례가 나와 있습니다.

네트워크 트래픽 설정을 지정하는 방법은 [기본 또는 사용자 지정 보안 그룹 구성을 통한 프라이빗 액세스 가능성](#)에서 확인할 수 있습니다. 인터넷에서 클러스터를 사용할 수 없는 사용 사례가 나와 있습니다.

권한 부여, 추가 서비스에 대한 액세스 권한 부여, IAM 역할 생성을 비롯한 Amazon Redshift Serverless의 보안 연결에 대한 자세한 내용은 [Amazon Redshift Serverless의 Identity and Access Management](#) 섹션을 참조하세요.

Amazon Redshift Serverless에서 페더레이션 사용자에게 부여할 데이터베이스 역할 정의

조직의 직원에게는 관련된 역할 모음이 있습니다. 예를 들어 프로그래머 및 매니저와 같은 직무에 대한 역할이 있습니다. 역할에 따라 액세스할 수 있는 애플리케이션 및 데이터가 결정됩니다. 대부분의 조직에서는 Microsoft Active Directory와 같은 ID 공급업체를 사용하여 사용자 및 그룹에 역할을 할당합니다. 조직에서 개별 사용자를 관리할 필요성이 줄어들기 때문에 리소스 액세스를 제어하는 데 역할을 사용하는 경우가 늘어났습니다.

최근에 역할 기반 액세스 제어가 Amazon Redshift Serverless에 도입되었습니다. 예를 들어 데이터베이스 역할을 사용하면 스키마나 테이블과 같은 데이터 및 객체에 대한 액세스를 보호할 수 있습니다. 또는 역할을 사용하여 시스템 모니터 또는 데이터베이스 관리자와 같은 상승된 권한 집합을 정의할 수 있습니다. 그러나 데이터베이스 역할에 리소스 권한을 부여한 후에는 조직의 사용자 역할을 데이터베이스 역할에 연결하는 추가 단계가 있습니다. 처음 로그인할 때 SQL 문을 실행하여 각 사용자에게 데이터베이스 역할을 할당할 수 있지만, 이렇게 하려면 많은 노력이 필요합니다. 더 쉬운 방법은 부여할 데이터베이스 역할을 정의하고 이를 Amazon Redshift Serverless로 전달하는 것입니다. 이렇게 하면 초기 로그인 프로세스를 간소화할 수 있다는 이점이 있습니다.

GetCredentials를 사용하여 Amazon Redshift Serverless에 역할을 전달할 수 있습니다. 사용자가 Amazon Redshift Serverless 데이터베이스에 처음으로 로그인하면 연결된 데이터베이스 사용자가 생성되고 일치하는 데이터베이스 역할에 매핑됩니다. 이 주제에서는 Amazon Redshift Serverless에 역할을 전달하는 메커니즘을 자세히 설명합니다.

데이터베이스 역할 전달에는 몇 가지 주요 사용 사례가 있습니다.

- 사용자가 일반적으로 페더레이션이 구성된 타사 ID 공급업체를 통해 로그인하고 세션 태그를 통해 역할을 전달하는 경우
- 사용자가 IAM 로그인 보안 인증 정보를 통해 로그인하고 태그 키 및 값을 통해 역할이 전달되는 경우

역할 기반 액세스 제어에 대한 자세한 내용은 [역할 기반 액세스 제어\(RBAC\)](#)를 참조하세요.

데이터베이스 역할 정의

Amazon Redshift Serverless에 역할을 전달하려면 먼저 데이터베이스에서 데이터베이스 역할을 구성하고 데이터베이스 리소스에 대한 적절한 권한을 부여해야 합니다. 예를 들어 간단한 시나리오에서는 sales라는 데이터베이스 역할을 만들고 판매 데이터가 있는 쿼리 테이블에 대한 액세스 권한을 부여할 수 있습니다. 데이터베이스 역할을 생성하고 권한을 부여하는 방법에 대한 자세한 내용은 [CREATE ROLE](#) 및 [GRANT](#)를 참조하세요.

페더레이션 사용자에게 부여할 데이터베이스 역할을 정의하는 사용 사례

이 섹션에서는 Amazon Redshift Serverless에 데이터베이스 역할을 전달하여 데이터베이스 리소스에 대한 액세스를 간소화할 수 있는 몇 가지 사용 사례를 간략하게 설명합니다.

ID 공급업체를 사용하여 로그인

첫 번째 사용 사례에서는 조직의 ID 및 액세스 관리 서비스에 사용자 ID가 있다고 가정합니다. 이 서비스는 JumpCloud 또는 Okta와 같은 클라우드 기반이거나 Microsoft Active Directory와 같은 온프레미스일 수 있습니다. 목표는 사용자가 쿼리 편집기 V2 또는 JDBC와 같은 클라이언트에 로그인할 때 ID 공급업체의 사용자 역할을 데이터베이스 역할에 자동으로 매핑하는 것입니다. 이를 설정하려면 몇 가지 구성 작업을 완료해야 합니다. 여기에는 다음이 포함됩니다.

1. 신뢰 관계를 사용하여 ID 제공업체(IdP)와의 페더레이션 통합을 구성합니다. 이는 전제 조건입니다. 이를 설정하면 ID 공급업체는 SAML 어설션을 통해 사용자를 인증하고 로그인 보안 인증 정보를 제공할 책임이 있게 됩니다. 자세한 내용은 [AWS와 서드 파티 SAML 솔루션 공급업체 통합](#)을 참조하세요. [Active Directory Federation Services\(AD FS\)를 통해 Amazon Redshift 쿼리 편집기 V2에 대한 액세스 페더레이션](#) 또는 [Okta를 통해 Amazon Redshift 쿼리 편집기 v2에 대한 Single Sign-On 액세스 페더레이션](#)에서 자세한 내용을 확인할 수도 있습니다.

2. 사용자는 다음 정책 권한이 있어야 합니다.

- GetCredentials – Amazon Redshift Serverless에 로그인하기 위한 임시 인증용 보안 인증 정보 제공
- sts:AssumeRoleWithSAML – 엔터프라이즈 ID 저장소 또는 디렉터리를 역할 기반 AWS 액세스에 연결하는 메커니즘 제공
- sts:TagSession – ID 제공업체 보안 주체에 대한 태그 세션 작업 권한

이 경우 AssumeRoleWithSAML에서 SAML 인증 응답을 통해 인증된 사용자에게 대한 보안 인증 정보 집합을 반환합니다. 이 작업은 사용자별 보안 인증 정보 없이 ID 저장소 또는 디렉터리를 역할 기반 AWS 액세스에 연결하는 메커니즘을 제공합니다. AssumeRoleWithSAML에 대한 권한이 있는

사용자의 경우 ID 공급업체는 역할 정보를 전달하는 데 사용되는 SAML 어설션을 관리할 책임이 있습니다.

가장 좋은 방법은 권한 정책을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에 할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.

3. 콜론으로 구분된 역할 값을 사용하여 role1:role2 형식으로 태그 RedshiftDbRoles를 구성합니다. 예를 들어 manager:engineer입니다. 이는 ID 공급업체에 구성된 세션 태그 구현에서 검색할 수 있습니다. SAML 인증 요청은 프로그래밍 방식으로 역할을 전달합니다. 세션 태그 전달에 대한 자세한 내용은 [AWS STS에서 세션 태그 전달](#)을 참조하세요.

데이터베이스에 존재하지 않는 역할 이름을 전달하면 해당 역할 이름은 무시됩니다.

이 사용 사례에서는 사용자가 페더레이션 ID를 사용하여 로그인하면 권한 부여 요청에서 세션 태그 키와 값을 통해 해당 역할이 전달됩니다. 이후 권한이 부여된 다음 GetCredentials가 데이터베이스로 역할을 전달합니다. 연결에 성공하면 데이터베이스 역할이 매핑되고 사용자는 역할에 해당하는 데이터베이스 작업을 수행할 수 있습니다. 작업의 핵심은 초기 권한 부여 요청에서 RedshiftDbRoles 세션 태그에 역할이 할당되는 것입니다. 세션 태그 전달에 대한 자세한 내용은 [AssumeRoleWithSAML을 사용하여 세션 태그 전달](#)을 참조하세요.

IAM 보안 인증 정보를 사용하여 로그인

두 번째 사용 사례에서는 사용자의 역할을 전달하고 사용자는 IAM 보안 인증 정보를 통해 데이터베이스 클라이언트 애플리케이션에 액세스할 수 있습니다.

1. 이 경우 로그인하는 사용자에게는 다음 작업에 대한 정책 권한이 할당되어야 합니다.

- tag:GetResources – 지정된 태그와 연결된 태그 지정 리소스 반환
- tag:GetTagKeys – 현재 사용 중인 태그 키 반환

가장 좋은 방법은 권한 정책을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에 할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.

2. Amazon Redshift Serverless와 같은 데이터베이스 서비스에 액세스하려면 허용 권한도 필요합니다.
3. 이 사용 사례의 경우 AWS Identity and Access Management에서 역할에 대한 태그 값을 구성하세요. 태그를 편집하고 역할이 포함된 태그 값 문자열과 함께 RedshiftDbRoles라는 태그 키를 만들도록 선택할 수 있습니다. 예를 들어 manager:engineer를 만들 수 있습니다.

사용자가 로그인하면 해당 역할이 권한 부여 요청에 추가되고 데이터베이스로 전달됩니다. 이는 기존 데이터베이스 역할에 매핑됩니다.

추가적인 리소스

사용 사례에서 언급한 대로 IdP와 AWS 간의 신뢰 관계를 구성할 수 있습니다. 자세한 내용은 [신뢰 당사자 신뢰로 SAML 2.0 IdP 구성 및 클레임 추가](#)를 참조하세요.

Amazon Redshift Serverless의 Identity and Access Management

Amazon Redshift에 액세스하려면 AWS가 요청을 인증하는 데 사용할 수 있는 자격 증명이 필요합니다. 이러한 보안 인증 정보에는 Amazon Redshift Serverless와 같은 AWS 리소스에 액세스할 수 있는 권한이 있어야 합니다.

다음 섹션에서는 리소스에 액세스할 수 있는지 대상을 제어하여 리소스를 보호할 수 있도록 AWS Identity and Access Management(IAM) 및 Amazon Redshift를 사용하는 방법에 대한 세부 정보를 제공합니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#) 섹션을 참조하세요.

Amazon Redshift Serverless에 권한 부여

Amazon Redshift Spectrum에서 다른 AWS 서비스에 액세스하려면 권한이 필요합니다. 일부 Amazon Redshift 기능을 사용하려면 Amazon Redshift가 사용자 대신 다른 AWS 서비스에 액세스해야 합니다. Amazon Redshift Serverless 인스턴스가 자동으로 작동하도록 엔드포인트에 보안 인증 정보를 제공합니다. 보안 자격 증명을 제공하는 데 선호되는 방법은 AWS Identity and Access Management(IAM) 역할을 지정하는 것입니다. Amazon Redshift 콘솔을 통해 IAM 역할을 생성하고 이를 기본값으로 설정할 수도 있습니다. 자세한 내용은 [Amazon Redshift의 기본값으로 IAM 역할 생성](#) 섹션을 참조하세요.

다른 AWS 서비스에 액세스하려면 적절한 권한이 있는 IAM 역할을 생성합니다. 또한 역할을 Amazon Redshift Serverless와 연결해야 합니다. 또한 Amazon Redshift 명령을 실행할 때 역할의 Amazon 리소스 이름(ARN)을 지정하거나 default 키워드를 지정합니다.

<https://console.aws.amazon.com/iam/>에서 IAM 역할에 대한 신뢰 관계를 변경할 때 보안 주체 서비스 이름으로 redshift-serverless.amazonaws.com 및 redshift.amazonaws.com을 포함해야 합니다. 사용자를 대신하여 다른 AWS 서비스에 액세스하기 위해 IAM 역할을 관리하는 방법에 대한 내용은 [Amazon Redshift가 사용자를 대신하여 AWS 서비스에 액세스할 수 있도록 권한 부여](#) 섹션을 참조하세요.

Amazon Redshift의 기본값으로 IAM 역할 생성

Redshift 콘솔을 통해 IAM 역할을 생성하면 Amazon Redshift가 프로그래밍 방식으로 AWS 계정에 역할을 생성합니다. 또한 Amazon Redshift는 기존 AWS 관리형 정책을 여기에 자동으로 연결합니다. 이 접근 방식을 사용하면 Amazon Redshift 콘솔 내에 머물 수 있고 역할 생성을 위해 IAM 콘솔로 전환할 필요가 없습니다.

클러스터를 위해 콘솔을 통해 생성한 IAM 역할에는 AmazonRedshiftAllCommandsFullAccess 관리형 정책이 자동으로 연결됩니다. 이 IAM 역할을 통해 Amazon Redshift는 IAM 계정의 AWS 리소스에 대한 데이터를 복사, 언로드, 쿼리 및 분석할 수 있습니다. 관련 명령에는 COPY, UNLOAD, CREATE EXTERNAL FUNCTION, CREATE EXTERNAL TABLE, CREATE EXTERNAL SCHEMA, CREATE MODEL 및 CREATE LIBRARY가 포함됩니다. Amazon Redshift의 기본 역할로 IAM 역할을 생성하는 방법에 대한 자세한 내용은 [Amazon Redshift의 기본값으로 IAM 역할 생성](#) 섹션을 참조하세요.

Amazon Redshift의 기본값으로 IAM 역할을 생성하기 시작하려면 AWS Management Console을 열고 Amazon Redshift 콘솔을 선택한 다음 메뉴에서 Redshift Serverless를 선택합니다. Serverless 대시보드에서 새 작업 그룹을 생성할 수 있습니다. 생성 단계에서는 IAM 역할을 선택하거나 새 IAM 역할을 구성하는 방법을 안내합니다.

기존 Amazon Redshift Serverless 작업 그룹이 있고 이에 대한 IAM 역할을 구성하려는 경우 AWS Management Console을 엽니다. Amazon Redshift 콘솔을 선택한 다음 Redshift Serverless를 선택합니다. Amazon Redshift Serverless 콘솔에서 기존 작업 그룹에 대한 네임스페이스 구성을 선택합니다. 보안 및 암호화에서 권한을 편집할 수 있습니다.

네임스페이스에 IAM 역할 할당

각 IAM 역할은 각 역할이 AWS에서 수행할 수 있는 작업을 결정하는 권한 정책이 있는 AWS 자격 증명입니다. 역할은 역할이 필요한 사용자라면 누구나 맡을 수 있습니다. 또한 각 네임스페이스는 테이블, 스키마 등의 객체와 사용자의 모음입니다. Amazon Redshift Serverless를 사용하는 경우 여러 IAM 역할을 네임스페이스에 연결할 수 있습니다. 이렇게 하면 데이터베이스 객체 모음에 대한 권한을 적절히 구성하기가 더 쉬워져 역할들이 내부 데이터와 외부 데이터 모두에 대해 작업을 수행할 수 있습니다. 예를 들어 Amazon Redshift 데이터베이스에서 COPY 명령을 실행하여 Amazon S3에서 데이터를 검색하고 Redshift 테이블을 채울 수 있습니다.

이 섹션의 앞부분에서 설명한 대로 콘솔을 사용하여 여러 역할을 네임스페이스에 연결할 수 있습니다. API 명령 CreateNamespace 또는 CLI 명령 create-namespace를 사용할 수도 있습니다. API 또는 CLI 명령을 사용하면 IAMRoles에 하나 이상의 역할을 채워 IAM 역할을 네임스페이스에 할당할 수 있습니다. 특히 모음에 특정 역할에 대한 ARN을 추가합니다.

네임스페이스 관련 IAM 역할 관리

AWS Management Console에서 AWS Identity and Access Management의 역할에 대한 권한 정책을 관리할 수 있습니다. 네임스페이스 구성(Namespace configuration)에서 사용 가능한 설정을 사용하여 네임스페이스에 대한 IAM 역할을 관리할 수 있습니다. 네임스페이스 및 Amazon Redshift Serverless에서 네임스페이스를 사용하는 방법에 대한 자세한 내용은 [작업 그룹 및 네임스페이스](#) 섹션을 참조하세요.

Amazon Redshift용 IAM 자격 증명 시작하기

Amazon Redshift 콘솔에 처음 로그인하고 Amazon Redshift Serverless를 처음 사용하는 경우 필요한 정책을 포함하는 IAM 역할이 연결된 사용자로 로그인하는 것이 좋습니다. Amazon Redshift Serverless 인스턴스 생성을 시작한 후 Amazon Redshift는 로그인할 때 사용된 IAM 역할 이름을 기록합니다. 동일한 자격 증명을 사용하여 Amazon Redshift 콘솔과 Amazon Redshift Serverless 콘솔에 로그인할 수 있습니다.

Amazon Redshift Serverless 인스턴스를 생성하는 동안 데이터베이스를 생성할 수 있습니다. 쿼리 편집기 v2를 사용하여 임시 자격 증명 옵션으로 데이터베이스에 연결합니다.

데이터베이스에 대해 유지되는 새 관리자 이름과 암호를 추가하려면 관리자 자격 증명 사용자 지정(Customize admin user credentials)을 선택하고 새 관리자 이름과 관리자 암호를 입력합니다.

Amazon Redshift Serverless 사용을 시작하고 콘솔에서 처음으로 작업 그룹 및 네임스페이스를 생성하려면 권한 정책이 연결된 IAM 역할을 사용합니다. 이 역할에 관리자 권한 `arn:aws:iam::aws:policy/AdministratorAccess` 또는 IAM 정책에 연결된 전체 Amazon Redshift 권한 `arn:aws:iam::aws:policy/AmazonRedshiftFullAccess`가 있는지 확인합니다.

다음 시나리오에서는 Amazon Redshift Serverless 콘솔을 시작할 때 Amazon Redshift Serverless에서 IAM 자격 증명을 사용하는 방법을 간략하게 설명합니다.

- 기본 설정 사용(Use default settings)을 선택하면 Amazon Redshift Serverless가 현재 IAM 자격 증명을 데이터베이스 슈퍼 사용자로 변환합니다. Amazon Redshift Serverless 콘솔에서 동일한 IAM 자격 증명을 사용하여 Amazon Redshift Serverless의 데이터베이스에서 슈퍼유저 작업을 수행할 수 있습니다.
- Amazon Redshift Serverless에서 관리자 사용자 이름(Admin user name)과 암호를 지정하지 않고 설정 사용자 지정(Customize settings)을 선택하는 경우 현재 IAM 보안 인증 정보가 기본 관리자 보안 인증 정보로 사용됩니다.
- 설정 사용자 지정(Customize settings)을 선택하고 관리자 이름(Admin user name)과 암호를 지정하는 경우 Amazon Redshift Serverless는 현재 IAM 자격 증명을 데이터베이스 슈퍼 사용자로 변환합니다.

니다. Amazon Redshift Serverless는 슈퍼 사용자로도 또 다른 장기 로그인 사용자 이름과 암호 페어를 생성합니다. 현재 IAM 자격 증명 또는 생성된 사용자 이름과 암호 페어를 사용하여 슈퍼 사용자로 데이터베이스에 로그인할 수 있습니다.

데이터베이스 역할 권한으로 Amazon Redshift Serverless 데이터베이스 객체에 액세스

이 절차에서는 [Amazon Redshift 데이터베이스 역할](#)을 통해 테이블을 쿼리할 수 있는 권한을 부여하는 방법을 보여 줍니다. 역할은 IAM에서 사용자에게 첨부된 태그를 통해 할당되며, 이 태그는 사용자가 로그인할 때 Amazon Redshift에 전달됩니다. [Amazon Redshift Serverless에서 페더레이션된 사용자에게 부여할 데이터베이스 역할 정의](#)의 개념을 예시를 통해 설명합니다. 이 단계를 완료하면 사용자를 데이터베이스 역할에 연결하고 각 데이터베이스 객체에 대한 권한을 설정하지 않아도 된다는 이점이 있습니다. 이렇게 하면 사용자가 테이블에 데이터를 쿼리, 수정 또는 추가하고 다른 작업을 수행할 수 있는 기능을 간편하게 관리할 수 있습니다.

이 절차에서는 이미 Amazon Redshift Serverless 데이터베이스를 설정했으며 데이터베이스에 권한을 부여할 수 있는 기능이 있다고 가정합니다. 또한 AWS 콘솔에서 IAM 사용자를 만들고, IAM 역할을 만들고, 정책 권한을 할당할 수 있는 권한이 있다고 가정합니다.

1. IAM 콘솔을 사용하여 IAM 사용자를 만듭니다. 나중에 이 사용자로 데이터베이스에 연결합니다.
2. 쿼리 편집기 v2 또는 다른 SQL 클라이언트를 사용하여 Redshift 데이터베이스 역할을 만듭니다. 데이터베이스 역할 생성에 대한 자세한 내용은 [CREATE ROLE](#)을 참조하세요.

```
CREATE ROLE urban_planning;
```

[SVV_ROLES](#) 시스템 뷰를 쿼리하여 역할이 생성되었는지 확인합니다. 또한 시스템 역할도 반환합니다.

```
SELECT * from SVV_ROLES;
```

3. 생성한 데이터베이스 역할에 테이블에서 선택할 수 있는 권한을 부여합니다. (생성한 IAM 사용자는 결국 데이터베이스 역할을 통해 로그인하고 테이블에서 레코드를 선택하게 됩니다.) 다음 코드 예제의 역할 이름과 테이블 이름은 샘플입니다. 여기에서는 cities라는 테이블에서 선택할 수 있는 권한이 부여됩니다.

```
GRANT SELECT on TABLE cities to ROLE urban_planning;
```

4. AWS Identity and Access Management 콘솔을 사용하여 IAM 역할을 생성합니다. 이 역할은 쿼리 편집기 v2를 사용할 수 있는 권한을 부여합니다. 새 IAM 역할을 만들고 신뢰할 수 있는 엔터티 유형에 대해 AWS 계정을 선택합니다. 그런 다음 이 계정을 선택합니다. 역할에 다음 정책 권한을 부여합니다.
 - AmazonRedshiftReadOnlyAccess
 - tag:GetResources
 - tag:GetTagKeys
 - sqlworkbench:ListDatabases 및 sqlworkbench:UpdateConnection을 포함한 sqlworkbench에 대한 모든 작업.
5. IAM 콘솔에서 이전에 만든 IAM 사용자에게 RedshiftDbRoles 키가 있는 태그를 추가합니다. 태그의 값은 첫 번째 단계에서 만든 데이터베이스 역할과 일치해야 합니다. 샘플에서는 urban_planning입니다.

이 단계를 완료한 후 IAM 콘솔에서 만든 사용자에게 IAM 역할을 할당합니다. 사용자가 쿼리 편집기 v2로 데이터베이스에 로그인하면 태그에 있는 데이터베이스 역할 이름이 Amazon Redshift에 전달되어 해당 사용자와 연결됩니다. 따라서 데이터베이스 역할을 통해 적절한 테이블을 쿼리할 수 있습니다. 예를 들어 이 샘플의 사용자는 urban_planning 데이터베이스 역할을 통해 cities 테이블을 쿼리할 수 있습니다.

프로비저닝된 클러스터를 Amazon Redshift Serverless로 마이그레이션

기존의 프로비저닝된 클러스터를 Amazon Redshift Serverless로 마이그레이션하여 컴퓨팅 리소스 규모를 온디맨드 방식으로 자동 조정할 수 있습니다. 프로비저닝된 클러스터를 Amazon Redshift Serverless로 마이그레이션하면 사용한 리소스에 대해서만 비용을 지불하고 워크로드 수요에 따라 용량 규모를 자동으로 조정하여 비용을 최적화할 수 있습니다. 마이그레이션의 일반적인 사용 사례에는 임시 쿼리 실행, 주기적인 데이터 처리 작업 또는 리소스를 과도하게 프로비저닝하지 않고 예측할 수 없는 워크로드 처리 작업이 포함됩니다. 다음 태스크 세트를 수행하여 프로비저닝된 Amazon Redshift 클러스터를 서버리스 배포 옵션으로 마이그레이션하세요.

프로비저닝된 클러스터의 스냅샷 생성

Note

Amazon Redshift는 프로비저닝된 클러스터 스냅샷을 서버리스 네임스페이스로 복원할 때 인터티브된 키를 복합 키로 자동 변환합니다.

프로비저닝된 클러스터에서 Amazon Redshift Serverless로 데이터를 전송하려면 프로비저닝된 클러스터의 스냅샷을 생성한 후 Amazon Redshift Serverless에서 스냅샷을 복원합니다.

Note

데이터를 서버리스 작업 그룹으로 마이그레이션하기 전에, 프로비저닝된 클러스터 요구 사항이 Amazon Redshift Serverless에서 선택한 RPU 양과 호환되는지 확인하세요.

프로비저닝된 클러스터의 스냅샷을 생성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 Clusters(클러스터), Snapshots(스냅샷)을 선택한 다음 Create snapshot(스냅샷 생성)을 선택합니다.
3. 스냅샷 정의의 속성을 입력한 후 스냅샷 생성을 선택합니다. 스냅샷을 사용할 수 있을 때까지 시간이 다소 걸릴 수 있습니다.

프로비저닝된 클러스터 스냅샷을 서버리스 네임스페이스로 복원

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. Amazon Redshift 프로비저닝된 클러스터 콘솔에서 시작하여 Clusters(클러스터), Snapshots(스냅샷) 페이지로 이동합니다.
3. 사용할 스냅샷을 선택합니다.
4. Restore snapshot(스냅샷 복원), Restore to serverless namespace(서버리스 네임스페이스로 복원)을 선택합니다.
5. 스냅샷을 어느 네임스페이스로 복원할지 선택합니다.

6. 스냅샷에서 복원할 것인지 확인합니다. 이 작업은 서버리스 엔드포인트의 모든 데이터베이스를 프로비저닝된 클러스터의 데이터로 바꿉니다. 복원을 선택합니다.

프로비저닝된 클러스터 스냅샷에 대한 자세한 내용은 [Amazon Redshift 스냅샷](#)을 참조하세요.

드라이버를 사용하여 Amazon Redshift Serverless에 연결

원하는 SQL 클라이언트에 Amazon Redshift Serverless를 연결하려면, Amazon Redshift에서 제공하는 [JDBC 드라이버 버전 2.x](#)를 사용하면 됩니다. 최신 버전의 Amazon Redshift JDBC 드라이버 버전 2.x를 사용하여 Amazon Redshift에 연결하는 것이 좋습니다. 포트 번호는 선택에 따릅니다. 이 드라이버를 포함하지 않으면 Amazon Redshift Serverless는 기본적으로 포트 번호 5439로 설정됩니다. 포트 범위 5431~5455 또는 8191~8215에서 다른 포트로 변경할 수 있습니다. 서버리스 엔드포인트의 기본 포트를 변경하려면 AWS CLI 및 Amazon Redshift API를 사용하세요.

JDBC, ODBC 또는 Python 드라이버에 사용할 정확한 엔드포인트를 찾으려면 Amazon Redshift Serverless에서 작업 그룹 구성을 참조하세요. Amazon Redshift Serverless API 작업인 GetWorkgroup 또는 AWS CLI 작업인 get-workgroups를 사용하여 작업 그룹에 대한 정보를 반환한 다음 연결할 수도 있습니다.

암호 기반 인증을 사용하여 연결

암호 기반 인증과 함께 Amazon Redshift JDBC 드라이버 버전 2.x를 사용하여 연결을 설정하려면 다음 구문을 사용합니다.

```
jdbc:redshift://<workgroup-name>.<account-number>.<aws-region>.redshift-
serverless.amazonaws.com:5439/?username=<username>&password=<password>
```

암호 기반 인증과 함께 Amazon Redshift Python 커넥터를 사용하여 연결을 설정하려면 다음 구문을 사용합니다.

```
import redshift_connector
with redshift_connector.connect(
    host='<workgroup-name>.<account-number>.<aws-region>.redshift-
serverless.amazonaws.com',
    database='<database-name>',
    user='<username>',
    password='<password>'
    # port value of 5439 is specified by default
) as conn:
    pass
```

암호 기반 인증과 함께 Amazon Redshift ODBC 드라이버 버전 2.x를 사용하여 연결을 설정하려면 다음 구문을 사용합니다.

```
Driver={Amazon Redshift ODBC Driver (x64)}; Server=<workgroup-name>.<account-number>.<aws-region>.redshift-serverless.amazonaws.com; Database=database-name;
User=username; Password=password
```

IAM을 사용하여 연결

IAM으로 로그인하려면 Amazon Redshift Serverless [GetCredentials](#) API 작업을 사용합니다.

IAM 인증을 사용하려면 다음 예시에 보이는 것처럼 jdbc:redshift: 다음에 iam:을 Amazon Redshift JDBC URL에 넣습니다.

```
jdbc:redshift:iam://<workgroup-name>.<account-number>.<aws-region>.redshift-serverless.amazonaws.com:5439/<database-name>
```

이 Amazon Redshift Serverless 엔드포인트는 dbUser, dbGroup 또는 auto-create 사용자 지정을 지원하지 않습니다. 기본적으로 드라이버는 로그인 시 데이터베이스 사용자를 자동으로 생성합니다. 그런 다음 IAM에 지정된 태그 또는 ID 제공업체(idP)에 정의된 그룹을 기반으로 Amazon Redshift 데이터베이스 역할에 사용자를 할당합니다.

AWS 자격 증명에 redshift-serverless:GetCredentials 작업에 대한 올바른 IAM 정책이 있어야 합니다. 다음은 Amazon Redshift Serverless에 연결하기 위한 올바른 권한을 AWS 자격 증명에 부여하는 IAM 정책의 예입니다. IAM 권한에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하세요.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": "redshift-serverless:GetCredentials",
      "Resource": "*"
    }
  ]
}
```

}

IAM 기반 인증과 함께 Amazon Redshift Python 커넥터를 사용하여 연결을 설정하려면 다음 구문과 같이 코드에 `iam=true`를 사용합니다.

```
import redshift_connector
with redshift_connector.connect(
    iam=True,
    host='<workgroup-name>.<account-number>.<aws-region>.redshift-
serverless.amazonaws.com',
    database='<database-name>'
    <IAM credentials>
) as conn:
    pass
```

IAM credentials의 경우 다음을 포함한 모든 자격 증명을 사용할 수 있습니다.

- AWS 프로파일 구성
- IAM 자격 증명(엑세스 키 ID, 시크릿 액세스 키 및 선택적 세션 토큰)
- ID 제공업체 페더레이션

IAM 기반 인증 및 프로파일과 함께 Amazon Redshift ODBC 드라이버 버전 2.x를 사용하여 연결을 설정하려면 다음 구문을 사용합니다.

```
Driver={Amazon Redshift ODBC Driver (x64)}; IAM=true; Server=<workgroup-name>.<account-
number>.<aws-region>.redshift-serverless.amazonaws.com; Database=<database-name>;
Profile=<aws-profile-name>;
```

GetClusterCredentials API로 IAM을 사용하여 연결

Note

Amazon Redshift Serverless에 연결할 때는 [GetCredentials](#) API를 사용하는 것이 좋습니다. 이 API는 포괄적인 역할 기반 액세스 제어(RBAC) 기능과 GetClusterCredentials에서 사용할 수 없는 기타 새로운 기능을 제공합니다. 프로비저닝된 클러스터에서 서버리스 작업 그룹으로의 전환을 간소화하기 위해 GetClusterCredentials API를 지원하지만 최적의 호환성을 위해 가능한 한 빨리 GetCredentials를 사용하도록 마이그레이션하는 것이 좋습니다.

[GetClusterCredentials](#) API를 사용하여 Amazon Redshift Serverless에 대한 연결을 설정할 수 있습니다. 이 인증 방법을 구현하려면 다음 파라미터를 통합하여 클라이언트 또는 애플리케이션을 수정합니다.

- iam=true
- clusterid/cluster_identifier=redshift-serverless-*<workgroup-name>*
- region=*<aws-region>*

다음 예시는 세 드라이버 모두에서 BrowserSAML 플러그인을 사용하는 방법을 보여줍니다. 이는 사용 가능한 여러 인증 방식 중 하나입니다. 예시를 수정하여 특정 요구 사항에 따라 다른 인증 방식이나 플러그인을 사용할 수 있습니다.

GetClusterCredentials에 대한 IAM 정책 권한

다음은 Amazon Redshift Serverless에서 GetClusterCredentials를 사용하는 데 필요한 권한이 있는 샘플 IAM 정책입니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "redshift-serverless:GetWorkgroup",
        "redshift-serverless:GetNamespace",
        "redshift:GetClusterCredentials",
        "redshift>CreateClusterUser",
        "redshift:JoinGroup",
        "redshift:ExecuteQuery",
        "redshift:FetchResults",
        "redshift:DescribeClusters",
        "redshift:DescribeTable"
      ],
      "Resource": [
        "arn:aws:redshift-serverless:us-east-1:111122223333:workgroup/workgroup-name",

```

```

        "arn:aws:redshift-serverless:us-
east-1:111122223333:namespace/namespace-name",
        "arn:aws:redshift:us-east-1:111122223333:cluster:cluster-name",
        "arn:aws:redshift:us-east-1:111122223333:dbuser:database-name/
${redshift:DbUser}"
    ]
}
]
}

```

GetClusterCredentials로 Amazon Redshift JDBC 드라이버 버전 2.x를 사용하여 연결을 설정하려면 다음 구문을 사용합니다.

```

jdbc:redshift:iam://redshift-serverless-<workgroup-name>:<aws-region>/<database-name>?
plugin_name=com.amazon.redshift.plugin.BrowserSamlCredentialsProvider&login_url=<single
sign-on URL from IdP>"

```

GetClusterCredentials로 Amazon Redshift Python 커넥터를 사용하여 연결을 설정하려면 다음 구문을 사용합니다.

```

import redshift_connector
with redshift_connector.connect(
    iam=True,
    cluster_identifier='redshift-serverless-<workgroup-name>',
    region='<aws-region>',
    database='<database-name>',
    credentials_provider='BrowserSamlCredentialsProvider'
    login_url='<single sign-on URL from IdP>'
    # port value of 5439 is specified by default
) as conn:
    pass

```

GetClusterCredentials로 Amazon Redshift ODBC 드라이버 버전 2.x를 사용하여 연결을 설정하려면 다음 구문을 사용합니다.

```

Driver= {Amazon Redshift ODBC Driver (x64)}; IAM=true; isServerless=true;
ClusterId=redshift-serverless-<workgroup-name>; region=<aws-region>;
plugin_name=BrowserSAML;login_url=<single sign-on URL from IdP>

```

다음은 Windows에서 ODBC DSN을 구성하는 예입니다.

Amazon Redshift ODBC Driver Setup v2.1.7.0(64 bit) X

Connection SSL Proxy Cursor Advanced

Data Source Name: TestBrowserSAML

Description: TestBrowserSAML

Server:

Port: 5439

Database: dev

Authentication

Auth Type: Identity Provider: Browser SAML

User Name:

Password:

Auth Profile: ☐ Group Federation

Cluster ID: redshift-serverless-serverless Region: us-east-1

DB User: ☐ User Auto Create ☒ Serverless

DB Groups: ☐ Force Lower Case

Endpoint URL:

STS Endpoint URL:

WorkGroup:

DB Groups Filter: |

Login URL: https://dev-01234567.okta.com/i

Preferred Role:

Listen Port:

Timeout (sec):

Managed VPC URL:

OK Cancel Test

Amazon Redshift Serverless SDK 사용

Amazon Redshift SDK를 사용하여 관리 스크립트를 작성한 경우 새로운 Amazon Redshift Serverless SDK를 사용하여 Amazon Redshift Serverless 및 관련 리소스를 관리해야 합니다. 사용 가능한 API 작업에 대한 자세한 내용은 [Amazon Redshift Serverless API 참조 가이드](#)를 참조하세요.

작업 그룹 및 네임스페이스

Amazon Redshift Serverless에서 워크로드를 격리하고 다양한 리소스를 관리하기 위해 네임스페이스와 작업 그룹을 생성하고 스토리지 리소스와 컴퓨팅 리소스를 별도로 관리할 수 있습니다.

네임스페이스는 데이터베이스 객체와 사용자의 모음입니다. 스토리지 관련 네임스페이스는 스키마, 테이블, 사용자 또는 데이터 암호화용 AWS Key Management Service 키를 함께 그룹화합니다. 스토리지 속성에는 관리자의 데이터베이스 이름 및 암호, 권한, 암호화, 보안이 포함됩니다. 네임스페이스 아래에 그룹화된 기타 리소스에는 데이터 공유, 복구 시점, 사용량 한도 등이 있습니다. Amazon Redshift Serverless 콘솔, AWS Command Line Interface 또는 Amazon Redshift Serverless API를 사용하여 특정 리소스에 대해 이러한 스토리지 속성을 구성할 수 있습니다.

작업 그룹은 컴퓨팅 리소스의 모음입니다. 컴퓨팅 관련 작업 그룹은 RPU, VPC 서브넷 그룹 및 보안 그룹과 같은 컴퓨팅 리소스를 함께 그룹화합니다. 작업 그룹의 속성에는 네트워크 및 보안 설정이 포함됩니다. 작업 그룹 아래에 그룹화된 기타 리소스에는 액세스 및 사용량 한도 등이 있습니다. Amazon Redshift Serverless 콘솔, AWS Command Line Interface 또는 Amazon Redshift Serverless API를 사용하여 이러한 스토리지 속성을 구성할 수 있습니다.

하나 이상의 네임스페이스와 작업 그룹을 생성할 수 있습니다. 각 네임스페이스에는 하나의 작업 그룹만 연결할 수 있습니다. 역으로 각 작업 그룹은 하나의 네임스페이스에만 연결할 수 있습니다.

콘솔을 사용하는 작업 그룹 및 네임스페이스

Amazon Redshift Serverless를 설정하려면 몇 가지 구성 단계를 거쳐야 합니다. Amazon Redshift Serverless를 설정하는 단계에 따라 네임스페이스와 작업 그룹을 생성하고 서로 연결합니다. Amazon Redshift Serverless 콘솔을 사용하여 Amazon Redshift Serverless 구성을 설정하려는 경우 Amazon Redshift Serverless 시작하기(Get started with Amazon Redshift Serverless)를 선택하여 Amazon Redshift Serverless를 설정하고 상호 작용을 시작할 수 있습니다. 더 빠른 설정을 위해 기본 설정이 있는 환경을 선택하거나 조직의 요구 사항에 따라 설정을 명시적으로 구성할 수 있습니다. 이 과정에서 작업 그룹 및 네임스페이스에 대한 설정을 지정합니다.

환경을 설정한 후 [작업 그룹 속성](#) 및 [네임스페이스 속성](#)을 통해 설정에 익숙해질 수 있습니다.

AWS Command Line Interface 및 Amazon Redshift Serverless API를 사용하는 작업 그룹 및 네임스페이스

AWS 콘솔을 사용하는 것 외에도 AWS CLI 또는 Amazon Redshift Serverless API를 사용하여 작업 그룹 및 네임스페이스와 상호 작용할 수 있습니다. 아래 테이블에는 스냅샷 및 복구 시점을 관리하는 데 사용할 수 있는 API 및 CLI 작업이 나와 있습니다.

API 작업	CLI 명령	설명
CreateNamespace	create-namespace	네임스페이스를 생성합니다. Amazon Redshift Serverless는 기본 AWS Key Management Service 키를 사용하여 네임스페이스를 생성하지만 데이터를 암호화하기 위해 다른 키를 지정할 수 있습니다. 스냅샷을 복원하여 네임스페이스를 생성할 수도 있습니다. 자세한 내용은 스냅샷 및 복구 시점 작업 을 참조하세요.
UpdateNamespace	update-namespace	네임스페이스를 업데이트합니다.
GetNamespace	get-namespace	네임스페이스에 대한 정보를 검색합니다.
ListNamespaces	list-namespaces	네임스페이스 목록에 대한 정보를 검색합니다.
DeleteNamespace	delete-namespace	네임스페이스를 삭제합니다.
CreateWorkgroup	create-workgroup	작업 그룹을 생성합니다. 작업 그룹을 만들 때는 작업 그룹과 연결할 수 있는 기존 네임스페이스가 있는지 확인하세요. 작업 그룹을 만들 때 서브넷, 보안 그룹 또는 RPU와 같은 컴퓨

API 작업	CLI 명령	설명
		팅 리소스를 지정할 수 있습니다.
UpdateWorkgroup	update-workgroup	작업 그룹을 업데이트합니다.
GetWorkgroup	get-workgroup	작업 그룹에 대한 정보를 검색합니다.
ListWorkgroups	list-workgroups	작업 그룹 목록에 대한 정보를 검색합니다.
DeleteWorkgroup	delete-workgroup	작업 그룹을 삭제합니다.

작업 그룹

Amazon Redshift Serverless를 사용하면 작업 그룹을 생성하고 관리하여 다양한 워크로드 또는 사용자의 컴퓨팅 리소스를 분리하고 제어할 수 있습니다. 작업 그룹을 사용하면 메모리 및 동시성 규모 조정 제한과 같은 구성 옵션을 설정하고 워크로드 전반에서 쿼리 실행의 우선순위를 지정할 수 있습니다. 컴퓨팅 관련 작업 그룹은 RPU 및 VPC 서브넷 그룹과 같은 컴퓨팅 리소스를 함께 그룹화합니다.

네임스페이스가 있는 작업 그룹 생성

다음 단계에 따라 작업 그룹을 생성하세요. 작업 그룹 구성에 대한 자세한 내용은 [작업 그룹 속성](#) 섹션을 참조하세요.

1. Serverless dashboard(Serverless 대시보드)를 선택합니다. Create workgroup(작업 그룹 생성)을 선택합니다.
2. 작업 그룹 이름을 입력합니다.
3. 작업 그룹의 IP 주소 유형을 선택합니다. 선택 사항은 다음과 같습니다.
 - IPv4 - 이 옵션을 사용하면 AWS 리소스는 IPv4 주소 지정 프로토콜을 통해서만 통신합니다.
 - 듀얼 스택 모드 - 이 옵션을 사용하면 AWS 리소스가 IPv4, IPv6 또는 두 가지 주소 지정 프로토콜을 통해 통신할 수 있습니다. 또한 Amazon VPC에서 작업 그룹에 사용되는 VPC 및 서브넷에 IPv6 CIDR 블록을 연결해야 합니다. Amazon VPC 콘솔을 사용하여 Amazon VPC를 만들거나 기존 Amazon VPC를 업데이트하여 IPv6 주소 지정을 사용할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서에서 [VPC에 대한 IPv6 지원](#)을 참조하세요.

4. Amazon Redshift Serverless용 Virtual Private Cloud(VPC)를 선택합니다. 이렇게 하면 작업 그룹이 AWS 환경의 특정 가상 네트워크에 할당됩니다. 사용 시 듀얼 스택 모드를 사용하는 경우 선택한 Amazon VPC는 IPV6 주소 지정을 지원해야 합니다. Amazon VPC에 대한 자세한 내용은 [VPC 및 서브넷 개요](#)를 참조하세요.
5. 하나 이상의 VPC 보안 그룹(VPC security groups)을 선택합니다. 자세한 내용은 [보안 그룹을 사용하여 리소스에 대한 트래픽 제어](#)를 참조하세요.
6. 자동 최적화를 위해 추가 컴퓨팅 리소스를 활성화할지 여부를 선택합니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [자동 데이터베이스 최적화를 위한 추가 컴퓨팅 리소스 할당](#)을 참조하세요.
7. 서브넷(Subnet)에서 데이터베이스와 연결할 서브넷을 하나 이상 지정합니다. 이러한 서브넷은 이전에 선택한 Amazon VPC에 포함되어 있으며 세 개의 개별 가용 영역에 있어야 합니다. 자세한 내용은 [Amazon Redshift Serverless 사용 시 고려 사항](#)을 참조하세요.
8. 요구 사항에 맞는 기본 RPU 용량을 선택합니다.

네임스페이스 선택

1. 새 네임스페이스 생성(Create a new namespace)을 선택하고 네임스페이스 이름을 입력하거나 기존 네임스페이스에 추가(Add to an existing namespace)를 선택하고 드롭다운 목록에서 네임스페이스를 선택합니다.
2. 데이터베이스 이름 및 암호(Database name and password)에 첫 번째 데이터베이스의 이름을 지정합니다. 관리자 보안 인증 정보(Admin user credentials)를 편집하여 기본 콘솔 관리자가 아닌 관리자를 지정할 수도 있습니다.
3. 권한(Permissions)에서 IAM 역할 연결(Associate IAM role)을 선택하여 특정 IAM 역할을 네임스페이스 및 작업 그룹과 연결합니다. IAM 역할을 Amazon Redshift와 연결하는 방법에 대한 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.
4. 새 키를 생성하거나 기본값이 아닌 다른 키를 선택하여 암호화 설정을 사용자 지정할 수 있습니다. 감사 로깅(Audit logging)의 경우 내보낼 로그를 선택합니다. 각 로그 유형은 서로 다른 메타데이터를 지정합니다. 계속(Continue)을 선택하여 선택 사항을 검토합니다.

작업 그룹 선택 검토

1. 검토 및 생성(Review and create)에서 설정을 검토합니다. 이전 단계에서 선택한 설정이 표시됩니다.
2. 저장을 선택합니다.

생성한 작업 그룹은 작업 그룹(Workgroups) 목록에 추가됩니다.

작업 그룹 속성 보기

Amazon Redshift Serverless에서 작업 그룹은 사용할 수 있는 컴퓨팅 리소스 모음입니다.

Amazon Redshift Serverless를 선택하면 AWS 콘솔의 탐색 메뉴에서 작업 그룹 구성(Workgroup configuration)을 선택하여 목록을 볼 수 있습니다. 검색(Search) 상자를 사용하여 검색 조건에 맞는 작업 그룹을 찾을 수 있습니다. 각 작업 그룹 항목에는 다음과 같은 몇 가지 속성이 표시됩니다.

- 작업 그룹(Workgroup) - 작업 그룹의 이름입니다. 이 옵션을 선택하여 작업 그룹의 속성을 보고 편집할 수 있습니다.
- 상태(Status) - 작업 그룹을 사용할 수 있는지 여부를 표시합니다.
- 네임스페이스(Namespace) - 작업 그룹과 연결된 네임스페이스입니다. 각 작업 그룹은 하나의 네임스페이스와 연결됩니다.
- 생성 날짜 - 작업 그룹을 만든 날짜입니다(UTC).
- 태그 - 작업 그룹과 연결된 태그입니다.

또한 작업 그룹 구성에는 AWS Glue에서 관리하는 Amazon Redshift Serverless 작업 그룹인 관리형 작업 그룹에 대한 또 다른 목록이 있습니다. 관리형 작업 그룹에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Managed workgroups](#) 섹션을 참조하세요.

작업 그룹 속성

왼쪽 메뉴에서 작업 그룹 구성(Workgroup configuration)을 선택하여 작업 그룹을 나열할 수 있습니다. 그런 다음 목록에서 작업 그룹을 선택할 수 있습니다. 여러 패널에 작업 그룹의 속성이 표시됩니다. 작업을 수행할 수도 있습니다. 일반 정보(General information)에는 다음 정보가 표시됩니다.

- 작업 그룹(Workgroup) - 작업 그룹의 이름입니다.
- 네임스페이스(Namespace) - 작업 그룹과 연결된 네임스페이스입니다. 선택하여 해당 속성을 볼 수 있습니다. 작업 그룹은 단일 네임스페이스와 연결됩니다.
- 생성된 날짜(Date created) - 작업 그룹이 생성된 날짜입니다.
- 상태(Status) - 작업 그룹 리소스를 사용할 수 있는지 여부를 나타냅니다. 사용 가능한 경우 클라이언트와 Amazon Redshift Serverless 인스턴스에 연결하여 데이터를 쿼리하거나 데이터베이스 리소스를 생성하거나 쿼리 편집기 v2를 사용하여 연결할 수 있습니다.
- 엔드포인트(Endpoint) - URL입니다.

- JDBC URL - JDBC 클라이언트 연결을 설정하기 위한 URL입니다. Amazon Redshift용 JDBC 드라이버에 연결할 때 이 URL을 사용할 수 있습니다. 자세한 내용은 [Amazon Redshift용 JDBC 드라이버 버전 2.x 연결 구성](#)을 참조하세요.
- ODBC URL - ODBC 클라이언트 연결을 설정하기 위한 URL입니다. 데이터베이스 및 사용자 ID와 같은 속성과 해당 값이 포함되어 있습니다.
- 작업 그룹 버전 및 패치 버전 - Amazon Redshift Serverless는 정기적으로 새 버전과 패치를 출시합니다. 작업 그룹 버전과 패치 버전 번호를 사용하여 Amazon Redshift Serverless 작업 그룹에 대한 소프트웨어 업데이트를 추적할 수 있습니다. 특정 패치에 포함된 변경 사항 및 기능에 대한 자세한 내용은 [Amazon Redshift에 대한 클러스터 버전](#)을 참조하세요.
- 작업 그룹 ARN - 작업 그룹의 Amazon 리소스 이름입니다.
- 자동 최적화를 위한 추가 컴퓨팅 리소스 - Amazon Redshift가 사용량이 많은 기간에도 추가 컴퓨팅 리소스를 할당하여 자동 최적화를 수행하는지 여부입니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [자동 데이터베이스 최적화를 위한 추가 컴퓨팅 리소스 할당](#)을 참조하세요.

데이터 액세스(Data access) 탭에는 여러 패널이 있습니다.

- 네트워크 및 보안 - Virtual Private Cloud(VPC) 식별자, VPC 보안 그룹 목록, 향상된 VPC 라우팅, IP 주소 유형, 퍼블릭 액세스가 가능한 설정 등의 네트워크 속성을 볼 수 있습니다. 편집(Edit)을 선택하면 이러한 설정을 변경할 수 있습니다. 또한 향상된 개인 정보 보호 및 보안을 위해 VPC를 통해 서버리스 데이터베이스와 데이터 리포지토리 간의 네트워크 트래픽을 라우팅하는 향상된 VPC 라우팅 켜기(Turn on enhanced VPC routing)를 선택할 수 있습니다. 또한 퍼블릭 액세스 가능 켜기(Turn on Public Accessible)를 선택하면 VPC 외부에서 데이터베이스에 퍼블릭 액세스가 가능하므로 인스턴스와 디바이스가 연결할 수 있습니다.

IP 주소 유형을 듀얼 스택 모드로 설정하여 IPv4와 IPv6 모두에서 동시에 작업 그룹에 액세스할 수 있도록 지원할 수 있습니다. 네트워크 계층 통신 인터넷 프로토콜(IP)에 대한 자세한 내용은 Wikipedia에서 [인터넷 프로토콜](#)을 참조하세요.

- Redshift 관리형 VPC 엔드포인트(Redshift managed VPC endpoints) - 관리형 VPC 엔드포인트를 생성하여 다른 VPC에서 Amazon Redshift Serverless에 액세스할 수 있습니다.

한도(Limits) 탭에는 Amazon Redshift Serverless의 용량 및 사용량 한도를 제어하는 설정이 있습니다. 여기에는 다음과 같은 패널이 있습니다.

- Redshift 처리 장치(RPU)의 기본 용량(Base capacity in Redshift processing units (RPU)) - 워크로드를 처리하는 데 사용되는 컴퓨팅 리소스의 기본 용량을 설정할 수 있습니다. 자세한 내용은 [Amazon Redshift Serverless의 컴퓨팅 용량](#) 섹션을 참조하세요.

- **사용 한도** - Amazon Redshift Serverless 인스턴스가 일정 기간 동안 사용할 수 있는 최대 컴퓨팅 리소스에 대해 최대 4개의 한도를 설정하고, 해당 한도에 도달했을 때 Amazon Redshift Serverless에서 수행할 작업을 선택할 수 있습니다. 예를 들어, 작업 그룹에 500RPU 시간과 900RPU 시간이라는 두 가지 한도를 두도록 설정할 수 있습니다. Amazon Redshift Serverless가 첫 번째 한도인 500RPU에 도달하면 알림을 보내고 두 번째 한도인 900시간에 도달하면 사용자 쿼리를 끄도록 할 수 있습니다. 이러한 한도는 비용을 제어하고 보다 비용 예측을 용이하게 하는 데 도움이 됩니다.
- **쿼리 한도(Query limits)** - 시간 초과 설정과 같은 쿼리에 대한 한도를 설정할 수 있습니다. 이러한 한도는 비용과 성능을 최적화하는 데 도움이 됩니다.

탭 탭에는 작업 그룹에 대해 생성한 모든 태그를 보여주는 태그 패널이 있습니다. 리소스 태깅에 대한 자세한 내용은 섹션을 참조하세요 [Amazon Redshift Serverless에서 리소스에 태그 지정](#)

관리형 작업 그룹 속성

관리형 작업 그룹 목록에서 AWS Glue Data Catalog에서 관리하는 작업 그룹을 선택할 수도 있습니다.

관리형 작업 그룹은 일반 작업 그룹과 속성이 다릅니다. 관리형 작업 그룹에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Managed workgroups](#) 섹션을 참조하세요.

일반 정보(General information)에는 다음 정보가 표시됩니다.

- **작업 그룹** - 작업 그룹의 이름입니다.
- **생성된 날짜** - 관리형 작업 그룹이 만들어진 날짜(UTC)입니다.
- **카탈로그 ARN** - AWS Glue Data Catalog에 있는 관리형 작업 그룹의 Amazon 리소스 이름(ARN)입니다.
- **상태** - 관리형 작업 그룹의 컴퓨팅 리소스를 사용할 수 있는지를 나타냅니다. 리소스를 사용할 수 있는 경우, 데이터를 쿼리하거나 데이터베이스 리소스를 만들기 위해 Apache Iceberg 호환 SQL 클라이언트와 함께 관리형 작업 그룹을 사용하는 카탈로그에 연결할 수 있습니다. Amazon Redshift Query Editor V2를 사용하여 카탈로그에 연결할 수도 있습니다.

쿼리 및 데이터베이스 모니터링에는 시간 경과에 따른 작업 그룹의 모든 쿼리의 평균 경과 시간을 보여주는 관리형 작업 그룹 성능 그래프가 포함되어 있습니다.

쿼리 기록 탭은 관리형 작업 그룹의 모든 쿼리 목록입니다. 세부 정보에는 쿼리를 실행한 사용자, 쿼리가 시작된 클라이언트 엔진, 쿼리의 ID 및 상태와 같은 정보가 있습니다. 사용자 탭은 작업 그룹의 모든 사용자 목록입니다. 성능 지표 탭에는 평균 쿼리 시간, 완료된 쿼리 수, 사용된 스토리지 용량 백분율과 같은 다양한 지표가 표시됩니다.

작업 그룹 삭제

콘솔을 사용하여 작업 그룹을 삭제할 수 있습니다. 이 작업을 수행하기 전에 데이터가 백업되고 스냅샷이 있어야 합니다. 대부분의 경우 작업 그룹의 일부로 삭제된 리소스는 검색할 수 없습니다.

다음 단계를 완료합니다.

1. Amazon Redshift Serverless를 선택하고 작업 그룹 구성(Workgroup configuration)을 선택한 다음 Amazon Redshift Serverless 인스턴스 삭제>Delete Amazon Redshift Serverless instance)를 선택합니다.
2. 대화 상자가 열립니다. 작업 그룹을 삭제하도록 선택하면 모든 사용량 한도가 제거되고 모든 VPC 엔드포인트가 제거되며 VPC 엔드포인트에 대한 액세스가 제거됩니다.

delete를 입력하고 삭제>Delete)를 선택하여 확인합니다.

단계를 모두 완료하면 작업 그룹의 상태가 삭제 중(Deleting)이 되며 작업 그룹이 삭제 중이라는 배너가 표시됩니다. 삭제 프로세스가 진행되는 동안 서버리스 대시보드(Serverless dashboard) 아래 일부 기능은 비활성화됩니다. 그러나 프로비저닝된 클러스터 대시보드(Provisioned clusters dashboard)에서 프로비저닝된 클러스터를 구성할 수 있습니다.

작업 그룹이 삭제된 후에는 네임스페이스와 함께 표시되지 않습니다. 작업 그룹 생성(Create workgroup) 버튼을 선택하여 새 그룹을 생성할 수 있습니다.

기존 작업 그룹을 삭제하고 구성이 다른 새 작업 그룹을 동일한 네임스페이스에 연결할 수 있습니다. 새 작업 그룹을 생성할 때는 네임스페이스에 연결된 데이터의 크기에 맞는 기본 용량을 선택합니다.

작업 그룹을 고객 관리형 키(CMK)로 생성된 네임스페이스와 연결할 수 있습니다. AWS KMS에 대한 자세한 내용은 [AWS KMS 개념](#)을 참조하세요.

네임스페이스

Amazon Redshift Serverless에서 네임스페이스는 데이터베이스 객체에 대한 논리적 컨테이너를 정의합니다. 네임스페이스에는 테이블, 작업 그룹 및 기타 데이터베이스 리소스가 보관될 수 있습니다. 작업 그룹과 네임스페이스를 생성하지 않았고 Amazon Redshift Serverless를 시작하는 방법에 대한 지침을 찾고 있는 경우 [Amazon Redshift Serverless 최초 설정](#)을 참조하세요.

네임스페이스 속성

Amazon Redshift Serverless에서 네임스페이스는 데이터베이스 객체에 대한 컨테이너를 정의합니다. 탐색 목록에서 네임스페이스 구성(Namespace configuration)을 선택하고 목록에서 네임스페이스를 선택한 다음 설정을 편집할 수 있습니다.

네임스페이스에 대한 일반 정보에는 다음이 포함됩니다.

- 네임스페이스(Namespace) - 이름입니다.
- 네임스페이스 ID(Namespace ID) - 고유 식별자입니다.
- ARN - AWS 전반의 리소스를 지정하는 데 사용되는 고유 식별자입니다. 리전 및 서비스와 같은 속성이 포함되어 있습니다.
- 상태(Status) - 사용 가능(Available)과 같은 상태입니다.
- 생성된 날짜 - 네임스페이스가 만들어진 날짜입니다(UTC).
- 사용된 스토리지(Storage used) - 네임스페이스 및 모든 객체가 사용하는 스토리지 공간입니다.
- 관리자 사용자 이름(Admin user name) - 관리자 계정입니다. 일반적으로 네임스페이스를 생성하는 데 사용되는 계정입니다.
- 데이터베이스 이름(Database name) - 네임스페이스에 포함된 데이터베이스의 이름입니다.
- 총 테이블 수(Total table count) - 모든 스키마의 테이블 수입니다.

네임스페이스에 대한 추가 설정 및 속성은 여러 탭에 있습니다. 여기에는 다음이 포함됩니다.

- 작업 그룹(Workgroup) - 네임스페이스와 연결된 작업 그룹을 표시합니다.
- 데이터 백업(Data back up) - 이 패널에서 스냅샷을 구성 및 생성하고 복구 시점을 구성할 수 있습니다.
- 보안 및 암호화(Security and encryption) - IAM 역할 권한을 관리하고 보안 및 암호화 설정을 보거나 편집할 수 있습니다. 여기에는 암호화 키 상태 및 감사 로깅을 켜는 설정이 포함됩니다. Amazon Redshift Serverless의 감사 로깅에 대한 자세한 내용은 [Amazon Redshift Serverless에 대한 감사 로깅](#)을 참조하세요.
- 데이터 공유(Datashares) - 데이터 공유를 표시합니다. 데이터 공유를 사용하면 데이터를 복사하거나 이동할 필요 없이 데이터에 대한 액세스를 제공할 수 있습니다. 데이터 공유에 대한 자세한 내용은 [Amazon Redshift Serverless에서 데이터 공유](#)를 참조하세요.

네임스페이스 검색

Amazon Redshift 메뉴의 네임스페이스(Namespace) 목록에서 선택하여 네임스페이스의 속성을 보거나 편집할 수 있습니다. 콘솔의 정보에는 네임스페이스 이름, 관리자 이름 및 기타 속성이 포함됩니다.

네임스페이스의 설정 및 속성은 여러 탭에 있습니다. 여기에는 다음이 포함됩니다.

- 작업 그룹(Workgroup) - 네임스페이스와 연결된 작업 그룹을 표시합니다.
- 데이터 백업(Data back up) - 스냅샷을 구성 및 생성하고 복구 시점을 구성할 수 있습니다.
- 보안 및 암호화(Security and encryption) - IAM 역할 권한을 관리하고 보안 및 암호화 설정을 보거나 편집할 수 있습니다. 여기에는 암호화 키 상태 및 감사 로깅 설정이 포함됩니다.
- 데이터 공유(Datashares) - 데이터 공유를 표시합니다.

보안 및 암호화 편집

Amazon Redshift Serverless는 KMS 암호화를 통해 보안을 유지합니다. 콘솔을 통해 암호화 설정을 업데이트할 수 있습니다.

1. 콘솔의 기본 메뉴에서 Namespace configuration(네임스페이스 구성)을 선택하고 편집할 네임스페이스를 선택한 다음 Security and encryption(보안 및 암호화) 탭에 있는 Edit(편집)을 선택합니다. 대화 상자가 나타납니다.
2. Customize encryption settings(암호화 설정 사용자 지정)을 선택한 다음 AWS Choose an customer managed key(고객 관리형 키 선택)를 선택하여 리소스를 암호화하는 데 사용되는 키를 변경할 수 있습니다.
3. 감사 로깅(Audit logging)의 경우 내보낼 로그를 선택합니다. 각 로그 유형은 서로 다른 메타데이터를 지정합니다.
4. 구성 업데이트를 완료하려면 변경 사항 저장(Save changes)을 선택합니다.

네임스페이스의 AWS KMS 키 변경

Amazon Redshift에서는 암호화를 통해 저장 데이터가 보호됩니다. Amazon Redshift Serverless는 AWS KMS 키 암호화를 자동으로 사용하여 Amazon Redshift Serverless 리소스와 스냅샷을 모두 암호화합니다. 모범 사례로서 대부분의 조직은 저장하는 데이터 유형을 검토하고 일정애 따라 암호화 키를 교체하는 계획을 가지고 있습니다. 키 교체 빈도는 데이터 보안 정책에 따라 다를 수 있습니다. Amazon Redshift Serverless는 조직의 보안 정책을 준수할 수 있도록 네임스페이스의 AWS KMS 키 변경을 지원합니다.

AWS KMS 키를 변경해도 데이터는 변경되지 않은 상태로 유지됩니다.

콘솔을 사용하여 AWS KMS 키 변경

Amazon Redshift에서는 암호화를 통해 저장 데이터가 보호됩니다. Amazon Redshift Serverless는 AWS KMS 키 암호화를 자동으로 사용하여 Amazon Redshift Serverless와 스냅샷을 모두 암호화합니다. 모범 사례로서 대부분의 조직은 저장하는 데이터 유형을 검토하고 일정에 따라 암호화 키를 교체하는 계획을 가지고 있습니다. 키 교체 빈도는 데이터 보안 정책에 따라 다를 수 있습니다. Amazon Redshift Serverless는 조직의 보안 정책을 준수할 수 있도록 네임스페이스의 AWS KMS 키 변경을 지원합니다.

AWS KMS 키를 변경해도 데이터는 변경되지 않은 상태로 유지됩니다.

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 창에서 네임스페이스 구성(Namespace configurations)을 선택합니다. 목록에서 데이터를 선택합니다.
3. 보안 및 암호화(Security and encryption) 탭에서 편집(Edit)을 선택합니다.
4. 암호화 설정 사용자 지정(Customize encryption settings)을 선택한 후 네임스페이스의 키를 선택합니다. 선택적으로 새 키를 생성할 수 있습니다.

AWS CLI를 사용하여 AWS KMS 암호화 키 변경

네임스페이스의 AWS KMS 키를 변경하려면 update-namespace를 사용합니다. 다음은 명령의 구문을 보여 줍니다.

```
aws redshift-serverless update-namespace
--namespace-name
[--kms-key-id <id-of-kms-key>]
// other parameters omitted here
```

네임스페이스가 생성되어 있어야 합니다. 그렇지 않으면 이 CLI 명령에서 오류가 발생합니다.

키 변경에 걸리는 시간은 Amazon Redshift Serverless의 데이터 양에 따라 달라집니다. 일반적으로 저장된 데이터 8TB당 15분이 소요됩니다.

제한 사항

고객 관리형 KMS 키에서 AWS KMS 키로 변경할 수 있습니다. 이 경우 새 네임스페이스를 생성해야 합니다.

키를 변경하는 동안에는 다른 작업을 수행할 수 없습니다.

네임스페이스 삭제

연결된 작업 그룹이 있는 네임스페이스를 삭제하려면 먼저 작업 그룹을 삭제해야 합니다.

Amazon Redshift Serverless 콘솔에서 다음 단계를 완료합니다.

1. 왼쪽 메뉴에서 네임스페이스 구성(Namespace configuration)을 선택한 후 목록에서 삭제할 네임스페이스를 선택합니다.
2. 작업(Actions)을 선택한 후 네임스페이스 삭제>Delete namespace)를 선택합니다.
3. 대화 상자가 열립니다. 삭제 작업을 완료하기 전에 수동 스냅샷을 생성하여 데이터를 유지할 수 있습니다.

delete를 입력하고 삭제>Delete)를 선택하여 확인합니다.

Amazon Redshift Serverless로 쿼리 및 워크로드 모니터링

제공된 시스템 보기를 사용하여 Amazon Redshift Serverless 쿼리 및 워크로드를 모니터링할 수 있습니다.

모니터링 보기는 쿼리 및 워크로드 사용량을 모니터링하는 데 사용되는 Amazon Redshift Serverless의 시스템 보기입니다. 이러한 보기는 pg_catalog 스키마에 있습니다. 사용 가능한 시스템 보기는 프로비저닝된 클러스터에 필요한 것보다 훨씬 간단한 Amazon Redshift Serverless를 모니터링하는 데 필요한 정보를 제공하도록 설계되었습니다. SYS 시스템 보기는 Amazon Redshift Serverless에서 작동하도록 설계되었습니다. 이러한 보기에서 제공하는 정보를 표시하려면 SQL SELECT 문을 실행합니다.

시스템 보기는 다음 모니터링 목표를 지원하도록 정의됩니다.

워크로드 모니터링

시간 경과에 따른 쿼리 활동을 모니터링하여 다음을 수행할 수 있습니다.

- 워크로드 패턴을 이해하여 정상(기준)과 비즈니스 서비스 수준 계약(SLA) 내에 있는 것을 알 수 있습니다.
- 일시적인 문제이거나 추가 조치가 필요한 문제일 수 있는 정상과의 편차를 신속하게 식별합니다.

데이터 로드 및 언로드 모니터링

Amazon Redshift Serverless 안팎으로 데이터를 이동하는 것은 중요한 기능입니다. COPY 및 UNLOAD를 사용하여 데이터를 로드 또는 언로드하고 비즈니스 SLA 준수를 추적하기 위해 전송된 바이트/행 수 및 완료된 파일 수를 기준으로 진행 상황을 면밀히 모니터링해야 합니다. 이는 일반적으로 시스템 테이블 쿼리를 자주(즉, 매분) 실행하여 진행 상황을 추적하고 심각한 편차가 감지되는 경우 조사/수정 조치에 대한 경고를 발생시켜 수행됩니다.

실패 및 문제 진단

쿼리 또는 런타임 실패에 대해 조치를 취해야 하는 경우가 있습니다. 개발자는 시스템 테이블에 의존하여 문제를 자가 진단하고 올바른 해결책을 결정합니다.

성능 튜닝

처음부터 SLA 요구 사항을 충족하지 않거나 시간 경과에 따라 성능이 저하된 쿼리를 조정해야 할 수 있습니다. 튜닝하려면 실행 계획, 통계, 기간 및 리소스 소비를 비롯한 런타임 세부 정보가 있어야 합니다. 편차의 원인을 파악하고 성능 개선 방법을 안내하기 위해 잘못된 쿼리에 대한 기준 데이터가 필요합니다.

사용자 객체 이벤트 모니터링

구체화된 보기 새로 고침, 정리 및 분석과 같은 사용자 객체에 대한 작업 및 활동을 모니터링해야 합니다. 여기에는 구체화된 뷰에 대한 자동 새로 고침과 같은 시스템 관리형 이벤트가 포함됩니다. 사용자가 시작한 경우 이벤트가 종료되는 시점을 모니터링하고 시스템이 시작한 경우 마지막으로 성공한 실행을 모니터링하려고 합니다.

청구를 위한 사용량 추적

시간 경과에 따른 사용량 추세를 모니터링하여 다음을 수행할 수 있습니다.

- 예산 계획과 비즈니스 확장 추정치를 알립니다.
- 콜드 데이터 제거와 같은 잠재적인 비용 절감 기회를 식별합니다.

SYS 시스템 뷰를 사용하여 Amazon Redshift Serverless를 모니터링합니다. SYS 모니터링 뷰에 관한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [SYS 모니터링 뷰](#)를 참조하시기 바랍니다.

쿼리 모니터링 정책 추가

슈퍼 사용자는 모든 사용자에게 쿼리 모니터링을 수행할 수 있도록 슈퍼 사용자가 아닌 사용자에게 액세스 권한을 제공할 수 있습니다. 먼저 쿼리 모니터링 액세스를 제공하기 위해 사용자 또는 역할에 대한 정책을 추가합니다. 그런 다음 사용자 또는 역할에 쿼리 모니터링 권한을 부여합니다.

쿼리 모니터링 정책을 추가하려면

1. <https://console.aws.amazon.com/iam/>을 선택합니다.
2. 액세스 관리(Access management)에서 정책(Policies)을 선택합니다.
3. 정책 생성(Create Policy)을 선택합니다.
4. JSON을 선택하고 다음 정책 정의를 붙여 넣습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift-data:ExecuteStatement",
        "redshift-data:DescribeStatement",
        "redshift-data:GetStatementResult",
        "redshift-data:ListDatabases"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "redshift-serverless:GetCredentials",
      "Resource": "*"
    }
  ]
}
```

5. 정책 검토를 선택합니다.
6. 이름에 정책의 이름을 입력합니다(예: query-monitoring).
7. [정책 생성]을 선택합니다.

정책을 생성한 후 적절한 권한을 부여할 수 있습니다.

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요.

- AWS IAM Identity Center의 사용자 및 그룹:

권한 세트를 생성합니다. AWS IAM Identity Center 사용자 안내서에서 [권한 세트 생성](#)의 지침을 따릅니다.

- ID 제공업체를 통해 IAM에서 관리되는 사용자:

ID 페더레이션을 위한 역할을 생성합니다. IAM 사용자 설명서의 [Create a role for a third-party identity provider \(federation\)](#)의 지침을 따릅니다.

- IAM 사용자:

- 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용자 설명서에서 [Create a role for an IAM user](#)의 지침을 따릅니다.
- (권장되지 않음) 정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용자 설명서에서 [사용자\(콘솔\)에 권한 추가](#)의 지침을 따릅니다.

사용자에 대한 쿼리 모니터링 권한 부여

sys:monitor 권한이 있는 사용자는 모든 쿼리를 볼 수 있습니다. 또한 sys:operator 권한이 있는 사용자는 쿼리를 취소하고 쿼리 기록을 분석하고 vacuum 작업을 수행할 수 있습니다.

사용자에게 쿼리 모니터링 권한을 부여하려면

1. 시스템 모니터 액세스를 제공하려면 다음 명령을 입력합니다. 여기서 user-name은 액세스를 제공하려는 사용자의 이름입니다.

```
grant role sys:monitor to "IAM:user-name";
```

2. (선택 사항) 시스템 운영 액세스를 제공하려면 다음 명령을 입력합니다. 여기서 user-name은 액세스를 제공하려는 사용자의 이름입니다.

```
grant role sys:operator to "IAM:user-name";
```

역할에 대한 쿼리 모니터링 권한 부여

sys:monitor 권한이 있는 역할의 사용자는 모든 쿼리를 볼 수 있습니다. 또한 sys:operator 권한이 있는 역할의 사용자는 쿼리를 취소하고 쿼리 기록을 분석하고 vacuum 작업을 수행할 수 있습니다.

역할에 쿼리 모니터링 권한을 부여하려면

1. 시스템 모니터 액세스를 제공하려면 다음 명령을 입력합니다. 여기서 `role-name`은 액세스를 제공하려는 사용자의 이름입니다.

```
grant role sys:monitor to "IAMR:role-name";
```

2. (선택 사항) 시스템 운영 액세스를 제공하려면 다음 명령을 입력합니다. 여기서 `role-name`은 액세스를 제공하려는 사용자의 이름입니다.

```
grant role sys:operator to "IAMR:role-name";
```

RPU 제한 설정을 포함한 사용 제한 설정

작업 그룹의 한도(Limits) 탭에서 하나 이상의 사용량 한도를 추가하여 지정된 기간 동안 사용하는 최대 RPU를 제어하거나 데이터 공유 사용량 한도를 설정할 수 있습니다.

1. 사용량 한도 관리(Manage usage limits)를 선택합니다. 한도 섹션은 기간별 컴퓨팅 사용량 패널 하단에 표시됩니다.
2. 사용량 한도를 RPU 시간 단위로 설정합니다.
3. 빈도를 일별, 주별 또는 월별 중에서 선택합니다. 이로써 사용량 한도의 기간이 설정됩니다. 이 인스턴스에서 일별(Daily)을 선택하면 보다 세부적으로 제어할 수 있습니다.
4. 사용량 한도를 시간 단위로 설정합니다.
5. 작업을 설정합니다. 다음과 같은 설정이 있습니다.
 - 시스템 테이블에 기록 - 시스템 뷰 [SYS_QUERY_HISTORY](#)에 레코드를 추가합니다. 이 뷰의 `usage_limit` 열을 쿼리하여 쿼리가 제한을 초과했는지 확인할 수 있습니다.
 - 알림(Alert) - Amazon SNS를 사용하여 알림 구독을 설정하고 한도가 위반된 경우 알림을 전송합니다. 기존 Amazon SNS 주제를 선택하거나 새로운 주제를 생성할 수 있습니다.
 - 사용자 쿼리 끄기(Turn off user queries) - 쿼리를 비활성화하여 Amazon Redshift Serverless 사용을 중지합니다. 또한 알림도 전송합니다.

처음 두 작업은 정보 제공용이지만 마지막 작업은 쿼리 처리를 해제합니다.

6. 선택적으로, 소비자가 쿼리할 수 있는, 생산자 리전에서 소비자 리전으로 전송되는 데이터의 양을 제한하는 교차 리전 데이터 공유 사용량 한도(Cross-Region data sharing usage limit)를 설정할 수 있습니다. 이렇게 하려면 한도 추가(Add limit)를 선택하고 단계를 따릅니다.

7. 페이지 하단에서 변경 사항 저장을 선택하여 한도를 저장합니다.
8. 필요에 따라 한도를 3개까지 더 설정합니다.

RPU 및 청구에 대한 개념 정보에 대한 자세한 내용은 [Amazon Redshift Serverless에 대한 청구](#)를 참조하세요.

쿼리 제한 설정

작업 그룹의 Limits(한도) 탭에서 한도를 추가하여 성능 및 한도를 모니터링할 수 있습니다. 쿼리 모니터링 제한 사항에 대한 자세한 내용은 [WLM 쿼리 모니터링 규칙](#)을 참조하세요.

1. 쿼리 한도 관리(Manage query limits)를 선택합니다. Manage query limits(쿼리 한도 관리) 대화 상자에서 Add limit(한도 추가)를 선택합니다.
2. 설정하려는 한도 유형을 선택하고 해당 한도 값을 입력합니다.
3. 변경 사항 저장(Save changes)을 선택하여 한도를 저장합니다.

쿼리 한도 및 구성 파라미터를 변경하면 데이터베이스가 다시 시작됩니다.

쿼리 대기열 설정

Amazon Redshift Serverless는 대기열 기반 쿼리 리소스 관리를 지원합니다. 다양한 워크로드에 대한 사용자 지정 모니터링 규칙을 사용하여 전용 쿼리 대기열을 생성할 수 있습니다. 이 기능으로 리소스 사용량에 대한 세부적인 제어가 가능합니다.

쿼리 모니터링 규칙(QMR)은 Redshift Serverless 작업 그룹 수준에서만 적용되며 이 작업 그룹에서 실행되는 모든 쿼리에 균일하게 영향을 미칩니다. 대기열 기반 접근 방식을 통해 고유한 모니터링 규칙을 사용하여 대기열을 생성할 수 있습니다. 이러한 대기열을 특정 사용자 역할 및 쿼리 그룹에 할당할 수 있습니다. 각 대기열은 독립적으로 작동하며, 규칙은 해당 대기열 내의 쿼리에만 영향을 미칩니다.

대기열을 사용하면 지표 기반 조건자 및 자동화된 응답을 설정할 수 있습니다. 예를 들어 시간 제한을 초과하거나 너무 많은 리소스를 소비하는 쿼리를 자동으로 중단하도록 규칙을 구성할 수 있습니다.

고려 사항

서버리스 쿼리를 사용할 때는 다음을 고려하세요.

- Amazon Redshift 프로비저닝 클러스터에 사용되는 다음 워크로드 관리(WLM) 구성 키는 Redshift Serverless 대기열에서 지원되지 않습니다. max_execution_time, short_query_queue,

auto_wlm, concurrency_scaling, priority, queue_type, query_concurrency, memory_percent_to_use, user_group, user_group_wild_card.

또한 change_query_priority 작업은 Serverless에서 지원되지 않습니다.

- 흡 작업(대기열 간 쿼리 이동)은 Amazon Redshift Serverless에서 지원되지 않습니다.
- 대기열 우선순위는 Amazon Redshift 프로비저닝 클러스터에서만 지원됩니다.
- Amazon Redshift Serverless는 최적의 성능을 위해 조정 및 리소스 할당을 자동으로 관리하므로 대기열 우선순위를 수동으로 구성할 필요가 없습니다.

쿼리 대기열 설정

AWS Management Console, AWS CLI 또는 Redshift Serverless API를 사용하여 서버리스 작업 그룹의 제한 탭에서 대기열을 생성할 수 있습니다.

Console

다음 단계에 따라 서버리스 작업 그룹에 대한 대기열을 생성합니다.

1. Redshift Serverless 작업 그룹으로 이동합니다.
2. 제한 탭을 선택합니다.
3. 쿼리 대기열에서 대기열 활성화를 선택합니다.

Important

쿼리 대기열을 활성화하는 것은 영구적인 변경입니다. 활성화되면 대기열 없는 모니터링으로 되돌릴 수 없습니다.

4. 다음 파라미터를 사용하여 대기열을 구성합니다.

대기열 수준 파라미터

- name - 대기열 식별자(필수, 고유, 비어 있지 않음)
- user_role - 사용자 역할 배열(선택 사항)
- query_group - 쿼리 그룹 배열(선택 사항)
- query_group_wild_card - 와일드카드 일치를 활성화하려면 0 또는 1(선택 사항)
- user_group_wild_card - 와일드카드 일치를 활성화하려면 0 또는 1(선택 사항)
- rules - 모니터링 규칙 배열(선택 사항)

규칙 수준 파라미터

- `rule_name` - 고유 식별자, 최대 32자(필수)
- `predicate` - 조건 배열, 1~3개의 조건자(필수)
- `action` - "중단" 또는 "로그"(필수)

조건자 수준 파라미터

- `metric_name` - 모니터링할 지표(필수)
- `operator` - "=", "<" 또는 ">"(필수)
- `value` - 숫자 임계값(필수)

한도

- 최대 8개의 대기열
- 모든 대기열에서 최대 25개의 규칙
- 규칙당 최대 3개의 조건자
- 이름은 전역적으로 고유해야 합니다.

구성의 예제

대기열 3개 예: 제한 시간이 짧은 쿼리를 대시보드로 처리하기 위한 대기열 1개, 제한 시간이 긴 ETL 쿼리와 관리자 대기열을 위한 대기열 1개:

```
[
  {
    "name": "dashboard",
    "user_role": ["analyst", "viewer"],
    "query_group": ["reporting"],
    "query_group_wild_card": 1,
    "rules": [
      {
        "rule_name": "short_timeout",
        "predicate": [
          {
            "metric_name": "query_execution_time",
```

```

        "operator": ">",
        "value": 60
      }
    ],
    "action": "abort"
  }
]
},
{
  "name": "ETL",
  "user_role": ["data_scientist"],
  "query_group": ["analytics", "ml"],
  "rules": [
    {
      "rule_name": "long_timeout",
      "predicate": [
        {
          "metric_name": "query_execution_time",
          "operator": ">",
          "value": 3600
        }
      ],
      "action": "log"
    },
    {
      "rule_name": "memory_limit",
      "predicate": [
        {
          "metric_name": "query_temp_blocks_to_disk",
          "operator": ">",
          "value": 100000
        }
      ],
      "action": "abort"
    }
  ]
},
{
  "name": "admin_queue",
  "user_role": ["admin"],
  "query_group": ["admin"]
}
]
```

이 예시는 다음과 같이 설정되어 있습니다.

- 대시보드 쿼리가 60초 이상 실행되면 중단됩니다.
- ETL 쿼리는 1시간 이상 실행되는 경우 로깅됩니다.
- 관리자 대기열에는 리소스 제한이 없습니다.

CLI

`wlm_json_configuration` 구성 파라미터와 함께 `CreateWorkgroup` 또는 `UpdateWorkgroup` APIs 사용하여 대기열을 관리하여 JSON 형식으로 대기열을 지정할 수 있습니다.

```
aws redshift-serverless create-workgroup \
  --workgroup-name test-workgroup \
  --namespace-name test-namespace \
  --config-parameters '[{"parameterKey": "wlm_json_configuration", "parameterValue":
  "[{\\"name\\":\\"dashboard\\",\\"user_role\\":[\\"analyst\\",\\"viewer\\"],\\"query_group
\\":[\\"reporting\\"],\\"query_group_wild_card\\":1,\\"rules\\":[{\\"rule_name\\":
\\"short_timeout\\",\\"predicate\\":[{\\"metric_name\\":\\"query_execution_time\\",
\\"operator\\":\\">\\",\\"value\\":60}],\\"action\\":\\"abort\\"}]}],{\\"name\\":\\"ETL
\\",\\"user_role\\":[\\"data_scientist\\"],\\"query_group\\":[\\"analytics\\",\\"ml
\\"],\\"rules\\":[{\\"rule_name\\":\\"long_timeout\\",\\"predicate\\":[{\\"metric_name
\\":\\"query_execution_time\\",\\"operator\\":\\">\\",\\"value\\":3600}],\\"action\\":
\\"log\\"}],{\\"rule_name\\":\\"memory_limit\\",\\"predicate\\":[{\\"metric_name\\":
\\"query_temp_blocks_to_disk\\",\\"operator\\":\\">\\",\\"value\\":100000}],\\"action\\":
\\"abort\\"}]}],{\\"name\\":\\"admin_queue\\",\\"user_role\\":[\\"admin\\"],\\"query_group\\":
[\\"admin\\"]}"]']'
```

모범 사례

서버리스 대기열을 사용할 때는 다음 모범 사례를 염두에 두세요.

- 고유한 제한 요구 사항(예: ETL, 보고 또는 임시 분석)이 있는 워크로드에는 별도의 대기열을 사용합니다.
- 간단한 임계값으로 시작하고 쿼리 동작 및 사용 패턴에 따라 조정합니다. [쿼리 모니터링 규칙에 대해 시스템 테이블 및 뷰](#)에 문서화된 테이블 및 뷰를 사용하여 쿼리 사용 패턴을 모니터링할 수 있습니다.

대시보드에서 Amazon Redshift Serverless 요약 데이터 확인

Amazon Redshift Serverless 대시보드에는 작업 그룹 및 네임스페이스에 대한 지표와 정보를 한눈에 보여주는 패널 모음이 포함되어 있습니다. 다음과 같은 패널 모음이 있습니다.

- Resources summary(리소스 요약) - 사용된 스토리지 및 기타 지표와 같은 Amazon Redshift Serverless에 대한 상위 수준 정보를 표시합니다.
- 쿼리 요약(Query summary) - 완료된 쿼리 및 실행 중인 쿼리를 포함하여 쿼리에 대한 정보를 표시합니다. 세부 정보 보기(View details)를 선택하여 추가 필터가 있는 화면으로 이동합니다.
- 사용된 RPU 용량(RPU capacity used) - 이전 10시간과 같은 지정된 기간 동안 사용된 전체 용량을 표시합니다.
- 데이터 공유 - AWS 계정과 같이 데이터 공유에 사용되는 데이터 공유 수를 표시합니다. 이 지표는 권한 부여가 필요한 Datashare 및 기타 정보를 보여줍니다.
- 총 컴퓨팅 사용량 - 선택한 시간 범위(최대 지난 7일) 동안 선택한 작업 그룹에서 소비한 총 RPU 시간을 표시합니다.

대시보드에서 이러한 사용 가능한 지표를 빠르게 살펴보고 Amazon Redshift Serverless에 대한 세부 정보를 확인하거나 쿼리를 검토하거나 작업 항목을 추적할 수 있습니다.

Amazon Redshift Serverless에 대한 감사 로깅

연결, 사용자 및 사용자 활동 로그 데이터를 Amazon CloudWatch Logs의 로그 그룹으로 내보내도록 Amazon Redshift Serverless를 구성할 수 있습니다. Amazon CloudWatch Logs를 통해 로그 데이터에 대한 실시간 분석을 수행할 수 있고, CloudWatch를 사용하여 경보를 만들고 지표를 볼 수 있습니다. CloudWatch Logs를 사용하여 내구성이 우수한 스토리지에 로그 레코드를 저장할 수 있습니다.

Amazon Redshift 콘솔을 사용하여 지표를 추적하는 CloudWatch 경보를 생성할 수 있습니다. 경보 생성에 대한 자세한 내용은 [경보 관리](#)를 참조하세요.

생성된 로그 데이터를 Amazon CloudWatch Logs로 내보내려면 콘솔의 Amazon Redshift Serverless 구성 설정에서 내보낼 각 로그를 선택해야 합니다. 보안 및 암호화에서 네임스페이스 구성 설정을 선택하여 이 작업을 수행할 수 있습니다.

CloudWatch의 로그 이벤트

내보낼 Redshift 로그를 선택한 후 Amazon CloudWatch Logs에서 이벤트를 모니터링할 수 있습니다. Amazon Redshift Serverless에 대한 새 로그 그룹이 자동으로 생성됩니다. 여기서 `log_type`은 로그 유형을 나타냅니다.

```
/aws/redshift/<namespace>/<log_type>
```

첫 번째 작업 그룹과 네임스페이스를 만들 때 기본값은 네임스페이스 이름입니다. 로그 그룹 이름은 네임스페이스의 이름에 따라 달라집니다.

예를 들어 연결 로그를 내보내면 로그 데이터는 다음 로그 그룹에 저장됩니다.

```
/aws/redshift/default/connectionlog
```

로그 이벤트는 서버리스 로그 스트림을 사용하여 로그 그룹으로 내보내집니다. 다음 조건 중 어디에 해당하는지에 따라 달라집니다.

- 지정된 이름의 로그 그룹이 존재합니다. Redshift는 기존 로그 그룹을 사용하여 로그 데이터를 내보냅니다. AWS CloudFormation에서 제공하는 것과 같은 자동 구성을 사용하여 미리 정의된 로그 보존 기간, 지표 필터 및 고객 액세스 권한이 있는 로그 그룹을 생성할 수 있습니다.
- 지정된 이름의 로그 그룹이 존재하지 않습니다. 인스턴스의 로그에서 일치하는 로그 항목이 감지되면 Amazon Redshift Serverless는 Amazon CloudWatch Logs에 새 로그 그룹을 자동으로 생성합니다. 이 로그 그룹에는 기본 로그 보존 기간인 만료되지 않음(Never Expire)이 사용됩니다. 로그 보존 기간을 변경하려면 Amazon CloudWatch Logs 콘솔, AWS CLI 또는 Amazon CloudWatch Logs API를 사용합니다. CloudWatch Logs의 로그 보존 기간 변경에 대한 자세한 내용은 [로그 그룹 및 로그 스트림 작업](#)의 로그 데이터 보존 변경을 참조하세요.

로그 이벤트 내에서 정보를 검색하려면 CloudWatch Logs 콘솔, AWS CLI 또는 Amazon CloudWatch Logs API를 사용합니다. 로그 데이터 검색 및 필터링에 대한 자세한 내용은 [로그 데이터 검색 및 필터](#)를 참조하세요.

CloudWatch 지표

Amazon Redshift Serverless 지표는 컴퓨팅 지표와 데이터 및 스토리지 지표로 나뉘며, 각각 작업 그룹 및 네임스페이스 차원 세트에 속합니다. 작업 그룹 및 네임스페이스에 대한 자세한 내용은 [작업 그룹 및 네임스페이스](#)를 참조하세요.

CloudWatch 컴퓨팅 지표는 다음과 같습니다.

지표 이름	단위	설명	차원 세트
QueriesCompletedPerSecond	쿼리 수	초당 완료된 쿼리 수입니다.	{Database, LatencyRange, Workgroup}, {LatencyRange, Workgroup}
QueryDuration	마이크로초	쿼리를 완료하는데 걸리는 평균 시간입니다.	{Database, LatencyRange, Workgroup}, {LatencyRange, Workgroup}
QueriesRunning	쿼리 수	특정 시점에 실행 중인 쿼리의 수입니다.	{Database, QueryType, Workgroup}, {QueryType, Workgroup}
QueriesQueued	쿼리 수	특정 시점에 대기열에 있는 쿼리의 수입니다.	{Database, QueryType, Workgroup}, {QueryType, Workgroup}
DatabaseConnections	연결 수	특정 시점에 데이터베이스의 연결 수입니다.	{Database, Workgroup}, {Workgroup}
QueryRuntimeBreakdown	밀리초	쿼리 단계에서 실행한 총 쿼리 시간입니다.	{Database, Stage, Workgroup}, {Stage, Workgroup}
ComputeCapacity	RPU	지난 30분간 할당된 평균 컴퓨	{Workgroup}

지표 이름	단위	설명	차원 세트
		팅 단위 수로, 가장 가까운 정수로 반올림됩니다.	
ComputeSeconds	RPU-seconds	지난 30분간 사용된 누적 컴퓨팅 단위(초)입니다.	{Workgroup}
QueriesSucceeded	쿼리 수	지난 5분간 성공한 쿼리 수입니다.	{Database, QueryType, Workgroup}, {QueryType, Workgroup}
QueriesFailed	쿼리 수	지난 5분간 실패한 쿼리 수입니다.	{Database, QueryType, Workgroup}, {QueryType, Workgroup}

지표 이름	단위	설명	차원 세트
UsageLimitAvailable	RPU 시간 또는 TB	UsageType에 따라 UsageLimitAvailable은 다음을 반환합니다. - UsageType이 SERVERLESS_COMPUTE 인 경우 UsageLimitAvailable은 작업 그룹이 지정된 한도 내에서 쿼리할 수 있는 남은 RPU 시간을 반환합니다. - UsageType이 CROSS_REGION_DATA_LOADING인 경우 UsageLimitAvailable은 고객이 지정된 한도 내에서 스캔할 수 있는 남은 TB 수를 반환합니다.	{UsageLimitId, UsageType, Workgroup}

지표 이름	단위	설명	차원 세트
UsageLimitConsumed	RPU 시간 또는 TB	UsageType에 따라 UsageLimitConsumed는 다음을 반환합니다. - UsageType이 SERVERLESS_COMPUTE 인 경우 UsageLimitConsumed는 작업 그룹이 지정된 한도 내에서 이미 쿼리한 RPU 시간 수를 반환합니다. - UsageType이 CROSS_REGION_DATA_HARING인 경우 UsageLimitConsumed는 고객이 지정된 한도 내에서 이미 스캔에 사용한 TB 수를 반환합니다.	{UsageLimitId, UsageType, Workgroup}

지표 이름	단위	설명	차원 세트
ExtraComputeForAutomaticOptimizationChargedSeconds	RPU-seconds	지난 30분 동안 자동 최적화 작업에 대해 청구된 컴퓨팅 단위 초 수입입니다.	{Workgroup}

CloudWatch 데이터 및 스토리지 지표는 다음과 같습니다.

지표 이름	단위	설명	차원 세트
TotalTableCount	테이블 수	특정 시점에 존재하는 사용자 테이블의 수입입니다. 이 집계에는 Amazon Redshift Spectrum 테이블이 포함되지 않습니다.	{Database, Namespace}
DataStorage	MB	디스크 또는 스토리지 공간에서 Redshift 데이터에 사용된 메가바이트 수입입니다.	{Namespace}

SnapshotStorage 지표는 네임스페이스 및 작업 그룹에 구애받지 않습니다. CloudWatch의 SnapshotStorage 지표는 다음과 같습니다.

지표 이름	단위	설명	차원 세트
SnapshotStorage	MB	디스크 또는 스토리지 공간에서	{}

지표 이름	단위	설명	차원 세트
		스냅샷에 사용된 메가바이트 수입니다.	

차원 세트는 지표에 적용되는 그룹화 차원입니다. 이러한 차원 그룹을 사용하여 통계 검색 방법을 지정할 수 있습니다.

다음 표에서는 특정 지표에 대한 차원 및 차원 값에 대해 자세히 설명합니다.

측정 기준	설명 및 값
DatabaseName	데이터베이스의 이름입니다. 사용자 지정 값입니다.
Latency	가능한 값은 다음과 같습니다. - Short - 10초 미만 - Medium - 10초~10분 - Long - 10분 초과
QueryType	가능한 값은 INSERT, DELETE, UPDATE, UNLOAD, LOAD, SELECT, CTAS, OTHER입니다.
stage	쿼리 실행 단계 가능한 값은 다음과 같습니다. - QueryPlanning: SQL 문을 구문 분석하고 최적화하는 데 소요된 시간 - QueryWaiting: WLM 대기열에서 대기하는 데 소요된 시간 - QueryExecutingRead: 읽기 쿼리를 실행하는 데 소요된 시간 - QueryExecutingInsert: 삽입 쿼리를 실행하는 데 소요된 시간 - QueryExecutingDelete: 삭제 쿼리를 실행하는 데 소요된 시간

측정 기준	설명 및 값
	- QueryExecutingUpdate: 업데이트 쿼리를 실행하는 데 소요된 시간 - QueryExecutingCtas: 쿼리로 테이블을 생성하는 데 소요된 시간 - QueryExecutingUnload: 업로드 쿼리를 실행하는 데 소요된 시간 - QueryExecutingCopy: 복사 쿼리를 실행하는 데 소요된 시간 - QueryCommit: 커밋하는 데 소요된 시간
Namespace	네임스페이스의 이름입니다. 사용자 지정 값입니다.
Workgroup	작업 그룹의 이름입니다. 사용자 지정 값입니다.
UsageLimitId	사용량 한도의 식별자입니다.
UsageType	제한되는 Amazon Redshift Serverless 기능입니다. 가능한 값은 다음과 같습니다. - SERVERLESS_COMPUTE - CROSS_REGION_DATASHARING

스냅샷 및 복구 시점

Amazon Redshift Serverless의 백업은 네임스페이스에 있는 객체와 데이터를 특정 시점으로 표현한 것입니다. 백업에는 두 가지 유형이 있습니다. 하나는 수동으로 생성하는 스냅샷이고 다른 하나는 Amazon Redshift Serverless가 자동으로 생성하는 복구 시점입니다.

Amazon Redshift Serverless는 30분마다 또는 노드당 5GB의 데이터가 변경될 때마다(먼저 발생하는 시점 기준)에 복구 시점을 자동으로 생성합니다. 더 큰 데이터세트(5GB × 노드 수 초과)의 경우 복구 시점 간의 최소 간격은 15분입니다. 이러한 복구 시점은 24시간 동안 유지됩니다.

Note

복구 시점이 생성되는 시기를 제어하기 위해 자체 스냅샷 일정을 생성할 수 없습니다.

Amazon Redshift Serverless는 Redshift Managed Storage(RMS)에서 스냅샷을 만듭니다. 자세한 내용은 [Amazon Redshift Serverless의 컴퓨팅 용량](#) 섹션을 참조하세요.

Note

RA3 프로비저닝된 클러스터 및 Amazon Redshift Serverless 작업 그룹에는 백업 없음 테이블이 지원되지 않습니다. RA3 클러스터 또는 서버리스 작업 그룹에서 백업 없음으로 표시된 테이블은 스냅샷을 생성하는 동안 항상 백업되고 스냅샷에서 복원할 때 항상 복원되는 영구 테이블로 처리됩니다. 백업이 없는 테이블의 스냅샷 비용을 방지하려면 스냅샷을 생성하기 전에 자릅니다.

스냅샷이나 복구 시점에서 데이터를 검색하려는 경우 스냅샷을 서버리스 네임스페이스나 프로비저닝된 클러스터로 복원할 수 있습니다. 다음과 같은 세 가지 시나리오로 스냅샷을 복원할 수 있습니다.

- 서버리스 스냅샷을 서버리스 네임스페이스로 복원합니다.
- 서버리스 스냅샷을 프로비저닝된 클러스터로 복원합니다.
- 프로비저닝된 클러스터 스냅샷을 서버리스 네임스페이스로 복원합니다.

서버리스 스냅샷을 프로비저닝된 클러스터로 복원할 때 사용할 노드 유형(예: RA3)과 노드 수를 선택하여 클러스터 또는 노드 수준에서 설정을 제어해야 합니다.

프로비저닝된 클러스터 스냅샷을 서버리스 네임스페이스로 복원하려면 Redshift 프로비저닝된 콘솔에서 시작하여 복원할 스냅샷을 선택한 다음 스냅샷에서 복원(Restore from snapshot), 서버리스 네임스페이스로 복원(Restore to serverless)을 선택합니다. Amazon Redshift는 프로비저닝된 클러스터 스냅샷을 서버리스 네임스페이스로 복원할 때 인터리브된 키가 있는 테이블을 복합 정렬 키로 변환합니다. 정렬 키에 대한 자세한 내용은 [정렬 키 작업](#)을 참조하세요.

추가 컨텍스트를 추가하려는 경우 스냅샷 및 복구 지점에 메타데이터 및 정보를 제공하는 키-값 쌍으로 스냅샷 및 복구 지점에 태그를 지정할 수 있습니다. 리소스 태그 지정에 대한 자세한 내용은 [리소스 태그 지정 개요](#) 단원을 참조하세요.

또한 다른 AWS 계정과 스냅샷을 공유하여 해당 계정이 스냅샷 내의 데이터에 액세스하고 쿼리를 실행할 수 있습니다.

AWS Backup 통합

AWS 서비스, 클라우드 및 온프레미스에서 데이터 보호를 중앙 집중화하고 자동화하는 데 도움이 되는 완전관리형 서비스인 AWS Backup을 사용하여 스냅샷을 생성하고 복원할 수도 있습니다. 자세한 내용은 [Amazon Redshift와 AWS Backup의 통합](#) 섹션을 참조하세요. AWS Backup에 관한 자세한 내용은 AWS Backup 개발자 안내서의 [AWS Backup이란?](#)을 참조하세요.

스냅샷 생성

Note

RA3 프로비저닝된 클러스터 및 Amazon Redshift Serverless 작업 그룹에는 백업 없음 테이블이 지원되지 않습니다. RA3 클러스터 또는 서버리스 작업 그룹에서 백업 없음으로 표시된 테이블은 스냅샷을 생성하는 동안 항상 백업되고 스냅샷에서 복원할 때 항상 복원되는 영구 테이블로 처리됩니다. 백업이 없는 테이블의 스냅샷 비용을 방지하려면 스냅샷을 생성하기 전에 자릅니다.

스냅샷을 만들려면 다음 절차에 따라 단계를 수행합니다.

스냅샷을 생성하려면

1. Amazon Redshift Serverless 콘솔에서 데이터 백업(Data backup)을 선택합니다.
2. 스냅샷 생성(Create snapshot)을 선택합니다.
3. 스냅샷을 생성할 네임스페이스를 선택합니다.
4. 스냅샷 식별자를 입력합니다.
5. (선택 사항) 보존 기간을 선택합니다. 사용자 지정 값(Custom value)을 선택한 경우 일수를 선택합니다. 1일부터 3653일까지 범위에서 값을 선택해야 합니다. 기본값은 무기한 유지입니다.
6. 생성(Create)을 선택합니다.

네임스페이스 구성에서 스냅샷을 생성하려면

1. Amazon Redshift Serverless 콘솔에서 네임스페이스 구성(Namespace configuration)을 선택합니다.

2. 스냅샷을 생성할 네임스페이스를 선택합니다. 작업 그룹과 연결되어 있고 사용 가능(Available) 상태인 네임스페이스의 스냅샷만 생성할 수 있습니다.
3. 데이터 백업(Data backup) 탭을 선택합니다.
4. 스냅샷 생성(Create snapshot)을 선택합니다.
5. 스냅샷 식별자를 입력합니다.
6. (선택 사항) 보존 기간을 선택합니다. 사용자 지정 값(Custom value)을 선택한 경우 일수를 선택합니다. 1일부터 3653일까지 범위에서 값을 선택해야 합니다.
7. 생성(Create)을 선택합니다.

최종 스냅샷 생성

네임스페이스를 삭제하기 전에 네임스페이스 내의 모든 데이터에 대한 최종 스냅샷을 만들려면 다음 절차에 따라 단계를 수행합니다.

최종 스냅샷을 생성하려면

1. Amazon Redshift Serverless 콘솔에서 네임스페이스 구성(Namespace configuration)을 선택합니다.
2. 삭제할 네임스페이스를 선택합니다.
3. 작업, 삭제를 선택합니다.
4. 최종 스냅샷 생성(Create final snapshot)을 선택합니다.
5. 스냅샷 이름을 입력합니다.
6. delete를 입력합니다.
7. 삭제를 선택합니다.

스냅샷 공유 또는 스냅샷 권한 제거

스냅샷을 다른 AWS 계정과 공유하거나 스냅샷에 대한 계정의 액세스 권한을 제거하려면 다음 절차를 수행합니다.

스냅샷에 대한 액세스 권한을 공유하거나 제거하려면

1. Amazon Redshift Serverless 콘솔에서 데이터 백업(Data backup)을 선택합니다.
2. 공유할 스냅샷을 선택합니다.
3. 작업(Actions), 액세스 관리(Manage access)를 선택합니다.

4. 다른 계정과 스냅샷을 공유하려면 AWS 계정 ID를 입력합니다. 계정에서 액세스 권한을 제거하려면 제거를 선택합니다.
5. 변경 사항 저장을 선택합니다.

스냅샷 일정 예약

스냅샷 생성 시점을 정확하게 제어하려면 특정 네임스페이스에 대한 스냅샷 일정을 생성하면 됩니다. 스냅샷 생성 일정을 예약할 때 일회성 이벤트를 만들거나 Unix cron 표현식을 사용하여 반복 일정을 만들 수 있습니다. Cron 표현식은 세 개의 필드를 지원하며 공백으로 구분됩니다.

```
cron(Minutes Hours Day-of-month Month Day-of-week Year)
```

필드:	값	와일드카드:
Minutes	0~59	, - * /
Hours	0~23	, - * /
Day-of-month	1~31	, - * ? / L W
월	1~12 또는 JAN-DEC	, - * /
요일	1~7 또는 SUN~SAT	, - * ? L #
년	1970~2199	, - * /

와일드카드

- ,(쉼표) 와일드카드는 추가 값을 포함합니다. Day-of-week 필드에서 MON, WED, FRI는 월요일, 수요일 및 금요일을 포함합니다. 총 값은 필드당 24로 제한됩니다.
- -(대시) 와일드카드는 범위를 지정합니다. Hour 필드에서 1~15는 지정된 일의 1~15시간을 포함합니다.
- *(별표) 와일드카드는 필드의 모든 값을 포함합니다. Hours 필드에서 *는 모든 시간을 포함한다는 의미입니다.
- /(슬래시) 와일드카드로 증분을 지정합니다. Hours 필드에 **1/10**을 입력하면 지정된 일의 첫 시간부터 시작해서 10시간 간격(예: 01:00, 11:00, 21:00 등)을 지정할 수 있습니다.

- ?(물음표) 와일드카드는 어떤 한 가지나 다른 것을 지정합니다. Day-of-month 필드에 7을 입력하고 Day-of-week 필드에는 ?을 입력하면 매월 7일이 무슨 요일이든 상관없이 7번째 되는 날을 지정한다는 의미입니다.
- Day-of-month 또는 Day-of-week 필드에서 L 와일드카드는 해당 월 또는 주의 마지막 날을 지정합니다.
-] 필드에서는 W 와일드카드로 어떤 한 평일을 지정할 수 있습니다. Day-of-month Day-of-month 필드에서 3W를 해당 월의 세 번째 평일에 가장 가까운 날을 지정할 수 있습니다.
- '요일' 필드의 # 와일드카드는 그 달에 속한 정해진 요일의 특정 인스턴스를 지정합니다. 예를 들어, 3#2는 그 달의 두 번째 화요일입니다. 3은 각 주의 셋째 날이므로 화요일을 나타내고 2는 그 달의 두 번째 해당 요일입니다.

Note

'#' 문자를 사용하는 경우 요일(day-of-week) 필드에 하나의 표현식만 정의할 수 있습니다. 예를 들어 '3#1,6#3'은 두 개의 표현식으로 해석되기 때문에 유효하지 않습니다.

한도

- 동일한 cron 표현식에 Day-of-month와 Day-of-week 필드를 지정할 수 없습니다. 이 필드 중 하나에 값을 지정하는 경우에는 다른 필드에서 반드시 ?(물음표)를 사용해야 합니다.
- 스냅샷 일정은 다음 빈도를 지원하지 않습니다.
 - 시간당 1개보다 더 높은 빈도로 예약된 스냅샷
 - 1일(24시간)당 1개보다 더 낮은 빈도로 예약된 스냅샷

1시간 기간 내 스냅샷을 예약하게 되는 중복된 일정이 있으면 검증 오류가 발생합니다.

다음 테이블에는 몇 가지 샘플 cron 문자열이 나와 있습니다.

분	시간	요일	의미			
0	14-20/1	화	화요일 오후 2시~오후 8시 사이 매시간			
0	21	월-금	월요일~금요일 매일 밤 9시입니다.			

분	시간	요일	의미			
30	0/6	토-일	당일 자정 이후 30분 (00:30)에 시작해서 토요일과 일요일에 6시간씩 증분 이 결과 매일 [00:30, 06:30, 12:30 및 18:30]에 스냅샷이 생성됩니다.			
30	12/4	*	매일 12:30에 시작해서 4시간씩 증분 이 경우 [12:30, 16:30, 20:30]이 됩니다.			

다음 예시에서는 매일 15:15에 시작하여 2시간 단위로 실행되는 일정을 생성하는 방법을 보여줍니다.

```
cron(15 15/2 *)
```

Amazon Redshift Serverless 콘솔, API 또는 AWS CLI를 사용해서만 스냅샷 일정을 생성할 수 있습니다.

스냅샷을 예약하려면

1. Amazon Redshift Serverless 콘솔에서 데이터 백업(Data backup)을 선택합니다.
2. 스냅샷 일정을 선택합니다.
3. 일정 생성을 선택합니다.
4. 스냅샷 일정에 이름을 입력합니다.
5. 스냅샷을 생성할 네임스페이스를 선택합니다.
6. 일정에 대한 cron 표현식을 입력하거나 일정 작성기를 사용하여 일정을 생성합니다.
7. (선택 사항) 보존 기간을 선택합니다. 사용자 지정 값을 선택한 경우 일수를 지정합니다.
8. 일정 생성을 선택합니다.

스냅샷 보존 기간 업데이트

스냅샷 보존 기간을 업데이트하려면 다음 절차를 수행합니다.

스냅샷 보존 기간을 업데이트하려면

1. Amazon Redshift Serverless 콘솔에서 데이터 백업(Data backup)을 선택합니다.
2. 업데이트할 스냅샷을 선택합니다.
3. 작업(Actions), 수동 스냅샷 설정 지정(Set manual snapshot settings)을 선택합니다.
4. 보존 기간을 선택합니다. 사용자 지정 값(Custom value)을 선택한 경우 일수를 선택합니다.
5. 변경 사항 저장을 선택합니다.

스냅샷 삭제

스냅샷을 삭제하려면 다음 절차를 수행합니다.

스냅샷을 삭제하는 방법

Note

다른 계정과 공유된 스냅샷은 삭제할 수 없습니다. 스냅샷을 삭제하기 전에 먼저 해당 계정의 스냅샷에 대한 액세스 권한을 제거해야 합니다.

1. Amazon Redshift Serverless 콘솔에서 데이터 백업(Data backup)을 선택합니다.
2. 삭제할 스냅샷을 선택합니다.
3. 작업(Actions), 삭제(Delete)를 선택합니다.
4. 삭제를 선택합니다.

스냅샷 복원

Note

RA3 프로비저닝된 클러스터 및 Amazon Redshift Serverless 작업 그룹에는 백업 없음 테이블이 지원되지 않습니다. RA3 클러스터 또는 서버리스 작업 그룹에서 백업 없음으로 표시된 테이블은 스냅샷을 생성하는 동안 항상 백업되고 스냅샷에서 복원할 때 항상 복원되는 영구 테이블입니다.

블로 처리됩니다. 백업이 없는 테이블의 스냅샷 비용을 방지하려면 스냅샷을 생성하기 전에 자릅니다.

스냅샷을 서버리스 네임스페이스로 복원하면 현재 데이터베이스가 스냅샷의 데이터베이스로 바뀝니다.

스냅샷을 서버리스 네임스페이스로 복원하는 작업은 두 단계로 완료됩니다. 첫 번째 단계는 몇 분 안에 완료되며, 데이터를 네임스페이스로 복원하고 쿼리에 사용할 수 있도록 합니다. 복원의 두 번째 단계에서는 데이터베이스가 튜닝되며 이로 인해 경미한 성능 문제가 나타날 수 있습니다. 두 번째 단계는 몇 시간에서 며칠, 어떤 경우에는 몇 주까지 지속될 수 있습니다. 지속 시간은 데이터 크기에 따라 다르지만 데이터베이스가 튜닝됨에 따라 성능은 점진적으로 향상됩니다. 이 단계가 끝나면 서버리스 네임스페이스가 완전히 튜닝되어 성능 문제 없이 쿼리를 제출할 수 있습니다.

스냅샷을 서버리스 네임스페이스로 복원하려면

1. Amazon Redshift Serverless 콘솔에서 데이터 백업(Data backup)을 선택합니다.
2. 복원할 스냅샷을 선택합니다. 한 번에 하나의 스냅샷만 복원할 수 있습니다.
3. 작업(Actions), 서버리스 네임스페이스로 복원(Restore to serverless namespace)을 선택합니다.
4. 복원할 사용 가능한 네임스페이스를 선택합니다. 상태가 사용 가능(Available)인 네임스페이스로만 복원할 수 있습니다.
5. 복원을 선택합니다.

스냅샷을 프로비저닝된 클러스터로 복원하려면

1. Amazon Redshift Serverless 콘솔에서 데이터 백업(Data backup)을 선택합니다.
2. 복원할 스냅샷을 선택합니다.
3. 작업(Action), 프로비저닝된 클러스터로 복원(Restore to provisioned cluster)을 선택합니다.
4. 클러스터 식별자를 입력합니다.
5. 노드 유형(Node type)을 선택합니다. 노드 수는 노드 유형에 따라 다릅니다.
6. 콘솔 페이지의 지침에 따라 클러스터 구성(Cluster configuration)에 속성을 입력합니다. 자세한 내용은 [클러스터 생성](#)을 참조하십시오.

프로비저닝된 클러스터의 스냅샷에 대한 자세한 내용은 [Amazon Redshift 스냅샷 및 백업](#)을 참조하세요.

복구 시점 변환

Amazon Redshift Serverless의 복구 시점은 약 30분마다 생성되고 24시간 동안 저장됩니다. 복구 시점을 스냅샷으로 변환하려면 다음 절차에 따라 단계를 수행합니다.

복구 시점을 스냅샷으로 변환하려면

1. Amazon Redshift Serverless 콘솔에서 데이터 백업(Data backup)을 선택합니다.
2. 복구 시점(Recovery points) 아래에서 스냅샷을 변환할 복구 시점의 생성 시간(Creation time)을 선택합니다.
3. 복구 시점에서 스냅샷 생성(Create snapshot from recovery point)을 선택합니다.
4. 스냅샷 식별자(Snapshot identifier)를 입력합니다.
5. 생성(Create)을 선택합니다.

복구 시점 복원

Amazon Redshift Serverless의 복구 시점은 약 30분마다 생성되고 24시간 동안 저장됩니다. 복구 시점을 서버리스 네임스페이스로 복원하려면 다음 절차에 따라 단계를 수행합니다.

복구 시점을 서버리스 네임스페이스로 복원하려면

1. Amazon Redshift Serverless 콘솔에서 데이터 백업(Data backup)을 선택합니다.
2. 복구 시점(Recovery points) 아래에서 복원할 복구 시점의 생성 시간(Creation time)을 선택합니다.
3. 복원(Restore)을 선택합니다. 상태가 사용 가능(Available)인 네임스페이스로만 복원할 수 있습니다.
4. 텍스트 입력 필드에 복원(restore)을 입력하고 복원(Restore)을 선택합니다.

다른 AWS 리전에 백업 복사

스냅샷과 복구 시점을 다른 AWS 리전에 자동으로 복사하도록 Amazon Redshift Serverless를 구성할 수 있습니다. 소스 AWS 리전에서 스냅샷을 생성하면 대상 리전에 복사됩니다. 한 번에 한 대상 AWS 리전에만 스냅샷과 복구 시점을 복사하도록 네임스페이스를 구성할 수 있습니다. Amazon Redshift Serverless를 사용할 수 있는 AWS 리전 목록은 Amazon Web Services 일반 참조의 [Amazon Redshift Serverless API](#)에 나열된 엔드포인트를 참조하세요.

백업 복사를 구성할 때 Amazon Redshift Serverless가 복사한 스냅샷을 보관해야 하는 기간을 보존 기간으로 지정할 수도 있습니다. 복구 시점의 보존 기간은 1일이어야 하며 변경할 수 없습니다. 대상 리

전에 있는 스냅샷의 보존 기간은 소스 리전에 있는 스냅샷의 보존 기간과 별개입니다. 기본적으로 보존 기간은 스냅샷을 무기한 보관하기 위한 것입니다. 사용자 지정 값을 선택한 경우 일수를 선택합니다. 1일부터 3,653일까지의 범위에서 값을 선택해야 합니다.

스냅샷을 복사할 대상 리전을 변경하려면 먼저 백업 복사를 비활성화한 다음 복사를 다시 활성화할 때 새 대상 리전을 지정하세요.

스냅샷이나 복구 시점을 대상 리전에 복사한 후에는 이를 사용하여 데이터를 리전에 복원할 수 있습니다.

기본적으로 데이터는 AWS에서 사용자를 위해 관리하는 키로 암호화됩니다. 다른 키를 사용하려면 소스 AWS 리전에서 백업 복사를 구성할 때 사용할 키를 선택합니다. 그러면 Amazon Redshift Serverless가 자동으로 권한 부여를 생성하여 대상 AWS 리전에서 스냅샷 암호화를 활성화합니다.

다른 리전에 백업을 복사하려면 다음 IAM 권한이 있는지 확인합니다.

```
redshift-serverless:CreateSnapshotCopyConfiguration
redshift-serverless:UpdateSnapshotCopyConfiguration
redshift-serverless:ListSnapshotCopyConfigurations
redshift-serverless>DeleteSnapshotCopyConfiguration
```

자체 KMS 키를 사용하여 백업을 암호화하는 경우 다음 권한도 필요합니다.

```
kms:CreateGrant
kms:DescribeKey
```

스냅샷 또는 복구 시점을 다른 AWS 리전에 복사하도록 구성하는 방법

1. Amazon Redshift Serverless 콘솔에서 스냅샷 또는 복구 시점 복사를 구성할 네임스페이스를 선택합니다.
2. 작업, 리전 간 백업 구성을 선택합니다.
3. 스냅샷을 복사할 대상 AWS 리전을 선택합니다.
4. (선택 사항) 스냅샷을 보존할 기간을 선택합니다. 사용자 지정 값을 선택할 경우 일수를 선택합니다. 1일부터 3,653일까지 범위에서 값을 선택해야 합니다. 기본값은 무기한 보존입니다.
5. (선택 사항) 대상 리전에서 암호화를 위해 사용할 다른 AWS KMS 키를 선택합니다.
6. 구성 저장을 선택합니다.

테이블 복원

스냅샷이나 복구 시점에서 특정 테이블을 복원할 수 있습니다. 이 경우 소스 스냅샷 또는 복구 시점, 데이터베이스, 스키마, 테이블, 대상 데이터베이스 및 새 테이블 이름을 지정합니다. 이 새 테이블은 기존 테이블과 동일한 이름을 가질 수 없습니다. 테이블을 복원하여 기존 테이블을 바꾸려면 테이블을 복원하기 전에 먼저 테이블의 이름을 바꾸거나 테이블을 삭제해야 합니다.

Note

RA3 프로비저닝된 클러스터 및 Amazon Redshift Serverless 작업 그룹에는 백업 없음 테이블이 지원되지 않습니다. RA3 클러스터 또는 서버리스 작업 그룹에서 백업 없음으로 표시된 테이블은 스냅샷을 생성하는 동안 항상 백업되고 스냅샷에서 복원할 때 항상 복원되는 영구 테이블로 처리됩니다. 그러나 백업 없음 테이블의 선택적 복원은 지원되지 않습니다.

대상 테이블은 외래 키를 제외하고 원본 테이블의 열 정의, 테이블 속성 및 열 속성을 사용하여 생성됩니다. 종속성으로 인한 충돌을 방지하기 위해 대상 테이블은 원본 테이블에서 외래 키를 상속하지 않습니다. 뷰 또는 원본 테이블에서 부여된 권한 같은 종속성은 대상 테이블에 적용되지 않습니다.

원본 테이블의 소유자가 존재하는 경우, 해당 사용자가 복원된 테이블의 소유자가 됩니다. 단, 지정된 데이터베이스와 스키마에서 관계의 소유자가 되기에 충분한 권한이 있어야 합니다. 그렇지 않은 경우, 복원된 테이블은 클러스터가 시작될 때 생성된 관리자가 소유합니다.

복원된 테이블은 백업이 생성된 시점의 상태로 되돌아갑니다. 여기에는 Amazon Redshift의 [직렬화 격리](#) 준수에 의해 정의된 트랜잭션 가시성 규칙이 포함됩니다. 이는 백업 후 시작되는 인플라이트 트랜잭션을 즉시 볼 수 있음을 뜻합니다.

Amazon Redshift Serverless 콘솔을 사용하여 스냅샷에서 테이블을 복원할 수 있습니다.

데이터 백업에서의 테이블 복원에는 다음과 같은 제한이 있습니다.

- 한 번에 하나의 테이블만 복원할 수 있습니다.
- 뷰 또는 원본 테이블에서 부여된 권한 같은 종속성은 대상 테이블에 적용되지 않습니다.
- 복원하는 테이블에 대해 행 수준 보안이 켜져 있는 경우 Amazon Redshift Serverless에서 행 수준 보안이 켜진 상태로 테이블을 복원합니다.

Amazon Redshift Serverless 콘솔을 사용하여 테이블을 복원하는 방법

1. Amazon Redshift Serverless 콘솔에서 데이터 백업(Data backup)을 선택합니다.

2. 복원할 테이블이 있는 스냅샷이나 복구 시점을 선택합니다.
3. 작업, 스냅샷에서 테이블 복원 또는 복구 시점에서 테이블 복원을 선택합니다.
4. 소스 스냅샷 또는 복구 시점 및 대상 테이블에 대한 정보를 입력한 후 테이블 복원을 선택합니다.

Amazon Redshift Serverless에서 데이터 공유

데이터 공유를 사용하면 데이터에 실시간으로 액세스할 수 있으므로 Amazon Redshift Serverless에서 업데이트되는 최신 정보와 일관된 정보를 사용자가 볼 수 있습니다.

AWS 계정 내 또는 간의 여러 Amazon Redshift Serverless 인스턴스에서 읽기용으로 데이터를 공유할 수 있습니다.

SQL 인터페이스나 Amazon Redshift 콘솔을 사용하여 datashare를 시작할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서에서 [Amazon Redshift에서 데이터 공유](#)를 참조하세요.

데이터 공유를 통해 Amazon Redshift Serverless 네임스페이스와 프로비저닝된 클러스터는 AWS 계정 전체 또는 AWS 계정 전체에서 AWS 리전 내에 있는지 여부에 관계없이 서로 라이브 데이터를 공유할 수 있습니다. 자세한 내용은 [데이터 공유가 가능한 리전](#) 섹션을 참조하세요.

AWS 계정 내에서 데이터 공유를 시작하려면 AWS Management Console을 연 다음 Amazon Redshift 콘솔을 선택합니다. 네임스페이스 구성(Namespace configuration)을 선택한 다음 데이터 공유(Datashares)를 선택합니다.

데이터 공유에서 데이터 쿼리를 시작하려면 연결된 작업 그룹이 있는 네임스페이스에 데이터베이스를 생성합니다. 지정된 데이터 공유에서 연결된 작업 그룹이 있는 네임스페이스를 선택하고 데이터를 쿼리할 데이터베이스를 생성합니다.

고려 사항

Amazon Redshift Serverless에서 데이터 공유 작업을 수행할 때는 다음 사항을 고려하세요.

- Amazon Redshift는 인스턴스 유형 ra3.16xlarge, ra3.4xlarge 및 ra3.xlplus의 프로비저닝된 클러스터와 데이터 공유 생산자 또는 소비자로서 서버리스 엔드포인트만 지원합니다.
- Amazon Redshift Serverless는 기본적으로 암호화됩니다.

지원되는 데이터베이스 객체, 암호화 요구 사항 및 정렬 키 요구 사항을 비롯한 데이터 공유 제한 목록은 Amazon Redshift 데이터베이스 개발자 안내서의 [Considerations for data sharing in Amazon Redshift](#) 섹션을 참조하세요.

데이터 공유를 볼 수 있는 액세스 권한 부여

수퍼유저는 모든 사용자가 생성한 데이터 공유를 볼 수 있도록 수퍼유저가 아닌 사용자에게 액세스 권한을 제공할 수 있습니다.

사용자에게 데이터 공유에 대한 액세스 권한을 부여하려면 다음 명령을 사용하여 사용자에게 데이터 공유 액세스를 제공하세요. 여기서 `datashare_name`은 데이터 공유의 이름이고 `user-name`은 액세스를 제공하려는 사용자의 이름입니다.

```
grant share on datashare datashare_name to "IAM:test_user";
```

사용자 그룹의 데이터 공유에 대한 액세스 권한을 부여하려면 먼저 사용자로 사용자 그룹을 만듭니다. 사용자 그룹을 만드는 방법에 대한 자세한 내용은 [CREATE GROUP](#) 섹션을 참조하세요. 그런 다음 다음 명령을 사용하여 사용자에게 데이터 공유 액세스 권한을 부여합니다. 여기서 `datashare_name`은 데이터 공유의 이름이고 `user-group`은 액세스 권한을 부여하려는 사용자 그룹의 이름입니다.

```
grant share on datashare datashare_name to group user_group;
```

GRANT 문을 사용하는 방법에 대한 자세한 내용은 [GRANT](#) 섹션을 참조하세요.

AWS Glue Data Catalog에 네임스페이스 등록

AWS Glue Data Catalog에 전체 네임스페이스를 등록하고 AWS Glue에서 관리하는 카탈로그를 만들 수 있습니다. Apache Iceberg REST API를 지원하는 SQL 엔진을 사용하여 이러한 카탈로그에 액세스할 수 있습니다. Amazon Redshift에서 Apache Iceberg 호환 카탈로그를 만드는 방법에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Apache Iceberg compatibility for Amazon Redshift](#) 섹션을 참조하세요.

AWS Glue Data Catalog에 서버리스 네임스페이스를 등록하는 방법

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 Redshift Serverless를 선택합니다. Serverless 대시보드가 나타납니다. 네임스페이스/작업 그룹 섹션에는 현재 AWS 리전의 계정에 대한 네임스페이스 및 작업 그룹 목록이 있습니다.

- 다. 네임스페이스가 없는 경우 작업 그룹 생성을 선택하여 작업 그룹과 해당 네임스페이스를 만듭니다.
3. 등록하려는 네임스페이스의 이름을 선택합니다.
4. 작업에서 AWS Glue Data Catalog에 등록을 선택합니다. AWS Glue Data Catalog에 등록 팝업 상자가 나타납니다.
5. 대상 계정 ID 아래에서 네임스페이스를 등록할 AWS 계정 ID를 입력합니다. AWS Glue Data Catalog에 카탈로그를 보관할 계정 ID입니다.
6. 네임스페이스 등록 이름 아래에 이름을 입력합니다. 이것이 Data Catalog에서 네임스페이스 이름이 됩니다.
7. 등록을 선택합니다. 이제 AWS Lake Formation 콘솔로 이동하게 됩니다.
8. AWS Lake Formation의 카탈로그 만들기 프로세스를 따릅니다. 카탈로그 만들기에 대한 자세한 내용은 AWS Lake Formation 개발자 안내서의 [Bringing Amazon Redshift data into the AWS Glue Data Catalog](#) 섹션을 참조하세요.

Amazon Redshift Serverless에서 리소스에 태그 지정

AWS에서 태그는 키-값 페어로 구성된 사용자 정의 레이블입니다. Amazon Redshift Serverless는 리소스에 대한 메타데이터를 한눈에 파악할 수 있도록 태그 지정을 지원합니다.

리소스에서는 태그가 필요하지 않지만 컨텍스트를 이해하는 데 도움이 될 수 있습니다. 리소스와 관련된 정보가 있는 메타데이터를 사용하여 리소스에 태그를 지정할 수 있습니다. 예를 들어 테스트 환경과 프로덕션 환경에 속하는 리소스를 추적한다고 가정하겠습니다. 그렇다면 environment라는 이름의 키를 생성한 후 test 또는 production 값을 입력하여 각 환경에서 사용하는 리소스를 식별할 수 있습니다. 다른 AWS 서비스에서 태그를 지정하거나, 혹은 기본적인 비즈니스 카테고리가 있는 경우에는 리소스에서도 일관성 유지를 위해 동일한 키-값 페어를 생성하는 것이 좋습니다.

리소스를 삭제하는 경우에는 연결된 태그까지 모두 삭제됩니다. AWS CLI와 Amazon Redshift Serverless 콘솔을 모두 사용하여 서버리스 리소스에 태그를 지정할 수 있습니다. 사용 가능한 API 작업은 TagResource, UntagResource 및 ListTagsForResource입니다.

각 리소스마다 태그 집합이 하나씩 있습니다. 여기에서 태그 집합이란 리소스에 할당되는 태그들이 하나 이상 모인 집합을 말합니다. 각 리소스는 태그 집합당 최대 50개의 태그를 가질 수 있습니다. 리소스를 생성할 때, 그리고 리소스가 생성된 후에도 태그 추가는 가능합니다. 태그를 추가할 수 있는 서버리스 리소스 유형은 다음과 같습니다.

- 작업 그룹

- 네임스페이스
- 스냅샷
- 복구 시점

태그를 지정할 때 요건은 다음과 같습니다.

- 키는 접두사로 `aws:`를 사용할 수 없습니다.
- 키는 태그 집합에 대해 고유해야 합니다.
- 키는 1~128자 사이의 허용된 문자이어야 합니다.
- 값은 0~256자 사이의 허용된 문자이어야 합니다.
- 값은 태그 집합마다 고유할 필요는 없습니다.
- 키와 값의 문자로는 Unicode 문자, 숫자, 공백 그리고 다음 기호가 허용됩니다. `_ . : / = + - @`.
- 키와 값은 대/소문자를 구분합니다.

Amazon Redshift Serverless 리소스에 대한 태그를 관리하려면

1. Amazon Redshift Serverless 콘솔에서 Manage Tags(태그 관리)를 선택합니다.
2. 검색할 리소스 유형을 입력하고 Search resources(리소스 검색)를 선택합니다. 태그를 관리할 리소스를 선택한 다음 Manage tags(태그 관리)를 선택합니다.
3. 리소스에 추가하려는 키와 선택적 값을 지정합니다. 태그를 수정할 때 태그의 값은 변경할 수 있지만 키는 변경할 수 없습니다.
4. 태그 추가, 제거 또는 수정을 완료한 후 Save changes(변경 사항 저장)를 선택한 다음 Apply(적용)를 선택하여 변경 내용을 저장합니다.

Amazon Redshift 프로비저닝 클러스터

Amazon Redshift 데이터 웨어하우스는 노드라는 컴퓨팅 리소스의 모음으로, 노드는 클러스터라는 그룹을 구성합니다. 각 클러스터는 Amazon Redshift 엔진을 실행하며, 하나 이상의 데이터베이스를 포함합니다.

Note

현재 Amazon Redshift 버전 1.0 엔진을 사용할 수 있습니다. 하지만 엔진이 업데이트되면 여러 Amazon Redshift 엔진 버전을 선택하여 사용할 수 있습니다.

Amazon Redshift의 클러스터 및 노드

Amazon Redshift 클러스터는 노드로 구성됩니다. 각 클러스터마다 리더 노드 1개와 컴퓨팅 노드 1개 이상이 있습니다. 리더 노드는 클라이언트 애플리케이션에서 쿼리를 수신하여 구문을 분석한 후 쿼리 실행 계획을 작성합니다. 그런 다음 리더 노드는 이러한 계획의 병렬 실행을 컴퓨팅 노드에 맞춰 조정하고 이들 노드에서 중간 결과를 집계합니다. 그런 다음 최종적으로 결과를 다시 클라이언트 애플리케이션에 반환합니다.

컴퓨팅 노드는 쿼리 실행 플랜을 실행하고 쿼리를 처리하기 위해 노드 간에 데이터를 전송합니다. 중간 결과는 클라이언트 애플리케이션으로 보내지기 전에 집계를 위해 리더 노드로 보내집니다. 리더 노드와 컴퓨팅 노드에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [데이터 웨어하우스 시스템 아키텍처](#)를 참조하세요.

Note

Amazon Redshift 콘솔(<https://console.aws.amazon.com/redshiftv2/>)에서 클러스터를 생성할 때 데이터 크기 및 쿼리 특성을 기반으로 클러스터 구성 권장 사항을 제공 받을 수 있습니다. 이 크기 조정 계산기를 사용하려면 RA3 노드 유형을 지원하는 AWS 리전의 콘솔에서 선택에 대한 도움말(Help me choose)을 찾습니다. 자세한 내용은 [클러스터 생성](#) 섹션을 참조하세요.

클러스터를 시작할 때 지정하는 한 가지 옵션이 노드 유형입니다. 노드 유형에서는 각 노드의 CPU, RAM, 스토리지 용량, 및 스토리지 드라이브 유형을 결정합니다.

Amazon Redshift는 워크로드를 수용할 수 있도록 다양한 노드 유형을 제공하며 필요한 성능, 데이터 크기 및 예상 데이터 증가에 따라 RA3 또는 DC2를 선택하는 것이 좋습니다.

관리형 스토리지가 포함된 RA3 노드를 사용하면 컴퓨팅 및 관리형 스토리지를 개별적으로 조절하고 비용을 지불하는 방식으로 데이터 웨어하우스를 최적화할 수 있습니다. RA3에서는 성능 요구 사항에 따라 노드 수를 선택하며, 사용하는 관리형 스토리지에 대해서만 비용을 지불합니다. RA3 클러스터는 매일 처리하는 데이터 크기에 따라 크기를 조정합니다. Virtual Private Cloud(VPC)에서 RA3 노드 유형을 사용하는 클러스터를 시작합니다. 자세한 내용은 [VPC에서 Redshift 프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹 생성](#) 섹션을 참조하세요.

Amazon Redshift 관리형 스토리지는 각 RA3 노드에서 빠른 로컬 스토리지에는 대용량 고성능 SSD를 사용하며, 내구성이 뛰어난 장기 스토리지에는 Amazon S3를 사용합니다. 한 노드의 데이터가 대용량 로컬 SSD의 크기 이상으로 증가하는 경우 Amazon Redshift 관리형 스토리지는 자동으로 해당 데이터를 Amazon S3에 오프로드합니다. 데이터가 고성능 SSD 또는 Amazon S3 중에 어디에 저장되는지에 관계없이, Amazon Redshift 관리형 스토리지에 대한 동일한 낮은 요금을 지불합니다. 지속적으로 확장하는 스토리지가 필요한 워크로드의 경우 관리형 스토리지를 사용하면 컴퓨팅 노드에서 별도로 데이터 웨어하우스 스토리지 용량을 자동으로 규모 조정할 수 있습니다.

DC2 노드를 사용하면 로컬 SSD 스토리지가 포함된 컴퓨팅 집약적인 데이터 웨어하우스를 갖출 수 있습니다. 데이터 크기와 성능 요구 사항을 기반으로 필요한 노드의 수를 선택합니다. DC2 노드는 높은 성능을 위해 데이터를 로컬로 저장하며, 데이터 크기가 증가하면 컴퓨팅 노드를 더 추가하여 클러스터의 스토리지 용량을 늘릴 수 있습니다. 1TB 미만의 비압축 데이터 세트에 대해서는 가장 저렴한 가격으로 최고의 성능을 얻을 수 있도록 DC2 노드 유형을 권장합니다. 데이터가 증가할 것으로 예상되면 컴퓨팅 및 스토리지 크기를 독립적으로 조정하여 더 좋은 요금과 성능을 얻을 수 있도록 RA3 노드를 사용하는 것이 좋습니다. Virtual Private Cloud(VPC)에서 DC2 노드 유형을 사용하는 클러스터를 시작합니다. 자세한 내용은 [VPC에서 Redshift 프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹 생성](#) 섹션을 참조하세요.

노드 유형은 사용할 수 있는 크기도 다양합니다. 클러스터의 전체 스토리지도 노드 유형과 노드 수에 따라 결정됩니다. 자세한 내용은 [노드 유형 세부 정보](#) 섹션을 참조하세요.

일부 노드 유형에서는 노드 1개(단일 노드) 또는 노드 2개 이상(다중 노드)이 허용됩니다. 일부 노드 유형의 클러스터에 대한 최소 노드 수는 노드 2개입니다. 단일 노드 클러스터에서는 리더 기능과 컴퓨팅 기능이 노드 하나로 공유됩니다. 단일 노드 클러스터는 프로덕션 워크로드를 실행하는 데 사용하지 않는 것이 좋습니다. 반면에 다중 노드 클러스터에서는 리더 노드와 컴퓨팅 노드가 분리됩니다. 리더 노드는 컴퓨팅 노드와 동일한 노드 유형입니다. 컴퓨팅 노드에 대한 요금만 지불합니다.

Amazon Redshift는 각 AWS 리전의 AWS 계정마다 리소스에 할당량을 적용합니다. 할당량은 노드 또는 스냅샷과 같이 AWS 리전 내에서 지정된 리소스 유형에 대해 계정에서 생성할 수 있는 리소스 수를 제한합니다. Amazon Redshift 리소스에 적용되는 기본 할당량에 대한 자세한 내용은 [Amazon Redshift의 할당량 및 제한](#) 섹션을 참조하세요.

클러스터 비용은 AWS 리전, 노드 유형, 노드 수, 노드의 사전 예약 여부에 따라 다릅니다. 노드 비용에 대한 자세한 내용은 [Amazon Redshift 요금](#) 페이지에서 확인할 수 있습니다.

노드 유형 세부 정보

아래 표는 각 노드 유형 및 크기에 따라 노드 명세를 요약한 것입니다. 표의 제목에는 다음과 같은 의미가 있습니다.

- vCPU는 각 노드의 가상 CPU 수입니다.
- RAM은 각 노드의 메모리 크기(기비바이트: GiB)입니다.
- 노드당 기본 조각은 클러스터를 생성하거나 클래식 크기 조정으로 크기를 조정할 때 컴퓨팅 노드가 분할되는 조각 수입니다.

탄력적 크기 조정을 사용하여 클러스터 크기를 조정할 경우 노드당 조각 수가 변경될 수 있습니다. 그러나 탄력적 크기 조정 후에도 클러스터의 모든 컴퓨팅 노드에 있는 총 조각 수는 동일하게 유지됩니다.

스냅샷에서 복원 작업을 사용하여 클러스터를 생성할 때 노드 유형을 변경하면 결과 클러스터의 조각 수가 원래 클러스터에서 변경될 수 있습니다.

- 스토리지는 각 노드의 스토리지 용량 및 유형입니다.
- 노드 범위는 노드 유형 및 크기에 따라 Amazon Redshift에서 지원되는 노드의 최소 수와 최대 수입니다.

Note

선택한 AWS 리전의 AWS 계정에 적용되는 할당량에 따라 제한되는 노드 수가 줄어들 수도 있습니다. Amazon Redshift 리소스에 적용되는 기본 할당량에 대한 자세한 내용은 [Amazon Redshift의 할당량 및 제한](#) 섹션을 참조하세요.

- 총 용량은 노드 범위에서 지정하는 최대 노드 수를 배포했을 때 클러스터의 전체 스토리지 용량을 말합니다.

다음 테이블에서는 RA3 노드의 사양을 설명합니다.

노드 유형	vCPU	RAM(GB)	노드당 기본 스토리지	노드당 관리형 스토리지 한도 ¹	클러스터 생성을 사용한 노드 범위	총 관리형 스토리지 용량 ²	
ra3.large(단일 노드)	2	16	2	1TB	1	1TB ³	
ra3.large(다중 노드)	2	16	2	8TB	2~16	128TB	
ra3.xlplus(단일 노드)	4	32	2	4TB	1	4TB ³	
ra3.xlplus(다중 노드)	4	32	2	32TB	2~32 ⁴	1,024TB ⁴	
ra3.4xlarge	12	96	4	128TB	2~32 ⁵	8,192TB ⁵	
ra3.16xlarge	48	384	16	128TB	2~128	16,384TB	

¹ Amazon Redshift 관리형 스토리지의 스토리지 한도입니다. 이것은 하드 제한입니다.

² 총 관리형 스토리지 한도는 노드당 관리 스토리지 한도를 곱한 최대 노드 수입니다.

³ 단일 노드 클러스터를 다중 노드로 크기 조정하려면 클래식 크기 조정만 지원됩니다.

⁴ 최대 32개의 노드가 있는 ra3.xlplus(다중 노드) 노드 유형으로 클러스터를 생성할 수 있습니다. 다중 노드 클러스터의 경우 탄력적 크기 조정을 통해 최대 32개 노드까지 크기를 조정할 수 있습니다.

⁵ 최대 32개의 노드가 있는 ra3.4xlarge 노드 유형으로 클러스터를 생성할 수 있습니다. 탄력적 크기 조정으로 최대 64개 노드까지 크기를 조정할 수 있습니다.

다음 테이블에서는 고밀도 컴퓨팅 노드의 사양을 설명합니다.

노드 유형	vCPU	RAM(GB)	노드당 기본 스토리지	노드당 스토리지	노드 범위	총 용량	
dc2.large	2	15	2	160GB NVMe-SSD	1~32	5.12TB	
dc2.8xlarge	32	244	16	2.56TB NVMe-SSD	2~128	326TB	

 Note

고밀도 스토리지(DS2) 노드 유형은 더 이상 사용할 수 없습니다.

이전 노드 유형 이름

이전 버전의 Amazon Redshift에서는 특정 노드 유형의 이름이 달랐습니다. Amazon Redshift API 및 AWS CLI에서 이전 이름을 사용할 수 있습니다. 하지만 그보다는 이러한 이름을 참조하는 스크립트를, 현재 이름을 사용하도록 업데이트하는 것이 좋습니다. 현재 이름과 이전 이름은 다음과 같습니다.

현재 이름	이전 이름
ds2.xlarge	ds1.xlarge, dw.hs1.xlarge, dw1.xlarge
ds2.8xlarge	ds1.8xlarge, dw.hs1.8xlarge, dw1.8xlarge
dc1.large	dw2.large
dc1.8xlarge	dw2.8xlarge

노드 수 결정

Amazon Redshift는 클러스터의 모든 컴퓨팅 노드에 걸쳐 쿼리를 배포하고 병렬식으로 실행하기 때문에 노드를 클러스터에 추가하여 쿼리 성능을 높일 수 있습니다. 두 개 이상의 컴퓨팅 노드에서 클러스터를 실행할 경우 각 노드의 데이터가 다른 노드의 디스크에 미러링되므로 데이터가 손실될 위험이 줄어듭니다.

Amazon Redshift 콘솔 및 Amazon CloudWatch 지표를 사용하여 쿼리 성능을 모니터링할 수 있습니다. 클러스터에 대한 요금과 성능 간의 균형을 맞추기 위해 필요에 따라 노드를 추가하거나 제거할 수도 있습니다. 추가 노드를 요청하면 Amazon Redshift가 배포, 로드 밸런싱 및 데이터 유지 보수에 필요한 모든 세부 정보를 관리합니다. 성능에 대한 자세한 내용은 [Amazon Redshift 클러스터 성능 모니터링](#) 섹션을 참조하세요.

예약 노드는 일정한 상태의 프로덕션 워크로드에 적합하며 온디맨드 요금보다 상당히 많은 할인을 제공합니다. 실험 및 개념 증명을 실행한 후 예약 노드를 구입하여 프로덕션 구성을 검증할 수 있습니다. 자세한 내용은 [예약 노드](#) 섹션을 참조하세요.

클러스터를 일시 중지하면 클러스터가 일시 중지되는 시간 동안 온디맨드 결제가 일시 중지됩니다. 일시 중지된 이 시간 동안에는 백업 스토리지에 대해서만 요금을 지불합니다. 이렇게 하면 필요한 데이터 웨어하우스 용량을 미리 계획하고 구매할 필요가 없으며 개발 또는 테스트를 위해 환경을 비용 효과적으로 관리할 수 있습니다.

온디맨드 및 예약 노드의 요금에 대한 자세한 내용은 [Amazon Redshift 요금](#)을 참조하세요.

EC2를 사용하여 클러스터 생성

Amazon Redshift 클러스터는 선택하는 Amazon Redshift 노드 유형 및 크기에 대해 구성된 Amazon EC2 인스턴스에서 실행됩니다. 이러한 네트워킹 플랫폼에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [지원되는 플랫폼](#)을 참조하세요.

Note

SQL 클라이언트 도구와 Amazon Redshift 데이터베이스 간 연결 문제를 방지하기 위해 다음 2가지 방법 중 하나를 사용하는 것이 좋습니다. 호스트가 패킷 크기를 협상하도록 인바운드 규칙을 구성할 수 있습니다. 또는 Amazon EC2 인스턴스 네트워크 인터페이스(NIC)의 최대 전송 단위(MTU)를 1500으로 설정해 TCP/IP 점보 프레임을 사용 중지할 수 있습니다. 이러한 옵션에 대한 자세한 내용은 [쿼리가 중단되거나, 간혹 클러스터까지 전송되지 않습니다](#) 섹션을 참조하세요.

Amazon Virtual Private Cloud(Amazon VPC)

Amazon VPC를 사용하면 클러스터가 AWS 계정에 논리적으로 격리된 가상 프라이빗 클라우드(VPC)에서 실행됩니다. Amazon VPC에서 클러스터를 프로비저닝하는 경우 하나 이상의 VPC 보안 그룹을 클러스터와 연결하여 클러스터에 대한 액세스를 제어합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [VPC의 보안 그룹](#) 섹션을 참조하세요.

VPC에서 클러스터를 생성하려면 먼저 VPC의 서브넷 정보를 입력하여 Amazon Redshift 클러스터 서브넷 그룹을 생성한 다음 클러스터를 시작할 때 서브넷 그룹을 입력해야 합니다. 자세한 내용은 [Redshift 리소스의 서브넷](#) 섹션을 참조하세요.

Amazon Virtual Private Cloud(Amazon VPC)에 대한 자세한 내용은 [Amazon VPC 제품 세부 정보 페이지](#)를 참조하세요.

기본 디스크 공간 경보

Amazon Redshift 클러스터를 생성할 때는 옵션으로 Amazon CloudWatch 경보를 구성하여 모든 클러스터 노드에서 사용 중인 디스크 공간의 평균 비율을 모니터링할 수 있습니다. 이러한 경보를 기본 디스크 공간 경보라고 부릅니다.

기본 디스크 공간 경보의 목적은 클러스터의 스토리지 용량을 모니터링하는 데 있습니다. 이 경보는 데이터 웨어하우스의 요건에 따라 구성할 수 있습니다. 예를 들어, 이 경보를 클러스터 크기 조정에 필요한 지표로 사용할 수 있습니다. 여러 노드 유형에 대해 크기를 조정하거나 노드를 추가할 수 있으며, 또는 향후 확장을 고려하여 예약 노드를 구매할 수 있습니다.

기본 디스크 공간 경보는 디스크 사용량이 일정한 횟수와 지속 시간으로 지정한 비율에 도달하거나, 이를 초과할 때 트리거됩니다. 기본적으로는 지정한 비율에 도달할 때 트리거되며, 해당 비율 이상에서 5분 이상 지속됩니다. 기본 값은 클러스터를 시작한 후에도 편집할 수 있습니다.

CloudWatch 경보가 트리거되면 Amazon Simple Notification Service(Amazon SNS)는 지정된 수신자에게 알림을 보내 백분율 임계값에 도달했음을 경고합니다. Amazon SNS는 주제를 사용하여 수신자와 알림으로 전송되는 메시지를 지정합니다. 기존 Amazon SNS 주제를 사용할 수도 있지만 그렇지 않고 클러스터를 시작할 때 지정한 설정에 따라 주제를 생성할 수도 있습니다. 경보 주제는 클러스터를 시작한 후에도 편집 가능합니다. Amazon SNS 주제 생성에 대한 자세한 내용은 [Amazon Simple Notification Service 시작하기](#) 섹션을 참조하세요.

클러스터를 시작한 후 클러스터 상태(Status) 창의 CloudWatch 경보(CloudWatch Alarms)에서 경보를 보고 편집할 수 있습니다. 이름은 `percentage-disk-space-used-default-<string>`입니다. 먼저 경보를 열어 연결되어 있는 Amazon SNS 주제를 확인한 후 경보 설정을 편집하면 됩니다. 기존 Amazon SNS 주제를 선택하지 않은 경우에는 직접 생성한 주제의 이름은 `<clustername>-default-`

alarms(<**recipient**>)가 됩니다. 예를 들면 examplecluster-default-alarms (notify@example.com)와 같습니다.

기본 디스크 공간 경보의 구성 및 편집에 대한 자세한 내용은 [클러스터 생성](#) 및 [디스크 공간 경보 생성](#) 섹션을 참조하세요.

Note

클러스터를 삭제해도 클러스터와 연결된 경보가 삭제되지는 않지만 트리거되지도 않습니다. 더 이상 필요 없는 경보는 CloudWatch 콘솔에서 삭제할 수 있습니다.

클러스터 상태

클러스터 상태에는 현재 클러스터 상태가 표시됩니다. 다음 표는 각 클러스터 상태를 설명한 것입니다.

상태 표시기	설명
available	클러스터가 실행 중이며 사용할 수 있습니다.
available, prep-for-resize	클러스터가 탄력적으로 크기를 조정할 수 있도록 준비 중입니다. 클러스터가 실행 중이고 읽기 및 쓰기 쿼리에 사용할 수 있지만 스냅샷 생성 등과 같은 클러스터 작업은 수행할 수 없습니다.
available, resize-cleanup	탄력적 크기 조정 작업이 새 클러스터 노드로 데이터 전송을 완료하는 중입니다. 클러스터가 실행 중이고 읽기 및 쓰기 쿼리에 사용할 수 있지만 스냅샷 생성 등과 같은 클러스터 작업은 수행할 수 없습니다.
cancelling- resize	크기 조정 작업을 취소하는 중입니다.
creating	Amazon Redshift가 클러스터를 생성하고 있습니다. 자세한 내용은 클러스터 생성 섹션을 참조하세요.
deleting	Amazon Redshift가 클러스터를 삭제하고 있습니다. 자세한 내용은 클러스터 종료 및 삭제 섹션을 참조하세요.
final-snapshot	Amazon Redshift가 삭제에 앞서 클러스터의 최종 스냅샷을 생성 중입니다. 자세한 내용은 클러스터 종료 및 삭제 섹션을 참조하세요.

상태 표시기	설명
hardware-failure	클러스터에 하드웨어 장애가 발생했습니다. 단일 노드 클러스터가 있는 경우 노드를 변경할 수 없습니다. 클러스터를 복구하려면 스냅샷을 복원하세요. 자세한 내용은 Amazon Redshift 스냅샷 및 백업 섹션을 참조하세요.
incompatible-hsm	Amazon Redshift가 하드웨어 보안 모듈(HSM)에 연결할 수 없습니다. 클러스터와 HSM 사이의 HSM 구성을 확인하세요. 자세한 내용은 하드웨어 보안 모듈을 사용한 암호화 섹션을 참조하세요.
incompatible-network	기본 네트워크 구성에 문제가 있습니다. 클러스터를 시작한 VPC가 존재하는지, 그리고 VPC 설정이 올바른지 확인하세요. 자세한 내용은 VPC의 Redshift 리소스 섹션을 참조하세요.
incompatible-parameters	연결되어 있는 파라미터 그룹에서 1개 이상의 파라미터 값에 문제가 있어서 파라미터 값을 적용할 수 없습니다. 파라미터 그룹을 수정하고 잘못된 값은 모두 업데이트하세요. 자세한 내용은 Amazon Redshift 파라미터 그룹 섹션을 참조하세요.
incompatible-restore	스냅샷에서 클러스터를 복원하는 데 문제가 발생했습니다. 다른 스냅샷을 이용해 클러스터를 다시 복원하세요. 자세한 내용은 Amazon Redshift 스냅샷 및 백업 섹션을 참조하세요.
modifying	Amazon Redshift가 클러스터에 변경 사항을 적용 중입니다. 자세한 내용은 클러스터 수정 섹션을 참조하세요.
paused	클러스터가 일시 중지되었습니다. 자세한 내용은 클러스터 일시 중지 및 다시 시작 섹션을 참조하세요.
rebooting	Amazon Redshift가 클러스터를 재부팅하고 있습니다. 자세한 내용은 클러스터 재부팅 섹션을 참조하세요.
renaming	Amazon Redshift가 클러스터에 새 이름을 적용 중입니다. 자세한 내용은 클러스터 이름 변경 섹션을 참조하세요.
resizing	Amazon Redshift가 클러스터 크기를 조정하고 있습니다. 자세한 내용은 클러스터 크기 조정 섹션을 참조하세요.

상태 표시기	설명
rotating-keys	Amazon Redshift가 클러스터의 암호 키를 교체하고 있습니다. 자세한 내용은 키 교체 암호화 섹션을 참조하세요.
storage-full	클러스터에 할당된 스토리지 용량이 거의 다 찼습니다. 클러스터 크기를 조정하여 노드를 추가하거나 다른 노드 크기를 선택하세요. 자세한 내용은 클러스터 크기 조정 섹션을 참조하세요.
updating-hsm	Amazon Redshift가 HSM 구성을 업데이트 중입니다.

Amazon Redshift 프로비저닝된 클러스터 사용 시 고려 사항

클러스터를 생성한 후에는 이 섹션에서 기능을 사용할 수 있는 리전, 유지 관리 작업, 노드 유형 및 사용 제한에 대한 정보를 찾을 수 있습니다.

리전 및 가용 영역 고려 사항

Amazon Redshift는 여러 AWS 리전에서 사용할 수 있습니다. 기본적으로 Amazon Redshift는 사용자가 선택한 AWS 리전 내에서 가용 영역(AZ)을 무작위로 선택하여 클러스터를 프로비저닝합니다. 모든 클러스터 노드는 동일한 AZ에 프로비저닝됩니다.

해당 영역에서 Amazon Redshift를 사용할 수 있는 경우 선택적으로 특정 가용 영역을 요청할 수 있습니다. 예를 들어 한 가용 영역에서 실행 중인 Amazon EC2 인스턴스가 이미 있는 경우 지연 시간을 줄이기 위해 동일한 영역에서 Amazon Redshift 클러스터를 생성할 수 있습니다. 반면에 더 높은 가용성을 위해 다른 가용 영역을 선택할 수도 있습니다. Amazon Redshift는 AWS 리전의 일부 가용 영역에서 사용 가능하지 않을 수 있습니다.

Amazon Redshift 클러스터를 프로비저닝할 수 있는 지원되는 AWS 리전 목록은 Amazon Web Services 일반 참조의 [Amazon Redshift엔드포인트](#)를 참조하세요.

클러스터 유지 관리

Amazon Redshift는 정기적으로 유지 관리를 실행하여 클러스터를 업그레이드합니다. 이러한 업데이트 도중에는 Amazon Redshift 클러스터를 정상적으로 사용할 수 없습니다. 클러스터 유지 관리 방법은 다양한 방식으로 제어할 수 있습니다. 예를 들어, 클러스터에 업데이트를 배포하는 시점을 제어할 수 있습니다. 또한 클러스터에서 항상 최근에 릴리스된 버전을 실행할지 또는 최근 릴리스 버전 바로 전에

릴리스된 버전을 실행할지를 선택할 수 있습니다. 마지막으로, 필수가 아닌 유지 관리 업데이트를 특정 기간 동안 연기할 수도 있습니다.

유지 관리 기간

Amazon Redshift는 AWS 리전마다 주중에 무작위로(월요일~일요일, 일요일 포함) 8시간의 주기를 두고 30분의 유지 관리 기간을 임의로 할당합니다.

기본 유지 관리 기간

다음은 기본 유지 관리 기간이 할당되는 각 AWS 리전의 시간 주기 목록입니다.

- 미국 동부(버지니아 북부) 리전: 03:00~11:00 UTC
- 미국 동부(오하이오) 리전: 03:00~11:00 UTC
- 미국 서부(캘리포니아 북부) 리전: 06:00~14:00 UTC
- 미국 서부(오레곤) 리전: 06:00~14:00 UTC
- 아프리카(케이프타운) 리전: 20:00~04:00 UTC
- 아시아 태평양(홍콩) 리전: 13:00~21:00 UTC
- 아시아 태평양(뭄바이) 리전: 16:30~00:30 UTC
- 아시아 태평양(자카르타) 리전: 15:00~23:00 UTC
- 아시아 태평양(말레이시아) 리전: 14:00~22:00 UTC
- 아시아 태평양(멜버른) 리전: 12:00~20:00 UTC
- 아시아 태평양(뭄바이) 리전: 16:30~00:30 UTC
- 아시아 태평양(뉴질랜드) 리전: 10:00~18:00 UTC
- 아시아 태평양(오사카) 리전: 13:00~21:00 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(타이베이) 리전: 14:00~22:00 UTC
- 아시아 태평양(태국) 리전: 15:00~23:00 UTC
- 아시아 태평양(도쿄) 리전: 13:00~21:00 UTC
- 캐나다(중부) 리전: 03:00~11:00 UTC

- 캐나다 서부(캘거리) 리전: 오전 4시~오후 12시(UTC)
- 중국(베이징) 리전: 13:00~21:00 UTC
- 중국(닝샤) 리전: 13:00~21:00 UTC
- 유럽(프랑크푸르트) 리전: 06:00~14:00 UTC
- 유럽(아일랜드) 리전: 22:00~06:00 UTC
- 유럽(런던) 리전: 22:00~06:00 UTC
- 유럽(밀라노) 리전: 21:00~05:00 UTC
- 유럽(파리) 리전: 23:00~07:00 UTC
- 유럽(스톡홀름) 리전: 23:00~07:00 UTC
- 유럽(취리히) 리전: 20:00~04:00 UTC
- 이스라엘(텔아비브) 리전: 20:00~04:00 UTC
- 멕시코(중부) 리전: 04:00~12:00 UTC
- 유럽(스페인) 리전: 21:00~05:00 UTC
- 중동(바레인) 리전: 13:00~21:00 UTC
- 중동(UAE) 리전: 18:00~02:00 UTC
- 남아메리카(상파울루) 리전: 19:00~03:00 UTC

유지 관리 이벤트가 지정된 주에 예약된 경우 할당된 30분의 유지 관리 기간 중에 시작됩니다. 유지 관리가 시작되면 Amazon Redshift가 진행 중인 모든 쿼리와 작업을 종료합니다. 계획된 유지 관리 중에 발생한 가동 중지 시간은 [Amazon Redshift 서비스 수준 계약](#) 내에서 월별 가동 중지 시간 또는 가용성 부족으로 간주되지 않습니다. 대부분 유지 관리는 30분 유지 관리 기간 내에 완료되지만 일부 유지 관리 작업은 기간이 완료된 후에도 계속될 수도 있습니다. 예정된 유지 관리 기간 중에 유지 관리 작업이 없으면 다음 유지 관리 기간에 이를 때까지 클러스터가 계속해서 정상적으로 실행됩니다.

예정된 유지 관리 기간은 프로그래밍 방식으로, 혹은 Amazon Redshift 콘솔에서 클러스터를 수정하여 변경할 수 있습니다. 유지 관리 탭에서 클러스터의 유지 관리 기간을 확인하고 해당 유지 관리 기간이 발생하는 날짜 및 시간을 설정할 수 있습니다.

클러스터가 유지 관리 기간 외에도 다시 시작될 수 있습니다. 이러한 현상이 발생하는 데에는 몇 가지 이유가 있습니다. 또 다른 일반적 이유는 클러스터에서 문제가 감지되어 클러스터를 정상 상태로 되돌리기 위한 유지 관리 작업이 수행되고 있기 때문입니다. 자세한 내용은 이 문제가 발생할 수 있는 이유에 대한 자세한 내용은 [Amazon Redshift 클러스터가 유지 관리 기간이 아닌데 재부팅되는 이유는 무엇입니까?](#) 문서를 참조하세요.

유지 관리 연기

클러스터의 유지 관리 기간을 다시 예약해야 하는 경우 유지 관리를 최대 60일까지 지연할 수 있습니다. 예를 들어, 클러스터의 유지 관리 기간이 수요일 8:30~9:00 UTC로 설정되어 있고 이 시간에 클러스터에 액세스해야 하는 경우 유지 관리를 이후 기간으로 연기할 수 있습니다.

유지 관리를 연기하더라도 Amazon Redshift는 여전히 하드웨어 업데이트 또는 기타 필수 보안 업데이트를 클러스터에 적용합니다. 이러한 업데이트 기간 중에는 클러스터를 사용할 수 없습니다.

다가오는 유지 관리 기간에 하드웨어 업데이트 또는 기타 필수 보안 업데이트가 예정되어 있는 경우 Amazon Redshift는 보류 중 카테고리로 사전 알리를 보냅니다. 보류 중 이벤트 알리에 대한 자세한 내용은 [Amazon Redshift 프로비저닝 클러스터 이벤트 알림](#) 섹션을 참조하세요.

Amazon Simple Notification Service(Amazon SNS)에서 이벤트 알리를 수신할 수도 있습니다. Amazon SNS에서 이벤트 알림 구독에 대한 자세한 내용은 [Amazon Redshift 클러스터 이벤트 알림 구독](#) 섹션을 참조하세요.

클러스터의 유지 관리를 연기하면 연기한 기간 이후 유지 관리 기간은 연기할 수 없습니다.

Note

시작한 후에는 유지 관리를 연기할 수 없습니다.

클러스터 유지 관리에 대한 자세한 내용은 다음 설명서를 참조하세요.

- [유지 관리 기간](#)
- [클러스터 작업](#)
- [클러스터 수정](#)

클러스터 유지 관리 트랙 선택

Amazon Redshift에서 새 클러스터 버전을 릴리스한 경우 유지 관리 기간 중에 클러스터가 업데이트됩니다. 클러스터를 최신 릴리스로 업데이트할지 이전 릴리스로 업데이트할지 여부를 제어할 수 있습니다.

트랙은 유지 관리 기간 중 적용되는 클러스터 버전을 제어합니다. Amazon Redshift에서 새 클러스터 버전을 릴리스한 경우 버전은 현재 트랙에 할당되고, 이전 버전은 후행 트랙에 할당됩니다.

클러스터 트랙에 대한 자세한 내용은 [Amazon Redshift 프로비저닝된 클러스터 및 서버리스 작업 그룹에 대한 트랙](#) 섹션을 참조하세요.

RA3 노드가 컴퓨팅과 스토리지를 분리하는 방법 이해

이 섹션에서는 RA3 노드 유형에 사용할 수 있는 태스크를 자세히 설명하며, 사용 사례 모음에 대한 적용 가능성을 보여주고 이전에 사용 가능한 노드 유형에 비해 이점을 자세히 설명합니다.

RA3 노드의 장점 및 가용성

RA3 노드는 다음과 같은 장점을 제공합니다.

- 스토리지 비용을 늘리지 않으면서 컴퓨팅 용량을 유연하게 증가시킵니다. 또한 컴퓨팅 용량을 과도하게 프로비저닝하지 않으면서 스토리지를 확장합니다.
- 핫 데이터에는 고성능 SSD를 사용하고 콜드 데이터에는 Amazon S3를 사용합니다. 따라서 사용 편의성, 비용 효과인 스토리지, 높은 쿼리 성능을 제공합니다.
- AWS Nitro System에 내장된 고대역폭 네트워킹을 사용하여 Amazon S3에 데이터를 오프로드하고 검색하는 데 걸리는 시간을 더욱 단축합니다.

다음과 같은 경우 RA3 노드 유형을 선택하는 것이 좋습니다.

- 스토리지와 별도로 컴퓨팅을 확장하고 요금을 결제할 수 있는 유연성이 필요합니다.
- 전체 데이터의 일부를 쿼리합니다.
- 데이터 볼륨이 빠르게 증가하고 있거나 빠르게 증가할 것으로 예상됩니다.
- 성능 필요에 따라서만 클러스터의 크기를 유연하게 조정하려고 합니다.

RA3 노드 유형을 사용하려면 해당 AWS 리전에서 RA3를 지원해야 합니다. 자세한 내용은 [AWS 리전의 RA3 노드 유형 가용성](#) 섹션을 참조하세요.

Important

ra3.xlplus 노드 유형은 클러스터 버전 1.0.21262 이상에서만 사용할 수 있습니다. Amazon Redshift 콘솔을 사용하여 기존 클러스터의 버전을 볼 수 있습니다. 자세한 내용은 [작업 그룹 또는 클러스터 버전 확인](#) 섹션을 참조하세요.

RA3 노드 유형으로 작업할 경우 새 Amazon Redshift 콘솔을 사용해야 합니다.

또한 트랙을 사용하는 Amazon Redshift 작업에서 RA3 노드 유형을 사용하려면 유지 관리 트랙 값을 RA3를 지원하는 클러스터 버전으로 설정해야 합니다. 트랙에 대한 자세한 내용은 [클러스터 유지 관리 트랙 선택](#) 섹션을 참조하세요.

단일 노드 RA3 노드 유형을 사용할 때 다음 사항을 고려하세요.

- 데이터 공유 생산자와 소비자가 지원됩니다.
- 노드 유형을 변경하려면 클래식 크기 조정만 지원됩니다. 탄력적 크기 조정 또는 스냅샷 복원을 사용한 노드 유형 변경은 지원되지 않습니다. 다음 시나리오가 지원됩니다.
 - 1노드 dc2.xlarge를 1노드 ra3.xplus로 또는 그 반대로 클래식 크기 조정.
 - 1노드 dc2.xlarge를 다중 노드 ra3.xplus로 또는 그 반대로 클래식 크기 조정.
 - 다중 노드 dc2.xlarge를 1노드 ra3.xplus로 또는 그 반대로 클래식 크기 조정.

Amazon Redshift 관리형 스토리지 작업

Amazon Redshift 관리형 스토리지를 사용하면 컴퓨팅 및 스토리지 용량을 별도로 조정할 수 있는 유연성을 확보하면서 Amazon Redshift에서 모든 데이터를 저장하고 처리할 수 있습니다. COPY 또는 INSERT 명령을 사용하여 데이터를 계속 수집합니다. 스토리지 계층 간에 성능을 최적화하고 자동 데이터 배치를 관리하기 위해 Amazon Redshift에서는 데이터 블록 온도, 데이터 블록 기간, 워크로드 패턴과 같은 최적화를 활용합니다. 필요한 경우 Amazon Redshift에서는 수동 작업 없이 스토리지의 크기를 Amazon S3로 자동 조정합니다.

스토리지 비용에 대한 자세한 내용은 [Amazon Redshift 요금](#)을 참조하세요.

RA3 노드 유형 관리

컴퓨팅을 스토리지와 분리하는 이점을 활용하려면 RA3 노드 유형으로 클러스터를 생성하거나 업그레이드할 수 있습니다. RA3 노드 유형을 사용하려면 Virtual Private Cloud(EC2-VPC)에서 클러스터를 생성합니다.

RA3 노드 유형을 사용하여 Amazon Redshift 클러스터의 노드 수를 변경하려면 다음 중 하나를 수행합니다.

- 탄력적인 크기 조정 작업을 사용하여 노드를 추가하거나 제거합니다. 일부 경우에 RA3 클러스터에서 노드를 제거하는 것은 탄력적 크기 조정에서 허용되지 않습니다. 예를 들면 2:1 노드 수 업그레이드를 통해 노드당 슬라이스 수가 32개로 설정되는 경우입니다. 자세한 내용은 [클러스터 크기 조정](#) 섹션을 참조하세요. 탄력적 크기 조정을 사용할 수 없는 경우 클래식 크기 조정을 사용합니다.

- 클래식 크기 조정 작업을 사용하여 노드를 추가하거나 제거합니다. 탄력적 크기 조정을 통해 사용할 수 없는 구성으로 크기를 조정하는 경우 이 옵션을 선택합니다. 탄력적 크기 조정은 클래식 크기 조정보다 빠릅니다. 자세한 내용은 [클러스터 크기 조정](#) 섹션을 참조하세요.

AWS 리전의 RA3 노드 유형 가용성

RA3 노드 유형은 다음 AWS 리전에서만 사용할 수 있습니다.

- 미국 동부(버지니아 북부) 리전(us-east-1)
- 미국 동부(오하이오) 리전(us-east-2)
- 미국 서부(캘리포니아 북부) 리전(us-west-1)
- 미국 서부(오레곤) 리전(us-west-2)
- 아프리카(케이프타운) 리전(af-south-1)
- 아시아 태평양(홍콩) 리전(ap-east-1)
- 아시아 태평양(하이데라바드) 리전(ap-south-2)
- 아시아 태평양(자카르타) 리전(ap-southeast-3)
- 아시아 태평양(말레이시아) 리전(ap-southeast-5)
- 아시아 태평양(멜버른) 리전(ap-southeast-4)
- 아시아 태평양(뭄바이) 리전(ap-south-1)
- 아시아 태평양(뉴질랜드) 리전(ap-southeast-6)
- 아시아 태평양(오사카) 리전(ap-northeast-3)
- 아시아 태평양(서울) 리전(ap-northeast-2)
- 아시아 태평양(싱가포르) 리전(ap-southeast-1)
- 아시아 태평양(시드니) 리전(ap-southeast-2)
- 아시아 태평양(타이베이) 리전(ap-east-2)
- 아시아 태평양(태국) 리전(ap-southeast-7)
- 아시아 태평양(도쿄) 리전(ap-northeast-1)
- 캐나다(중부) 리전(ca-central-1)
- 캐나다 서부(캘거리) 리전(ca-west-1)
- 중국(베이징) 리전(cn-north-1)
- 중국(닝샤) 리전(cn-northwest-1)

- 유럽(프랑크푸르트) 리전(eu-central-1)
- 유럽(취리히) 리전(eu-central-2)
- 유럽(아일랜드) 리전(eu-west-1)
- 유럽(런던) 리전(eu-west-2)
- 유럽(밀라노) 리전(eu-south-1)
- 유럽(스페인) 리전(eu-south-2)
- 유럽(파리) 리전(eu-west-3)
- 유럽(스톡홀름) 리전(eu-north-1)
- 이스라엘(텔아비브) 리전(il-central-1)
- 멕시코(중부) 리전(mx-central-1)
- 중동(바레인) 리전(me-south-1)
- 중동(UAE) 리전(me-central-1)
- 남아메리카(상파울루) 리전(sa-east-1)
- AWS GovCloud(미국 동부)(us-gov-east-1)
- AWS GovCloud(미국 서부)(us-gov-west-1)

RA3 노드 유형으로 업그레이드

기존 노드 유형을 RA3으로 업그레이드하려면 다음 옵션을 선택하여 노드 유형을 변경할 수 있습니다.

- 스냅샷에서 복원 – Amazon Redshift는 클러스터의 최신 스냅샷을 사용하고 이 스냅샷을 복원하여 새 RA3 클러스터를 생성합니다. 클러스터 생성이 완료되면(일반적으로 몇 분 이내) RA3 노드는 전체 프로덕션 워크로드를 실행할 준비가 됩니다. 컴퓨팅이 스토리지와 별개이므로, 대규모 네트워킹 대역폭 덕분에 핫 데이터는 빠른 속도로 로컬 캐시에 도입됩니다. 최신 DC2 스냅샷에서 복원하는 경우 RA3은 DC2 워크로드의 핫 블록 정보를 보존하고 로컬 캐시를 가장 핫한 블록으로 채웁니다. 자세한 내용은 [스냅샷에서 클러스터 복원](#) 섹션을 참조하세요.

애플리케이션 및 사용자에게 대해 동일한 엔드포인트를 유지하기 위해 새 RA3 클러스터의 이름을 원래 DC2 클러스터와 같은 이름으로 바꿀 수 있습니다. 클러스터의 이름을 바꾸려면 Amazon Redshift 콘솔 또는 ModifyCluster API 작업에서 클러스터를 수정합니다. 자세한 내용은 Amazon Redshift API Reference의 [ModifyCluster API 작업](#) 또는 [클러스터 이름 변경](#) 섹션을 참조하세요.

- 탄력적 크기 조정 - 탄력적 크기 조정을 사용하여 클러스터의 크기를 조정합니다. 탄력적 크기 조정을 사용하여 노드 유형을 변경하면 Amazon Redshift는 자동으로 스냅샷을 생성하고, 새 클러스터

를 생성하며, 이전 클러스터를 삭제하고, 새 클러스터의 이름을 바꿉니다. 탄력적 크기 조정 작업은 온디맨드로 실행하거나 나중에 실행하도록 예약할 수 있습니다. 탄력적 크기 조정을 사용하여 기존 DC2 노드 유형 클러스터를 RA3으로 빠르게 업그레이드할 수 있습니다. 자세한 내용은 [탄력적 크기 조정](#) 섹션을 참조하세요.

다음 표에서는 RA3 노드 유형으로 업그레이드할 때의 권장 사항을 보여줍니다. (이러한 권장 사항은 예약 노드에도 적용됩니다.)

이 테이블의 권장 사항은 시작 클러스터 노드 유형 및 크기이지만 워크로드의 컴퓨팅 요구 사항에 따라 다릅니다. 요구 사항을 더 잘 예측하려면 [테스트 드라이브](#)를 사용하여 잠재적 구성을 실행하는 개념 증명(POC)을 수행하는 것이 좋습니다. Redshift Serverless 대신 POC 데이터 웨어하우스용 클러스터를 프로비저닝하세요. 개념 증명 수행에 대한 자세한 내용은 Amazon Redshift Database 개발자 안내서에 [서 Amazon Redshift에 대한 개념 증명\(POC\) 수행](#)을 참조하세요.

기존 노드 유형	기존 노드 수	권장되는 새 노드 유형	업그레이드 작업
dc2.8xlarge	2~15	ra3.4xlarge	dc2.8xlarge 노드 1개마다 ra3.4xlarge 노드 2개로 시작합니다 ¹ .
dc2.8xlarge	16~128	ra3.16xlarge	dc2.8xlarge 노드 2개마다 ra3.16xlarge 노드 1개로 시작합니다 ¹ .
dc2.large	1~4	ra3.large	dc2.large 노드 1개마다 ra3.large 노드 1개로 시작합니다 ¹ . dc2.large 노드 2개마다 ra3.large 노드 2개로 시작합니다 ¹ .

기존 노드 유형	기존 노드 수	권장되는 새 노드 유형	업그레이드 작업	
			dc2.large 노드 3개마다 ra3.large 노드 3개로 시작합니다 ¹ .	
			dc2.large 노드 4개마다 ra3.large 노드 3개로 시작합니다 ¹ .	
dc2.large	5~15	ra3.xlplus	dc2.large 노드 8개마다 ra3.xlplus 노드 3개로 시작합니다 ¹ .	
dc2.large	16~32	ra3.4xlarge	dc2.large 노드 8개마다 ra3.4xlarge 노드 1개로 시작합니다 ^{1,2} .	

¹워크로드 요구 사항에 따라 추가 노드가 필요할 수 있습니다. 필요한 쿼리 성능의 컴퓨팅 요구 사항을 기반으로 노드를 추가하거나 제거합니다.

²dc2.large 노드 유형을 포함하는 클러스터는 32개 노드로 제한됩니다.

일부 RA3 노드 유형의 최소 노드 수는 2개 노드입니다. RA3 클러스터를 생성할 때는 다음을 고려해 보세요.

RA3 노드에서 지원하는 네트워킹 기능

RA3 노드는 다른 노드 유형에서 사용할 수 없는 네트워킹 기능 모음을 지원합니다. 이 섹션에서는 각 기능에 대한 간략한 설명과 추가 설명서 링크를 제공합니다.

- 프로비저닝된 클러스터 VPC 엔드포인트 - RA3 클러스터를 생성하거나 복원할 때 Amazon Redshift는 5431~5455 또는 8191~8215 범위 내의 포트를 사용합니다. 클러스터가 이러한 범위 중 하나의 포트로 설정되면 Amazon Redshift는 클러스터의 AWS 계정에 VPC 엔드포인트를 자동으로 생성하고

여기에 프라이빗 IP 주소를 연결합니다. 클러스터에 퍼블릭 액세스가 가능하도록 설정하면 Redshift는 AWS 계정에 탄력적 IP 주소를 생성하여 VPC 엔드포인트에 연결합니다. 자세한 내용은 [Amazon Redshift 클러스터에 대한 보안 그룹 통신 설정 구성 또는 Amazon Redshift Serverless 작업 그룹](#)을 참조하세요.

- 단일 서브넷 RA3 클러스터 - 단일 서브넷으로 RA3 클러스터를 만들 수 있지만 재해 복구 기능은 사용할 수 없습니다. 서브넷에 가용 영역(AZ)이 여러 개가 아닌 경우 클러스터 재배포치를 사용하도록 설정하면 예외가 발생합니다.
- 다중 서브넷 RA3 클러스터 및 서브넷 그룹 - Virtual Private Cloud(VPC)에서 클러스터를 프로비저닝할 때 서브넷 그룹을 생성하여 서브넷이 여러 개인 RA3 클러스터를 만들 수 있습니다. 클러스터 서브넷 그룹을 사용하면 VPC에 서브넷 세트를 지정할 수 있으며 Amazon Redshift는 그 중 하나에 클러스터를 생성합니다. 서브넷 그룹을 생성한 후 이전에 추가한 서브넷을 제거하거나 서브넷을 더 추가할 수 있습니다. 자세한 내용은 [Amazon Redshift 클러스터 서브넷 그룹](#)을 참조하세요.
- 계정 간 또는 VPC 간 엔드포인트 액세스 - Redshift 관리형 VPC 엔드포인트를 설정하여 프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹에 액세스할 수 있습니다. 예를 들면 클러스터나 작업 그룹을 포함하는 VPC와 클라이언트 도구를 실행하는 VPC 간의 프라이빗 연결로 설정할 수 있습니다. 이렇게 하면 퍼블릭 IP 주소를 사용하거나 인터넷을 통해 트래픽을 라우팅하지 않고도 데이터 웨어하우스에 액세스할 수 있습니다. 자세한 내용은 [Redshift 관리형 VPC 엔드포인트 작업](#)을 참조하세요.
- 클러스터 재배포치 - 서비스가 중단되는 경우 데이터 손실 없이 클러스터를 다른 가용 영역(AZ)으로 이동할 수 있습니다. 콘솔에서 이 기능을 활성화합니다. 자세한 내용은 [클러스터 재배포치](#) 섹션을 참조하세요.
- 사용자 지정 도메인 - 이름사용자 지정 URL이라고도 하는 사용자 지정 도메인 이름을 Amazon Redshift 클러스터에 만들 수 있습니다. SQL 클라이언트 연결을 클러스터 엔드포인트로 라우팅하는 읽기 쉬운 DNS 레코드입니다. 자세한 내용은 [클라이언트 연결의 사용자 지정 도메인 이름](#) 섹션을 참조하세요.

클러스터 작업

클러스터를 생성한 후 클러스터 작업을 수행하여 성능을 최적화하고, 비용을 제어하며, 고가용성을 보장할 수 있습니다. 클러스터 작업을 통해 데이터 웨어하우징 요구 사항의 변화에 따라 클러스터를 크기 조정, 일시 중지, 재개 또는 재생성할 수 있습니다.

일반적인 사용 사례로는 피크 워크로드에 맞게 컴퓨팅 용량 규모를 조정하고, 비활성 기간 동안 클러스터를 일시 중지하여 비용을 절감하고, 재해 복구를 위해 구성이 다르거나 가용 영역이 다른 클러스터를 재생성하는 작업 등이 있습니다. 다음 섹션에서는 Amazon Redshift 환경을 효과적으로 관리하기 위한 여러 클러스터 작업 수행의 세부 정보를 다룹니다.

클러스터 생성

Amazon Redshift를 사용하면 프로비저닝된 클러스터를 생성하여 새 데이터 웨어하우스를 시작할 수 있습니다. 프로비저닝된 클러스터는 노드라는 컴퓨팅 리소스의 모음으로, 노드는 단일 대량 병렬 처리 (MPP) 시스템을 구성합니다.

클러스터를 생성하기 전에 [Amazon Redshift 프로비저닝 클러스터](#) 및 [Amazon Redshift의 클러스터 및 노드](#) 섹션을 읽어 보세요.

클러스터 생성

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다.
3. 클러스터를 생성하려면 클러스터 생성(Create cluster)을 선택합니다.
4. 콘솔 페이지의 지침에 따라 클러스터 구성(Cluster configuration)에 속성을 입력합니다.

다음 단계에서는 RA3 노드 유형을 지원하는 AWS 리전에서 실행되는 Amazon Redshift 콘솔에 대해 설명합니다. RA3 노드 유형을 지원하는 AWS 리전 목록은 Amazon Redshift 관리 안내서의 [RA3 노드 유형 개요](#)를 참조하세요.

클러스터 크기를 얼마로 조정해야 할지 모르는 경우 선택 도움말(Help me choose)을 선택합니다. 이렇게 하면 데이터 웨어하우스에 저장하려는 데이터의 크기 및 쿼리 특성에 대한 질문을 하는 크기 조정 계산기가 시작됩니다. 클러스터의 필요한 크기(즉, 노드 유형 및 노드 수)를 알고 있는 경우 [직접 선택(I'll choose)]을 선택합니다. 그런 다음 개념 증명을 위해 클러스터 크기를 조정할 [노드 유형(Node type)]과 [노드(Nodes)] 수를 선택합니다.

Note

조직에 자격이 있으며 Amazon Redshift Serverless를 사용할 수 없는 AWS 리전에 클러스터가 생성되고 있는 경우, Amazon Redshift 무료 평가판 프로그램에서 클러스터를 생성하는 것이 가능할 수 있습니다. 이 클러스터를 어떤 용도로 사용할 계획입니까?라는 질문에 대해 프로덕션 또는 무료 평가판을 선택하여 답합니다. 무료 평가판을 선택하면 dc2.large 노드 유형으로 구성을 생성합니다. 무료 평가판 선택에 대한 자세한 내용은 [Amazon Redshift 무료 평가판](#)을 참조하세요. Amazon Redshift Serverless를 사용할 수 있

는 AWS 리전 목록은 Amazon Web Services 일반 참조의 [Redshift Serverless API](#)에 나열된 엔드포인트를 참조하세요.

5. 데이터베이스 구성 섹션에서 관리자 사용자 이름 값을 지정합니다. 관리자 암호의 경우 다음 옵션 중에서 선택할 수 있습니다.
 - 암호 생성 - Amazon Redshift에서 생성된 암호를 사용합니다.
 - 관리자 암호 수동 추가 - 내 암호를 사용합니다.
 - AWS Secrets Manager에서 관리자 보안 인증 정보 관리 - Amazon Redshift는 관리자 암호를 생성하고 관리하는 데 AWS Secrets Manager를 사용합니다. AWS Secrets Manager를 사용하여 암호의 보안 암호를 생성하고 관리하면 수수료가 발생합니다. AWS Secrets Manager 요금에 대한 자세한 내용은 [AWS Secrets Manager 요금](#)을 참조하세요.
6. (선택 사항) 콘솔 페이지의 지침에 따라 클러스터 권한(Cluster permissions)에 속성을 입력합니다. 클러스터가 다른 AWS 서비스에 액세스해야 하는 경우(예: Amazon S3에서 데이터 로드) 클러스터 권한을 제공합니다.
7. 클러스터를 생성하려면 클러스터 생성(Create cluster)을 선택합니다. 클러스터를 사용할 준비가 끝나려면 몇 분이 걸릴 수 있습니다.

추가 구성

클러스터를 생성할 때 추가 속성을 지정하여 사용자 지정할 수 있습니다. 다음 목록에서 이러한 속성 중 일부에 대한 자세한 내용을 찾을 수 있습니다.

IP 주소 유형

클러스터의 IP 주소 유형을 선택합니다. 리소스가 IPv4 주소 지정 프로토콜을 통해서만 통신하도록 선택하거나 리소스가 IPv4와 IPv6를 통해 통신할 수 있는 이중 스택 모드를 선택할 수 있습니다. 이 기능은 AWS GovCloud(미국 동부) 및 AWS GovCloud(미국 서부) 리전에서만 사용할 수 있습니다. AWS 리전에 대한 자세한 내용은 [리전 및 가용 영역](#)을 참조하세요.

Virtual Private Cloud(VPC)

클러스터 서브넷 그룹이 있는 VPC를 선택합니다. 클러스터가 생성된 후에는 클러스터 서브넷 그룹을 변경할 수 없습니다.

파라미터 그룹 수(Parameter groups)

클러스터와 연결할 클러스터 파라미터 그룹을 선택합니다. 선택하지 않으면 클러스터가 기본 파라미터 그룹을 사용합니다.

암호화(Encryption)

클러스터의 모든 데이터와 스냅샷에 대한 암호화 여부를 선택합니다. 기본 설정인 없음(None)을 그대로 사용하면 암호화가 활성화되지 않습니다. 암호화를 활성화하려면 먼저 AWS Key Management Service(AWS KMS) 또는 하드웨어 보안 모듈(HSM) 중에서 무엇을 사용할지 선택한 후 관련 설정을 구성해야 합니다. Amazon Redshift의 암호화에 대한 자세한 내용은 [Amazon Redshift 데이터베이스 암호화](#) 섹션을 참조하세요.

- KMS

암호화를 사용하고 AWS KMS로 암호화 키를 관리하려면 AWS Key Management Service 사용(AWS KMS)을 선택합니다. 또한 사용할 키를 선택합니다. 기본 키는 현재 계정의 키 또는 다른 계정의 키를 선택할 수 있습니다.

 Note

다른 AWS 계정의 키를 사용하려면 사용할 키의 Amazon 리소스 이름(ARN)을 입력합니다. 이때 해당 키에 대한 사용 권한이 있어야 합니다. AWS KMS의 키 액세스에 대한 자세한 내용은 AWS Key Management Service Developer Guide의 [Controlling Access to Your Keys](#)를 참조하세요.

Amazon Redshift에서 AWS KMS 암호화 키 사용에 대한 자세한 내용은 [AWS KMS로 암호화](#) 섹션을 참조하세요.

- HSM

암호화를 활성화한 후 하드웨어 보안 모듈(HSM)을 사용해 암호화 키를 관리하려면 HSM을 선택합니다.

HSM을 선택하는 경우에는 HSM 연결(HSM Connection) 및 HSM 클라이언트 인증서(HSM Client Certificate)에서 값을 선택합니다. 이 두 값은 Amazon Redshift와 HSM에서 클러스터 키를 믿고 전달할 수 있는 연결을 형성하는 데 필요합니다. 또한 클러스터를 시작하기 전에 Amazon Redshift에서 HSM 연결과 클라이언트 인증서를 설정해야 합니다. HSM 연결 및 클라이언트 인증서의 설정에 대한 자세한 내용은 [하드웨어 보안 모듈을 사용한 암호화](#) 섹션을 참조하세요.

유지 관리 트랙(Maintenance track)

사용된 클러스터 버전이 현재(Current), 검사(Trailing) 또는 미리 보기(Preview) 트랙인지 여부를 선택할 수 있습니다.

모니터링(Monitoring)

CloudWatch 경보 생성 여부를 선택할 수 있습니다.

교차 리전 스냅샷 구성(Configure cross-region snapshot)

리전 간 스냅샷을 활성화할지 여부를 선택할 수 있습니다.

자동 스냅샷 보존 기간

이러한 스냅샷을 보존할 일수를 35일 이내의 범위에서 선택할 수 있습니다. 노드 유형이 DC2인 경우 0일을 선택하여 자동 스냅샷을 생성하지 않을 수 있습니다.

수동 스냅샷 보존 기간

이러한 스냅샷을 보존할 기간으로 일수 또는 Indefinitely를 선택할 수 있습니다.

자동 최적화를 위한 추가 컴퓨팅 리소스

사용량이 많은 기간에도 추가 컴퓨팅 리소스를 할당하여 자동 최적화를 수행할지 여부를 선택할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [자동 데이터베이스 최적화를 위한 추가 컴퓨팅 리소스 할당](#)을 참조하세요.

디스크 공간 경보 생성

디스크 공간 사용량을 모니터링하고, 디스크 공간이 클러스터의 지정된 임계값을 초과할 때 알림을 받도록 경보를 설정할 수 있습니다. 디스크 공간 사용량 경보를 생성하면 스토리지 용량을 사전에 관리하고, 쿼리 실패 또는 데이터 모으기 오류 등 디스크 공간 부족으로 인한 문제를 예방할 수 있습니다. 다음 절차는 디스크 공간 사용 경보를 생성하는 과정을 안내합니다.

클러스터에 대한 디스크 공간 사용 경보를 생성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 경보(Alarms)를 선택합니다.
3. 작업(Actions)에서 경보 생성(Create alarm)을 선택합니다. 경보 생성(Create alarm) 페이지가 표시됩니다.
4. 페이지의 지시 사항을 따릅니다.
5. 경보 생성(Create alarm)을 선택하세요.

클러스터 보기

클러스터를 보면 클러스터의 구성, 상태 및 성능 지표를 모니터링하고 관리할 수 있습니다. 클러스터 세부 정보를 보면 리소스 사용률, 쿼리 실행 시간 및 시스템 상태에 대한 인사이트를 얻을 수 있습니다. 다음 절차는 클러스터 정보에 액세스하는 방법을 보여줍니다.

클러스터를 보려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다. 클러스터가 없는 경우 클러스터 생성(Create cluster)을 선택하여 클러스터를 생성합니다.
3. 클러스터에 대한 자세한 내용을 보려면 목록에서 클러스터 이름을 선택합니다.

클러스터 수정

클러스터를 수정할 때 다음 옵션은 변경 사항이 바로 적용됩니다.

- VPC 보안 그룹(VPC security groups)
- 공개적으로 액세스할 수 있음(Publicly accessible)
- 관리자 사용자 암호(Admin user password)
- HSM 연결(HSM Connection)
- HSM 클라이언트 인증서(HSM Client Certificate)
- 유지 관리 세부 정보
- 스냅샷 기본 설정(Snapshot preferences)

다음 옵션의 변경은 클러스터를 다시 시작해야만 적용됩니다.

- 클러스터 식별자(Cluster identifier)

[클러스터 식별자(Cluster identifier)]를 변경하면 Amazon Redshift가 클러스터를 자동으로 다시 시작합니다.

- 향상된 VPC 라우팅(Enhanced VPC Routing)

[Enhanced VPC Routing]을 변경하면 Amazon Redshift가 클러스터를 자동으로 다시 시작합니다.

- 클러스터 파라미터 그룹(Cluster parameter group)
- IP 주소 유형

이 기능은 AWS GovCloud(미국 동부) 및 AWS GovCloud(미국 서부) 리전에서만 사용할 수 있습니다. AWS 리전에 대한 자세한 내용은 [리전 및 가용 영역](#)을 참조하세요.

자동 스냅샷의 보관 기간을 줄일 경우 기존 자동 스냅샷 중에서 새로운 보관 기간이 경과한 스냅샷은 삭제됩니다. 자세한 내용은 [Amazon Redshift 스냅샷 및 백업](#) 섹션을 참조하세요.

클러스터 속성에 대한 자세한 내용은 [추가 구성](#) 섹션을 참조하세요.

클러스터를 수정하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다.
3. 수정할 클러스터를 선택합니다.
4. 편집을 선택합니다. [클러스터 편집(Edit cluster)] 페이지가 표시됩니다.
5. 클러스터 속성을 업데이트합니다. 수정할 수 있는 일부 속성은 다음과 같습니다.
 - 클러스터 식별자
 - 스냅샷 보관
 - 클러스터 재배포

네트워크 및 보안, 유지 관리 및 데이터베이스 구성에 대한 설정 편집을 위해 콘솔은 해당 클러스터 세부 정보 탭에 대한 링크를 제공합니다.

6. 변경 사항 저장을 선택합니다.

클러스터 크기 조정

데이터 웨어하우징 용량 및 성능 요건이 변화를 거듭하면서 클러스터의 크기 조정을 통해 Amazon Redshift의 컴퓨팅 및 스토리지 옵션을 최대한 활용할 수 있게 되었습니다.

클러스터 크기를 조정할 때는 노드 수 또는 노드 유형을 현재 클러스터 구성과 다르게 지정합니다. 클러스터가 크기 조정을 진행하는 동안 클러스터에 대한 쓰기 또는 읽기/쓰기 쿼리를 실행할 수 없으며, 오직 읽기 쿼리만 가능합니다.

다른 방법을 사용한 클러스터 크기 조정 프로세스를 포함하여 클러스터 크기 조정에 대한 자세한 내용은 [클러스터 크기 조정](#) 섹션을 참조하세요.

클러스터 크기 조정 방법

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다.
3. 크기를 조정할 클러스터를 선택합니다.
4. 작업(Actions)에서 크기 조정(Resize)을 선택합니다. 클러스터 크기 조정(Resize cluster) 페이지가 표시됩니다.
5. 페이지의 지시 사항을 따릅니다. 이제 특정 시간에 한 번 클러스터의 크기를 조정하거나, 일정에 따라 클러스터의 크기를 늘리거나 줄일 수 있습니다.
6. 선택 사항에 따라 지금 크기 조정(Resize now) 또는 크기 조정 예약(Schedule resize)을 선택합니다.

예약 노드가 있는 경우 RA3 예약 노드로 업그레이드할 수 있습니다. 콘솔을 사용하여 스냅샷에서 복원하거나 탄력적 크기 조정을 수행할 때 이 작업을 수행할 수 있습니다. 콘솔을 사용하여 이 프로세스를 안내할 수 있습니다. RA3 노드로 업그레이드에 대한 자세한 내용은 [RA3 노드 유형으로 업그레이드](#)를 참조하세요.

DC2.large 노드 유형에서 RA3.large 노드 유형으로 업그레이드하기 위해 크기 조정 작업을 수행하면 Amazon Redshift가 자동으로 인터리브 정렬 키를 복합 정렬 키로 변환합니다. 이 변환을 통해 동시성 조정 기능에 액세스할 수 있지만, 이는 인터리브 정렬 키가 있는 테이블에 대한 쿼리를 지원하지 않습니다. 이 자동 변환은 RA3 기능과의 호환성을 보장하지만 기존 쿼리 성능 패턴에 영향을 미칠 수 있습니다.

RA3 노드로 업그레이드한 뒤 인터리브 정렬 키를 유지하려는 경우 크기 조정 작업이 완료된 후 원하는 정렬 키 구성으로 테이블을 다시 생성할 수 있습니다. 그러나 이 옵션을 선택하면 이러한 테이블에 동시성 조정을 사용할 수 없습니다.

크기 조정 작업에는 두 가지 유형이 있습니다.

- 탄력적 크기 조정 - 클러스터에 노드를 추가하거나 클러스터에서 노드를 제거할 수 있습니다. 또한 DC2 노드를 RA3 노드로 변경하는 등, 노드 유형을 변경할 수 있습니다. 탄력적 크기 조정은 일반적으로 빠르게 완료되며 평균 10분이 소요됩니다. 따라서 이를 첫 번째 옵션으로 사용하는 것이 좋습니다. 탄력적 크기 조정을 수행하면, 각 노드에 메모리와 디스크 공간이 할당되어 있는 파티션인 데이터 슬라이스가 재배포됩니다. 탄력적 크기 조정은 다음과 같은 경우에 적합합니다.

- 기존 클러스터에서 노드를 추가하거나 줄이지만 노드 유형은 변경하지 않음 - 이를 일반적으로 인플레이스 크기 조정이라고 합니다. 이 유형의 크기 조정을 수행하면 실행 중인 일부 쿼리는 성공적으로 완료되지만 다른 일부 쿼리는 작업의 일부로서 삭제될 수 있습니다.
- 클러스터의 노드 유형 변경 - 노드 유형을 변경하면 스냅샷이 생성되고 데이터가 소스 클러스터에서 새 노드 유형으로 구성된 클러스터로 재배포됩니다. 완료되면 실행 중인 쿼리가 삭제됩니다. 인플레이스 크기 조정과 마찬가지로 빠르게 완료됩니다.
- 클래식 크기 조정 - 노드 유형, 노드 수 또는 두 가지 모두를 탄력적 크기 조정과 유사한 방식으로 변경할 수 있습니다. 클래식 크기 조정은 완료하는 데 시간이 더 걸리지만 마이그레이션할 노드 수 또는 유형의 변경 폭이 탄력적 크기 조정의 범위를 벗어나는 경우에 유용할 수 있습니다. 예를 들어 노드 수가 매우 큰 폭으로 변경되는 경우에 적용될 수 있습니다.

주제

- [탄력적 크기 조정](#)
- [클래식 크기 조정](#)

탄력적 크기 조정

동일한 유형의 노드를 추가하거나 제거할 경우, 탄력적 크기 조정 작업의 단계는 다음과 같습니다.

1. 탄력적 크기 조정 시 클러스터 스냅샷이 생성됩니다. 백업 없음 테이블은 DC2 노드에만 지원됩니다. 다른 모든 유형의 클러스터의 경우 백업 없음 테이블이 스냅샷에 포함됩니다. 자세한 내용은 [스냅샷에서 테이블 제외](#) 섹션을 참조하세요. 자동 스냅샷을 비활성화해 클러스터에 최신 스냅샷이 없는 경우 백업 작업에 시간이 오래 걸릴 수 있습니다. (크기 조정 작업을 시작하기 전에 시간을 최소화하기 위해 자동화된 스냅샷을 활성화하거나 크기 조정을 시작하기 전에 수동 스냅샷을 생성하는 것이 좋습니다.) 탄력적 크기 조정을 시작하고 스냅샷 작업이 진행 중인 경우 스냅샷 작업이 몇 분 내에 완료되지 않으면 크기 조정이 실패할 수 있습니다. 자세한 내용은 [Amazon Redshift 스냅샷 및 백업](#) 섹션을 참조하세요.
2. 이 작업은 클러스터 메타데이터를 마이그레이션합니다. 클러스터를 몇 분간 사용할 수 없습니다. 대부분의 쿼리가 일시적으로 중지되고 연결은 열린 상태로 유지됩니다. 하지만 일부 쿼리를 삭제하지 못할 수도 있습니다. 이 단계는 짧습니다.
3. 세션 연결이 복구되고 쿼리가 다시 시작됩니다.
4. 탄력적 크기 조정은 백그라운드에서 노드 슬라이스로 데이터를 다시 분산합니다. 클러스터는 읽기 및 쓰기 작업에 사용할 수 있지만 일부 쿼리의 경우 실행 시간이 길어질 수 있습니다.
5. 작업이 완료되면 Amazon Redshift에서 이벤트 알림을 보냅니다.

탄력적 크기 조정을 사용하여 노드 유형을 변경하는 경우 동일한 유형의 노드를 추가하거나 제거할 때와 유사하게 작동합니다. 먼저 스냅샷이 생성됩니다. 새로운 대상 클러스터는 스냅샷의 최신 데이터를 사용하여 프로비저닝되고, 데이터는 백그라운드에서 새 클러스터로 전송됩니다. 이 기간 동안 데이터는 읽기 전용입니다. 크기 조정이 거의 끝나가면 Amazon Redshift가 새 클러스터를 가리키도록 엔드포인트를 업데이트하고, 소스 클러스터의 모든 연결이 종료됩니다.

탄력적 크기 조정이 실패할 가능성은 거의 없습니다. 그러나 장애가 발생한 경우 대부분은 수동 개입 없이 자동으로 롤백이 이뤄집니다.

예약 노드(예: DC2 예약 노드)가 있는 경우 크기 조정을 수행할 때 RA3 예약 노드로 업그레이드할 수 있습니다. 탄력적 크기 조정을 수행할 때 또는 콘솔을 사용하여 스냅샷에서 복원할 때 이 작업을 수행할 수 있습니다. 콘솔 가이드에서는 이 프로세스를 안내합니다. RA3 노드로 업그레이드에 대한 자세한 내용은 [RA3 노드 유형으로 업그레이드](#)를 참조하세요.

탄력적 크기 조정은 테이블을 정렬하지 않고 디스크 스페이스를 회수하지 않기 때문에 vacuum 작업을 대신할 수 없습니다. 자세한 내용은 [테이블 Vacuum](#)을 참조하세요.

탄력적 크기 조정에는 다음과 같은 제약이 적용됩니다.

- 탄력적 크기 조정 및 데이터 공유 클러스터 - 데이터 공유의 생산자인 클러스터에서 노드를 추가하거나 뺄 때 Amazon Redshift가 클러스터 메타데이터를 마이그레이션하는 동안에는 소비자로부터 연결할 수 없습니다. 마찬가지로 탄력적 크기 조정을 수행하고 새 노드 유형을 선택하면 연결이 끊어지고 새 대상 클러스터로 전송되는 동안에는 데이터 공유를 사용할 수 없습니다. 두 가지 유형의 탄력적 크기 조정 모두에서 생산자는 몇 분 동안 사용할 수 없습니다.
- 공유 스냅샷에서 데이터 전송 - 공유 스냅샷에서 데이터를 전송하는 클러스터에서 탄력적 크기 조정을 실행하려면 클러스터에 대해 하나 이상의 백업을 사용할 수 있어야 합니다. Amazon Redshift 콘솔 스냅샷 목록, describe-cluster-snapshots CLI 명령 또는 DescribeClusterSnapshots API 작업에서 백업을 볼 수 있습니다.
- 플랫폼 제한 사항 - 탄력적 크기 조정은 EC2-VPC 플랫폼을 사용하는 클러스터에만 사용할 수 있습니다. 자세한 내용은 [EC2를 사용하여 클러스터 생성](#) 섹션을 참조하세요.
- 스토리지 고려 사항 - 새 노드 구성에 기존 데이터를 위한 충분한 스토리지가 있는지 확인합니다. 노드를 추가하거나 구성을 변경해야 할 수 있습니다.
- 소스와 대상 클러스터 크기 비교 - 탄력적 크기 조정으로 크기를 조정할 수 있는 노드 수 및 노드 유형은 소스 클러스터의 노드 수와 크기 조정된 클러스터에 대해 선택한 노드 유형에 따라 결정됩니다. 사용 가능한 구성을 확인하려면 콘솔을 사용합니다. 또는 action-type resize-cluster 옵션과 함께 describe-node-configuration-options AWS CLI 명령을 사용할 수 있습니다. Amazon Redshift 콘솔을 사용한 크기 조정에 대한 자세한 내용은 [클러스터 크기 조정](#) 섹션을 참조하세요.

다음 예제 CLI 명령은 사용 가능한 구성 옵션을 설명합니다. 이 예에서 mycluster라는 클러스터는 dc2.large 8노드 클러스터입니다.

```
aws redshift describe-node-configuration-options --cluster-identifier mycluster --region eu-west-1 --action-type resize-cluster
```

이 명령은 각 옵션의 권장 노드 유형, 노드 수, 디스크 사용률이 포함된 옵션 목록을 반환합니다. 반환되는 구성은 특정 입력 클러스터에 따라 다를 수 있습니다. resize-cluster CLI 명령의 옵션을 지정할 때 반환되는 구성 중 하나를 선택할 수 있습니다.

- 추가 노드의 한도 - 탄력적 크기 조정에는 클러스터에 추가할 수 있는 노드에 대한 제한이 있습니다. 예를 들어 dc2 클러스터는 노드 수의 최대 2배까지 탄력적 크기 조정을 지원합니다. 설명을 위해 4노드 dc2.8xlarge 클러스터에 노드를 하나 추가하여 5노드 클러스터로 만들거나 8개가 될 때까지 노드를 더 추가할 수 있습니다.

Note

증가 및 감소 한도는 원래 노드 유형과 원래 클러스터의 노드 수 또는 마지막 클래식 크기 조정을 기반으로 합니다. 탄력적 크기 조정이 증가 및 감소 한도를 초과하는 경우 클래식 크기 조정을 사용하세요.

일부 ra3 노드 유형의 경우 노드 수를 기존 수의 최대 4배까지 늘릴 수 있습니다. 특히 클러스터가 ra3.4xlarge 또는 ra3.16xlarge 노드로 구성되어 있다고 가정합니다. 그런 다음 탄력적 크기 조정을 사용하여 8노드 클러스터의 노드 수를 32개로 늘릴 수 있습니다. 또는 한도 미만의 값을 선택할 수 있습니다. (클러스터를 4배씩 확장하는 기능은 소스 클러스터 크기에 따라 달라진다는 점에 유의하세요.) 클러스터에 ra3.xplus 노드가 있는 경우 제한은 2배입니다.

모든 ra3 노드 유형은 기존 수의 1/4로 노드 수 감소를 지원합니다. 예를 들어 ra3.4xlarge 노드가 있는 클러스터의 크기를 12개 노드에서 3개 또는 최솟값 이상으로 줄일 수 있습니다.

다음 표에는 탄력적 크기 조정을 지원하는 각 노드 유형에 대한 증가 및 감소 제한이 나열되어 있습니다.

원래 노드 유형	증가 한도	감소 한도
ra3.16xlarge	4x(예: 노드 4개에서 16개로)	기존 수의 1/4까지(예: 노드 16개에서 4개로)
ra3.4xlarge	4x	기존 수의 1/4까지
ra3.xlplus	2x(예: 노드 4개에서 8개로)	기존 수의 1/4까지
ra3.large	2x	기존 수의 1/2까지
dc2.8xlarge	2x	기존 수의 1/2까지(예: 노드 16개에서 8개로)
dc2.large	2x	기존 수의 1/2까지

Note

RA3 클러스터의 크기를 조정할 때 레거시 노드 유형 선택 - RA3 노드가 있는 클러스터에서 DC2 등의 다른 노드 유형으로 크기를 조정하려고 하면 콘솔에 유효성 검사 경고 메시지가 나타나고 크기 조정 작업이 완료되지 않습니다. 이는 레거시 노드 유형으로의 크기 조정이 지원되지 않기 때문에 발생합니다. 이는 고객이 더 이상 사용되지 않거나 곧 사용되지 않을 노드 유형으로 크기를 조정하는 것을 방지하기 위한 것입니다. 이는 탄력적 크기 조정과 클래식 크기 조정 모두에 적용됩니다.

클래식 크기 조정

클래식 크기 조정은 클러스터 크기 또는 노드 유형의 변경 폭이 탄력적 크기 조정에서 지원되지 않는 경우를 처리합니다. 클래식 크기 조정을 수행하면 Amazon Redshift는 대상 클러스터를 생성하고 원본 클러스터에서 대상 클러스터로 데이터와 메타데이터를 마이그레이션합니다.

RA3로 클래식 크기 조정을 수행하면 가용성이 향상될 수 있습니다.

대상 노드 유형이 RA3일 때 클래식 크기 조정 기능이 향상되었습니다. 원본 및 대상 클러스터 간의 백업 및 복원 작업을 사용하여 이를 수행합니다. 크기 조정이 시작되면 원본 클러스터가 재시작되며 몇

분간 사용할 수 없습니다. 그 후에는 백그라운드에서 크기 조정 작업을 하는 동안 클러스터의 읽기 및 쓰기 작업을 사용할 수 있습니다.

클러스터 검사

RA3으로 클러스터의 클래식 크기 조정을 수행할 때 최상의 성능과 결과를 얻으려면 이 체크리스트를 완료하세요. 체크리스트를 따르지 않으면 읽기 및 쓰기 작업 수행 기능과 같은 RA3 노드를 사용한 클래식 크기 조정의 일부 이점을 얻지 못할 수 있습니다.

1. 데이터 크기는 2페타바이트 미만이어야 합니다. 1페타바이트는 1,000테라바이트와 같습니다. 데이터 크기를 확인하려면 스냅샷을 만들고 크기를 확인합니다. 또한 다음 쿼리를 실행하여 크기를 확인할 수 있습니다.

```
SELECT
sum(case when lower(diststyle) like ('%key%') then size else 0 end) distkey_blocks,
sum(size) as total_blocks,
((distkey_blocks/(total_blocks*1.00)))*100 as Blocks_need_redist
FROM svv_table_info;
```

svv_table_info 테이블은 슈퍼 사용자에게만 표시됩니다.

2. 클래식 크기 조정을 시작하기 전에 10시간 이상 경과하지 않은 수동 스냅샷이 있는지 확인하세요. 그렇지 않은 경우 스냅샷을 만듭니다.
3. 클래식 크기 조정을 수행하는 데 사용된 스냅샷은 테이블 복원이나 다른 용도로는 사용할 수 없습니다.
4. 클러스터는 VPC에 속해야 합니다.

RA3으로 클래식 크기 조정에 따른 정렬 및 배포 작업

RA3으로 클래식 크기 조정을 수행하는 동안 Even 배포로 마이그레이션되었으며 KEY 배포가 있는 테이블은 원래 배포 스타일로 다시 변환됩니다. 이 기간은 데이터 크기와 클러스터의 사용량에 따라 달라집니다. 쿼리 워크로드를 데이터 마이그레이션보다 실행 우선 순위가 높습니다. 자세한 내용은 [배포 스타일](#)을 참조하세요. 이 마이그레이션 과정에서는 데이터베이스에 대한 읽기와 쓰기가 모두 작동합니다. 하지만 쿼리를 완료하는 데 시간이 더 오래 걸릴 수 있습니다. 그러나 동시성 조정은 이 과정에서 쿼리 워크로드에 리소스를 추가하여 성능을 높일 수 있습니다. [SYS_RESTORE_STATE](#) 및 [SYS_RESTORE_LOG](#) 보기의 결과를 보면 데이터 마이그레이션 진행 상황을 확인할 수 있습니다. 모니터링에 대한 자세한 내용은 다음과 같습니다.

클러스터 크기가 완전히 조정되면 다음과 같은 정렬 동작이 발생합니다.

- 크기 조정으로 인해 클러스터의 슬라이스가 더 많아지면 KEY 분배 테이블은 부분적으로 정렬되지 않지만 EVEN 테이블은 정렬된 상태로 유지됩니다. 또한 크기 조정 직후에는 정렬되는 데이터의 양에 대한 정보가 최신 상태가 아닐 수도 있습니다. 키 복구 후 자동 vacuum이 시간이 지남에 따라 테이블을 정렬합니다.
- 크기 조정으로 인해 클러스터의 슬라이스 수가 줄어들면 KEY 분포 테이블과 EVEN 분포 테이블은 부분적으로 정렬되지 않습니다. 자동 vacuum은 시간이 지남에 따라 테이블을 정렬합니다.

자동 테이블 vacuum 작업에 대한 자세한 내용은 [테이블 Vacuuming](#)을 참조하세요. 컴퓨팅 노드의 슬라이스에 대한 자세한 내용은 [데이터 웨어하우스 시스템 아키텍처](#)를 참조하세요.

대상 클러스터가 RA3인 경우의 클래식 크기 조정 단계

클래식 크기 조정은 대상 클러스터 유형이 RA3이고 이전 섹션에 설명된 사전 요구 사항을 충족한 경우 다음 단계로 구성됩니다.

1. 원본 클러스터에서 대상 클러스터로 마이그레이션을 시작합니다. 새로운 대상 클러스터가 프로비저닝되면 Amazon Redshift가 크기 조정이 시작되었다는 이벤트 알림 메시지를 보냅니다. 그러면 기존 클러스터가 다시 시작되고 모든 연결이 닫힙니다. 기존 클러스터가 데이터 공유 생산자 클러스터인 경우 소비자 클러스터와의 연결도 끊어집니다. 다시 시작하는 데 몇 분이 걸립니다.
2. 다시 시작한 후에는 데이터베이스를 읽고 쓸 수 있습니다. 또한 데이터 공유가 재개되며 이 작업에는 몇 분이 더 걸립니다.
3. 데이터가 대상 클러스터로 마이그레이션됩니다. 대상 노드 유형이 RA3인 경우 데이터 마이그레이션 중에 읽기 및 쓰기가 가능합니다.
4. 크기 조정 프로세스가 거의 끝나가면 Amazon Redshift가 엔드포인트를 대상 클러스터로 업데이트하고, 소스 클러스터의 모든 연결이 종료됩니다. 대상 클러스터는 데이터 공유를 위해 생산자가 됩니다.
5. 크기 조정이 완료됩니다. Amazon Redshift에서 이벤트 알림을 전송합니다.

Amazon Redshift 콘솔에서 크기 조정 진행률을 볼 수 있습니다. 클러스터 크기 조정에 걸리는 시간은 데이터 크기에 따라 다릅니다.

Note

RA3 클러스터의 크기를 조정할 때 레거시 노드 유형 선택 - RA3 노드가 있는 클러스터에서 DC2 등의 다른 노드 유형으로 크기를 조정하려고 하면 콘솔에 유효성 검사 경고 메시지가 나타나고 크기 조정 작업이 완료되지 않습니다. 이는 레거시 노드 유형으로의 크기 조정이 지원

되지 않기 때문에 발생합니다. 이는 고객이 더 이상 사용되지 않거나 곧 사용되지 않을 노드 유형으로 크기를 조정하는 것을 방지하기 위한 것입니다. 이는 탄력적 크기 조정과 클래식 크기 조정 모두에 적용됩니다.

대상 클러스터가 RA3일 때 클래식 크기 조정 모니터링

KEY 배포를 포함하여 진행 중인 프로비저닝된 클러스터의 클래식 크기 조정을 모니터링하려면 [SYS_RESTORE_STATE](#)를 사용하세요. 변환 중인 테이블의 완료율을 보여 줍니다. 데이터에 액세스하려면 수퍼 사용자여야 합니다.

클래식 크기 조정을 수행할 때 필요하지 않은 테이블을 드롭합니다. 이렇게 하면 기존 테이블을 더 빠르게 배포할 수 있습니다.

대상 클러스터가 RA3가 아닌 경우의 클래식 크기 조정 단계

예를 들어 대상 노드 유형이 RA3가 아닌 유형(예: DC2)인 경우 클래식 크기 조정은 다음과 같이 구성됩니다.

1. 원본 클러스터에서 대상 클러스터로 마이그레이션을 시작합니다. 새로운 대상 클러스터가 프로비저닝되면 Amazon Redshift가 크기 조정이 시작되었다는 이벤트 알림 메시지를 보냅니다. 그러면 기존 클러스터가 다시 시작되고 모든 연결이 닫힙니다. 기존 클러스터가 데이터 공유 생산자 클러스터인 경우 소비자 클러스터와의 연결도 끊어집니다. 다시 시작하는 데 몇 분이 걸립니다.

BACKUP NO에서 생성한 테이블 또는 구체화된 뷰와 같은 데이터베이스 관계는 클래식 크기 조정 중에는 유지되지 않는다는 점에 유의하세요. 자세한 내용은 [구체화된 뷰 생성](#)을 참조하세요.

2. 다시 시작한 후에는 데이터베이스를 읽기 전용으로 사용할 수 있습니다. 데이터 공유가 재개되는 데 몇 분 정도 더 걸립니다.
3. 데이터가 대상 클러스터로 마이그레이션됩니다. 데이터베이스는 읽기 전용으로 유지됩니다.
4. 크기 조정 프로세스가 거의 끝나가면 Amazon Redshift가 엔드포인트를 대상 클러스터로 업데이트하고, 소스 클러스터의 모든 연결이 종료됩니다. 대상 클러스터는 데이터 공유를 위해 생산자가 됩니다.
5. 크기 조정이 완료됩니다. Amazon Redshift에서 이벤트 알림을 전송합니다.

Amazon Redshift 콘솔에서 크기 조정 진행률을 볼 수 있습니다. 클러스터 크기 조정에 걸리는 시간은 데이터 크기에 따라 다릅니다.

Note

대상 클러스터가 RA3가 아니거나 이전 섹션에서 자세히 설명한 RA3 대상 클러스터의 사전 요구 사항을 충족하지 않는 경우 많은 양의 데이터가 포함된 클러스터의 크기를 조정하는 데 며칠 또는 몇 주가 걸릴 수 있습니다.

또한 클래식 크기 조정 후에는 클러스터에 사용된 스토리지 용량이 늘어날 수 있습니다. 이는 클래식 크기 조정으로 인해 클러스터에 추가 데이터 슬라이스가 있을 때 발생하는 일반적인 시스템 동작입니다. 클러스터의 노드 수가 동일하게 유지되는 경우에도 이러한 추가 용량 사용이 발생할 수 있습니다.

탄력적 크기 조정 vs 클래식 크기 조정

다음 표에서는 두 크기 조정 유형의 동작을 비교합니다.

동작	탄력적 크기 조정	클래식 크기 조정	설명				
시스템 데이터 보존	탄력적 크기 조정은 시스템 로그 데이터를 유지합니다.	클래식 크기 조정은 시스템 테이블과 데이터를 유지하지 않습니다.	소스 클러스터에 감사 로깅이 활성화되어 있으면 크기 조정 후에 Amazon S3 또는 CloudWatch에서 계속				

동작	탄력적 크기 조정	클래식 크기 조정	설명				
			해서 로그에 액세스할 수 있습니다. 이러한 로그는 데이터 정책에 따라 보관하거나 삭제할 수 있습니다.				

동작	탄력적 크기 조정	클래식 크기 조정	설명				
노드 유형 변경	노드 유형이 변경되지 않는 경우의 탄력적 크기 조정: 인플레이스 크기 조정 및 대부분의 쿼리가 유지됩니다. 새 노드 유형을 선택한 경우의 탄력적 크기 조정: 새 클러스터가 생성됩니다. 크기 조정 프로세스가 완료되면 쿼리가 삭제됩니다.	클래식 크기 조정: 새 클러스터가 생성됩니다. 크기 조정 프로세스 중에 쿼리가 삭제됩니다.					

동작	탄력적 크기 조정	클래식 크기 조정	설명				
세션 및 쿼리 보존	탄력적 크기 조정은 소스 클러스터와 대상의 노드 유형이 동일할 때 세션과 쿼리를 유지합니다. 새 노드 유형을 선택하면 쿼리가 삭제됩니다.	클래식 크기 조정은 세션과 쿼리를 유지하지 않습니다. 쿼리가 삭제됩니다.	쿼리가 삭제되면 성능이 어느 정도 저하될 수 있습니다. 따라서 사용량이 적은 시간대에 크기 조정 작업을 수행하는 것이 가장 좋습니다.				

동작	탄력적 크기 조정	클래식 크기 조정	설명				
크기 조정 작업 취소	탄력적 크기 조정은 취소할 수 없습니다.	Amazon Redshift 콘솔의 클러스터 세부 정보에서 [크기 조정 취소(Cancel resize)]를 선택하여 완료되기 전에 기존 크기 조정 작업을 취소할 수 있습니다.	크기 조정 작업 취소에 걸리는 시간은 취소 시 크기 조정 작업의 단계에 따라 달라 집니다. 취소하는 경우 취소 작업이 완료 될 때까지 클러스터를 사용할 수 없습니다. 크기 조				

동작	탄력적 크기 조정	클래식 크기 조정	설명				
			정작업이 마지막 단계에 있는 경우에는 취소할 수 없습니다.				
			RA3 클러스터에 대한 클래식 크기 조정의 경우 취소할 수 없습니다.				

크기 조정 예약

클러스터의 크기 조정 작업을 예약하여 사용량이 많을 것으로 예상되는 경우 스케일 업하거나 비용 절감을 위해 스케일 다운할 수 있습니다. 예약은 탄력적 크기 조정과 클래식 크기 조정 모두에 적용됩니다. 이제 Amazon Redshift 콘솔에서 일정을 설정할 수 있습니다. 자세한 내용은 콘솔을 사용한 클러스터 관리에서 [클러스터 크기 조정](#) 섹션을 참조하세요. AWS CLI 또는 Amazon Redshift API 작업을 사

용하여 크기 조정을 예약할 수도 있습니다. 자세한 내용은 AWS CLI Command Reference의 [create-scheduled-action](#) 또는 Amazon Redshift API Reference의 [CreateScheduledAction](#)을 참조하세요.

스냅샷, 복원 및 크기 조정

[탄력적 크기 조정](#)은 Amazon Redshift 클러스터의 크기를 조정할 수 있는 가장 빠른 방법입니다. 탄력적 크기 조정을 선택할 수 없고 클러스터에 대해 거의 일정한 쓰기 액세스가 필요하다면 클래식 크기 조정과 함께 다음 섹션에서 설명하는 스냅샷 및 복원 작업을 사용할 수 있습니다. 단, 이 방법을 사용하려면 스냅샷 생성 이후 원본 클러스터에 작성되는 데이터를 전환을 마친 대상 클러스터에 수동으로 복사해야 합니다. 복사에 걸리는 시간에 따라 두 클러스터의 데이터가 동일해질 때까지 이 프로세스를 몇 차례 반복해야 하는 경우도 있습니다. 그런 다음 대상 클러스터로 전환할 수 있습니다. 이때 전체 데이터 집합이 대상 클러스터로 복사될 때까지는 기존 쿼리에 부정적인 영향을 미칠 수도 있습니다. 그러나 데이터베이스에 쓸 수 없는 시간이 최소화됩니다.

스냅샷, 복원 및 클래식 크기 조정 작업에서는 사용되는 프로세스는 다음과 같습니다.

1. 기존 클러스터의 스냅샷을 생성합니다. 기존 클러스터란 원본 클러스터를 말합니다.
2. 스냅샷을 만든 시간입니다. 이렇게 하면 나중에 ETL(추출, 변환, 로드) 프로세스를 다시 실행해 스냅샷 이후 데이터를 대상 데이터베이스로 로드해야 하는 시점을 파악할 수 있습니다.
3. 스냅샷을 새로운 클러스터로 복원합니다. 새로운 클러스터는 대상 클러스터를 말합니다. 대상 클러스터에 샘플 데이터가 존재하는지 확인합니다.
4. 대상 클러스터의 크기를 조정합니다. 대상 클러스터에서 새로운 노드 유형과 노드 수, 그리고 기타 설정을 선택합니다.
5. 원본 클러스터의 스냅샷을 생성한 이후 발생한 ETL 프로세스의 데이터 로드를 확인합니다. 똑같은 순서에 따라 동일한 데이터를 대상 클러스터에 다시 로드해야 할 수도 있습니다. 데이터 로드가 계속되는 경우에는 원본 클러스터와 대상 클러스터의 데이터가 동일해질 때까지 이 프로세스를 몇 차례 반복합니다.
6. 원본 클러스터에서 실행 중인 모든 쿼리를 중단합니다. 이를 위해서는 클러스터를 재부팅하거나, 슈퍼유저로 로그인하여 [PG_CANCEL_BACKEND](#) 및 [PG_TERMINATE_BACKEND](#) 명령을 사용해야 할 수도 있습니다. 클러스터 재부팅이 클러스터 사용을 중단할 수 있는 가장 쉬운 방법입니다.
7. 원본 클러스터 이름을 변경합니다. 예를 들어 이름을 `examplecluster`에서 `examplecluster-source`로 바꿉니다.
8. 대상 클러스터 이름을 변경 이전 원본 클러스터의 이름으로 변경합니다. 예를 들어 대상 클러스터 이름을 `examplecluster`로 바꿉니다. 이 시점부터 `examplecluster`가 포함된 엔드포인트를 사용하는 애플리케이션이 모두 대상 클러스터에 연결됩니다.
9. 대상 클러스터로 전환을 마친 후 원본 클러스터를 삭제하고 모든 프로세스가 정상적으로 실행되는지 확인합니다.

또는 대상 클러스터로 데이터를 다시 로드하기 전에 소스 및 대상 클러스터의 이름을 바꿀 수 있습니다. 이 방법은 종속 시스템 및 보고서를 대상 클러스터의 시스템 및 보고서로 즉시 업데이트할 필요가 없는 경우에 효과적입니다. 이 경우에는 6단계가 위에서 설명한 프로세스의 끝으로 이동합니다.

이름 변경 프로세스는 애플리케이션에서 동일한 엔드포인트를 사용하여 클러스터에 연결해야 할 때만 필요합니다. 이럴 필요가 없다면 클러스터에 연결되는 모든 애플리케이션을 업데이트하여 이름을 변경하지 않고도 대상 클러스터의 엔드포인트를 사용할 수 있습니다.

클러스터 이름을 재사용하면 몇 가지 이점이 있습니다. 첫째, 기본 클러스터를 변경하더라도 엔드포인트가 바뀌지 않기 때문에 애플리케이션 연결 문자열을 업데이트할 필요가 없습니다. 둘째, Amazon CloudWatch 경보 및 Amazon Simple Notification Service(Amazon SNS) 알림과 같은 관련 항목이 클러스터 이름에 연결됩니다. 이러한 연결은 클러스터에 대해 설정한 것과 동일한 경보와 알림을 계속해서 사용할 수 있음을 나타냅니다. 프로덕션 환경에서는 경보나 알림 같은 관련 항목을 재구성하지 않고도 클러스터 크기를 조정할 수 있는 유연성이 필요하기 때문에 이러한 기능이 많이 사용됩니다.

클러스터 이름 변경

클러스터에 다른 이름을 사용하고 싶다면 이름을 변경할 수 있습니다. 클러스터 엔드포인트에는 클러스터 이름(클러스터 식별자라고도 함)이 포함되어 있기 때문에 이름 변경을 마치면 엔드포인트도 새로운 이름을 사용하도록 바뀝니다. 예를 들어, `examplecluster` 클러스터의 이름을 `newcluster`로 변경하면 엔드포인트도 `newcluster` 식별자를 사용하도록 바뀝니다. 클러스터에 연결되는 애플리케이션은 모두 새로운 엔드포인트로 업데이트되어야 합니다.

이러한 애플리케이션의 엔드포인트를 변경하지 않고 애플리케이션이 연결되는 클러스터를 변경하고 싶다면 클러스터의 이름을 바꿀 수 있습니다. 이 경우에는 먼저 원래 클러스터 이름을 변경한 후에 두 번째 클러스터를 변경하여 원래 클러스터의 변경 전 이름을 재사용해야 합니다. 이는 클러스터 식별자는 계정 및 리전 내에서 고유해야, 즉 원래 클러스터와 두 번째 클러스터의 이름이 동일할 수 없기 때문입니다. 스냅샷에서 클러스터를 복원할 때 종속된 애플리케이션의 연결 속성을 변경하지 않으려는 경우 이렇게 할 수 있습니다.

Note

원래 클러스터를 삭제할 때 원하지 않는 클러스터 스냅샷까지 삭제되는 경우 사용자 본인에게 책임이 있습니다.

클러스터 이름을 변경할 때는 프로세스를 마칠 때까지 클러스터 상태가 `renaming`으로 바뀝니다. 클러스터에서 이전에 사용된 DNS 이름은 바로 삭제되지만 캐시는 몇 분 더 남을 수도 있습니다. 이름이

바뀐 클러스터의 새로운 DNS 이름은 약 10분 내에 적용됩니다. 이름이 바뀐 클러스터를 사용하려면 새로운 이름이 적용될 때까지 기다려야 합니다. 클러스터가 재부팅되고 기존 클러스터 연결은 모두 종료됩니다. 이 모든 것이 완료되면 엔드포인트가 새로운 이름을 사용하도록 바뀝니다. 따라서 이름을 변경하기 전에 실행 중인 쿼리를 모두 종료한 후 이름 변경이 완료된 후에 쿼리를 다시 시작해야 합니다.

이름이 변경된 후에도 클러스터 스냅샷은 그대로 유지되며, 클러스터와 연결된 스냅샷 모두 클러스터와 연결된 상태가 지속됩니다. 예를 들어, 프로덕션 데이터베이스 역할을 수행하는 클러스터에 다수의 스냅샷이 있다고 가정하겠습니다. 이때 클러스터 이름을 변경한 후 프로덕션 환경에서 스냅샷으로 교체하더라도 이름이 바뀐 클러스터는 여전히 기존 스냅샷과 연결됩니다.

Amazon CloudWatch 경보와 Amazon Simple Notification Service(Amazon SNS) 이벤트 알림은 클러스터 이름과 연결됩니다. 클러스터 이름을 변경하면 이 경보와 알림도 업데이트해야 합니다. CloudWatch 콘솔에서 CloudWatch 경보를 업데이트하고 [이벤트(Events)] 창의 Amazon Redshift 콘솔에서 Amazon SNS 이벤트 알림을 업데이트할 수 있습니다. 클러스터의 로드 및 쿼리 데이터는 이름 변경 전후의 데이터를 계속해서 표시하지만 성능 데이터는 이름 변경 이후 재설정됩니다.

자세한 내용은 [클러스터 수정](#) 섹션을 참조하세요.

클러스터의 릴리스 버전 업그레이드

릴리스 상태(Release Status) 값이 새로운 릴리스 이용 가능(New release available)인 클러스터의 릴리스 유지 관리 버전을 업그레이드할 수 있습니다. 유지 관리 버전을 업그레이드할 때 즉시 업그레이드하거나 다음 유지 관리 기간에 업그레이드하도록 선택할 수 있습니다.

Important

즉시 업그레이드하는 경우, 업그레이드가 완료될 때까지 클러스터가 오프라인입니다.

클러스터를 새 릴리스 버전으로 업그레이드하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다.
3. 업그레이드할 클러스터를 선택합니다.
4. 작업(Actions)에서 클러스터 버전 업그레이드(Upgrade cluster version)을 선택합니다. 클러스터 버전 업그레이드(Upgrade cluster version) 페이지가 표시됩니다.
5. 페이지의 지시 사항을 따릅니다.

6. 클러스터 버전 업그레이드(Upgrade cluster version)을 선택합니다.

클러스터 일시 중지 및 다시 시작

특정 시간에만 사용 가능해야 하는 클러스터가 있는 경우 클러스터를 일시 중지하고 나중에 다시 시작할 수 있습니다. 클러스터가 일시 중지되는 동안 온디맨드 결제가 일시 중지됩니다. 클러스터의 스토리지에만 요금이 부과됩니다. 요금에 대한 자세한 내용은 [Amazon Redshift 요금](#) 페이지를 참조하세요.

클러스터를 일시 중지하면 Amazon Redshift에서 스냅샷을 생성하고 쿼리를 종료하기 시작한 다음 클러스터를 일시 중지 상태로 만듭니다. 최종 스냅샷을 요청하지 않고 일시 중지된 클러스터를 삭제하면 클러스터를 복원할 수 없습니다. 일시 중지 또는 다시 시작 작업이 시작된 후에는 취소 또는 롤백할 수 없습니다.

AWS CLI 또는 Amazon Redshift API 작업을 사용하여 Amazon Redshift 콘솔에서 클러스터를 일시 중지하고 다시 시작할 수 있습니다.

클러스터를 일시 중지하고 다시 시작하도록 작업을 예약할 수 있습니다. 새 Amazon Redshift 콘솔을 사용하여 일시 중지하고 다시 시작할 반복 일정을 생성하면 선택한 날짜 범위에 대해 2개의 예약된 작업이 생성됩니다. 예약된 작업 이름에는 접미사로 -pause 및 -resume이 붙습니다. 이름의 총 길이는 예약된 작업 이름의 최대 크기 내에 있어야 합니다.

다음 유형의 클러스터는 일시 중지할 수 없습니다.

- EC2-Classic 클러스터
- 활성 상태가 아닌 클러스터(예: 현재 수정 중인 클러스터)
- 하드웨어 보안 모듈(HSM) 클러스터
- 자동 스냅샷이 비활성화된 클러스터

클러스터를 일시 중지할 때 다음 사항을 고려하세요.

- 클러스터에 대한 연결 또는 쿼리를 사용할 수 없습니다.
- Amazon Redshift 콘솔에서 일시 중지된 클러스터의 쿼리 모니터링 정보를 볼 수 없습니다.
- 일시 중지된 클러스터는 수정할 수 없습니다. 클러스터에서 예약된 작업은 수행되지 않습니다. 여기에는 스냅샷 생성, 클러스터 크기 조정 및 클러스터 유지 관리 작업이 포함됩니다.
- 하드웨어 지표가 생성되지 않았습니다. 누락된 지표에 대한 경보가 설정된 경우 CloudWatch 경보를 업데이트합니다.

- 일시 중지된 클러스터의 최신 자동 스냅샷을 수동 스냅샷으로 복사할 수 없습니다.
- 클러스터가 일시 중지되는 동안에는 일시 중지 작업이 완료될 때까지 클러스터를 다시 시작할 수 없습니다.
- 클러스터를 일시 중지하면 결제가 일시 중단됩니다. 그러나 일시 중지 작업은 일반적으로 클러스터의 크기에 따라 15분 이내에 완료됩니다.
- 감사 로그는 보관되고 재개 시 복원되지 않습니다.
- 클러스터가 일시 중지된 후에는 일시 중지 전에 발생한 문제를 해결하기 위해 추적 및 로그를 사용하지 못할 수 있습니다.
- AWS Secrets Manager를 사용하여 관리자 보안 인증 정보를 관리하고 클러스터를 일시 중지하면 클러스터의 보안 암호는 삭제되지 않으며 보안 암호에 대한 요금이 계속 청구됩니다. AWS Secrets Manager를 사용하여 Redshift 관리자 암호를 관리하는 방법에 대한 자세한 내용은 [AWS Secrets Manager를 사용하여 Amazon Redshift 관리자 암호를 관리하는 방법](#) 섹션을 참조하세요.
- RA3 인스턴스 유형의 경우 클러스터의 no-backup 테이블은 다시 시작할 때 복원됩니다. DC2 인스턴스 유형에는 재개 시 복원되지 않습니다. 백업 없음 테이블에 대한 자세한 내용은 [스냅샷에서 테이블 제외](#) 섹션을 참조하세요.

클러스터를 다시 시작할 때 다음 사항을 고려하세요.

- 다시 시작된 클러스터의 클러스터 버전은 클러스터의 유지 관리 기간에 따라 유지 관리 버전으로 업데이트됩니다.
- 일시 중지된 클러스터와 연결된 서브넷을 삭제하면 호환되지 않는 네트워크가 있을 수 있습니다. 이 경우 최신 스냅샷에서 클러스터를 복원합니다.
- 클러스터가 일시 중지된 상태에서 탄력적 IP 주소를 삭제하면 새 탄력적 IP 주소가 요청됩니다.
- Amazon Redshift에서 이전 탄력적 네트워크 인터페이스로 클러스터를 다시 시작할 수 없는 경우 Amazon Redshift에서 새 인터페이스를 할당하려고 시도합니다.
- 클러스터를 다시 시작하면 노드 IP 주소가 변경될 수 있습니다. SSH(Secure Shell)에서 COPY 또는 Amazon EMR에서 COPY 지원과 같은 기능에 대해 이러한 새 IP 주소를 지원하도록 VPC 설정을 업데이트해야 할 수 있습니다.
- 일시 중지되지 않은 클러스터를 다시 시작하려고 하면 다시 시작 작업이 오류를 반환합니다. 다시 시작 작업이 예약된 작업의 일부인 경우 이후의 오류를 방지하기 위해 예약된 작업을 수정하거나 삭제합니다.
- 클러스터의 크기에 따라 쿼리 처리 전 클러스터를 다시 시작하는 데 몇 분 정도 걸릴 수 있습니다. 또한 다시 시작 완료 후 클러스터를 다시 수화하는 동안 일정 기간 동안 쿼리 성능에 영향을 줄 수 있습니다.

클러스터 재부팅

클러스터 재부팅은 재부팅 전과 동일한 구성으로 클러스터를 재시작하는 클러스터 작업입니다. 클러스터를 재부팅하여 보류 중인 유지 관리 업데이트를 적용하거나, 구성 변경을 재설정하거나, 특정 문제에서 복구하거나, 클러스터 문제를 해결할 수 있습니다. 클러스터를 재부팅하면 Amazon Redshift 환경의 성능, 보안 및 안정성을 최적화하는 데 도움이 될 수 있습니다. 다음 절차는 Amazon Redshift 클러스터를 재부팅하기 위한 세부 단계를 제공합니다.

클러스터를 재부팅하면 클러스터 상태가 `rebooting`으로 설정되고, 재부팅을 마치면 클러스터 이벤트가 생성됩니다. 대기 중인 클러스터 변경 사항은 이때 적용됩니다.

클러스터를 재부팅하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다.
3. 재부팅할 클러스터를 선택합니다.
4. 작업(Actions)에서 클러스터 재부팅(Reboot cluster)을 선택합니다. 클러스터 재부팅(Reboot cluster) 페이지가 표시됩니다.
5. 클러스터 재부팅(Reboot cluster)을 선택합니다.

클러스터 재부팅

Amazon Redshift에서 재배포를 사용하면 데이터 손실이나 애플리케이션 변경 없이 Amazon Redshift가 클러스터를 다른 가용 영역으로 이동할 수 있습니다. 재배포를 사용하면 클러스터에서 서비스 중단이 발생할 때 영향을 최소화하면서 작업을 계속할 수 있습니다.

클러스터 재배포가 켜지면 Amazon Redshift가 일부 상황에서 클러스터를 재배포하도록 선택할 수 있습니다. 특히 현재 가용 영역의 문제로 인해 최적의 클러스터 작동이 방해받거나 서비스 가용성을 향상시키려는 경우 클러스터가 재배포됩니다. 지정된 가용 영역의 리소스 제약으로 인해 클러스터 작업이 중단되는 경우에도 재배포 기능을 호출할 수 있습니다. 클러스터를 재개하거나 크기를 조정하는 기능을 예로 들 수 있습니다. Amazon Redshift는 추가 비용 없이 재배포 기능을 제공합니다.

Amazon Redshift 클러스터가 새 가용 영역으로 재배포되면 새 클러스터에는 원래 클러스터와 동일한 엔드포인트가 있습니다. 애플리케이션은 엔드포인트에 다시 연결하고 데이터 수정 또는 손실 없이 작업을 계속할 수 있습니다. 그러나 지정된 가용 영역에서 잠재적인 리소스 제약으로 인해 재배포가 불가능한 경우도 있을 수 있습니다.

Amazon Redshift 클러스터 재배포는 RA3 인스턴스 유형에 대해서만 지원됩니다. RA3 인스턴스 유형은 Redshift 관리형 스토리지(RMS)를 내구성 있는 스토리지 계층으로 사용합니다. 클러스터 데이터의 최신 복사본은 AWS 리전의 다른 가용 영역에서 항상 사용할 수 있습니다. 즉, 데이터 손실 없이 Amazon Redshift 클러스터를 다른 가용 영역으로 재배포할 수 있습니다.

클러스터에 대한 재배포를 켜면 Amazon Redshift가 클러스터를 프록시 뒤에 있도록 마이그레이션합니다. 이렇게 하면 클러스터의 컴퓨팅 리소스에 대한 위치 독립적 액세스를 구현하는 데 도움이 됩니다. 마이그레이션으로 인해 클러스터가 재부팅됩니다. 클러스터가 다른 가용 영역으로 재배포되면 새 클러스터가 새 가용 영역에서 다시 온라인 상태가 되는 동안 중단이 발생합니다. 그러나 클러스터가 새 가용 영역으로 재배포된 후에도 클러스터 엔드포인트가 변경되지 않은 상태로 유지되므로 애플리케이션을 변경할 필요가 없습니다.

새로 생성되거나 복원된 RA3 클러스터의 서브넷 그룹에 여러 가용 영역이 포함된 경우 클러스터 재배포는 기본적으로 활성화됩니다. Amazon Redshift는 프로비저닝된 클러스터를 생성하는 동안 5439를 기본 포트로 할당합니다. 포트 범위 5431~5455 또는 8191~8215에서 다른 포트로 변경할 수 있습니다. 범위를 벗어난 포트로 변경하지 마세요. 오류가 발생합니다. 프로비저닝된 클러스터의 기본 포트를 변경하려면 Amazon Redshift 콘솔, AWS CLI 또는 Amazon Redshift API를 사용하세요. 서버리스 작업 그룹의 기본 포트를 변경하려면 AWS CLI 또는 Amazon Redshift Serverless API를 사용하세요.

재배포를 켜고 현재 리더 노드 IP 주소를 사용하여 클러스터 또는 향상된 VPC 라우팅에 액세스하는 경우 해당 액세스를 변경해야 합니다. 대신 클러스터의 Virtual Private Cloud(VPC) 엔드포인트와 연결된 IP 주소를 사용합니다. 이 클러스터 IP 주소를 찾으려면 클러스터 세부 정보 페이지의 [네트워크 및 보안(Network and security)] 섹션에서 VPC 엔드포인트를 찾아서 사용합니다. VPC 엔드포인트에 대한 세부 정보를 보려면 Amazon VPC 콘솔에 로그인합니다.

AWS Command Line Interface(AWS CLI) 명령 `describe-vpc-endpoints`를 사용하여 엔드포인트와 연결된 탄력적 네트워크 인터페이스를 가져올 수도 있습니다. `describe-network-interfaces` 명령을 사용하여 연결된 IP 주소를 가져올 수 있습니다. Amazon Redshift AWS CLI 명령에 대한 자세한 내용은 AWS CLI Command Reference의 [Available commands](#)를 참조하세요.

제한 사항

Amazon Redshift 재배포를 사용할 때 다음 제한 사항에 유의합니다.

- 지정된 가용 영역의 잠재적인 리소스 제한으로 인해 일부 시나리오에서는 클러스터 재배포가 불가능할 수 있습니다. 이 경우 Amazon Redshift가 원래 클러스터를 변경하지 않습니다.
- DC2 인스턴스 패밀리에서는 재배포가 지원되지 않습니다.
- AWS 리전 간에 재배포를 수행할 수 없습니다.

- Amazon Redshift 재배포는 기본적으로 포트 5439로 설정됩니다. 5431~5455 또는 8191~8215 범위에서 다른 포트로 변경할 수도 있습니다.

콘솔을 사용하여 재배포 관리

Amazon Redshift 콘솔을 사용하여 클러스터 재배포에 대한 설정을 관리할 수 있습니다.

새 클러스터 생성 시 재배포 끄기

새 클러스터를 생성할 때 재배포를 끄려면 다음 절차를 따릅니다.

새 클러스터의 재배포를 끄려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다.
3. [클러스터 생성(Create cluster)]을 선택하여 새 클러스터를 생성합니다. 클러스터를 생성하는 방법에 대한 자세한 내용은 Amazon Redshift 시작 안내서의 [Get started with Amazon Redshift provisioned data warehouses](#)를 참조하세요.
4. 백업에서 클러스터 재배포에 대해 비활성화됨을 선택합니다. 재배포는 기본적으로 켜져 있습니다.
5. 클러스터 생성을 선택합니다.

기존 클러스터에 대한 재배포 수정

기존 클러스터에 대한 재배포 설정을 변경하려면 다음 절차를 따릅니다.

기존 클러스터에 대한 재배포 설정을 수정하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다.
3. 목록에서 수정할 클러스터의 이름을 선택합니다. 클러스터 세부 정보 페이지가 나타납니다.
4. [유지 관리(Maintenance)] 탭을 선택하고 [백업 세부 정보(Backup details)] 섹션에서 [편집(Edit)]을 선택합니다.
5. 백업에서 비활성화됨을 선택합니다. 재배포는 기본적으로 켜져 있습니다.

6. 클러스터 수정을 선택합니다.

클러스터 재부팅

다른 가용 영역으로 클러스터를 수동으로 재배포하려면 다음 절차를 따릅니다. 이는 보조 가용 영역에서 네트워크 설정을 테스트하려는 경우 또는 현재 가용 영역에서 리소스 제약 조건을 실행할 때 특히 유용합니다.

다른 가용 영역으로 클러스터를 재배포하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다.
3. 목록에서 이동할 클러스터의 이름을 선택합니다. 클러스터 세부 정보 페이지가 나타납니다.
4. [작업(Actions)]에서 [재배치(Relocate)]를 선택합니다. [클러스터 재배포(Relocate cluster)] 페이지가 표시됩니다.
5. (옵션) [가용 영역(Availability Zone)]을 선택합니다. 가용 영역을 선택하지 않으면 Amazon Redshift가 자동으로 선택합니다.

Amazon Redshift가 재배포를 시작하고 클러스터를 재배포 중으로 표시합니다. 재배포가 완료되면 클러스터 상태가 사용 가능으로 바뀝니다.

Amazon Redshift CLI를 사용하여 재배포 관리

AWS 명령줄 인터페이스(CLI)를 사용하여 클러스터 재배포에 대한 설정을 관리할 수 있습니다.

AWS CLI 사용 시 다음 예제 명령은 재배포가 켜져 있는 **mycluster**라는 Amazon Redshift 클러스터를 생성합니다.

```
aws redshift create-cluster --cluster-identifier mycluster --number-of-nodes 2 --
master-username enter a username --master-user-password enter a password --node-type
ra3.4xlarge --port 5439 --no-availability-zone-relocation
```

현재 클러스터가 다른 포트를 사용하는 경우에는 포트 범위 5431~5455 또는 8191~8215를 사용하도록 수정한 다음 배치를 켜도록 수정해야 합니다. 기본값은 5439입니다. 다음 예제 명령은 클러스터가 지정된 범위에 속하는 포트를 사용하지 않는 경우 포트를 수정합니다.

```
aws redshift modify-cluster --cluster-identifier mycluster --port 5439
```

다음 예제 명령은 Amazon Redshift 클러스터에서 availability-zone-relocation 파라미터를 포함합니다.

```
aws redshift modify-cluster --cluster-identifier mycluster --availability-zone-relocation
```

다음 예제 명령은 Amazon Redshift 클러스터에서 availability-zone-relocation 파라미터를 끕니다.

```
aws redshift modify-cluster --cluster-identifier mycluster --no-availability-zone-relocation
```

다음 예제 명령은 Amazon Redshift 클러스터에서 재배치를 호출합니다.

```
aws redshift modify-cluster --cluster-identifier mycluster --availability-zone us-east-1b
```

클러스터에 대한 사용량 한도 설정

최대 4개의 사용량 한도를 추가하여 다음 각 항목의 사용량을 제어할 수 있습니다.

- 동시성 확장
- 추가 컴퓨팅 리소스를 사용하여 자동 최적화 실행
- Redshift Spectrum 사용량
- 리전 간 데이터 공유

프로비저닝된 클러스터의 사용량 한도 설정

다음은 프로비저닝된 클러스터에서 사용량 한도를 설정하는 절차입니다.

클러스터의 사용량 한도를 설정하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 제한을 설정할 프로비저닝된 클러스터로 이동합니다.
3. 클러스터의 세부 정보 페이지의 작업 드롭다운 메뉴에서 사용량 한도 관리를 선택합니다. 클러스터의 유지 관리 탭을 선택한 다음 아래로 스크롤하여 사용량 한도 생성을 선택할 수도 있습니다.

4. 설정하려는 사용량 한도에 대해 한도 추가를 선택합니다. 지정된 기능에 대해 최대 4개의 한도를 추가할 수 있습니다.
5. 사용량 한도의 기간을 일별, 주별 또는 월별로 설정합니다.
6. 사용량 한도를 설정합니다.
 - 추가 컴퓨팅 리소스 제한을 사용하여 실행되는 동시성 규모 조정 및 자동 최적화의 경우 사용량 한도는 Amazon Redshift가 지정된 기간 동안 기능을 사용하여 소비하는 시간입니다. 이 경우 사용량 한도는 시간 및 분으로 설정됩니다.
 - Redshift Spectrum의 경우 사용량 한도는 Amazon S3에서 스캔한 데이터의 양입니다. 이 경우 사용량 한도는 테라바이트(TB)로 설정됩니다.
 - 리전 간 데이터 공유의 사용량 한도는 생산자 리전에서 소비자 리전으로 전송되어 소비자가 쿼리할 수 있는 데이터의 양입니다. 이 경우 사용량 한도는 테라바이트(TB)로 설정됩니다.
7. 클러스터가 한도에 도달할 때 수행할 Amazon Redshift에 대한 작업을 설정합니다. 다음과 같은 설정이 있습니다.
 - 시스템 테이블에 기록 - 시스템 뷰 [SYS_QUERY_HISTORY](#)에 레코드를 추가합니다. 이 뷰의 `usage_limit` 열을 쿼리하여 쿼리가 한도를 초과했는지 확인할 수 있습니다.
 - 알림 - Amazon SNS를 사용하여 알림 구독을 설정하고 한도가 위반된 경우 알림을 전송합니다. 기존 Amazon SNS 주제를 선택하거나, 새로운 주제를 생성하거나, 주제 없이 진행할 수 있습니다.
 - 기능 비활성화 - 기능을 비활성화합니다. Amazon SNS를 사용하여 알림을 보내도록 선택할 수도 있습니다. 사용자는 다른 작업에 클러스터를 계속 사용할 수 있습니다.

처음 두 작업은 정보 제공용이지만 마지막 작업은 기능 사용을 해제합니다.
8. 페이지 하단에서 변경 사항 저장을 선택하여 한도를 저장합니다. 한 번에 두 개 이상의 한도를 설정하면 변경 사항 저장이 모든 한도를 한 번에 저장합니다.

클러스터 종료 및 삭제

실행을 중단하여 요금을 지불하지 않으려면 클러스터를 종료할 수 있습니다. 클러스터를 종료할 때는 옵션으로 최종 스냅샷을 생성할 수 있습니다. 최종 스냅샷을 생성하는 경우에는 Amazon Redshift가 종료 전에 클러스터의 수동 스냅샷을 생성합니다. 삭제하는 클러스터와 동일한 데이터 및 구성으로 새로운 클러스터를 프로비저닝할 계획이라면 수동 스냅샷이 필요합니다. 수동 스냅샷을 사용하여 나중에 스냅샷을 복원하고 클러스터 사용을 다시 시작할 수 있습니다.

더 이상 클러스터와 데이터가 필요하지 않다면 최종 스냅샷을 생성하지 않고 종료할 수도 있습니다. 이 때는 클러스터와 데이터가 영구적으로 삭제됩니다.

최종 수동 스냅샷을 포함한 클러스터의 종료 여부와 상관없이 클러스터가 종료되면 클러스터와 연결된 자동 스냅샷도 모두 삭제됩니다. 하지만 클러스터와 연결된 수동 스냅샷은 모두 유지됩니다. 옵션으로 제공되는 최종 스냅샷을 포함하여 종료 이후 남는 수동 스냅샷은 클러스터 종료 시 실행 중인 다른 클러스터가 없는 경우, 혹은 실행 중인 Amazon Redshift 클러스터에 제공되는 무료 스토리지를 초과하는 경우 Amazon Simple Storage Service 스토리지 요율에 따라 요금이 부과됩니다. 스냅샷 스토리지 요금에 대한 자세한 내용은 [Amazon Redshift 요금 페이지](#)를 참조하세요.

클러스터를 삭제하면 연결된 AWS Secrets Manager 보안 암호도 모두 삭제됩니다.

클러스터를 삭제하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다.
3. 삭제할 클러스터를 선택합니다.
4. 작업(Actions)에 대해 삭제>Delete)를 선택합니다. 클러스터 삭제>Delete cluster) 페이지가 표시됩니다.
5. 클러스터 삭제>Delete cluster)를 선택합니다.

Note

클러스터를 삭제하고 최종 스냅샷을 생성하기로 선택할 때 클러스터에서 복원 작업이 진행 중인 경우 Amazon Redshift가 삭제 요청을 중지합니다. 이 경우 최종 스냅샷 없이 클러스터를 삭제하거나 복원이 완료된 후 최종 스냅샷과 함께 클러스터를 삭제할 수 있습니다.

Amazon Redshift 스냅샷 및 백업

스냅샷은 클러스터의 특정 시점 백업입니다. 자동 및 수동의 2가지 유형의 스냅샷이 있습니다.

Amazon Redshift는 암호화된 보안 소켓 계층(SSL) 연결을 사용하여 이러한 스냅샷을 Amazon S3에 내부적으로 저장합니다.

Amazon Redshift는 이전 자동 스냅샷 이후 클러스터의 변경 사항을 추적하는 증분 스냅샷을 자동으로 생성합니다. 자동 스냅샷은 클러스터를 복원하는 데 필요한 모든 데이터를 유지합니다. 자동 스냅샷을 만든 시기를 제어하는 스냅샷 일정을 생성하거나 언제든지 수동 스냅샷을 생성할 수 있습니다.

스냅샷에서 복원하면 Amazon Redshift에서 새 클러스터가 생성되고 모든 데이터가 로드되기 전에 새 클러스터를 사용할 수 있으므로 즉각 새 클러스터 쿼리를 시작할 수 있습니다. 클러스터는 활성 쿼리에 응답하여 스냅샷에서 온디맨드로 데이터를 스트리밍한 다음 백그라운드에서 남은 데이터를 로드합니다.

클러스터를 시작할 때 자동 및 수동 스냅샷의 보관 기간을 설정할 수 있습니다. 자동 및 수동 스냅샷의 기본 보존 기간은 클러스터를 수정해 변경할 수 있습니다. 수동 스냅샷의 보존 기간은 스냅샷을 생성할 때 또는 스냅샷 설정을 수정해 변경할 수 있습니다.

AWS Management Console에서 스냅샷 세부 정보를 확인하거나 CLI에서 [describe-cluster-snapshots](#) 또는 [DescribeClusterSnapshots](#) API 작업을 호출하여 스냅샷 진행 상황을 모니터링할 수 있습니다. 진행 중인 스냅샷의 경우, 증분 스냅샷 크기, 전송 속도, 경과 시간, 남은 예상 시간 같은 정보가 표시됩니다.

클러스터에서 백업을 항상 사용할 수 있도록 Amazon Redshift는 Amazon Redshift에서 관리하는 내부적으로 관리되는 Amazon S3 버킷에 스냅샷을 저장합니다. 스토리지 요금을 관리하려면 자동화된 스냅샷을 보관해야 하는 일 수를 평가하고 그에 따라 보관 기간을 구성합니다. 더 이상 필요하지 않은 수동 스냅샷을 삭제합니다. 백업 스토리지 비용에 대한 자세한 내용은 [Amazon Redshift 요금](#) 페이지를 참조하세요.

AWS 서비스, 클라우드 및 온프레미스에서 데이터 보호를 중앙 집중화하고 자동화하는 데 도움이 되는 완전관리형 서비스인 AWS Backup을 사용하여 스냅샷을 생성하고 복원할 수도 있습니다. 자세한 내용은 [Amazon Redshift와 AWS Backup의 통합](#) 섹션을 참조하세요. AWS Backup에 관한 자세한 내용은 AWS Backup 개발자 안내서의 [AWS Backup이란?](#)을 참조하세요.

Amazon Redshift Serverless에서 스냅샷 및 백업 작업

Amazon Redshift Serverless를 사용하면 프로비저닝된 클러스터에서와 마찬가지로 네임스페이스에 있는 객체와 데이터를 특정 시점으로 표현한 백업을 생성할 수 있습니다. Amazon Redshift Serverless의 백업에는 두 가지 유형이 있습니다. 하나는 수동으로 생성하는 스냅샷이고 다른 하나는 Amazon Redshift Serverless가 자동으로 생성하는 복구 시점입니다. Amazon Redshift Serverless의 스냅샷 사용에 대한 자세한 내용은 [스냅샷 및 복구 시점](#)에서 확인할 수 있습니다.

프로비저닝된 클러스터에서 서버리스 네임스페이스로 스냅샷을 복원할 수도 있습니다. 자세한 내용은 [스냅샷에서 서버리스 네임스페이스 복원](#)을 참조하세요.

자동 스냅샷

클러스터에 자동 스냅샷이 사용되면 Amazon Redshift는 정기적으로 해당 클러스터의 스냅샷을 생성합니다. 기본적으로 Amazon Redshift는 8시간마다 또는 데이터 노드당 5GB가 변경된 이후 중 먼저 오

는 때를 기준으로 스냅샷을 생성합니다. 데이터가 5GB * 노드 수보다 큰 경우 자동 스냅샷 생성 사이의 최소 시간은 15분입니다. 또는 자동 스냅샷을 만든 시기를 제어하는 스냅샷 일정을 생성할 수 있습니다. 사용자 지정 일정을 사용하는 경우 자동 스냅샷 사이의 최소 시간은 1시간입니다. 클러스터를 생성하면 기본적으로 자동 스냅샷이 활성화됩니다.

자동 스냅샷은 보존 기간이 끝날 때 삭제됩니다. 기본 보존 기간은 하루지만 Amazon Redshift 콘솔을 사용하거나 Amazon Redshift API 또는 CLI를 통해 프로그래밍 방식으로 수정할 수 있습니다.

자동 스냅샷을 비활성화하려면 보존 기간을 0으로 설정합니다. 자동 스냅샷을 사용 중지하면 Amazon Redshift는 스냅샷 생성을 중단하며 클러스터의 기존 자동 스냅샷을 모두 삭제합니다. RA3 노드 유형에 대한 자동 스냅샷은 비활성화할 수 없습니다. RA3 노드 유형 자동 보존 기간을 1~35일로 설정할 수 있습니다.

Amazon Redshift만 자동 스냅샷을 삭제할 수 있습니다. 사용자가 수동으로 삭제할 수 없습니다. Amazon Redshift는 스냅샷의 보존 기간이 끝날 때, 클러스터에 대해 자동 스냅샷을 사용 중지할 때 또는 클러스터를 삭제할 때 자동 스냅샷을 삭제합니다. Amazon Redshift는 자동 스냅샷을 사용 중지하거나 클러스터를 삭제할 때까지 최신 자동 스냅샷을 유지합니다.

자동 스냅샷을 장기간 보관하려는 경우, 수동 스냅샷으로 자동 스냅샷의 사본을 만들 수 있습니다. 자동 스냅샷은 보존 기간이 끝날 때까지 보존되지만 해당 수동 스냅샷은 수동으로 삭제할 때까지 또는 보존 기간이 종료될 때까지 보존됩니다.

자동 스냅샷 일정

스냅샷이 생성되는 시기를 정확하게 제어하려면 스냅샷 일정을 생성하고 해당 일정을 하나 이상의 클러스터에 연결합니다. 스냅샷 일정을 수정할 때 연관된 모든 클러스터에 대한 일정이 수정됩니다. 클러스터에 스냅샷 일정이 연결되어 있지 않으면 클러스터가 기본 자동 스냅샷 일정을 사용합니다.

스냅샷 일정은 일련의 일정 규칙입니다. 8시간마다 또는 12시간마다 등 지정된 간격을 기반으로 간단한 일정 규칙을 정의할 수 있습니다. 또한 특정 요일, 특정 시간 또는 특정 기간 동안 스냅샷을 생성하는 규칙을 추가할 수 있습니다. 규칙은 Unix 종류의 cron 식을 사용하여 정의할 수도 있습니다.

스냅샷 일정 형식

Amazon Redshift 콘솔에서 스냅샷 일정을 생성할 수 있습니다. 그런 다음 일정을 클러스터에 연결하여 시스템 스냅샷의 생성을 트리거할 수 있습니다. 일정은 여러 클러스터에 연결할 수 있으며 스냅샷을 트리거하기 위해 일정에서 여러 cron 정의를 생성할 수 있습니다.

cron 구문을 사용하여 스냅샷에 대한 일정을 정의할 수 있습니다. 이러한 일정의 정의는 수정된 Unix 종류의 [cron](#) 구문을 사용합니다. 시간을 [협정 세계시\(UTC\)](#)로 지정합니다. 최대 1시간 주기 및 최소 1분 정밀도의 일정을 생성할 수 있습니다.

Amazon Redshift에서 수정한 cron 식에는 각각 공백으로 구분되는 필수 필드 3개가 있습니다.

구문:

```
cron(Minutes Hours Day-of-month Month Day-of-week Year)
```

필드:	값	와일드카드:
Minutes	0~59	, - * /
Hours	0~23	, - * /
Day-of-month	1~31	, - * ? / L W
월	1~12 또는 JAN-DEC	, - * /
요일	1~7 또는 SUN~SAT	, - * ? L #
년	1970~2199	, - * /

와일드카드

- ,(쉼표) 와일드카드는 추가 값을 포함합니다. Day-of-week 필드에서 MON, WED, FRI는 월요일, 수요일 및 금요일을 포함합니다. 총 값은 필드당 24로 제한됩니다.
- -(대시) 와일드카드는 범위를 지정합니다. Hour 필드에서 1~15는 지정된 일의 1~15시간을 포함합니다.
- *(별표) 와일드카드는 필드의 모든 값을 포함합니다. Hours 필드에서 *는 모든 시간을 포함한다는 의미입니다.
- /(슬래시) 와일드카드로 증분을 지정합니다. Hours 필드에 **1/10**을 입력하면 지정된 일의 첫 시간부터 시작해서 10시간 간격(예: 01:00, 11:00, 21:00 등)을 지정할 수 있습니다.
- ?(물음표) 와일드카드는 어떤 한 가지나 다른 것을 지정합니다. Day-of-month 필드에 7을 입력하고 Day-of-week 필드에는 ?을 입력하면 매월 7일이 무슨 요일이든 상관없이 7번째 되는 날을 지정한다는 의미입니다.
- Day-of-month 또는 Day-of-week 필드에서 L 와일드카드는 해당 월 또는 주의 마지막 날을 지정합니다.
-] 필드에서는 W 와일드카드로 어떤 한 평일을 지정할 수 있습니다. Day-of-month Day-of-month 필드에서 3W를 해당 월의 세 번째 평일에 가장 가까운 날을 지정할 수 있습니다.

- '요일' 필드의 # 와일드카드가 그 달에 속한 정해진 요일의 특정 인스턴스를 지정합니다. 예를 들어, 3#2는 그 달의 두 번째 화요일입니다. 3은 각 주의 셋째 날이므로 화요일을 나타내고 2는 그 달의 두 번째 해당 요일입니다.

Note

'#' 문자를 사용하는 경우 요일(day-of-week) 필드에 하나의 표현식만 정의할 수 있습니다. 예를 들어 '3#1,6#3'은 두 개의 표현식으로 해석되기 때문에 유효하지 않습니다.

한도

- 동일한 cron 표현식에 Day-of-month와 Day-of-week 필드를 지정할 수 없습니다. 이 필드 중 하나에 값을 지정하는 경우에는 다른 필드에서 반드시 ?(물음표)를 사용해야 합니다.
- 스냅샷 일정은 다음 빈도를 지원하지 않습니다.
 - 시간당 1개보다 더 높은 빈도로 예약된 스냅샷
 - 1일(24시간)당 1개보다 더 낮은 빈도로 예약된 스냅샷

1시간 기간 내 스냅샷을 예약하게 되는 중복된 일정이 있으면 검증 오류가 발생합니다.

일정을 생성할 때는 다음과 같은 Cron 문자열을 사용할 수 있습니다.

분	시간	요일	의미			
0	14-20/1	화	화요일 오후 2시~오후 8시 사이 매시간			
0	21	월-금	월요일~금요일 매일 밤 9시입니다.			
30	0/6	토-일	당일 자정 이후 30분 (00:30)에 시작해서 토요일과 일요일에 6시간씩 증분 이 결과 매일 [00:30, 06:30, 12:30 및 18:30]에 스냅샷이 생성됩니다.			

분	시간	요일	의미			
30	12/4	*	매일 12:30에 시작해서 4시간씩 증분 이 경우 [12:30, 16:30, 20:30]이 됩니다.			

예를 들어 매일 15:15에 시작해서 2시간씩 증분한 시간에 일정을 실행하려는 경우 [15:15, 17:15, 19:15, 21:15, 23:15]가 됩니다. 다음을 지정하십시오.

```
cron(15 15/2 *)
```

일정 내에서 여러 cron 일정 정의를 생성할 수 있습니다. 예를 들어 다음 AWS CLI 명령은 한 일정에 두 cron 일정을 포함시킵니다.

```
create-snapshot-schedule --schedule-identifier "my-test" --schedule-definition "cron(0 17 SAT,SUN)" "cron(0 9,17 MON-FRI)"
```

수동 스냅샷 수

수동 스냅샷은 언제든지 생성할 수 있습니다. 기본적으로 수동 스냅샷은 클러스터를 삭제한 후에도 무제한으로 보존됩니다. 수동 스냅샷의 보존 기간은 스냅샷을 생성할 때 지정할 수 있으며, 스냅샷을 수정해 변경할 수 있습니다. 보존 기간 변경에 대한 자세한 내용은 [수동 스냅샷 보존 기간 수정](#) 섹션을 참조하세요.

스냅샷을 삭제하면 해당 스냅샷을 참조하는 새 작업을 시작할 수 없습니다. 다만 복원 작업이 진행 중인 경우에는 해당 복원 작업은 완료됩니다.

Amazon Redshift에는 생성할 수 있는 수동 스냅샷의 총 개수를 제한하는 할당량이 있습니다. 이 할당량은 AWS 리전별로 AWS 계정마다 주어집니다. 기본 할당량은 [Amazon Redshift의 할당량 및 제한](#)에 나열됩니다.

스냅샷 스토리지

스냅샷에는 스토리지 요금이 발생하므로 더 이상 필요하지 않을 때 삭제하는 것이 중요합니다.

Amazon Redshift는 해당 스냅샷 보존 기간이 끝나면 자동 및 수동 스냅샷을 삭제합니다. 또한 AWS

Management Console 또는 [batch-delete-cluster-snapshots](#) CLI 명령을 사용하여 수동 스냅샷을 삭제할 수 있습니다.

수동 스냅샷의 보존 기간은 수동 스냅샷 설정을 수정해 변경할 수 있습니다.

Amazon Redshift 콘솔 또는 [describe-storage](#) CLI 명령을 사용하여 스냅샷에서 소비하는 스토리지 양에 대한 정보를 얻을 수 있습니다.

스냅샷에서 테이블 제외

기본적으로 모든 사용자 정의 영구 테이블은 스냅샷에 포함됩니다. 스테이징 테이블 같은 테이블을 백업하지 않아도 된다면 스냅샷을 생성하고 스냅샷에서 복원하는 데 필요한 시간을 크게 줄일 수 있습니다. 또 비백업 테이블을 사용하면 Amazon S3에서 스토리지 공간을 줄일 수 있습니다. 비백업 테이블을 생성하려면 테이블을 생성할 때 BACKUP NO 파라미터를 포함시키십시오. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [CREATE TABLE](#) 및 [CREATE TABLE AS](#)를 참조하세요.

Note

RA3 프로비저닝된 클러스터 및 Amazon Redshift Serverless 작업 그룹에는 백업 없음 테이블이 지원되지 않습니다. RA3 클러스터 또는 서버리스 작업 그룹에서 백업 없음으로 표시된 테이블은 스냅샷을 생성하는 동안 항상 백업되고 스냅샷에서 복원할 때 항상 복원되는 영구 테이블로 처리됩니다. 백업이 없는 테이블의 스냅샷 비용을 방지하려면 스냅샷을 생성하기 전에 자릅니다.

수동 스냅샷 생성

다음과 같이 스냅샷 목록에서 클러스터의 수동 스냅샷을 생성할 수 있습니다. 그 밖에 클러스터 구성 창에서 클러스터 스냅샷을 생성하는 방법도 있습니다. 자세한 내용은 [Amazon Redshift 스냅샷 및 백업](#) 섹션을 참조하세요.

Note

RA3 프로비저닝된 클러스터 및 Amazon Redshift Serverless 작업 그룹에는 백업 없음 테이블이 지원되지 않습니다. RA3 클러스터 또는 서버리스 작업 그룹에서 백업 없음으로 표시된 테이블은 스냅샷을 생성하는 동안 항상 백업되고 스냅샷에서 복원할 때 항상 복원되는 영구 테이블로 처리됩니다. 백업이 없는 테이블의 스냅샷 비용을 방지하려면 스냅샷을 생성하기 전에 자릅니다.

수동 스냅샷을 생성하는 방법

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters), 스냅샷(Snapshots)을 선택한 다음 스냅샷 생성(Create snapshot)을 선택합니다. 수동 스냅샷을 생성하는 스냅샷 페이지가 표시됩니다.
3. 스냅샷 정의의 속성을 입력한 후 스냅샷 생성을 선택합니다. 스냅샷을 사용할 수 있을 때까지 시간이 다소 걸릴 수 있습니다.

스냅샷 일정 생성

Amazon Redshift는 데이터의 증분 스냅샷을 정기적으로 자동 생성하여 Amazon S3에 저장합니다. 또한 원할 때에는 직접 수동으로 데이터 스냅샷을 생성할 수도 있습니다.

Amazon Redshift 콘솔에서 생성된 스냅샷은 모두 스냅샷 목록부터 시작됩니다. 이 목록은 시간 범위, 스냅샷 유형 그리고 스냅샷과 연결된 클러스터를 기준으로 필터링이 가능합니다. 또한 날짜, 크기 및 스냅샷 유형을 기준으로 정렬할 수도 있습니다. 선택하는 스냅샷 유형에 따라 스냅샷 작업에서 사용할 수 있는 옵션이 달라집니다.

스냅샷이 생성되는 시기를 정확하게 제어하려면 스냅샷 일정을 생성하고 해당 일정을 하나 이상의 클러스터에 연결합니다. 클러스터를 생성할 때 또는 클러스터를 수정하여 일정을 연결할 수 있습니다. 자세한 내용은 [자동 스냅샷 일정](#) 섹션을 참조하세요.

스냅샷 일정을 생성하는 방법

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters), 스냅샷(Snapshots)을 선택한 다음 스냅샷 일정(Snapshot schedules) 탭을 선택합니다. 스냅샷 일정이 표시됩니다.
3. Add schedule(일정 추가)을 선택하여 일정을 추가할 페이지를 표시합니다.
4. 일정 정의의 속성을 입력한 후 Add schedule(일정 추가)을 선택합니다.
5. 나타나는 페이지에서 클러스터를 새 스냅샷 일정에 연결한 다음 확인을 선택합니다.

스냅샷 공유

스냅샷에 대한 액세스 권한을 부여하여 기존의 수동 스냅샷을 다른 AWS 고객 계정과 공유할 수 있습니다. 각 스냅샷당 최대 20개 계정, 각각의 AWS Key Management Service(AWS KMS) 키당 최대 100

개 계정에 권한을 부여할 수 있습니다. 다시 말해서 단일 KMS 키로 암호화되어 있는 스냅샷이 10개라고 가정하면 AWS 계정 10개에 각 스냅샷을 복원할 수 있는 권한을 부여하거나, 혹은 최대 100개까지 계정을 추가하지만 각 스냅샷마다 계정 20개를 초과하지 않는 기타 조합에 복원 권한을 부여할 수 있습니다. 그러면 권한이 부여된 계정 중 하나에 사용자로 로그인한 사람이 스냅샷을 설명하거나 복원하여 계정 아래 새로운 Amazon Redshift 클러스터를 생성할 수 있습니다. 예를 들어 프로덕션과 테스트에 별개의 AWS 고객 계정을 사용하는 경우, 사용자가 프로덕션 계정을 사용해 로그인하고 테스트 계정에 있는 사용자들과 스냅샷을 공유할 수 있습니다. 그런 다음 테스트 계정 사용자로 로그인한 사람이 스냅샷을 복원하여 테스트 또는 진단 작업을 위해 테스트 계정이 소유하는 새로운 클러스터를 생성할 수 있습니다.

수동 스냅샷은 스냅샷을 생성한 AWS 고객 계정이 영구적으로 소유합니다. 스냅샷을 소유한 계정의 사용자만이 다른 계정에게 스냅샷에 액세스할 권한을 부여하거나 권한 부여를 취소할 수 있습니다. 권한이 부여된 계정의 사용자는 공유된 스냅샷의 설명 또는 복원만 가능하며 공유된 스냅샷을 복사하거나 삭제할 수는 없습니다. 권한 부여는 스냅샷 소유자가 취소하기 전까지 유효합니다. 권한 부여가 취소되면 이전에 권한이 부여된 사용자는 스냅샷을 볼 수 없으며 해당 스냅샷을 참조하는 새로운 작업을 시작할 수 없습니다. 액세스가 취소될 때 계정에서 스냅샷 복원이 진행 중인 경우에는 복원이 실행되어 완료됩니다. 스냅샷에 대한 권한 부여가 활성화되어 있는 동안은 스냅샷을 삭제할 수 없습니다. 먼저 모든 권한 부여를 취소해야 합니다.

AWS 고객 계정에는 계정이 소유한 스냅샷에 액세스할 권한이 항상 부여됩니다. 소유자 계정에 대한 액세스 권한을 부여하거나 취소하려고 하면 오류가 반환됩니다. 비활성 AWS 고객 계정이 소유한 스냅샷은 복원하거나 설명할 수 없습니다.

AWS 고객 계정에 대한 액세스 권한을 부여한 후에는 해당 계정의 사용자는 작업을 허용하는 정책이 있는 역할을 맡지 않는 한 스냅샷에 대한 어떠한 작업도 수행할 수 없습니다.

- 스냅샷 소유자 계정의 사용자는 스냅샷이 포함된 리소스 사양을 사용하여 스냅샷에 대한 액세스 권한 부여 또는 취소 작업을 허용하는 IAM 정책이 있는 역할을 맡는 경우에만 이러한 작업을 수행할 수 있습니다. 예를 들어 다음 정책은 AWS 계정 012345678912의 사용자 또는 역할이 다른 계정에게 my-snapshot20130829라는 이름의 스냅샷에 대한 액세스 권한을 부여하도록 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift:AuthorizeSnapshotAccess",
```



```

        "redshift:RevokeSnapshotAccess"
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:012345678912:snapshot:*/my-
snapshot20130829"
      ]
    }
  ]
}

```

- 스냅샷을 공유한 AWS 계정의 사용자는 작업을 허용하는 권한이 없는 한 스냅샷에 대한 어떠한 작업도 수행할 수 없습니다. 정책을 역할에 할당하고 역할을 위임하여 이 작업을 수행할 수 있습니다.
- 사용자가 스냅샷을 나열 또는 설명하기 위해서는 DescribeClusterSnapshots 작업을 허용하는 IAM 정책을 가지고 있어야 합니다. 다음 코드에 예가 나와 있습니다:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift:DescribeClusterSnapshots"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}

```

- 스냅샷을 복원하기 위해서는 사용자가 RestoreFromClusterSnapshot 작업을 허용하고 생성하려는 클러스터와 스냅샷이 모두 포함된 리소스 요소가 있는 IAM 정책이 있는 역할을 맡아야 합니다. 예를 들어 계정 012345678912의 사용자가 스냅샷 my-snapshot20130829를 복원하여 클러스터를 생성하기 위해 해당 스냅샷을 계정 219876543210과 공유한 경우, 계정 219876543210의 사용자는 다음과 같은 정책이 있는 역할을 맡아야 합니다.

JSON

```

{
  "Version": "2012-10-17",

```

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "redshift:RestoreFromClusterSnapshot"
    ],
    "Resource": [
      "arn:aws:redshift:us-east-1:012345678912:snapshot:*/my-snapshot20130829",
      "arn:aws:redshift:us-east-1:219876543210:cluster:from-another-account"
    ]
  }
]
}

```

- AWS 계정에서 스냅샷에 대한 액세스가 취소된 후에는 해당 계정의 사용자가 스냅샷에 액세스할 수 없습니다. 해당 계정에 이전에 공유된 스냅샷 리소스에 대한 작업을 허용하는 IAM 정책이 있더라도 마찬가지입니다.

콘솔을 사용하여 클러스터 스냅샷 공유

콘솔에서 다른 사용자에게 자신의 수동 스냅샷에 대한 액세스 권한을 부여하거나, 혹은 나중에 더 이상 필요 없을 때는 해당 액세스 권한을 취소할 수도 있습니다.

다른 계정과 스냅샷을 공유하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters), 스냅샷(Snapshots)을 선택한 다음 공유하려는 수동 스냅샷을 선택합니다.
3. 작업(Actions)에서 수동 스냅샷 설정(Manual snapshot settings)를 선택하여 수동 스냅샷의 속성을 표시합니다.
4. 액세스 관리 섹션에서 공유할 계정을 입력한 다음 저장을 선택합니다.

암호화된 스냅샷 공유에 대한 보안 고려 사항

암호화된 스냅샷에 대한 액세스를 제공하는 경우 Redshift를 사용하려면 스냅샷을 생성하는 데 사용되는 AWS KMS 고객 관리형 키를 복원을 수행하는 계정 또는 계정과 공유해야 합니다. 키를 공유하지 않는 경우 스냅샷을 복원하려고 하면 액세스 거부 오류가 발생합니다. 수신 계정에는 공유 스냅샷을 복원

하기 위한 추가 권한이 필요하지 않습니다. 스냅샷 액세스 권한을 부여하고 키를 공유하는 경우 액세스 권한 부여 ID에 스냅샷을 암호화하는 데 사용되는 키에 대한 `kms:DescribeKey` 권한이 있어야 합니다. 이 권한에 대한 자세한 내용은 [AWS KMS 권한](#)을 참조하세요. 자세한 내용은 Amazon Redshift API 참조 설명서의 [DescriptionKey](#)를 참조하세요.

고객 관리형 키 정책은 프로그래밍 방식으로 또는 AWS Key Management Service 콘솔에서 업데이트할 수 있습니다.

Note

기본 KMS 키를 사용하는 경우 스냅샷을 공유하기 위해 AWS KMS에서 아무런 조치를 취하거나 아무것도 변경할 필요가 없습니다.

암호화된 스냅샷의 AWS KMS 키에 대한 액세스 허용

암호화된 스냅샷의 AWS KMS 고객 관리형 키를 공유하려면 다음 단계를 수행하여 키 정책을 업데이트합니다.

1. KMS 키 정책에서 Principal로 공유하고 있는 AWS 계정의 Amazon 리소스 이름(ARN)을 사용하여 KMS 정책을 업데이트합니다.
2. `kms:Decrypt` 작업을 허용합니다.

다음 키 정책 예에서는 사용자 111122223333이 KMS 키의 소유자이고 사용자 444455556666이 키를 공유하는 계정입니다. 이 키 정책은 사용자 444455556666의 루트 AWS 계정 자격 증명에 대한 ARN을 정책의 Principal로 포함하고 `kms:Decrypt` 작업을 허용하여 AWS 계정이 샘플 KMS 키에 액세스할 수 있도록 합니다.

JSON

```
{
  "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:Decrypt"
    ],
    "Resource": "*"
}
    ]
}

```

고객 관리형 KMS 키에 대한 액세스 권한이 부여된 후에는 암호화된 스냅샷을 복원하는 계정은 AWS Identity and Access Management(IAM) 역할 또는 사용자(아직 없는 경우)를 생성해야 합니다. 또한 해당 AWS 계정은 IAM 역할 또는 사용자가 해당 KMS 키를 사용하여 암호화된 데이터베이스 스냅샷을 복원할 수 있도록 허용하는 IAM 정책을 해당 역할 또는 사용자에게 연결해야 합니다.

AWS KMS 키에 대한 액세스 권한을 부여하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 가이드에서 [다른 계정의 사용자가 KMS 키를 사용하도록 허용](#) 섹션을 참조하세요.

키 정책의 개요는 [Amazon Redshift에서 AWS KMS 사용 방법](#)을 참조하세요.

자동 스냅샷 복사

자동 스냅샷은 보존 기간이 종료되었을 때, 자동 스냅샷을 비활성화할 때 또는 클러스터를 삭제할 때 자동으로 삭제됩니다. 하지만 자동 스냅샷을 보관하고 싶다면 수동 스냅샷으로 복사하면 됩니다.

자동 스냅샷을 복사하는 방법

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters), 스냅샷(Snapshots)을 선택한 다음 복사할 스냅샷을 선택합니다.
3. 작업에서 자동 스냅샷 복사를 선택하여 스냅샷을 복사합니다.
4. 새 스냅샷의 속성을 업데이트한 다음 복사를 선택합니다.

다른 AWS 리전에 스냅샷 복사

클러스터의 스냅샷(자동 또는 수동)을 다른 AWS 리전에 자동으로 복사하도록 Amazon Redshift를 구성할 수 있습니다. 클러스터의 기본 AWS 리전에서 스냅샷이 생성되면 보조 AWS 리전으로 복사됩니다. 이 두 AWS 리전을 각각 소스 AWS 리전과 대상 AWS 리전이라고 합니다. 다른 AWS 리전에 스냅샷 사본을 저장하면, 무언가 기본 AWS 리전에 영향을 줄 때 최근 데이터에서 클러스터를 복원할 수 있습니다. 한 번에 하나의 대상 AWS 리전으로만 스냅샷을 복사하도록 클러스터를 구성할 수 있습니다. Amazon Redshift 리전 목록은 Amazon Web Services 일반 참조의 [리전 및 엔드포인트](#)를 참조하세요.

Amazon Redshift가 스냅샷을 다른 AWS 리전으로 자동 복사하도록 하려면, 스냅샷을 복사할 대상 AWS 리전을 지정합니다. 자동 스냅샷의 경우, 대상 AWS 리전에 스냅샷을 보관할 보존 기간을 지정할 수도 있습니다. 자동 스냅샷이 대상 AWS 리전에 복사되어 보존 기간에 도달한 후에는 대상 AWS 리전에서 삭제됩니다. 이렇게 하면 스냅샷 사용량이 줄어듭니다. 대상 AWS 리전에 자동 스냅샷을 더 짧게 또는 더 길게 보관하려면 이 보존 기간을 변경합니다.

대상 AWS 리전에 복사된 자동 스냅샷에 대해 설정하는 보존 기간은 원본 AWS 리전에 있는 자동 스냅샷의 보존 기간과 별개입니다. 복사된 스냅샷의 기본 보존 기간은 7일입니다. 7일의 기간은 자동 스냅샷에만 적용됩니다. 소스 및 대상 AWS 리전 모두에서 스냅샷 보존 기간 종료 시 또는 수동으로 삭제할 때 수동 스냅샷이 삭제됩니다.

언제든 클러스터의 자동 스냅샷 복사를 비활성화할 수 있습니다. 이 기능을 사용 중지하면 더 이상 스냅샷이 원본 AWS 리전에서 대상 AWS 리전으로 복사되지 않습니다. 대상 AWS 리전에 복사한 자동 스냅샷은 수동 스냅샷 사본을 생성하지 않는 한 보존 기한 한도에 도달하면 삭제됩니다. 이러한 수동 스냅샷과 대상 AWS 리전에서 복사한 수동 스냅샷은 수동으로 삭제할 때까지 대상 AWS 리전에서 보존됩니다.

스냅샷을 복사할 대상 AWS 리전을 변경하려면 먼저 자동 복사 기능을 사용 중지합니다. 그런 다음, 새 대상 AWS 리전을 지정하여 다시 사용합니다.

대상 AWS 리전으로 복사된 스냅샷은 활성 상태가 되고 복원에 사용할 수 있습니다.

AWS KMS 암호화된 클러스터의 스냅샷을 다른 AWS 리전으로 복사하려면 Amazon Redshift가 대상 AWS 리전에서 고객 관리형 키를 사용할 수 있는 권한을 생성합니다. 그런 다음 소스 AWS 리전에서 스냅샷 복사를 사용할 때 해당 권한을 선택합니다. 스냅샷 복사 권한 구성에 대한 자세한 내용은 [AWS KMS로 암호화된 스냅샷을 다른 AWS 리전으로 복사](#) 단원을 참조하십시오.

스냅샷에서 클러스터 복원

스냅샷에는 클러스터에서 실행 중인 모든 데이터베이스의 데이터가 포함됩니다. 또한 노드 수, 노드 유형, 관리자 이름을 포함한 클러스터에 대한 정보도 포함합니다. 스냅샷에서 클러스터를 복원하는 경우

Amazon Redshift는 클러스터 정보를 사용하여 새 클러스터를 생성합니다. 그런 다음 스냅샷 데이터에서 모든 데이터베이스를 복원합니다.

 Note

RA3 프로비저닝된 클러스터 및 Amazon Redshift Serverless 작업 그룹에는 백업 없음 테이블이 지원되지 않습니다. RA3 클러스터 또는 서버리스 작업 그룹에서 백업 없음으로 표시된 테이블은 스냅샷을 생성하는 동안 항상 백업되고 스냅샷에서 복원할 때 항상 복원되는 영구 테이블로 처리됩니다.

원래 스냅샷에서 생성된 새 클러스터의 경우 노드 유형과 노드 수 등의 구성을 선택할 수 있습니다. 요청에서 다른 가용 영역을 지정하지 않는 한 클러스터는 동일한 AWS 리전과 시스템에서 선택한 임의의 가용 영역에서 복원됩니다. 스냅샷에서 클러스터를 복원할 때, 필요한 경우 새 클러스터에 대해 호환되는 유지 관리 트랙을 선택할 수 있습니다.

 Note

다른 구성으로 클러스터에 스냅샷을 복원할 때 클러스터 버전이 1.0.10013 이상인 클러스터에서 스냅샷을 생성해야 합니다.

복원이 진행 중일 때 이벤트는 일반적으로 다음 순서로 발생합니다.

1. RESTORE_STARTED – 복원 프로세스가 시작되면 REDSHIFT-EVENT-2008이 전송됩니다.
2. RESTORE_SUCCEEDED – 새 클러스터가 생성되면 REDSHIFT-EVENT-3003이 전송됩니다.

클러스터는 쿼리에 사용할 수 있습니다.

3. DATA_TRANSFER_COMPLETED – 데이터 전송이 완료되면 REDSHIFT-EVENT-3537이 전송됩니다.

 Note

RA3 클러스터는 RESTORE_STARTED 및 RESTORE_SUCCEEDED 이벤트만 내보냅니다. RA3 노드 유형은 Amazon Redshift 관리형 스토리지에 데이터를 저장하기 때문에 RESTORE가 성공한 후에 수행할 명시적 데이터 전송이 없습니다. RA3 노드를 사용하면 일반 쿼리 처리의 일부로 RA3 노드와 Amazon Redshift 관리형 스토리지 간에 데이터가 지속적으로 전송됩니다.

다. RA3 노드는 핫 데이터를 로컬로 캐시하고 쿼리 빈도가 낮은 블록을 Amazon Redshift 관리형 스토리지에 자동으로 유지합니다.

[DescribeClusters](#) API 작업을 호출하거나 AWS Management Console에서 클러스터 세부 정보를 확인하면 복원 진행률을 모니터링할 수 있습니다. 진행 중인 복원의 경우, 이들은 스냅샷 크기, 전송 속도, 경과 시간, 남은 예상 시간 같은 정보를 표시합니다. 이들 지표에 대한 설명은 [RestoreStatus](#)를 참조하십시오.

활성 클러스터는 스냅샷을 사용하여 이전 상태로 되돌릴 수 없습니다.

 Note

새 클러스터로 스냅샷을 복원하는 경우, 다른 값을 지정하지 않는 한 기본 보안 그룹과 파라미터 그룹이 사용됩니다.

다음과 같은 이유로 다른 구성으로 클러스터에 스냅샷을 복원할 수 있습니다.

- 클러스터가 작은 노드 유형으로 구성되어 있는 상태에서, 더 적은 수의 노드로 더 큰 노드 유형에 통합하려는 경우.
- 워크로드를 모니터링한 결과, 더 큰 CPU 및 스토리지의 노드 유형으로 이동해야 한다고 판단한 경우.
- 다른 노드 유형으로 테스트 워크로드의 성능을 측정하려는 경우.

복원에는 다음과 같은 제약이 있습니다.

- 새 노드 구성에는 기존 데이터를 저장하기에 충분한 스토리지가 있어야 합니다. 노드를 추가하더라도 데이터가 다시 분산되는 방식 때문에 새 구성에 스토리지가 부족할 수 있습니다.
- 복원 작업은 새 클러스터의 클러스터 버전과 호환되는 클러스터 버전에서 스냅샷이 생성되었는지 확인합니다. 새 클러스터의 버전 수준이 너무 이른 경우 복원 작업이 실패하고 오류 메시지에 추가 정보가 보고됩니다.
- 복원할 수 있는 가능한 구성(노드 수 및 노드 유형)은 원래 클러스터의 노드 수와 새 클러스터의 대상 노드 유형에 따라 결정됩니다. 사용 가능한 구성을 확인하기 위해 Amazon Redshift 콘솔 또는 `describe-node-configuration-options` AWS CLI 명령을 `action-type restore-cluster`와 함께 사용할 수 있습니다. Amazon Redshift 콘솔을 사용한 복원에 대한 자세한 내용은 [스냅샷에서 클러스터 복원](#) 섹션을 참조하세요.

다음 단계는 AWS CLI를 사용하여 많은 노드가 있는 클러스터를 가져다 더 적은 노드 수의 더 큰 노드 유형으로 통합합니다. 이 예에서는 24개의 노드의 소스 클러스터로 시작합니다. 이 경우, 이미 이 클러스터의 스냅샷을 생성했고 더 큰 노드 유형으로 복원하려고 한다고 가정하십시오.

1. 다음 명령을 실행하여 24노드 클러스터의 세부 정보를 파악합니다.

```
aws redshift describe-clusters --region eu-west-1 --cluster-identifier
mycluster-123456789012
```

2. 다음 명령을 실행하여 스냅샷의 세부 정보를 파악합니다.

```
aws redshift describe-cluster-snapshots --region eu-west-1 --snapshot-identifier
mycluster-snapshot
```

3. 이 스냅샷에 사용 가능한 옵션을 설명하려면 다음 명령을 실행하십시오.

```
aws redshift describe-node-configuration-options --snapshot-identifier mycluster-
snapshot --region eu-west-1 --action-type restore-cluster
```

이 명령은 각 옵션의 권장 노드 유형, 노드 수, 디스크 사용률이 포함된 옵션 목록을 반환합니다. 이 예에서 앞의 명령은 다음과 같이 가능한 노드 구성을 나열합니다. 3노드 클러스터로 복원하기로 했습니다.

```
{
  "NodeConfigurationOptionList": [
    {
      "EstimatedDiskUtilizationPercent": 65.26134808858235,
      "NodeType": "dc2.large",
      "NumberOfNodes": 24
    },
    {
      "EstimatedDiskUtilizationPercent": 32.630674044291176,
      "NodeType": "dc2.large",
      "NumberOfNodes": 48
    },
    {
      "EstimatedDiskUtilizationPercent": 65.26134808858235,
      "NodeType": "dc2.8xlarge",
      "NumberOfNodes": 3
    },
    {
```



```

        "EstimatedDiskUtilizationPercent": 48.94601106643677,
        "NodeType": "dc2.8xlarge",
        "NumberOfNodes": 4
    },
    {
        "EstimatedDiskUtilizationPercent": 39.156808853149414,
        "NodeType": "dc2.8xlarge",
        "NumberOfNodes": 5
    },
    {
        "EstimatedDiskUtilizationPercent": 32.630674044291176,
        "NodeType": "dc2.8xlarge",
        "NumberOfNodes": 6
    }
]
}

```

4. 선택한 클러스터 구성으로 스냅샷을 복원하려면 다음 명령을 실행하십시오. 이 클러스터가 복원된 후 소스 클러스터와 내용은 동일하지만 데이터는 3개의 dc2.8xlarge 노드로 통합되었습니다.

```

aws redshift restore-from-cluster-snapshot --region eu-west-1 --snapshot-identifier
mycluster-snapshot --cluster-identifier mycluster-123456789012-x --node-type
dc2.8xlarge --number-of-nodes 3

```

예약 노드(예: DC2 예약 노드)가 있는 경우 RA3 예약 노드로 업그레이드할 수 있습니다. 스냅샷에서 복원하거나 탄력적 크기 조정을 수행할 때 이 작업을 수행할 수 있습니다. 콘솔을 사용하여 이 프로세스를 안내할 수 있습니다. RA3 노드로 업그레이드에 대한 자세한 내용은 [RA3 노드 유형으로 업그레이드](#)를 참조하세요.

콘솔에서 스냅샷의 클러스터를 복원하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters), 스냅샷(Snapshots)을 선택한 다음 복원할 스냅샷을 선택합니다.
3. [스냅샷에서 복원(Restore from snapshot)]을 선택하여 스냅샷 정보로 생성할 새 클러스터의 [클러스터 구성(Cluster configuration)] 및 [클러스터 세부 정보(Cluster details)] 값을 봅니다.
4. 새 클러스터의 속성을 업데이트한 다음 스냅샷에서 클러스터 복원을 선택합니다.

클러스터 스냅샷을 복원한 후 복원된 데이터 웨어하우스는 스냅샷을 생성할 때 사용한 것과 동일한 사용자 지정 AWS KMS 키로 암호화됩니다. 스냅샷에 사용자 지정 KMS 키가 없는 경우 Amazon Redshift의 백업 암호화 로직은 다음 요인에 따라 달라집니다.

- 스냅샷을 복원할 Amazon Redshift 데이터 웨어하우스의 유형입니다.
- 스냅샷이 생성된 시점의 클러스터 암호화 유형입니다.

클러스터 스냅샷에서 데이터 웨어하우스를 복원한 후 데이터 웨어하우스가 어떻게 암호화되는지 알아보려면 다음 표를 참조하세요.

대상 유형	스냅샷 암호화 유형	대상 암호화 유형
프로비저닝된 클러스터	AWS 관리형 키를 사용하여 암호화	AWS 관리형 키를 사용하여 암호화
프로비저닝된 클러스터	AWS 소유 키를 사용하여 암호화	AWS 소유 키를 사용하여 암호화
서버리스 네임스페이스	AWS 관리형 키를 사용하여 암호화	AWS 소유 키를 사용하여 암호화
서버리스 네임스페이스	AWS 소유 키를 사용하여 암호화	AWS 소유 키를 사용하여 암호화

스냅샷을 생성할 때 AWS Secrets Manager가 클러스터의 관리자 암호를 관리했다면 관리자 암호를 관리하는 데 AWS Secrets Manager를 계속 사용해야 합니다. 복원 후 클러스터 세부 정보 페이지에서 클러스터의 관리자 보안 인증 정보를 업데이트하여 보안 암호 사용을 취소할 수 있습니다.

예약 노드가 있는 경우 RA3 예약 노드로 업그레이드할 수 있습니다. 스냅샷에서 복원하거나 탄력적 크기 조정을 수행할 때 이 작업을 수행할 수 있습니다. 콘솔을 사용하여 이 프로세스를 안내할 수 있습니다. RA3 노드로 업그레이드에 대한 자세한 내용은 [RA3 노드 유형으로 업그레이드](#)를 참조하세요.

스냅샷에서 테이블 복원

전체 클러스터를 복원하는 대신 스냅샷에서 단일 테이블을 복원할 수 있습니다. 스냅샷에서 단일 테이블을 복원할 때는 원본 스냅샷, 데이터베이스, 스키마, 테이블 이름과 대상 데이터베이스, 스키마 및 복원될 테이블의 새 테이블 이름을 지정합니다.

Note

RA3 프로비저닝된 클러스터 및 Amazon Redshift Serverless 작업 그룹에는 백업 없음 테이블이 지원되지 않습니다. RA3 클러스터 또는 서버리스 작업 그룹에서 백업 없음으로 표시된 테이블은 스냅샷을 생성하는 동안 항상 백업되고 스냅샷에서 복원할 때 항상 복원되는 영구 테이블로 처리됩니다. 그러나 백업 없음 테이블의 선택적 복원은 지원되지 않습니다.

새 테이블 이름은 기존 테이블 이름과 동일하면 안 됩니다. 기존 테이블을 스냅샷에서 복원한 테이블로 교체하려면 스냅샷에서 테이블을 복원하기 전에 기존 테이블의 이름을 변경하고 끌어 놓습니다.

대상 테이블은 외래 키를 제외하고 원본 테이블의 열 정의, 테이블 속성 및 열 속성을 사용하여 생성됩니다. 종속성으로 인한 충돌을 방지하기 위해 대상 테이블은 원본 테이블에서 외래 키를 상속하지 않습니다. 뷰 또는 원본 테이블에서 부여된 권한 같은 종속성은 대상 테이블에 적용되지 않습니다.

원본 테이블의 소유자가 존재하는 경우, 해당 데이터베이스 사용자가 복원된 테이블의 소유자가 됩니다. 단, 지정된 데이터베이스와 스키마에서 관계의 소유자가 되기에 충분한 권한이 있어야 합니다. 그렇지 않은 경우, 복원된 테이블은 클러스터가 시작될 때 생성된 관리자가 소유합니다.

복원된 테이블은 백업이 생성된 시점의 상태로 되돌아갑니다. 여기에는 Amazon Redshift의 [직렬화 격리](#) 준수에 의해 정의된 트랜잭션 가시성 규칙이 포함됩니다. 이는 백업 후 시작되는 인플라이트 트랜잭션을 즉시 볼 수 있음을 뜻합니다.

스냅샷에서의 테이블 복원은 다음과 같은 제한이 있습니다.

- 테이블은 현재 실행 중인 활성 클러스터로만 복원할 수 있으며, 해당 클러스터에서 생성된 스냅샷에서만 복원할 수 있습니다.
- 한 번에 하나의 테이블만 복원할 수 있습니다.
- 클러스터 크기 조정 전에 생성된 클러스터 스냅샷에서는 테이블을 복원할 수 없습니다. 노드 유형이 변경되지 않은 경우 탄력적인 크기 조정 후 테이블을 복원할 수 있다는 점은 예외입니다.
- 뷰 또는 원본 테이블에서 부여된 권한 같은 종속성은 대상 테이블에 적용되지 않습니다.
- 복원하는 테이블에 대해 행 수준 보안이 켜져 있는 경우 Amazon Redshift에서 행 수준 보안이 켜진 상태로 테이블을 복원합니다.

스냅샷에서 테이블을 복원하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.

2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 테이블을 복원에 사용할 클러스터를 선택합니다.
3. 작업에서 테이블 복원을 선택하여 테이블 복원 페이지를 표시합니다.
4. 사용할 스냅샷, 소스 테이블 및 대상 테이블에 대한 정보를 입력한 후 테이블 복원을 선택합니다.

Example예: AWS CLI를 사용하여 스냅샷에서 테이블 복원

다음 예는 `restore-table-from-cluster-snapshot` AWS CLI 명령을 사용하여 `my-source-table`의 `sample-database` 스키마에서 `my-snapshot-id` 테이블을 복원합니다. AWS CLI 명령 `describe-table-restore-status`를 사용하여 복원 작업의 상태를 검토할 수 있습니다. 이 예는 새로운 테이블 이름 `mycluster-example`을 사용하여 `my-new-table` 클러스터로 스냅샷을 복원합니다.

```
aws redshift restore-table-from-cluster-snapshot --cluster-identifier mycluster-
example
--new-table-name my-new-table
--snapshot-identifier my-snapshot-id
--source-database-name sample-
database
--source-table-name my-source-table
```

스냅샷에서 서버리스 네임스페이스 복원

스냅샷에서 서버리스 네임스페이스를 복원하면 네임스페이스의 모든 데이터베이스가 스냅샷의 데이터베이스로 바뀝니다. 서버리스 스냅샷에 대한 자세한 내용은 [스냅샷 및 복구 시점](#)을 참조하세요. Amazon Redshift는 프로비저닝된 클러스터 스냅샷을 Amazon Redshift Serverless 네임스페이스로 복원할 때 인터리브된 키가 있는 테이블을 복합 키로 자동 변환합니다. 정렬 키에 대한 자세한 내용은 [정렬 키 작업](#)을 참조하세요.

프로비저닝된 클러스터에서 서버리스 네임스페이스로 스냅샷 복원

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 Clusters(클러스터), Snapshots(스냅샷)을 선택한 다음 사용할 스냅샷을 선택합니다.
3. Restore from snapshot(스냅샷에서 복원), Restore to serverless namespace(서버리스 네임스페이스로 복원)를 선택합니다.

4. 복원할 네임스페이스를 선택합니다.
5. 스냅샷에서 복원할 것인지 확인합니다. Restore(복원)를 선택합니다. 이 작업은 서버리스 네임스페이스의 모든 데이터베이스를 프로비저닝된 클러스터의 데이터로 바꿉니다.

암호화되지 않은 클러스터에 대한 리전 간 스냅샷 복사 구성

클러스터 스냅샷을 다른 AWS 리전으로 복사하도록 Amazon Redshift를 구성할 수 있습니다. 리전 간 스냅샷 복사를 구성하려면 각 클러스터에 대해 이 복사 기능을 사용하고 스냅샷을 복사할 위치와 복사된 자동 또는 수동 스냅샷을 대상 AWS 리전에 보관할 기간을 구성해야 합니다. 클러스터에 대해 리전 간 복사 기능을 사용하면 새로운 수동 및 자동 스냅샷이 지정된 AWS 리전으로 모두 복사됩니다. 복사된 스냅샷의 이름에는 **copy:**라는 접두사가 첨부됩니다.

리전 간 스냅샷을 구성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 스냅샷을 이동할 클러스터를 선택합니다.
3. [작업(Actions)]에서 [교차 리전 스냅샷 구성(Configure cross-region snapshot)]을 선택합니다.

[교차 리전 구성(Configure cross-Region)] 대화 상자가 나타납니다.
4. [스냅샷 복사(Copy snapshots)]에서 [예(Yes)]를 선택합니다.
5. 대상 AWS 리전에서 스냅샷을 복사할 AWS 리전을 선택합니다.
6. [자동 스냅샷 보존 기간(일)(Automated snapshot retention period (days))]에서 자동 스냅샷을 삭제 이전까지 대상 AWS 리전에 보관할 일 수를 선택합니다.
7. [수동 스냅샷 보존 기간(Manual snapshot retention period)]에서 수동 스냅샷을 삭제 이전까지 대상 AWS 리전에 보관할 일 수를 나타내는 값을 선택합니다. [사용자 정의 값(Custom value)]을 선택하는 경우 보존 기간은 1~3653일이어야 합니다.
8. 저장을 선택합니다.

AWS KMS 암호화 클러스터에 대한 리전 간 스냅샷 복사 구성

Amazon Redshift 클러스터를 시작할 때 대상 AWS 리전의 계정에서 루트 키에 대한 스냅샷 복사 권한 부여를 구성할 수 있습니다. 권한 부여를 구성하지 않으면 대상 리전의 스냅샷이 기본 AWS 소유 키로 암호화됩니다. 이렇게 하면 Amazon Redshift가 대상 AWS 리전에서 암호화 작업을 수행할 수 있습니다.

다음 절차에서는 AWS KMS 암호화 클러스터에 대한 리전 간 스냅샷 복사 기능을 활성화하는 프로세스를 설명합니다. Amazon Redshift의 암호화와 스냅샷 복사 권한에 대한 자세한 내용은 [AWS KMS로 암호화된 스냅샷을 다른 AWS 리전으로 복사](#) 섹션을 참조하세요.

AWS KMS 암호화 클러스터에 대한 리전 간 스냅샷 복사를 구성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 스냅샷을 이동할 클러스터를 선택합니다.
3. [작업(Actions)]에서 [교차 리전 스냅샷 구성(Configure cross-region snapshot)]을 선택합니다.

[교차 리전 구성(Configure cross-Region)] 대화 상자가 나타납니다.
4. [스냅샷 복사(Copy snapshots)]에서 [예(Yes)]를 선택합니다.
5. 대상 AWS 리전에서 스냅샷을 복사할 AWS 리전을 선택합니다.
6. [자동 스냅샷 보존 기간(일)(Automated snapshot retention period (days))]에서 자동 스냅샷을 삭제 이전까지 대상 AWS 리전에 보관할 일 수를 선택합니다.
7. [수동 스냅샷 보존 기간(Manual snapshot retention period)]에서 수동 스냅샷을 삭제 이전까지 대상 AWS 리전에 보관할 일 수를 나타내는 값을 선택합니다. [사용자 정의 값(Custom value)]을 선택하는 경우 보존 기간은 1~3653일이어야 합니다.
8. 저장을 선택합니다.

수동 스냅샷 보존 기간 수정

수동 스냅샷의 보존 기간은 스냅샷 설정을 수정해 변경할 수 있습니다.

수동 스냅샷 보존 기간을 변경하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters), 스냅샷(Snapshots)을 선택한 다음 변경하려는 수동 스냅샷을 선택합니다.
3. 작업(Actions)에서 수동 스냅샷 설정(Manual snapshot settings)를 선택하여 수동 스냅샷의 속성을 표시합니다.
4. 스냅샷 정의의 수정된 속성을 입력한 후 저장(Save)을 선택합니다.

리전 간 스냅샷 복사본의 보존 기간 수정

리전 간 스냅샷 복사를 구성한 후 나중에 설정을 변경해야 할 수 있습니다. 특히 보존 기간은 일 수를 새로 선택한 후 변경 내용을 저장하면 쉽게 변경할 수 있습니다.

Warning

리전 간 스냅샷 복사를 구성한 후에는 대상 AWS 리전을 수정할 수 없습니다. 스냅샷을 다른 AWS 리전에 복사하려면 먼저 리전 간 스냅샷 복사를 사용 중지합니다. 그런 다음 새로운 대상 AWS 리전 및 보존 기간으로 다시 사용합니다. 리전 간 스냅샷 복사를 비활성화한 후에는 복사된 자동 스냅샷이 모두 삭제됩니다. 따라서 리전 간 스냅샷 복사를 비활성화하기 전에 보존하려는 스냅샷이 있는지 확인한 후 수동 스냅샷에 복사해야 합니다.

리전 간 스냅샷을 수정하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 스냅샷을 수정할 클러스터를 선택합니다.
3. Actions(작업)에서 [Configure cross-region snapshot(교차 리전 스냅샷 구성)을 선택하여 스냅샷의 속성을 표시합니다.
4. 스냅샷 정의의 수정된 속성을 입력한 후 저장(Save)을 선택합니다.

수동 스냅샷 삭제

스냅샷 목록에서 스냅샷을 하나 이상 선택해 수동 스냅샷을 삭제할 수 있습니다.

스냅샷을 삭제하는 방법

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters), 스냅샷(Snapshots)을 선택한 다음 삭제할 스냅샷을 선택합니다.
3. 작업에서 스냅샷 삭제를 선택하여 스냅샷을 삭제합니다.
4. 나열된 스냅샷의 삭제를 확인한 다음 삭제를 선택합니다.

AWS Glue Data Catalog에 클러스터 등록

AWS Glue Data Catalog에 전체 클러스터를 등록하고 AWS Glue에서 관리하는 카탈로그를 만들 수 있습니다. Apache Iceberg REST API를 지원하는 SQL 엔진을 사용하여 이러한 카탈로그에 액세스할 수 있습니다. Amazon Redshift에서 Apache Iceberg 호환 카탈로그를 만드는 방법에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Apache Iceberg compatibility for Amazon Redshift](#) 섹션을 참조하세요.

AWS Glue Data Catalog에 클러스터를 등록하는 방법

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다. 클러스터가 없는 경우 클러스터 생성(Create cluster)을 선택하여 클러스터를 생성합니다.
3. 등록할 클러스터 이름을 선택합니다.
4. 작업에서 AWS Glue Data Catalog에 등록을 선택합니다. AWS Glue Data Catalog에 등록 팝업 상자가 나타납니다.
5. 대상 계정 ID 아래에서 클러스터를 등록할 AWS 계정 ID를 입력합니다. AWS Glue Data Catalog에 카탈로그를 보관할 계정 ID입니다.
6. 네임스페이스 등록 이름 아래에 이름을 입력합니다. 이것이 Data Catalog에서 클러스터 이름이 됩니다.
7. 등록을 선택합니다. 이제 AWS Lake Formation 콘솔로 이동하게 됩니다.
8. AWS Lake Formation의 카탈로그 만들기 프로세스를 따릅니다. 카탈로그 만들기에 대한 자세한 내용은 AWS Lake Formation 개발자 안내서의 [Bringing Amazon Redshift data into the AWS Glue Data Catalog](#) 섹션을 참조하세요.

다중 AZ 배포

Amazon Redshift는 프로비저닝된 RA3 클러스터에 다중 가용 영역(다중 AZ) 배포를 지원합니다. 다중 AZ 배포를 사용하면 Amazon Redshift 데이터 웨어하우스는 가용 영역에서 예기치 않은 이벤트가 발생하는 장애 시나리오에서 계속 작동할 수 있습니다. 다중 AZ 배포는 두 개의 가용 영역(AZ)에 컴퓨팅 리소스를 배포하며, 이 컴퓨팅 리소스는 단일 엔드포인트를 통해 액세스할 수 있습니다. 전체 가용 영역에 장애가 발생하는 경우 두 번째 가용 영역의 나머지 컴퓨팅 리소스를 계속해서 워크로드를 처리하는데 사용할 수 있습니다. Amazon Redshift는 다중 AZ 데이터 웨어하우스를 실행할 때 RA3에 대해 동일

한 시간당 컴퓨팅 요금을 부과합니다. 스토리지 비용은 AWS 리전 내의 모든 가용 영역에서 공유되므로 동일하게 유지됩니다.

현재 Amazon Redshift는 장애 발생 시 데이터를 최신 상태로 유지할 수 있는 제로 Recovery Point Objective(RPO)를 지원합니다. 다중 AZ 배포를 통해 Amazon Redshift는 기존 복구 기능을 더욱 향상하고 Recovery Time Objective(RTO)를 단축합니다. 이것이 가능한 이유는 다중 AZ 배포는 장애 또는 재해로부터 더 빠르게 복구할 수 있어 Amazon Redshift 서비스 수준에 관한 계약(SLA)을 99.99%까지 높일 수 있기 때문입니다. 단일 AZ 데이터 웨어하우스 경우는 99.9%입니다.

다중 AZ 배포 설정

다중 AZ 배포를 설정하려면 다중 AZ 옵션을 선택하고 가용 영역에서 프로비저닝할 컴퓨팅 노드 수를 지정합니다. Amazon Redshift는 두 가용 영역에 동일한 컴퓨팅 리소스를 자동으로 배포하며 정상 운영 중에는 모든 컴퓨팅 리소스를 읽기 및 쓰기 처리에 항상 사용할 수 있습니다. 따라서 다중 AZ 배포가 단일 엔드포인트가 있는 단일 데이터 웨어하우스의 역할을 할 수 있으므로 재해가 발생했을 때 애플리케이션을 변경할 필요가 없습니다. 다중 AZ 배포는 하나의 가용 영역에만 있는 컴퓨팅 리소스를 사용하여 개별 쿼리를 처리하지만 여러 개의 동시 쿼리 처리를 두 가용 영역에 자동으로 분산하여 동시성이 높은 워크로드의 전체 처리량을 높일 수 있습니다.

기존 단일 AZ 데이터 웨어하우스를 다중 AZ 데이터 웨어하우스로 변환하거나 그 반대로 변환할 수도 있습니다. 두 번째 가용 영역에 추가 컴퓨팅 리소스가 프로비저닝된다는 점을 제외하면 모든 것이 동일합니다. 기존 단일 AZ 클러스터에서 다중 AZ로 마이그레이션할 때 단일 쿼리 성능을 쉽게 유지할 수 있도록 필요한 클러스터 노드 수를 두 배로 늘려야 할 수 있습니다. 다중 AZ 데이터 웨어하우스를 사용하면 사용 가능한 컴퓨팅 리소스의 양이 두 배로 늘어나 대부분의 워크로드에서 전체 쿼리 처리 처리량이 증가하는 것을 볼 수 있습니다.

가용 영역에서 장애가 발생하는 경우 Amazon Redshift는 나머지 가용 영역의 리소스를 자동으로 사용하여 계속 작동합니다. 그러나 사용자 연결이 끊어질 수 있으므로 다시 연결해야 합니다. 또한 장애가 발생한 가용 영역에서 실행 중이던 쿼리는 실패할 수 있으며 재시도해야 합니다. 그러나 클러스터에 다시 연결하고 쿼리를 즉시 다시 예약할 수 있습니다. 그러면 Amazon Redshift가 나머지 가용 영역에서 쿼리를 처리합니다. 다중 AZ 데이터 웨어하우스가 복구되는 동안 오류가 발생한 시점 또는 이후에 실행된 쿼리에서 런타임 지연이 발생할 수 있습니다.

Note

성능과 가용성을 높이려면 다중 AZ 클러스터와 함께 SNAPSHOT ISOLATION을 사용하는 것이 좋습니다. 자세한 내용은 [Creating a database](#)(데이터베이스 생성)를 참조하세요.

제한 사항

다중 AZ 데이터 웨어하우스는 다중 AZ 데이터 웨어하우스에 적용되는 다음 제한을 제외하고 단일 AZ 데이터 웨어하우스와 동일한 기능을 갖습니다.

- 암호화되지 않은 다중 AZ 데이터 웨어하우스를 생성할 수 없습니다. 새 다중 AZ 데이터 웨어하우스를 생성하거나, 단일 AZ 데이터 웨어하우스를 다중 AZ 데이터 웨어하우스로 변환하거나, 단일 AZ 데이터 웨어하우스를 다중 AZ 데이터 웨어하우스로 변환할 때는 반드시 암호화를 추가해야 합니다.
- RA3 인스턴스 유형에 대해 단일 노드 다중 AZ 배포를 생성할 수 없습니다. 다중 AZ 배포를 생성하는 동안 가용 영역당 2개 이상의 노드를 선택합니다.
- Amazon Redshift는 세 개 미만의 가용 영역을 지원할 수 있는 서브넷 구성을 지원하지 않습니다. 즉, 구성된 서브넷 그룹에는 세 개 이상의 서브넷이 더 필요합니다.
- 다중 AZ 배포를 다른 가용 영역으로 재배포할 수 없습니다. 다중 AZ 배포를 사용하는 경우 Amazon Redshift에서 재배포를 자동으로 결정하고 수행합니다.
- 다중 AZ 배포를 일시 중지하거나 재개할 수 없습니다.
- 지원되는 포트 범위인 5431~5455 및 8191~8215를 벗어나면 다중 AZ 배포를 실행할 수 없습니다.
- 다중 AZ 배포에서는 시스템 모니터링 뷰(SYS_* 뷰)만 지원하므로 STL, SVCS, SVL, SVV, STV 뷰를 사용할 수 없습니다. 시스템 모니터링 뷰(SYS_* 뷰)를 사용하도록 모니터링 쿼리를 변경하세요.
- 다중 AZ가 활성화된 기존 클러스터에는 탄력적 IP 주소를 연결할 수 없습니다.
- 탄력적 IP 주소가 연결된 클러스터를 단일 AZ에서 다중 AZ로 변환할 수 없습니다.
- Amazon Redshift 다중 AZ 배포는 다음 AWS 리전에서 사용할 수 있습니다.
 - 미국 동부(오하이오)(us-east-2)
 - 미국 동부(버지니아 북부)(us-east-1)
 - 미국 서부(오리건)(us-west-2)
 - 아프리카(케이프타운)(af-south-1)
 - 아시아 태평양(홍콩)(ap-east-1)
 - 아시아 태평양(타이베이)(ap-east-2)
 - 아시아 태평양(하이데라바드)(ap-south-2)
 - 아시아 태평양(자카르타) (ap-southeast-3)
 - 아시아 태평양(말레이시아)(ap-southeast-5)
 - 아시아 태평양(멜버른)(ap-southeast-4)
 - 아시아 태평양(뭄바이)(ap-south-1)
 - 아시아 태평양(오사카) (ap-northeast-3)

- 아시아 태평양(서울)(ap-northeast-2)
- 아시아 태평양(싱가포르)(ap-southeast-1)
- 아시아 태평양(시드니)(ap-southeast-2)
- 아시아 태평양(뉴질랜드)(ap-southeast-6)
- 아시아 태평양(태국)(ap-southeast-7)
- 아시아 태평양(도쿄)(ap-northeast-1)
- 캐나다(중부)(ca-central-1)
- 중국(베이징)(cn-north-1)
- 중국(닝샤) (cn-northwest-1)
- 유럽(프랑크푸르트)(eu-central-1)
- 유럽(아일랜드)(eu-west-1)
- 유럽(런던) (eu-west-2)
- 유럽(밀라노) (eu-south-1)
- 유럽(파리) (eu-west-3)
- 유럽(스페인)(eu-south-2)
- 유럽(스톡홀름)(eu-north-1)
- 유럽(취리히)(eu-central-2)
- 이스라엘(텔아비브)(il-central-1)
- 멕시코(중부)(mx-central-1)
- 중동(바레인)(me-south-1)
- 중동(UAE)(me-central-1)
- 남아메리카(상파울루)(sa-east-1)
- AWS GovCloud(미국 동부)(us-gov-east-1)
- AWS GovCloud(미국 서부)(us-gov-west-1)
- 퍼블릭 액세스가 가능한 다중 AZ 데이터 웨어하우스는 단일 AZ 및 프라이빗 액세스가 가능한 다중 AZ 데이터 웨어하우스보다 1개 적은 수의 VPC 보안 그룹을 지원합니다.

새 클러스터 생성 시 다중 AZ 설정

~~Amazon Redshift 다중 AZ는 한 번에 두 개의 가용 영역을 지원합니다. Amazon Redshift는 선택한 서브넷 그룹 구성을 기반으로 가용 영역을 자동으로 선택합니다. 기존 단일 가용 영역 데이터 웨어하우스~~

를 다중 AZ 데이터 웨어하우스로 변환하거나 스냅샷에서 복원하여 다중 AZ 데이터 웨어하우스로 구성할 수 있습니다.

Amazon Redshift 콘솔을 사용하면 새로운 다중 AZ 배포를 쉽게 생성할 수 있습니다. Amazon Redshift 콘솔을 사용하여 새로운 다중 AZ 배포를 생성하려면 데이터 웨어하우스를 생성할 때 다중 AZ 옵션을 선택하세요. 단일 가용 영역에 필요한 컴퓨팅 노드 수를 지정하면 Amazon Redshift가 두 가용 영역 각각에 그만큼 노드를 배포합니다. 모든 노드는 정상 작동 중에 읽기 및 쓰기 워크로드 처리에 사용됩니다. AWS CLI `create-cluster` 명령에서 `multi-az` 파라미터를 사용하여 새 다중 AZ 데이터 웨어하우스를 만들 수도 있습니다.

기존 단일 AZ 데이터 웨어하우스를 다중 AZ 데이터 웨어하우스로 변환할 수 있습니다. Amazon Redshift 콘솔 또는 `multi-az` 파라미터를 사용하는 AWS CLI `modify-cluster` 명령을 사용할 수 있습니다. 또는 Amazon Redshift 콘솔 또는 `multi-az` 파라미터를 사용한 AWS CLI `restore-from-cluster-snapshot` 명령을 통해 스냅샷에서 복구하여 단일 AZ 데이터 웨어하우스를 다중 AZ 데이터 웨어하우스로 구성할 수 있습니다.

다중 AZ 배포는 Amazon Redshift 관리형 스토리지(RMS)를 사용하는 RA3 노드 유형만 지원합니다. Amazon Redshift는 Amazon S3를 사용하고 Amazon Redshift 수준에서 데이터를 복제하지 않고도 AWS 리전의 모든 가용 영역에서 액세스할 수 있는 RMS에 데이터를 저장합니다.

Amazon Redshift 콘솔 또는 AWS Command Line Interface를 사용하여 새 클러스터를 생성할 때 다중 AZ 배포를 설정할 수 있습니다.

콘솔 사용

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 Provisioned clusters dashboard(프로비저닝된 클러스터 대시보드)를 선택하고 Clusters(클러스터)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다.
3. 클러스터 생성 버튼을 선택하여 클러스터 생성 페이지를 엽니다.
4. 클러스터의 속성을 입력합니다. 클러스터 생성에 대한 일반적인 내용은 [클러스터 생성](#) 섹션을 참조하세요.
5. Node type(노드 유형) 드롭다운 목록에서 RA3 노드 유형 중 하나를 선택합니다. 다중 AZ 구성 옵션은 RA3 노드 유형을 선택한 경우에만 사용할 수 있습니다.
6. AZ 구성에서 다중 AZ를 선택합니다.
7. AZ당 노드 수에 클러스터의 노드를 두 개 이상 입력합니다.
8. 샘플 데이터를 로드하거나 자체 데이터를 가져올 수 있습니다.

- 샘플 데이터(Sample data)에서 샘플 데이터 로드(Load sample data)를 선택하여 Amazon Redshift 클러스터로 샘플 데이터 집합을 로드합니다. Amazon Redshift는 샘플 데이터 집합 Tackit을 기본 dev 데이터베이스와 퍼블릭 스키마로 로드합니다. Amazon Redshift는 샘플 데이터 집합을 Amazon Redshift 클러스터에 자동으로 로드합니다. 쿼리 편집기 v2를 사용하여 데이터 쿼리를 시작할 수 있습니다.
 - 자체 데이터를 Amazon Redshift 클러스터로 가져오려면 자체 데이터를 [Amazon Redshift로 가져오기](#)의 단계를 따르세요.
9. Additional configurations(추가 구성)까지 아래로 스크롤하고 Network and security(네트워크 및 보안)를 확장한 다음 기본 Cluster subnet group(클러스터 서브넷 그룹)을 수락하거나 다른 그룹을 선택해야 합니다. 다른 클러스터 서브넷 그룹을 선택하는 경우 선택한 서브넷 그룹에 3개의 가용 영역이 있는지 확인하세요.
 10. Additional configurations(추가 구성)에서 Database configurations(데이터베이스 구성)를 확장합니다.
 11. 기본 AWS 소유 키 대신 사용자 지정 AWS KMS 키를 사용하려면 데이터베이스 암호화에서 암호화 설정 사용자 지정을 클릭합니다.
 12. Choose an KMS key(KMS 키 선택)에서 AWS Key Management Service 키를 선택하거나 ARN을 입력할 수 있습니다. 또는 AWS Key Management Service 콘솔에서 AWS Key Management Service 키 생성을 클릭할 수 있습니다. KMS 키 생성에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [키 생성](#)을 참조하세요.
 13. Create Cluster(클러스터 생성)를 클릭합니다. 클러스터 생성에 성공하면 클러스터 세부 정보 페이지에서 세부 정보를 볼 수 있습니다. SQL 클라이언트를 사용하여 데이터를 로드하고 쿼리할 수 있습니다.

AWS Command Line Interface 사용

AWS Command Line Interface를 사용하여 클러스터 생성 시 다중 AZ를 설정하는 방법

- AWS CLI에서 create-cluster 명령과 multi-az 파라미터를 다음과 같이 사용하세요.

```
aws redshift create-cluster
  --port 5439
  --master-username master
  --master-user-password #####
  --node-type ra3.4xlarge
  --number-of-nodes 2
  --profile maz-test
```

```
--endpoint-url https://redshift.eu-west-1.amazonaws.com
--region eu-west-1
--cluster-identifier test-maz
--multi-az
--maintenance-track-name CURRENT
--encrypted
```

스냅샷에서 복원된 데이터 웨어하우스에 대한 다중 AZ 설정

스냅샷에서 복원하여 새 다중 AZ 클러스터를 만들려면 다음 절차를 완료하세요.

콘솔 사용

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 Clusters(클러스터), Snapshots(스냅샷)을 선택한 다음 사용할 스냅샷을 선택합니다.
3. Restore snapshot(스냅샷 복원), Restore to a provisioned cluster(프로비저닝된 클러스터로 복원)를 선택합니다.
4. 클러스터의 속성을 입력합니다. 클러스터 생성에 대한 일반적인 내용은 [클러스터 생성](#) 섹션을 참조하세요.
5. Node type(노드 유형) 드롭다운 목록에서 RA3 노드 유형 중 하나를 선택합니다. 다중 AZ 구성 옵션은 RA3 노드 유형을 선택한 경우에만 사용할 수 있습니다.
6. AZ 구성에서 다중 AZ를 선택합니다.
7. AZ당 노드 수에 클러스터의 노드를 두 개 이상 입력합니다.
8. 샘플 데이터를 로드하거나 자체 데이터를 가져올 수 있습니다.
 - 샘플 데이터(Sample data)에서 샘플 데이터 로드(Load sample data)를 선택하여 Amazon Redshift 클러스터로 샘플 데이터 집합을 로드합니다. Amazon Redshift는 샘플 데이터 집합 Tckit을 기본 dev 데이터베이스와 퍼블릭 스키마로 로드합니다. Amazon Redshift는 샘플 데이터 집합을 Amazon Redshift 클러스터에 자동으로 로드합니다. 쿼리 편집기 v2를 사용하여 데이터 쿼리를 시작할 수 있습니다.
 - 자체 데이터를 Amazon Redshift 클러스터로 가져오려면 [Amazon S3에서 Amazon Redshift로 데이터 로드](#)의 단계를 따르세요.
9. Additional configurations(추가 구성)까지 아래로 스크롤하고 Network and security(네트워크 및 보안)를 확장한 다음 기본 Cluster subnet group(클러스터 서브넷 그룹)을 수락하거나 다른 그룹을

선택해야 합니다. 다른 클러스터 서브넷 그룹을 선택하는 경우 선택한 서브넷 그룹에 3개의 가용 영역이 있는지 확인하세요.

10. Additional configurations(추가 구성)에서 Database configurations(데이터베이스 구성)를 확장합니다.
11. 데이터베이스 암호화에서 기본 AWS 소유 키가 아닌 사용자 지정 KMS 키를 사용하려면 암호화 설정 사용자 지정을 클릭합니다. 이 옵션은 기본적으로 선택 취소되어 있습니다.
12. Choose an KMS key(KMS 키 선택)에서 AWS Key Management Service 키를 선택하거나 ARN을 입력할 수 있습니다. 또는 AWS Key Management Service 콘솔에서 AWS Key Management Service 키 생성을 클릭할 수 있습니다. KMS 키 생성에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [키 생성](#)을 참조하세요.
13. Restore cluster from snapshot(스냅샷에서 클러스터 복원)을 클릭합니다. 클러스터 복원에 성공하면 클러스터 세부 정보 페이지에서 세부 정보를 볼 수 있습니다.

AWS Command Line Interface 사용

- AWS CLI에서 `restore-from-cluster-snapshot` 명령을 다음과 같이 사용하세요.

```
aws redshift restore-from-cluster-snapshot
--region eu-west-1
--multi-az
--snapshot-identifier test-snap1
--cluster-identifier test-saz-11
--endpoint-url https://redshift.eu-west-1.amazonaws.com/
```

단일 AZ 데이터 웨어하우스를 다중 AZ 데이터 웨어하우스로 변환

단일 AZ 데이터 웨어하우스를 다중 AZ 데이터 웨어하우스로 변환하면 SLA가 99.99% 보장되어 데이터 웨어하우스의 가용성이 높아집니다. 다중 AZ 데이터 웨어하우스를 사용하더라도 개별 쿼리의 성능은 동일하게 유지됩니다. 동시성이 높은 워크로드의 경우 Amazon Redshift가 두 가용 영역의 컴퓨팅 리소스를 사용하여 요청을 실행할 수 있으므로 전체 처리량이 향상됩니다.

Note

Amazon Redshift에서는 단일 AZ에서 다중 AZ로 또는 그 반대로 변환하는 동안 기존 컴퓨팅 리소스를 분할할 수 없습니다. 개별 쿼리 성능을 일관되게 유지하기 위해 이 작업은 지원되지 않습니다.

콘솔 사용

콘솔을 사용하여 단일 AZ 클러스터를 다중 AZ 데이터 웨어하우스로 변환하는 방법

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 Provisioned clusters dashboard(프로비저닝된 클러스터 대시보드)를 선택하고 Clusters(클러스터)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다.
3. 다중 AZ 배포로 변환하려는 클러스터를 선택합니다. 클러스터 세부 정보 페이지가 나타납니다.
4. 작업에서 다중 AZ 활성화를 선택합니다. 수정 요약 페이지가 나타납니다. 다중 AZ 활성화를 클릭합니다.
5. 오류가 발생하면 다음 중 하나를 수행한 다음 다중 AZ 활성화를 클릭합니다.
 - 클러스터 암호화 - 클러스터 세부 정보 페이지의 속성 탭 아래에 있는 데이터베이스 구성 섹션에서 속성을 선택하여 암호화 설정을 편집합니다.
 - 서브넷 그룹 - 서브넷 그룹 링크를 클릭하여 서브넷 그룹을 선택하고 클러스터 서브넷 그룹 설정을 편집합니다. 다른 클러스터 서브넷 그룹을 선택하는 경우 선택한 서브넷 그룹에 3개의 가용 영역이 있는지 확인하세요.
 - 포트 설정 - 클러스터 세부 정보 페이지의 속성 탭 아래에 있는 데이터베이스 구성 섹션에서 속성을 선택하여 포트 설정을 편집합니다.
6. SQL 클라이언트를 사용하여 데이터를 로드하고 쿼리할 수 있습니다.

AWS Command Line Interface 사용

- AWS CLI에서 `modify-cluster` 명령과 `multi-az` 파라미터를 다음과 같이 사용하세요.

```
aws redshift modify-cluster
  --profile maz-test
  --endpoint-url https://redshift.eu-west-1.amazonaws.com
  --region eu-west-1
  --cluster-identifier test-maz-11
  --multi-az
```

다중 AZ 배포에서는 시스템 모니터링 뷰(SYS_* 뷰)만 지원하므로 STL, SVCS, SVL, SVV 또는 STV 뷰를 사용할 수 없습니다. 시스템 모니터링 뷰(SYS_* 뷰)를 사용하도록 모니터링 쿼리를 변경하세요.

다중 AZ 데이터 웨어하우스를 단일 AZ 데이터 웨어하우스로 변환

다중 AZ 데이터 웨어하우스를 단일 AZ 데이터 웨어하우스로 변환하면 SLA를 99.99% 보장하는 다중 AZ와 달리 데이터 웨어하우스가 SLA를 99.99% 보장하지 않습니다. 개별 쿼리의 성능은 동일하게 유지되지만 두 번째 가용 영역의 컴퓨팅 리소스를 사용할 수 없으므로 전체 처리량에 영향을 미칩니다. 동시성 조정을 활성화하여 단일 AZ에서도 일관된 성능을 위해 처리량을 자동으로 확장할 수 있는 옵션이 있습니다.

Note

Amazon Redshift에서는 단일 AZ에서 다중 AZ로 또는 그 반대로 변환하는 동안 기존 컴퓨팅 리소스를 분할할 수 없습니다. 개별 쿼리 성능을 일관되게 유지하기 위해 이 작업은 지원되지 않습니다.

콘솔 사용

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 Provisioned clusters dashboard(프로비저닝된 클러스터 대시보드)를 선택하고 Clusters(클러스터)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다.
3. 다중 AZ 배포로 변환하려는 클러스터를 선택합니다. 클러스터 세부 정보 페이지가 나타납니다.
4. 작업에서 다중 AZ 비활성화를 선택합니다. 수정 요약 페이지가 나타납니다. 다중 AZ 비활성화를 클릭합니다.

AWS Command Line Interface 사용

- AWS CLI에서 `modify-cluster` 명령과 `no-multi-az` 파라미터를 다음과 같이 사용하세요.

```
aws redshift modify-cluster
  --profile maz-test
  --endpoint-url https://redshift.eu-west-1.amazonaws.com
  --region eu-west-1
  --cluster-identifier test-maz-11
  --no-multi-az
```

데이터 웨어하우스가 단일 AZ로 전환되면 99.99의 SLA 보장을 잃게 됩니다. 전체 처리량에도 영향을 미칩니다. 변경 사항을 저장하면 클러스터 세부 정보 페이지에서 세부 정보를 볼 수 있습니다.

다중 AZ 데이터 웨어하우스 크기 조정

다중 AZ 데이터 웨어하우스의 크기를 조정하고 데이터 웨어하우스의 현재 구성과 다르게 노드 또는 노드 유형의 수를 지정할 수 있습니다.

콘솔 사용

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 Provisioned clusters dashboard(프로비저닝된 클러스터 대시보드)를 선택하고 Clusters(클러스터)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다.
3. 다중 AZ 데이터 웨어하우스의 크기를 조정하려는 클러스터를 선택합니다. 클러스터 세부 정보 페이지가 나타납니다.
4. 작업(Actions)에서 크기 조정(Resize)을 선택합니다. 클러스터 크기 조정(Resize cluster) 페이지가 표시됩니다.
5. 페이지의 지시 사항을 따릅니다. 이제 특정 시간에 한 번 클러스터의 크기를 조정하거나, 일정에 따라 클러스터의 크기를 늘리거나 줄일 수 있습니다.
6. 새 구성의 노드 유형 드롭다운 목록에서 RA3 노드 유형 중 하나를 선택합니다.
7. 클러스터 크기 조정을 클릭합니다.

AWS Command Line Interface 사용

AWS Command Line Interface를 사용하여 다중 AZ 데이터 웨어하우스의 크기를 조정하는 방법

- AWS CLI에서 `resize-cluster` 명령을 사용하여 다음과 같이 단일 가용 영역의 노드 수를 변경하세요.

```
aws redshift resize-cluster \  
  --cluster-identifier test-maz-11  
  --cluster-type multi-node  
  --node-type ra3.4xlarge  
  --number-of-nodes 6
```

다중 AZ 배포를 통한 장애 조치

다중 AZ 데이터 웨어하우스는 두 가용 영역에 동시에 배포된 컴퓨팅 리소스 모음입니다. 기본 가용 영역에 배포된 컴퓨팅 리소스를 기본 컴퓨팅이라고 하고 보조 가용 영역에 배포된 컴퓨팅 리소스를 보조 컴퓨팅이라고 합니다. 다중 AZ 데이터 웨어하우스는 가용 영역 또는 인프라 장애와 같은 예상치 못한 이벤트가 발생할 경우 사용자 개입 없이 자동으로 복구할 수 있습니다. 복구 프로세스에는 기본 컴퓨팅에서 보조 컴퓨팅으로 장애 조치하고 보조 컴퓨팅 리소스를 기본 컴퓨팅으로 지정하는 작업이 포함됩니다. 또한 새로운 보조 컴퓨팅 리소스가 세 번째 가용 영역에 프로비저닝됩니다. 자동 복구 프로세스는 RTO 및 RPO 기준으로 측정됩니다.

- (Recovery Time Objective(RTO) – 재해 발생 후 시스템이 정상 작동 상태로 돌아가는 데 걸리는 시간입니다. 즉 RTO는 가동 중지 시간을 측정합니다.
- Recovery Point Objective(RPO) – 손실될 수 있는 데이터의 양입니다(시간으로 측정). Amazon Redshift 다중 AZ 데이터 웨어하우스의 경우 모든 데이터가 기본적으로 내구성이 뛰어나고 가용성이 뛰어난 Amazon Simple Storage Service의 지원을 받는 Amazon Redshift 관리형 스토리지(RMS)에 저장되므로 RPO는 일반적으로 0입니다.

Note

개별 쿼리 성능은 장애 조치가 발생한 후에도 변하지 않습니다. 가용 영역 중 하나에서 컴퓨팅 리소스를 사용할 수 없게 되면 잠시 동안 데이터 웨어하우스의 전체 처리량이 감소합니다. 하지만 Amazon Redshift는 다른 가용 영역의 용량을 자동으로 확보하여 동일한 데이터 웨어하우스 처리 용량이 복원되도록 합니다.

자동 복구 프로세스 외에도 기본 컴퓨팅 장애 조치 옵션을 사용하여 데이터 웨어하우스에 대해 이 프로세스를 수동으로 트리거할 수도 있습니다. 이 접근 방식을 사용하여 다중 AZ가 애플리케이션의 고가용성과 연속성 향상에 어떻게 도움이 되는지 테스트할 수 있습니다.

콘솔 사용

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 다음 중 하나를 수행하세요.
 - 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. Clusters(클러스터)에서 클러스터를 선택합니다. 클러스터 세부 정보 페이지가 나타납니다.

- 클러스터 대시보드에서 클러스터를 선택합니다.
3. 작업에서 기본 컴퓨팅 장애 조치를 선택합니다.
 4. 메시지가 나타나면 Confirm(확인)을 클릭합니다.

AWS Command Line Interface 사용

- AWS CLI에서 failover-primary-compute 명령을 다음과 같이 사용하세요.

```
aws redshift failover-primary-compute
  --profile maz-test
  --endpoint-url https://redshift.eu-west-1.amazonaws.com
  --region eu-west-1
  --cluster-identifier test-maz-11
```

위 작업이 확인되면 Amazon Redshift는 가용 영역 또는 인프라 장애로부터의 자동 복구와 동일한 단계를 수행합니다. 이 프로세스로 인해 기본 가용 영역의 컴퓨팅 노드를 사용할 수 없게 되고 보조 가용 영역의 컴퓨팅 리소스가 기본 컴퓨팅으로 지정됩니다. 클러스터 복구가 성공적으로 완료되면 다중 AZ 배포를 사용할 수 있게 됩니다. 또한 다중 AZ 데이터 웨어하우스는 사용 가능한 즉시 다른 세 번째 가용 영역에 새 보조 컴퓨팅을 자동으로 프로비저닝합니다.

이 프로세스 동안 클러스터가 자동으로 복구되고 다중 AZ 배포 설정으로 다시 재구성되기 때문에 콘솔의 클러스터 상태는 전체 시간 동안 수정 중으로 표시됩니다. 클러스터는 새 연결을 즉시 수락할 수 있습니다. 기존 연결 및 이동 중인 쿼리가 삭제될 수 있습니다. 즉시 다시 시도할 수 있습니다.

다중 AZ 데이터 웨어하우스의 쿼리 및 로드 보기

클러스터의 유형, 크기 및 상태(일시 중지 또는 재개)에 관계없이 지난 7일 동안 실행된 쿼리에 대한 정보를 볼 수 있습니다.

쿼리 및 로드 페이지에 표시되는 정보는 Amazon Redshift 시스템 테이블(SYS_* 보기)의 정보로 채워집니다. 이 정보를 사용하면 7일 동안 유지되는 쿼리 및 제안에 대한 추가 정보를 표시할 수 있습니다. 쿼리 진단이 빨라져 데이터베이스, 사용자 이름 또는 SQL 문 유형별로 데이터를 필터링할 수 있습니다. 실행된 모든 쿼리에 대한 이러한 추가 필터 및 정보를 보려면 다음 사전 요구 사항을 참고하세요.

- Connect to database(데이터베이스에 연결)를 선택하여 데이터베이스에 연결해야 합니다.
- 데이터베이스 사용자에게 sys:operator 또는 sys:monitor 역할 및 권한이 있어야 쿼리 모니터링을 수행할 수 있습니다. 시스템 역할에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Amazon Redshift 시스템 정의 역할](#)을 참조하세요.

데이터베이스에 연결하면 이러한 추가 필터 및 쿼리 정보를 볼 수 있습니다.

쿼리 및 로드의 쿼리 성능 데이터 표시

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 쿼리를 선택한 다음 쿼리 및 로드(Queries and loads)를 선택하여 계정에 대한 쿼리 목록을 표시합니다.
3. 추가 필터를 보려면 데이터베이스에 연결해야 할 수 있습니다. 필요한 경우 Connect to database(데이터베이스에 연결)를 클릭하고 프롬프트에 따라 데이터베이스에 연결합니다.

기본적으로 이 목록에는 지난 24시간 동안의 모든 클러스터에 대한 쿼리가 표시됩니다. 콘솔에서 표시된 날짜의 범위를 변경할 수 있습니다.

쿼리 모니터링에서 쿼리 성능 데이터 표시

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. Clusters(클러스터)에서 클러스터를 선택합니다.
3. Query monitoring(쿼리 모니터링)을 선택합니다.
4. 클러스터의 구성 또는 버전에 따라 추가 필터를 보려면 데이터베이스에 연결해야 할 수 있습니다. 필요한 경우 Connect to database(데이터베이스에 연결)를 클릭하고 프롬프트에 따라 데이터베이스에 연결합니다.

다중 AZ 배포에서 쿼리 모니터링

다중 AZ 배포는 두 가용 영역에 배포된 컴퓨팅 리소스를 사용하며 지정된 가용 영역의 리소스를 사용할 수 없는 경우에도 계속 작동할 수 있습니다. 모든 컴퓨팅 리소스는 항상 사용됩니다. 이를 통해 읽기 및 쓰기 작업 모두에 대해 액티브-액티브 방식으로 두 가용 영역에서 모든 작업을 수행할 수 있습니다.

pg_catalog 스키마에서 SYS_ 뷰를 쿼리하여 다중 AZ 배포에서 쿼리 런타임을 모니터링할 수 있습니다. SYS_ 뷰는 기본 및 보조 클러스터의 쿼리 런타임 활동 또는 통계를 표시합니다. 모니터링 뷰 목록은 [모니터링 뷰](#)를 참조하세요.

다음 단계에 따라 다중 AZ 배포 내의 각 가용 영역에 대한 쿼리 런타임을 모니터링합니다.

1. Amazon Redshift 콘솔로 이동하여 다중 AZ 배포의 데이터베이스에 연결하고 쿼리 편집기를 통해 쿼리를 실행합니다.

2. 다중 AZ Amazon Redshift 배포에서 샘플 쿼리를 실행합니다.
3. 다중 AZ 배포의 경우 SYS_QUERY_HISTORY 테이블의 compute_type 열을 사용하여 쿼리와 쿼리가 실행되는 가용 영역을 식별할 수 있습니다. primary는 다중 AZ 배포의 기본 클러스터에서 실행되는 쿼리를 나타내고 secondary는 다중 AZ 배포의 보조 클러스터에서 실행되는 쿼리를 나타냅니다.

다음 쿼리는 compute_type 열을 사용하여 쿼리를 모니터링합니다.

```
select (compute_type) as compute_type, left(query_text, 50) query_text from
sys_query_history order by start_time desc;
```

```
compute_type | query_text
-----+-----
secondary    | select count(*) from t1;
```

클러스터에 대한 쿼리 종료

이 절차는 다중 AZ 클러스터와 단일 AZ 클러스터 모두에 적용됩니다.

쿼리를 종료하는 방법

쿼리 페이지에서는 현재 진행 중인 쿼리를 종료할 수도 있습니다.

데이터베이스 사용자는 실행 중인 쿼리를 종료할 수 있는 sys:operator 역할 및 권한이 있어야 합니다. 시스템 역할에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Amazon Redshift 시스템 정의 역할](#)을 참조하세요.

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 쿼리를 선택한 다음 쿼리 및 로드(Queries and loads)를 선택하여 계정에 대한 쿼리 목록을 표시합니다.
3. 목록에서 종료할 실행 쿼리를 선택한 다음 쿼리 종료를 선택합니다.

Amazon Redshift 클러스터 성능 모니터링

Amazon Redshift는 클러스터와 데이터베이스의 상태 및 성능을 추적할 수 있도록 성능 지표와 데이터를 제공합니다. 이 섹션에서는 Amazon Redshift, 특히 Amazon Redshift 콘솔에서 사용할 수 있는 데이터 형식에 대해서 얘기하겠습니다.

Amazon Redshift 콘솔에서 사용할 수 있는 성능 데이터는 다음과 같이 2가지 범주로 나뉩니다.

- Amazon CloudWatch 지표 – Amazon CloudWatch 지표는 CPU 사용량, 지연 시간, 처리량 등 클러스터의 물리적 요소를 모니터링하는 데 유용합니다. 지표 데이터는 Amazon Redshift 콘솔에 직접 표시됩니다. CloudWatch 콘솔에서도 확인 가능합니다. 아니면 AWS CLI 또는 AWS SDK 중 하나와 같이 다른 방식으로 지표를 사용할 수도 있습니다.
- 쿼리/로드 성능 데이터 – 성능 데이터는 데이터베이스 작업 및 성능을 모니터링하는 데 유용합니다. 이 데이터는 Amazon Redshift 콘솔에 수집되어 CloudWatch 지표 결과와 특정 데이터베이스 쿼리 및 로드 이벤트의 상관관계를 쉽게 파악하는 데 도움이 됩니다. 또한 사용자 지정 성능 쿼리를 작성한 후 데이터베이스에서 직접 실행할 수도 있습니다. 쿼리 및 로드 성능 데이터는 Amazon Redshift 콘솔에만 표시되며 CloudWatch 지표로는 게시되지 않습니다.

성능 데이터는 Amazon Redshift 콘솔에 통합되어 다음과 같이 다양한 사용자 경험을 가능하게 합니다.

- 크기 조정 등 클러스터에 대한 여러 가지 결정이 필요한 경우 클러스터와 연결된 성능 데이터가 상황에 따라 함께 표시됩니다.
- CloudWatch와 비교했을 때 Amazon Redshift 콘솔에서는 일부 성능 지표가 더욱 적합한 눈금 단위로 표시됩니다. 예를 들어 WriteThroughput이 GB/s(CloudWatch에서는 바이트/초로 표시됨)로 표시되어 일반적인 노드의 스토리지 공간에 더욱 적합합니다.
- 동일한 그래프에서 클러스터의 여러 노드에 대한 성능 데이터를 쉽게 표시할 수 있습니다. 이러한 방식으로 클러스터의 모든 노드에 대한 성능을 손쉽게 모니터링할 수 있습니다. 노드별 성능 데이터를 확인할 수도 있습니다.

Amazon Redshift는 추가 비용 없이 성능 데이터(CloudWatch 지표와 쿼리 및 로드 데이터)를 제공합니다. 성능 데이터는 1분 주기로 기록되기 때문에 Amazon Redshift 콘솔에서 성능 데이터의 이력 값에 액세스할 수 있습니다. CloudWatch를 사용하여 CloudWatch 지표로 표시되는 Amazon Redshift 성능 데이터에 액세스하는 방법에 대한 자세한 내용은 Amazon CloudWatch User Guide의 [What is CloudWatch?](#)를 참조하세요.

Amazon Redshift의 성능 데이터

Amazon Redshift에 CloudWatch 지표를 사용하면 클러스터의 상태 및 성능에 대한 정보와 노드 수준의 정보를 얻을 수 있습니다. 단, 이 지표를 사용할 때는 각 지표마다 차원이 1개 이상 연결되어 있다는 점을 알고 있어야 합니다. 이러한 차원은 적용 대상 지표를 알려주고 지표의 범위를 나타냅니다. Amazon Redshift에는 다음과 같은 2가지 차원이 있습니다.

- NodeID 차원을 갖는 지표는 클러스터 노드의 성능 데이터를 제공하는 지표입니다. 이러한 지표 집합에는 리더와 컴퓨팅 노드가 포함됩니다. 예를 들면 CPUUtilization, ReadIOPS, WriteIOPS 등이 있습니다.
- ClusterIdentifier 차원만 갖고 있는 지표는 클러스터의 성능에 대한 데이터를 제공하는 지표입니다. 이러한 지표의 예로는 HealthStatus와 MaintenanceMode가 있습니다.

Note

일부 지표의 경우에는 각 클러스터의 지표가 노드 특성의 합계를 의미합니다. 이때는 리더 노드의 특성이 컴퓨팅 노드와 합산되기 때문에 지표 값을 해석할 때는 주의가 필요합니다.

CloudWatch 지표 및 차원에 대한 내용은 Amazon CloudWatch User Guide의 [CloudWatch concepts](#)을 참조하세요.

Amazon Redshift용 CloudWatch 지표에 대한 자세한 설명은 다음 섹션을 참조하세요.

주제

- [Amazon Redshift 지표](#)
- [Amazon Redshift 지표의 차원](#)
- [Amazon Redshift 쿼리 및 로드 성능 데이터](#)

Amazon Redshift 지표

AWS/Redshift 네임스페이스에는 다음과 같은 지표가 포함됩니다. 달리 명시되지 않는 한, 지표는 1분 간격으로 수집됩니다.

지표	설명
CommitQueueLength	주어진 시점에 커밋을 기다리는 트랜잭션의 수입니다. 단위: 개 차원: ClusterIdentifier
ConcurrencyScalingActiveClusters	주어진 시간에 능동적으로 쿼리를 처리 중인 동시성 확장 클러스터의 수입니다.

지표	설명
ConcurrencyScaling Seconds	단위: 개
	차원: ClusterIdentifier
	활성 쿼리 처리 활동이 있는 동시성 확장 클러스터에서 사용하는 시간(초)입니다.
	단위: 개
CPUUtilization	차원: ClusterIdentifier
	CPU 사용 백분율. 클러스터의 경우, 이 지표는 모든 노드(리더 및 컴퓨팅)의 CPU 사용 값을 집계한 결과를 의미합니다.
	단위: 백분율
	Dimensions: ClusterIdentifier , NodeID
DatabaseConnections	차원: ClusterIdentifier
	클러스터에 대한 데이터베이스 연결 수
	단위: 개
	차원: ClusterIdentifier

지표	설명
HealthStatus	클러스터가 데이터베이스에 연결되어 간단한 쿼리를 실행할 때마다 클러스터의 상태를 나타냅니다. 이 작업을 성공적으로 실행할 수 있다면 클러스터는 정상 상태입니다. 그렇지 않다면 비정상 상태입니다. 비정상 상태는 클러스터 데이터베이스의 데이터 로드 가 너무 많거나, 혹은 클러스터 데이터베이스에 구성 문제가 있을 때 발생합니다.  Note Amazon CloudWatch에서는 이 지표가 1 또는 0으로 보고되는 반면 Amazon Redshift 콘솔에서는 편의를 위해 이 지표가 HEALTHY 또는 UNHEALTHY 로 표시됩니다. 이 지표가 Amazon Redshift 콘솔에 표시될 때는 샘플 평균이 무시되고 HEALTHY 또는 UNHEALTHY 만 표시됩니다. 하지만 Amazon CloudWatch에서는 샘플링 문제로 인해 1과 0이 아닌 다른 값이 표시될 수도 있습니다. HealthStatus 가 1 미만인 값은 모두 0(UNHEALTHY)으로 보고됩니다. 단위: 개수 (1/0)(Amazon Redshift 콘솔의 경우 HEALTHY/UNHEALTHY) 차원: ClusterIdentifier

지표	설명
MaintenanceMode	클러스터가 현재 유지 관리 모드인지 여부를 나타냅니다.  Note Amazon CloudWatch에서는 이 지표가 1 또는 0으로 보고되는 반면 Amazon Redshift 콘솔에서는 편의를 위해 이 지표가 ON 또는 OFF로 표시됩니다. 이 지표가 Amazon Redshift 콘솔에 표시될 때는 샘플 평균이 무시되고 ON 또는 OFF만 표시됩니다. 하지만 Amazon CloudWatch에서는 샘플링 문제로 인해 1과 0이 아닌 다른 값이 표시될 수도 있습니다. MaintenanceMode 가 0보다 큰 값은 모두 1(ON)로 보고됩니다. 단위: 개수 (1/0)(Amazon Redshift 콘솔의 경우 ON/OFF) 차원: ClusterIdentifier
MaxConfiguredConcurrencyScalingClusters	파라미터 그룹에서 구성된 최대 동시성 확장 클러스터 수입니다. 자세한 내용은 Amazon Redshift 파라미터 그룹 섹션을 참조하세요. 단위: 개 차원: ClusterIdentifier
NetworkReceiveThroughput	노드 또는 클러스터의 데이터 수신 속도 단위: 바이트/초(Amazon Redshift 콘솔의 경우 MB/초) Dimensions: ClusterIdentifier , NodeID 차원: ClusterIdentifier

지표	설명
NetworkTransmitThroughput	노드 또는 클러스터의 데이터 쓰기 속도 단위: 바이트/초(Amazon Redshift 콘솔의 경우 MB/초) Dimensions: ClusterIdentifier , NodeID 차원: ClusterIdentifier
PercentageDiskSpaceUsed	사용된 디스크 공간 비율 단위: 백분율 차원: ClusterIdentifier Dimensions: ClusterIdentifier , NodeID
QueriesCompletedPerSecond	초당 완료된 평균 쿼리 수입니다. 5분 간격으로 보고됩니다. 단일 노드 클러스터에서는 이 지표가 지원되지 않습니다. 단위: 개수/초 Dimensions: ClusterIdentifier , latency Dimensions: ClusterIdentifier , wlmid
QueryDuration	쿼리를 완료하는 데 걸리는 평균 시간입니다. 5분 간격으로 보고됩니다. 단일 노드 클러스터에서는 이 지표가 지원되지 않습니다. 단위: 마이크로초 차원: ClusterIdentifier , NodeID, latency Dimensions: ClusterIdentifier , latency 차원: ClusterIdentifier , NodeID, wlmid

지표	설명
QueryRuntimeBreakdown	쿼리 단계에서 실행한 총 쿼리 시간입니다. 5분 간격으로 보고됩니다. 단위: 밀리초 차원: ClusterIdentifier, NodeID, stage 차원: ClusterIdentifier, stage
ReadIOPS	초당 평균 디스크 I/O 연산 수. 단위: 개수/초 Dimensions: ClusterIdentifier , NodeID 차원: ClusterIdentifier
ReadLatency	디스크 I/O 연산당 평균 처리 시간. 단위: 초 Dimensions: ClusterIdentifier , NodeID 차원: ClusterIdentifier
ReadThroughput	초당 디스크에서 읽은 평균 바이트 수입니다. 단위: 바이트(Amazon Redshift 콘솔의 경우 GB/초) Dimensions: ClusterIdentifier , NodeID 차원: ClusterIdentifier
RedshiftManagedStorageTotalCapacity	관리형 스토리지의 총 용량입니다. 단위: 메가바이트 차원: ClusterIdentifier

지표	설명
TotalTableCount	특정 시점에 열린 사용자 테이블의 수입니다. 이 합계에는 Amazon Redshift Spectrum 테이블이 포함되지 않습니다. 단위: 개 차원: ClusterIdentifier
WLMQueueLength	워크로드 관리(WLM) 대기열을 입력하기 위해 대기하는 쿼리 수입니다. 단위: 개 Dimensions: ClusterIdentifier , service class Dimensions: ClusterIdentifier , QueueName
WLMQueueWaitTime	워크로드 관리(WLM) 대기열에서 대기하는 데 소요된 총 쿼리 시간. 5분 간격으로 보고됩니다. 단위: 밀리초 Dimensions: ClusterIdentifier , QueryPriority Dimensions: ClusterIdentifier , wlmid Dimensions: ClusterIdentifier , QueueName
WLMQueriesCompletedPerSecond	워크로드 관리(WLM) 대기열에 대해 완료된 초당 평균 쿼리 수입니다. 5분 간격으로 보고됩니다. 단일 노드 클러스터에서는 이 지표가 지원되지 않습니다. 단위: 개수/초 Dimensions: ClusterIdentifier , wlmid Dimensions: ClusterIdentifier , QueueName

지표	설명
WLMQueryDuration	워크로드 관리(WLM) 대기열에 대해 쿼리를 완료하는 데 걸리는 평균 시간입니다. 5분 간격으로 보고됩니다. 단일 노드 클러스터에서는 이 지표가 지원되지 않습니다. 단위: 마이크로초 Dimensions: ClusterIdentifier , wlmid Dimensions: ClusterIdentifier , QueueName
WLMRunningQueries	WLM 대기열 1개당 기본 클러스터 및 동시성 확장 클러스터 모두에서 실행되는 쿼리의 수입니다. 단위: 개 Dimensions: ClusterIdentifier , wlmid Dimensions: ClusterIdentifier , QueueName
WriteIOPS	초당 커밋 작업의 평균 수. 단위: 개수/초 Dimensions: ClusterIdentifier , NodeID 차원: ClusterIdentifier
WriteLatency	디스크 I/O 연산당 평균 처리 시간. 단위: 초 Dimensions: ClusterIdentifier , NodeID 차원: ClusterIdentifier

지표	설명
WriteThroughput	초당 디스크에 쓴 평균 바이트 수. 단위: 바이트(Amazon Redshift 콘솔의 경우 GB/초) Dimensions: ClusterIdentifier , NodeID 차원: ClusterIdentifier
SchemaQuota	스키마에 대해 구성된 할당량입니다. 단위: 메가바이트 차원: ClusterIdentifier , Database, Schema 주기적/푸시: Periodic 빈도: 5 minutes 중지 조건: 스키마가 삭제되거나 할당량이 제거됨
NumExceededSchemaQuotas	할당량이 초과된 스키마 수입니다. 단위: 개 차원: ClusterIdentifier 주기적/푸시: Periodic 빈도: 5 minutes 중지 기준: 해당 사항 없음

지표	설명
StorageUsed	스키마에서 사용하는 디스크 또는 스토리지 공간입니다. 단위: 메가바이트 차원: ClusterIdentifier , Database, Schema 주기적/푸시: Periodic 빈도: 5 minutes 중지 조건: 스키마가 삭제되거나 할당량이 제거됨
PercentageQuotaUsed	구성된 스키마 할당량을 기준으로 사용된 디스크 또는 스토리지 공간의 백분율입니다. 단위: 백분율 차원: ClusterIdentifier , Database, Schema 주기적/푸시: Periodic 빈도: 5 minutes 중지 조건: 스키마가 삭제되거나 할당량이 제거됨

지표	설명
UsageLimitAvailable	FeatureType에 따라 UsageLimitAvailable은 다음을 반환합니다. - FeatureType이 CONCURRENCY_SCALING 인 경우 UsageLimitAvailable은 동시성 조정에 사용할 수 있는 총 시간을 1분 단위로 반환합니다. 이 시간은 CONCURRENCY_SCALING_AUTO_TASK 에 사용할 수 있는 시간으로 계산되지 않으며 그 반대의 경우도 마찬가지입니다. - FeatureType이 CONCURRENCY_SCALING_AUTO_TASK 인 경우 UsageLimitAvailable은 자동 최적화 작업을 위한 동시성 규모 조정에 사용할 수 있는 총 시간을 반환합니다. 이 시간은 CONCURRENCY_SCALING 에 사용할 수 있는 시간으로 계산되지 않으며 그 반대의 경우도 마찬가지입니다. - FeatureType이 EXTRA_COMPUTE_FOR_AUTOMATIC_OPTIMIZATION 인 경우 UsageLimitAvailable은 자동 최적화 작업에 사용할 수 있는 총 시간을 반환합니다. - FeatureType이 CROSS_REGION_DATASHARING 인 경우 UsageLimitAvailable은 스캔될 수 있는 데이터의 총량을 1TB 단위로 반환합니다. - FeatureType이 SPECTRUM인 경우 UsageLimitAvailable은 스캔될 수 있는 데이터의 총량을 1TB 단위로 반환합니다. 단위: 분 또는 TB 차원: ClusterIdentifier , FeatureType , UsageLimitId

지표	설명
UsageLimitConsumed	FeatureType에 따라 UsageLimitConsumed는 다음을 반환합니다. - FeatureType이 CONCURRENCY_SCALING 인 경우 UsageLimitConsumed는 동시성 규모 조정에서 사용된 총 시간을 1분 단위로 반환합니다. 이 시간은 CONCURRENCY_SCALING_AUTO_TASK 에 사용된 시간으로 계산되지 않으며 그 반대의 경우도 마찬가지입니다. - FeatureType이 CONCURRENCY_SCALING_AUTO_TASK 인 경우 UsageLimitConsumed는 자동 최적화 작업을 위한 동시성 규모 조정에서 사용된 총 시간을 반환합니다. 이 시간은 CONCURRENCY_SCALING 에 사용된 시간으로 계산되지 않으며 그 반대의 경우도 마찬가지입니다. - FeatureType이 EXTRA_COMPUTE_FOR_AUTOMATIC_OPTIMIZATION 인 경우 UsageLimitConsumed는 자동 최적화 작업에 사용된 총 시간을 반환합니다. - FeatureType이 CROSS_REGION_DATASHARING 인 경우 UsageLimitConsumed는 스캔된 데이터의 총량을 1TB 단위로 반환합니다. - FeatureType이 SPECTRUM인 경우 UsageLimitConsumed는 스캔된 데이터의 총량을 1TB 단위로 반환합니다. 단위: 분 또는 TB 차원: ClusterIdentifier , FeatureType , UsageLimitId

Amazon Redshift 지표의 차원

Amazon Redshift 데이터는 다음 표의 차원 중 하나와 함께 필터링할 수 있습니다.

차원	설명
latency	가능한 값은 다음과 같습니다.

차원	설명
	• short - 10초 미만 • medium - 10초~10분 • long - 10분 초과
NodeID	클러스터 노드 고유의 요청 데이터를 필터링합니다. NodeID는 "Leader", "Shared" 또는 "Compute-N"이 되고, 여기에서 N은 클러스터의 노드 번호를 가리키는 0, 1 등으로 매겨집니다. "Shared"는 클러스터의 노드 수가 1개, 즉 리더 노드와 컴퓨팅 노드가 결합되었다는 것을 의미합니다. 지표는 CPUUtilization , NetworkTransmitThroughput 및 ReadIOPS일 때에 한해 리더 노드와 컴퓨팅 노드에 대해 보고됩니다. NodeId 차원을 사용하는 다른 지표는 컴퓨팅 노드에 대해서만 보고됩니다.
ClusterIdentifier	클러스터 고유의 요청 데이터를 필터링합니다. 클러스터 고유의 지표로는 HealthStatus , MaintenanceMode 및 DatabaseConnections 가 있습니다. 노드 지표이기도 한 이 차원의 일반 지표(ReadIOPS 등)는 노드 지표 데이터를 모두 종합한 것입니다. 이러한 지표들은 리더 노드와 컴퓨팅 노드의 속성을 모두 수집하기 때문에 해석할 때 주의하십시오.
service class	WLM 서비스 클래스의 식별자입니다.

차원	설명
stage	쿼리 실행 단계 가능한 값은 다음과 같습니다. - QueryPlanning: SQL 문을 구문 분석하고 최적화하는 데 소요된 시간 - QueryWaiting: WLM 대기열에서 대기하는 데 소요된 시간 - QueryExecutingRead: 읽기 쿼리를 실행하는 데 소요된 시간 - QueryExecutingInsert: 삽입 쿼리를 실행하는 데 소요된 시간 - QueryExecutingDelete: 삭제 쿼리를 실행하는 데 소요된 시간 - QueryExecutingUpdate: 업데이트 쿼리를 실행하는 데 소요된 시간 - QueryExecutingCtas: 쿼리로 테이블을 생성하는 데 소요된 시간 - QueryExecutingUnload: 업로드 쿼리를 실행하는 데 소요된 시간 - QueryExecutingCopy: 복사 쿼리를 실행하는 데 소요된 시간 - QueryCommit: 커밋하는 데 소요된 시간
wlmid	워크로드 관리 대기열에 대한 식별자입니다.
QueryPriority	쿼리의 우선순위입니다. 가능한 값은 CRITICAL, HIGHEST, HIGH, NORMAL, LOW, LOWEST입니다.
QueueName	워크로드 관리 대기열의 이름입니다.
FeatureType	사용량 한도로 제한되는 기능입니다. 가능한 값은 CONCURRENT_SCALING, CROSS_REGION_DATASHARING 및 SPECTRUM입니다.
UsageLimitId	사용량 한도의 식별자입니다.

Amazon Redshift 쿼리 및 로드 성능 데이터

Amazon Redshift는 CloudWatch 지표 외에도 쿼리 및 로드 성능 데이터를 제공합니다. 쿼리/로드 성능 데이터는 데이터베이스 성능과 클러스터 지표의 관계를 이해하는 데 유용합니다. 예를 들어 클러스터

의 CPU가 급증한 것을 확인한 경우에는 클러스터 CPU 그래프에서 급증한 부분을 찾아 해당 시간에 실행된 쿼리를 살펴볼 수 있습니다. 반대로 특정 쿼리를 살펴볼 때는 지표 데이터(CPU 등)가 상황에 맞게 표시되어 쿼리가 클러스터 지표에 미치는 영향을 이해할 수 있습니다.

쿼리 및 로드 성능 데이터는 CloudWatch 지표로 게시되지 않기 때문에 Amazon Redshift 콘솔에서만 확인할 수 있습니다. 쿼리 및 로드 성능 데이터는 데이터베이스의 시스템 테이블을 사용한 쿼리에서 생성됩니다(자세한 내용은 Amazon Redshift 개발자 안내서의 [시스템 테이블 참조](#) 참조). 사용자 지정 데이터베이스 성능 쿼리를 직접 작성할 수도 있지만 처음에는 콘솔에서 제공되는 쿼리/로드 성능 데이터를 사용하는 것이 좋습니다. 데이터베이스 성능을 직접 측정하고 모니터링하는 방법에 대한 자세한 내용은 Amazon Redshift 개발자 안내서의 [성능 관리](#)를 참조하세요.

다음은 Amazon Redshift 콘솔에서 액세스할 수 있는 쿼리/로드 데이터의 여러 가지 요소를 설명한 표입니다.

쿼리/로드 데이터	설명
쿼리 요약	특정 시간대에 실행된 쿼리 목록입니다. 이 목록은 쿼리 ID, 쿼리 실행 시간, 상태 등의 값을 기준으로 정렬할 수 있습니다. 클러스터 세부 정보 페이지의 쿼리 모니터링 탭에 이 데이터가 표시됩니다.
쿼리 세부 정보	다음은 포함하여 특정 쿼리에 대한 세부 정보를 제공합니다. • 쿼리 속성(쿼리 ID, 유형, 쿼리가 실행된 클러스터, 실행 시간 등) • 세부 정보(쿼리 상태, 오류 수 등) • 실행된 SQL 문 • EXPLAIN 계획(해당되는 경우) • 쿼리 실행 중 클러스터 성능 데이터(자세한 내용은 쿼리 기록 데이터 보기 참조)
로드 요약	지정 시간대의 모든 로드 목록입니다. 이 목록은 쿼리 ID, 쿼리 실행 시간, 상태 등의 값을 기준으로 정렬할 수 있습니다. 클러스터 세부 정보 페이지의 쿼리 모니터링 탭에 이 데이터가 표시됩니다.
로드 세부 정보	다음은 포함하여 특정 로드 작업에 대한 세부 정보를 제공합니다. • 로드 속성(쿼리 ID, 유형, 쿼리가 실행된 클러스터, 실행 시간 등) • 세부 정보(로드 상태, 오류 수 등) • 실행된 SQL 문

쿼리/로드 데이터	설명
	- 로드된 파일 목록 - 로드 작업 중 클러스터 성능 데이터(자세한 내용은 쿼리 기록 데이터 보기 참조)

성능 데이터 보기

이 섹션에서는 Amazon Redshift 콘솔에서 클러스터 및 쿼리 성능에 대한 정보를 포함하여 성능 데이터를 확인하는 방법에 대해 설명합니다. 그 밖에 Amazon Redshift 콘솔에서 클러스터 지표에 대한 경보를 직접 생성할 수도 있습니다.

Amazon Redshift 콘솔에서는 성능 데이터가 클러스터를 기준으로 표시됩니다. 클러스터의 성능 데이터 그래프는 직접 데이터에 접근하여 가장 공통적인 성능 관련 질문에 대한 답변을 확인할 수 있도록 작성됩니다. 일부 성능 데이터([Amazon Redshift의 성능 데이터](#) 참조)의 경우 CloudWatch를 사용하여 지표 그래프를 추가로 사용자 정의할 수도 있습니다. 예를 들어, 더 긴 시간을 선택하거나 여러 클러스터에서 결합할 수 있습니다. CloudWatch 콘솔의 사용 방법에 대한 자세한 내용은 [CloudWatch 콘솔의 성능 지표](#) 섹션을 참조하세요.

Amazon Redshift 콘솔의 쿼리 모니터링 기능을 사용하여 쿼리를 모니터링, 격리 및 최적화하는 방법을 알아보려면 [Query Monitoring with Amazon Redshift](#) 동영상을 시청하세요.

주제

- [클러스터 성능 데이터 보기](#)
- [쿼리 기록 데이터 보기](#)
- [데이터베이스 성능 데이터 보기](#)
- [워크로드 동시성 및 동시성 확장 데이터 보기](#)
- [자동 최적화 데이터 보기](#)
- [쿼리 및 로드 보기](#)
- [쿼리 세부 정보 보기 및 분석](#)
- [쿼리 실행 시 클러스터 성능 보기](#)
- [로드 작업 중 클러스터 지표 보기](#)
- [클러스터 워크로드 분석 차트 보기](#)

클러스터 성능 데이터 보기

Amazon Redshift의 클러스터 지표를 사용하여 다음과 같이 일반적인 성능 태스크를 수행할 수 있습니다.

- 특정 시간 동안 클러스터 지표의 이상 여부를 확인하여 이상이 있는 경우에는 성능 히트의 원인이 되는 쿼리를 식별합니다.
- 이전 또는 현재 쿼리가 클러스터 성능에 영향을 미치는지 검사합니다. 문제가 있는 쿼리가 발견되면 쿼리 실행 시 클러스터 성능 등을 포함하여 세부 정보를 확인할 수 있습니다. 이 정보를 사용하여 쿼리 속도가 느려진 이유, 성능 개선을 위해 필요한 사항 등을 진단할 수 있습니다.

성능 데이터를 보려면

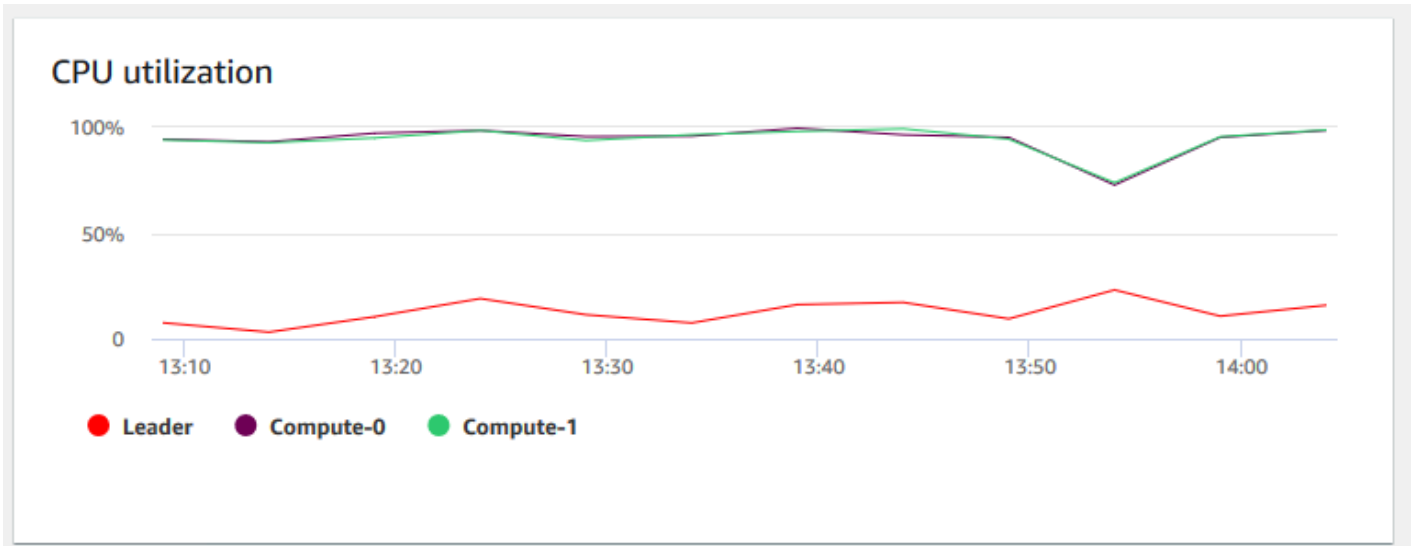
1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 목록에서 클러스터의 이름을 선택하여 세부 정보를 엽니다. 클러스터의 세부 정보가 표시됩니다. 여기에는 클러스터 성능, 쿼리 모니터링, 데이터베이스, 데이터 공유, 일정, 유지 관리 및 속성 탭이 포함될 수 있습니다.
3. 다음을 포함한 성능 정보를 보려면 Cluster performance(클러스터 성능) 탭을 선택합니다.
 - CPU 사용률
 - Percentage disk space used(사용된 디스크 공간 비율)
 - 데이터베이스 연결
 - 상태 확인
 - 쿼리 지속시간
 - 쿼리 처리량
 - 동시성 확장 활동

그 밖에도 많은 지표를 사용할 수 있습니다. 사용 가능한 지표를 보고 표시할 지표를 선택하려면 기본 설정 아이콘을 선택합니다.

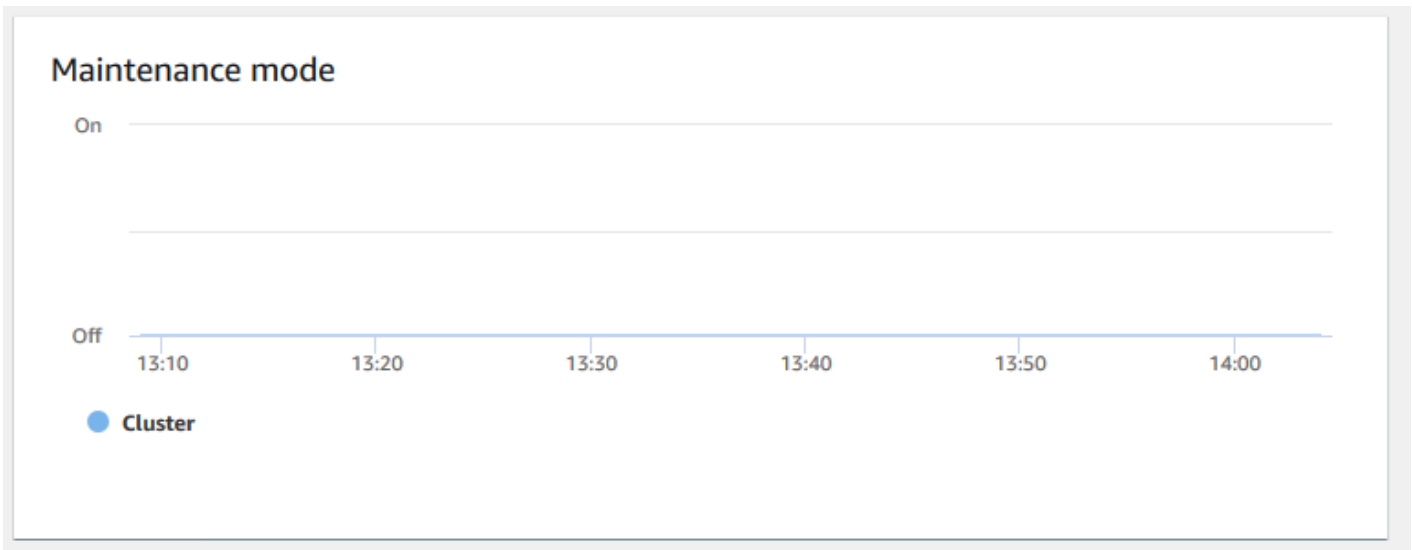
클러스터 성능 그래프

다음은 새 Amazon Redshift 콘솔에 표시되는 그래프의 예입니다.

- [CPU 사용률(CPU utilization)] – 모든 노드(리더 및 컴퓨팅)에 대한 CPU 사용률의 백분율을 표시합니다. 클러스터 마이그레이션 또는 기타 리소스를 많이 사용하는 작업을 예약하기 전에 클러스터 사용량이 가장 낮은 시간을 찾으려면 이 차트를 모니터하여 개별 노드 또는 모든 노드당 CPU 사용률을 확인합니다.

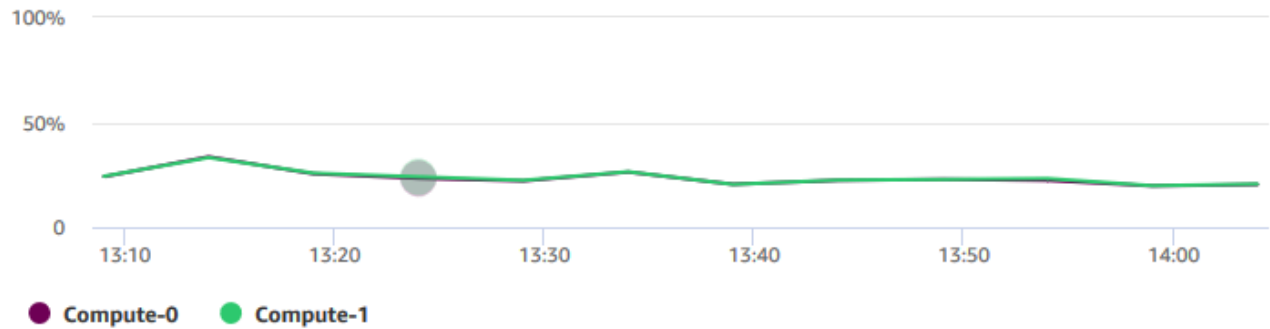


- [유지 관리 모드(Maintenance mode)] – On 및 Off 표시기를 사용하여 선택한 시간에 클러스터가 유지 관리 모드에 있는지 여부를 표시합니다. 클러스터가 유지 관리 진행 중인 시간을 확인할 수 있습니다. 그런 다음, 이 시간을 클러스터에 대해 완료된 작업과 연관시켜 반복 이벤트에 대한 향후 가동 중지 시간을 예측할 수 있습니다.



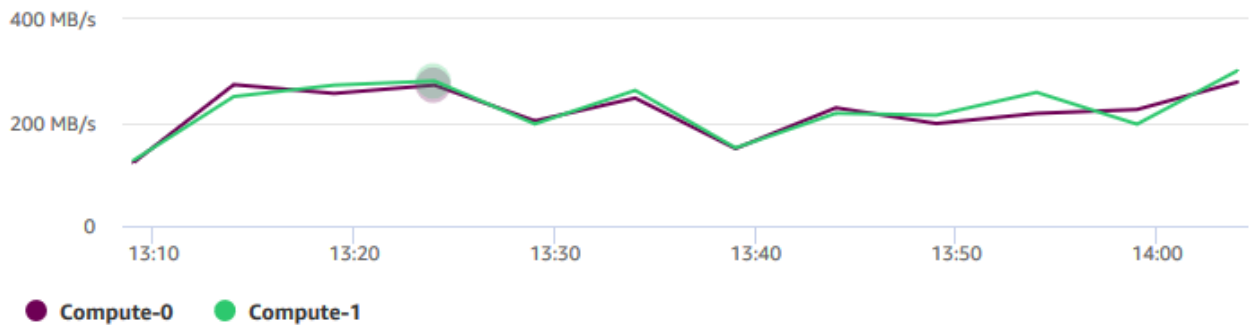
- [사용된 디스크 공간 백분율(Percentage disk space used)] – 클러스터 전체가 아니라 각 컴퓨팅 노드당 디스크 공간 사용량의 백분율을 표시합니다. 이 차트에서 디스크 사용률을 모니터링할 수 있습니다. VACUUM 및 COPY와 같은 유지 관리 작업은 정렬 작업에 대해 중간 임시 스토리지 공간을 사용하므로 디스크 사용량이 급증합니다.

Percentage disk space used



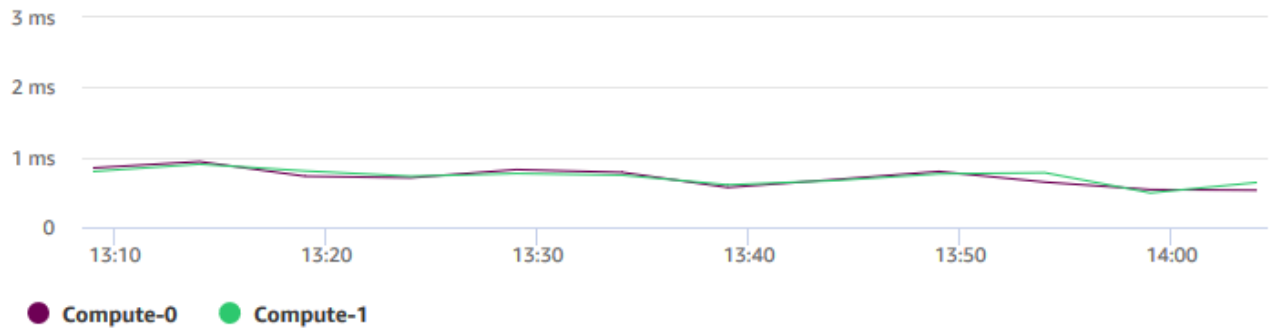
- [읽기 처리량(Read throughput)] – 초당 디스크에서 읽은 평균 메가바이트 수를 표시합니다. 이 차트를 평가하여 클러스터의 해당 물리적 측면을 모니터링할 수 있습니다. 클러스터의 인스턴스와 클러스터의 볼륨 간 네트워크 트래픽은 처리량에서 제외됩니다.

Read throughput



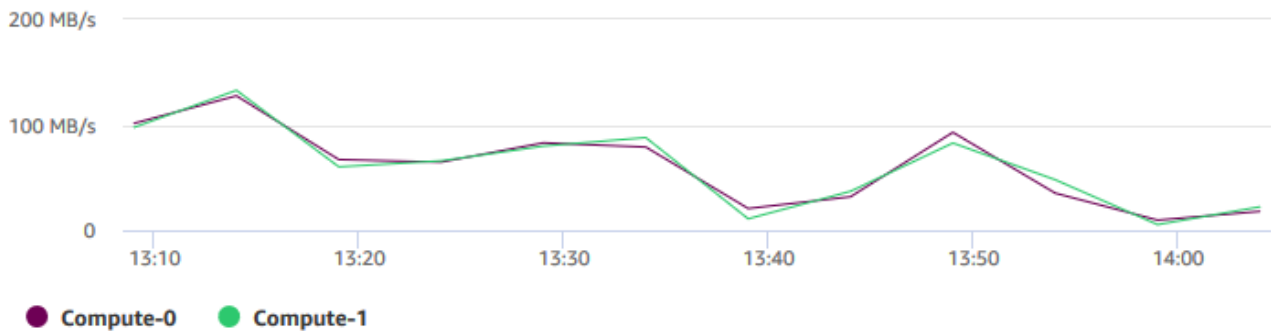
- [읽기 지연 시간(Read latency)] – 디스크 읽기 I/O 작업에 소요된 평균 시간을 밀리초 단위로 표시합니다. 반환할 데이터에 대한 응답 시간을 볼 수 있습니다. 대기 시간이 길면 발신자가 유틸 시간(새 패킷을 보내지 않음)을 더 많이 소비하므로 처리량이 증가하는 속도가 줄어듭니다.

Read latency



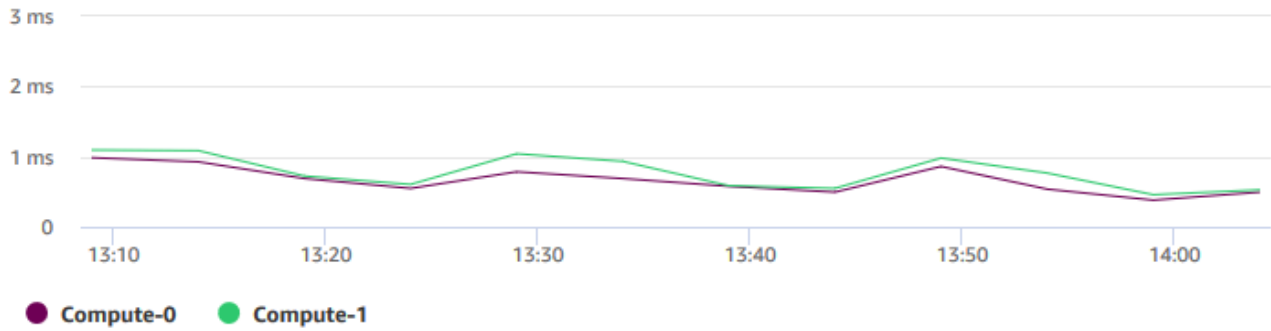
- [쓰기 처리량(Write throughput)] – 초당 디스크에 기록된 평균 메가바이트 수를 표시합니다. 이 지표를 평가하여 클러스터의 해당 물리적 측면을 모니터링할 수 있습니다. 클러스터의 인스턴스와 클러스터의 볼륨 간 네트워크 트래픽은 처리량에서 제외됩니다.

Write throughput



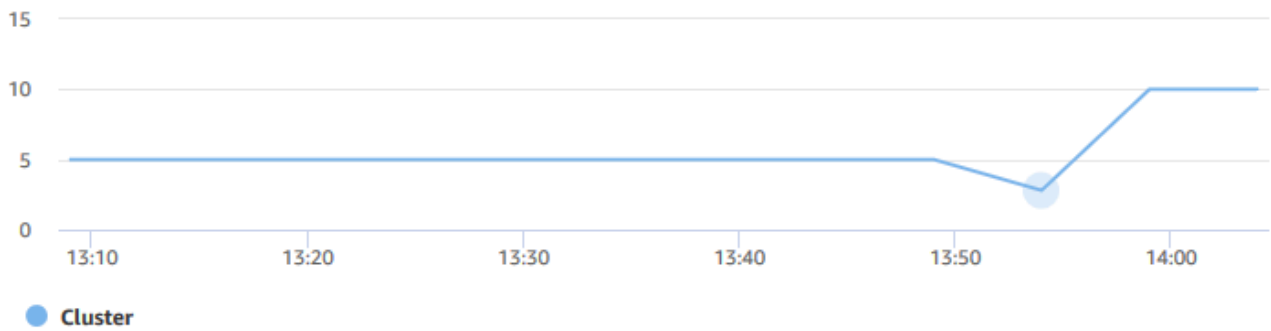
- [쓰기 지연 시간(Write latency)] – 디스크 쓰기 I/O 작업에 소요된 평균 시간을 밀리초 단위로 표시합니다. 쓰기 승인이 반환되는 시간을 평가할 수 있습니다. 대기 시간이 길면 발신자가 유틸 시간(새 패킷을 보내지 않음)을 더 많이 소비하므로 처리량이 증가하는 속도가 줄어듭니다.

Write latency

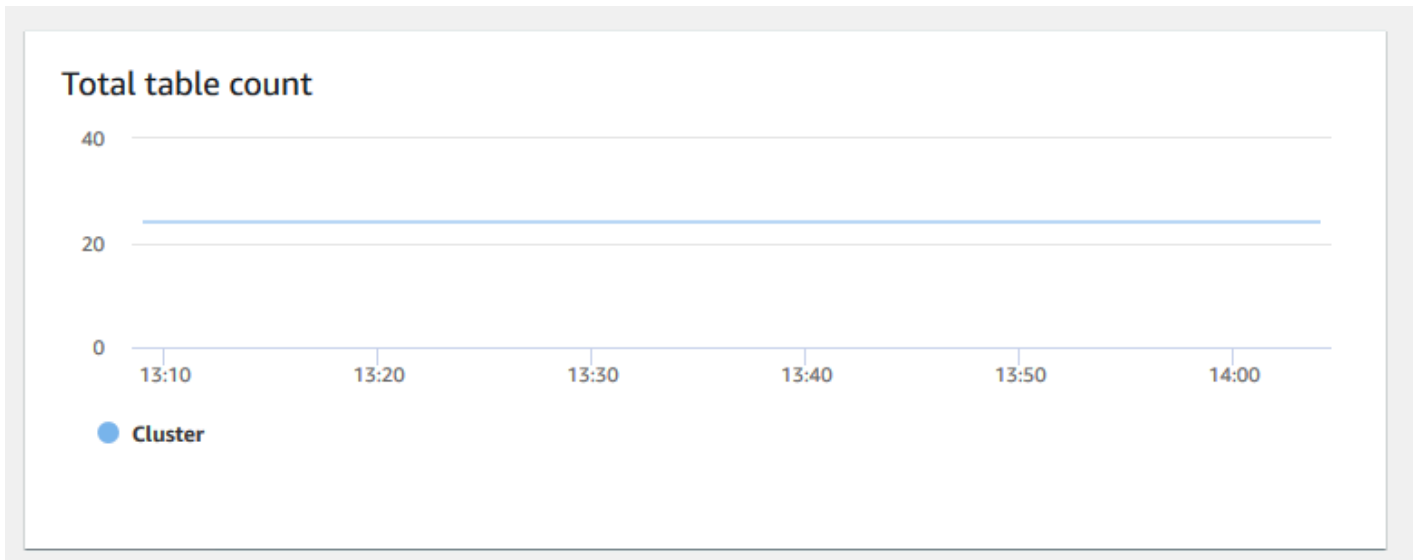


- [데이터베이스 연결(Database connections)] – 클러스터에 대한 데이터베이스 연결 수를 표시합니다. 이 차트를 사용하여 데이터베이스에 대해 설정된 연결 수를 확인하고 클러스터 사용량이 가장 낮은 시간을 찾을 수 있습니다.

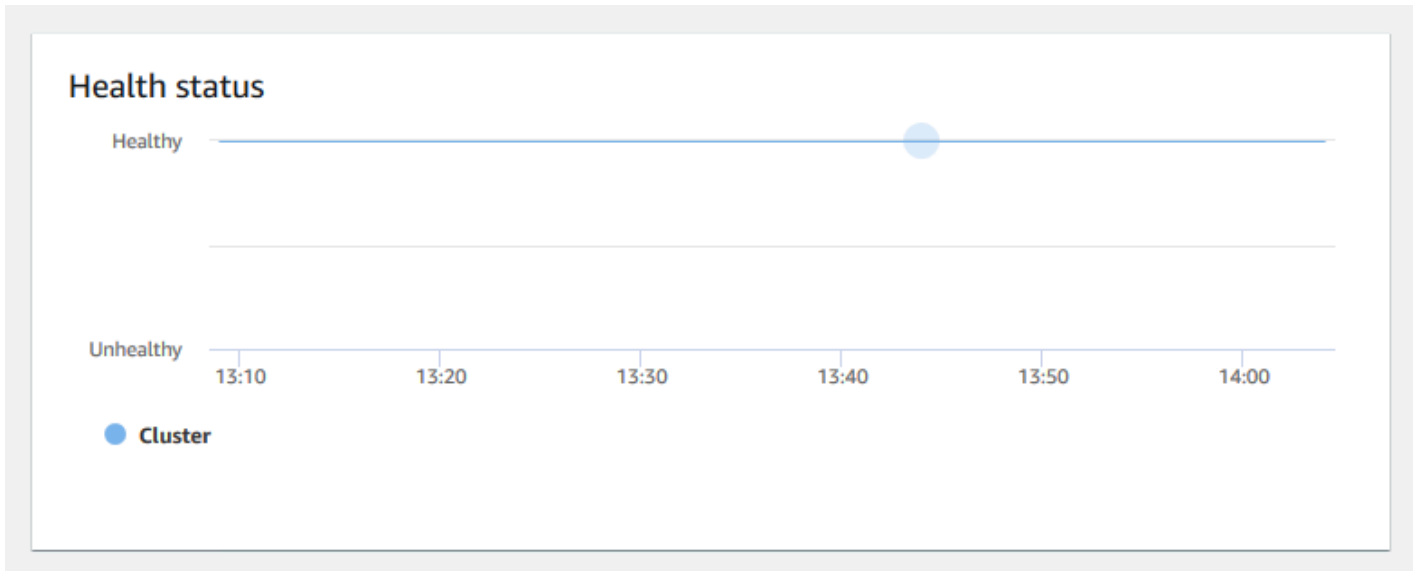
Database connections



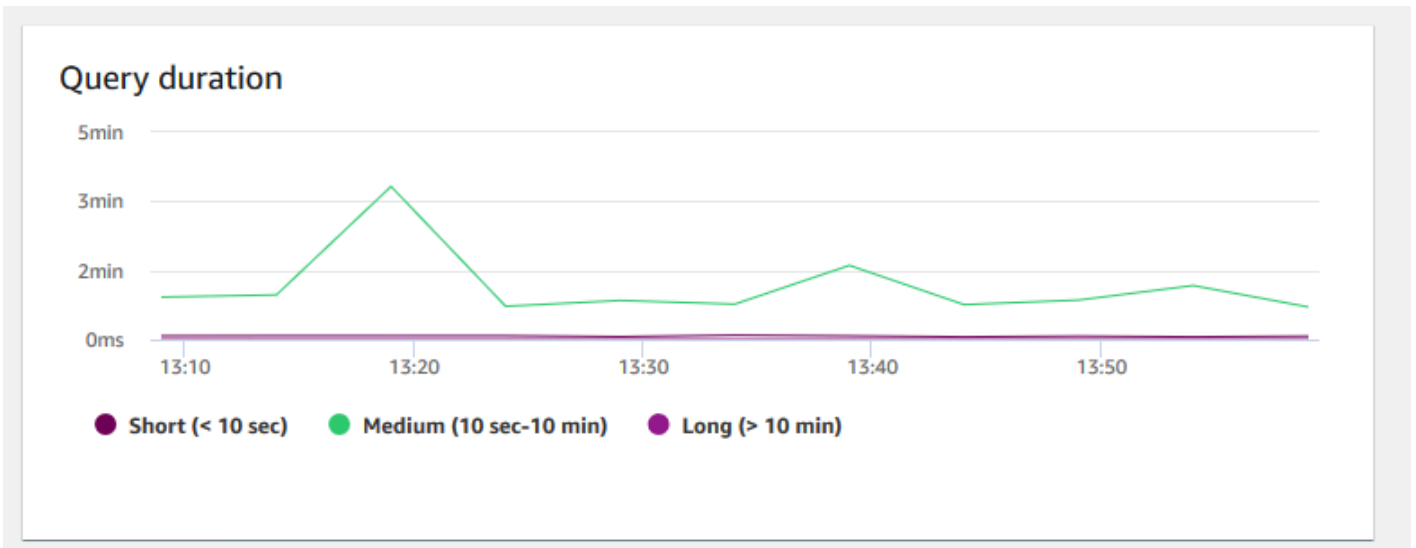
- [총 테이블 수(Total table count)] – 클러스터 내의 특정 시점에 열려 있는 사용자 테이블 수를 표시합니다. 열려 있는 테이블 수가 많을 때 클러스터 성능을 모니터링할 수 있습니다.



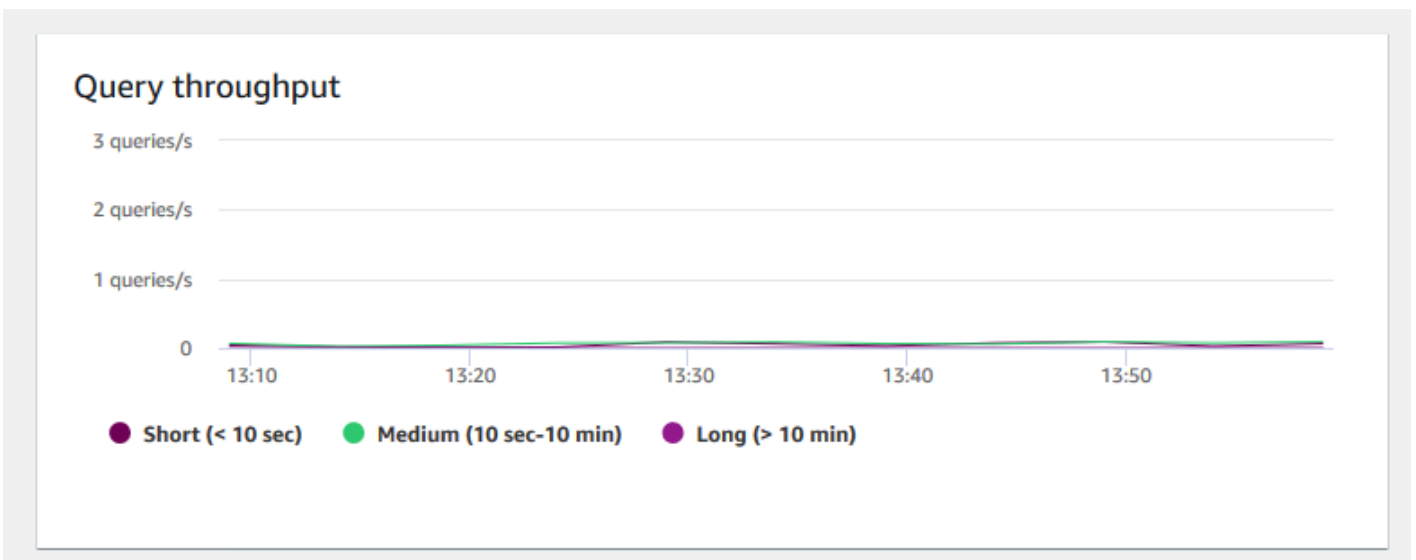
- [상태(Health status)] – 클러스터의 상태를 Healthy 또는 Unhealthy로 나타냅니다. 클러스터가 데이터베이스에 연결할 수 있고 간단한 쿼리를 성공적으로 수행하면 클러스터가 정상으로 간주됩니다. 그렇지 않다면 비정상 상태입니다. 비정상 상태는 클러스터 데이터베이스의 데이터 로드가 너무 많거나, 혹은 클러스터 데이터베이스에 구성 문제가 있을 때 발생합니다.



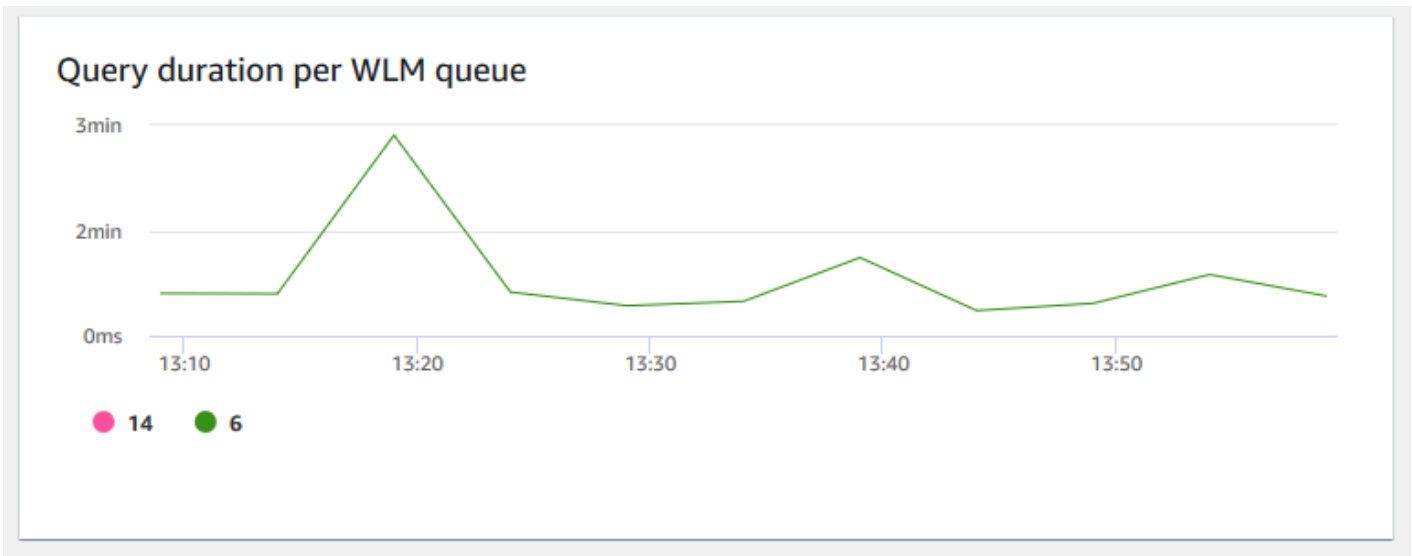
- [쿼리 기간(Query duration)] – 쿼리를 완료하는 걸리는 평균 시간을 마이크로 초 단위로 표시합니다. 이 차트의 데이터를 벤치마킹하여 클러스터 내 I/O 성능을 측정하고 필요한 경우 가장 시간이 많이 걸리는 쿼리를 조정할 수 있습니다.



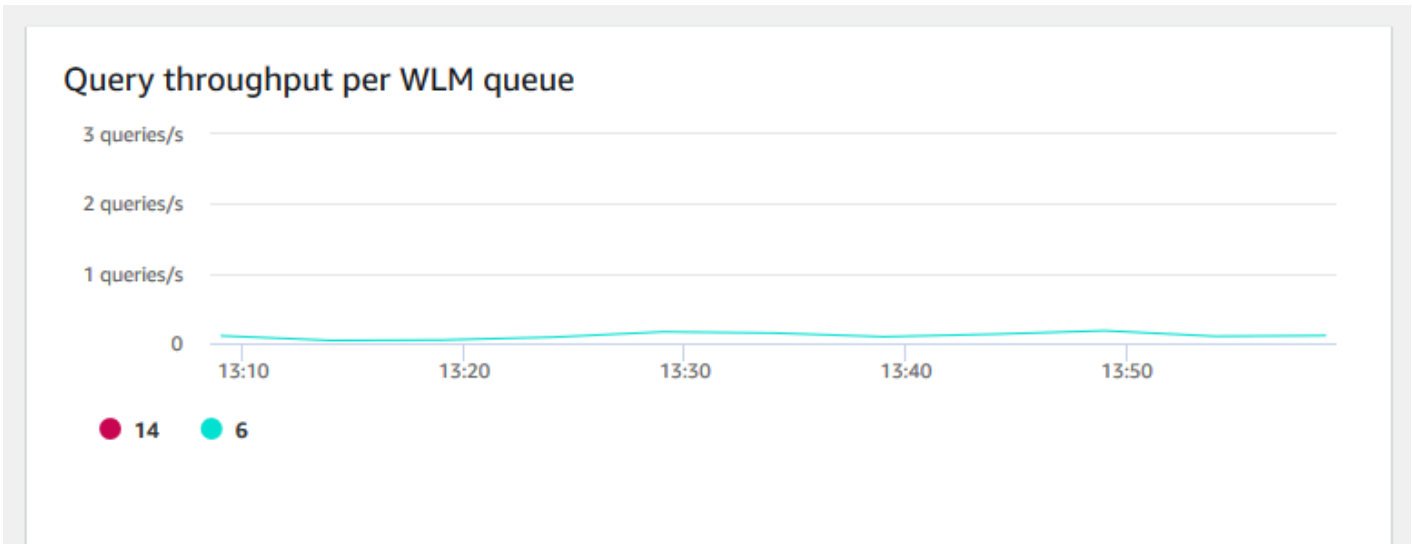
- [쿼리 처리량(Query throughput)] – 초당 완료된 평균 쿼리 수를 표시합니다. 이 차트의 데이터를 분석하여 데이터베이스 성능을 측정하고 균형 잡힌 방식으로 다중 사용자 워크로드를 지원하는 시스템의 기능을 특성화할 수 있습니다.



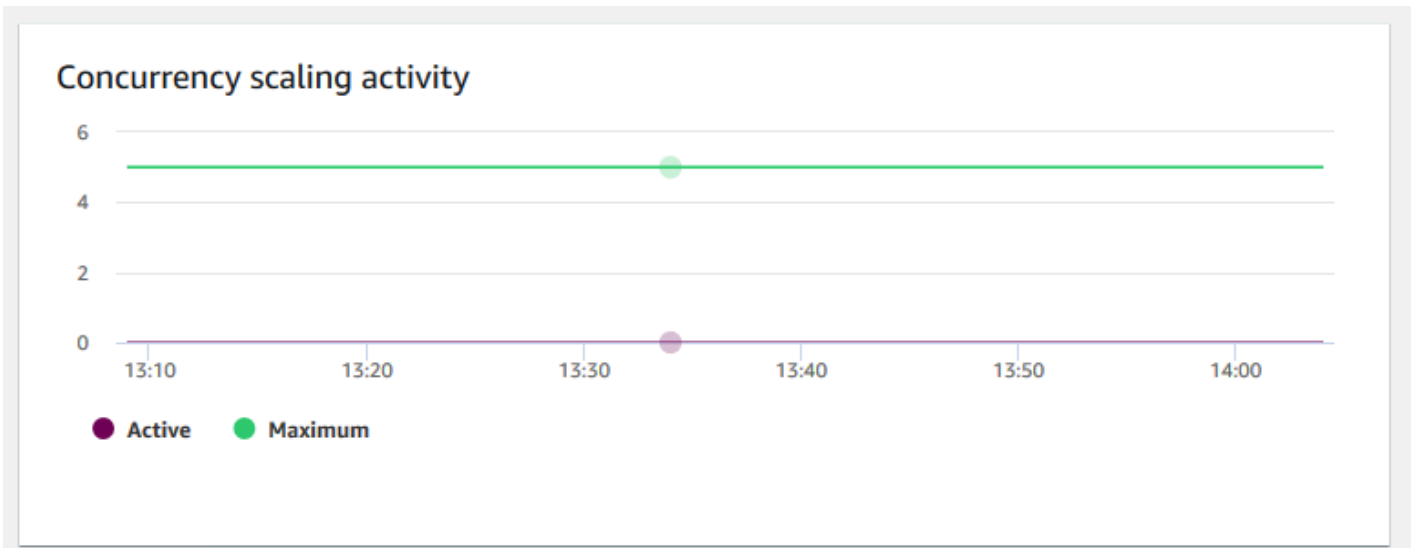
- [WLM 대기열당 쿼리 기간(Query duration per WLM queue)] – 쿼리를 완료하는 데 걸리는 평균 시간을 마이크로초 단위로 표시합니다. 이 차트의 데이터를 벤치마킹하여 WLM 대기열당 I/O 성능을 측정하고 필요한 경우 가장 시간이 많이 걸리는 쿼리를 조정할 수 있습니다.



- [WLM 대기열당 쿼리 처리량(Query throughput per WLM queue)] - 초당 완료된 평균 쿼리 수를 표시합니다. 이 차트의 데이터를 분석하여 WLM 대기열당 데이터베이스 성능을 측정할 수 있습니다.



- [동시성 조정 활동(Concurrency scaling activity)] - 활성 동시성 조정 클러스터의 수를 표시합니다. 동시성 조정이 사용되면 동시 읽기 쿼리의 증가를 처리하는 데 필요한 추가 클러스터 용량을 Amazon Redshift에서 자동으로 추가합니다.



쿼리 기록 데이터 보기

Amazon Redshift의 쿼리 기록 지표를 사용하여 다음을 수행할 수 있습니다.

- 쿼리 성능 문제를 격리하고 진단합니다.
- 동일한 타임라인에서 쿼리 실행 시간 지표와 클러스터 성능 지표를 비교하여 두 지표가 어떻게 관련되는지 확인합니다. 그러면 속도가 느린 쿼리를 식별하거나, 병목 현상 쿼리를 발견하거나, 워크로드 클러스터의 크기의 조정 필요성을 결정하는 데 도움이 됩니다.
- 타임라인에서 선택하여 특정 쿼리의 세부 정보로 드릴다운합니다. 쿼리 ID 및 기타 속성이 그래프 아래의 행에 표시되면 쿼리를 선택하여 쿼리 세부 정보를 볼 수 있습니다. 세부 정보에는 쿼리의 SQL 문, 실행 세부 정보 및 쿼리 계획 등이 포함됩니다. 자세한 내용은 [쿼리 세부 정보 보기 및 분석](#) 섹션을 참조하세요.
- 로드 작업이 성공적으로 완료되고 SLA(서비스 수준 계약)를 충족하는지 확인합니다.

쿼리 기록 데이터를 표시하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 목록에서 클러스터 이름을 선택하여 세부 정보를 엽니다. 클러스터의 세부 정보가 표시됩니다. 여기에는 클러스터 성능, 쿼리 모니터링, 데이터베이스, 데이터 공유, 일정, 유지 관리 및 속성 탭이 포함될 수 있습니다.
3. 쿼리에 대한 지표를 보려면 Query monitoring(쿼리 모니터링) 탭을 선택합니다.
4. 쿼리 모니터링 섹션에서 쿼리 기록 탭을 선택합니다.

창의 컨트롤을 사용하여 쿼리 목록과 클러스터 지표 간에 전환할 수 있습니다.

쿼리 목록을 선택하면 탭에 다음과 같은 그래프가 포함됩니다.

- [쿼리 실행 시간(Query runtime)] – 타임라인의 쿼리 활동입니다. 이 그래프를 사용하여 동일한 기간에 실행 중인 쿼리를 확인합니다. 쿼리를 선택하여 더 많은 쿼리 실행 세부 정보를 확인합니다. X축에는 선택한 기간이 표시됩니다. 그래프로 된 쿼리를 실행 중, 완료됨, 로드 등을 기준으로 필터링할 수 있습니다. 각 막대는 쿼리를 나타내고 막대의 길이는 막대의 시작부터 끝까지 실행 시간을 나타냅니다. 쿼리에는 SQL 데이터 조작 문(예: SELECT, INSERT, DELETE)과 로드(예: COPY)가 포함될 수 있습니다. 기본적으로 가장 긴 실행 쿼리 100개가 선택한 기간 동안 표시됩니다.
- [쿼리 및 로드(Queries and loads)] – 클러스터에서 실행된 쿼리 및 로드 목록입니다. 이 창에는 쿼리가 현재 실행 중인 경우 쿼리를 종료하는 옵션이 포함되어 있습니다.

클러스터 지표를 선택하면 이 탭에는 다음 그래프가 포함됩니다.

- [쿼리 실행 시간(Query runtime)] – 타임라인의 쿼리 활동입니다. 이 그래프를 사용하여 동일한 기간에 실행 중인 쿼리를 확인합니다. 쿼리를 선택하여 더 많은 쿼리 실행 세부 정보를 확인합니다.
- [CPU 사용률(CPU utilization)] – 리더 노드별 클러스터의 CPU 사용률 및 계산 노드의 평균입니다.
- [사용된 스토리지 용량(Storage capacity used)] – 사용된 스토리지 용량의 백분율입니다.
- [활성 데이터베이스 연결(Active database connections)] – 클러스터에 대한 활성 데이터베이스 연결 수입니다.

쿼리 기록 그래프로 작업할 때는 다음 사항을 고려하십시오.

- 쿼리 실행 시간 차트에서 특정 쿼리를 나타내는 막대를 선택하여 해당 쿼리에 대한 세부 정보를 확인합니다. 쿼리 및 로드 목록에서 쿼리 ID를 선택하여 세부 정보를 확인할 수도 있습니다.
- 스와이프하여 쿼리 실행 시간 차트의 섹션을 선택하여 특정 기간을 표시하도록 확대할 수 있습니다.
- 쿼리 실행 시간 차트에서 선택한 필터로 모든 데이터를 고려하려면 쿼리 및 로드 목록에 나열된 모든 페이지를 앞으로 이동합니다.
- 설정 기어 아이콘으로 표시되는 기본 설정 창을 사용하여 [쿼리 및 로드(Queries and loads)] 목록에 표시되는 열과 행 수를 변경할 수 있습니다.

- 쿼리 및 로드 목록은 왼쪽 탐색기의 쿼리 아이콘, 쿼리 및 로드를 탐색하여 표시할 수도 있습니다. 자세한 내용은 [쿼리 및 로드 보기](#) 섹션을 참조하세요.

쿼리 기록 그래프

다음은 새 Amazon Redshift 콘솔에 표시되는 그래프의 예입니다.

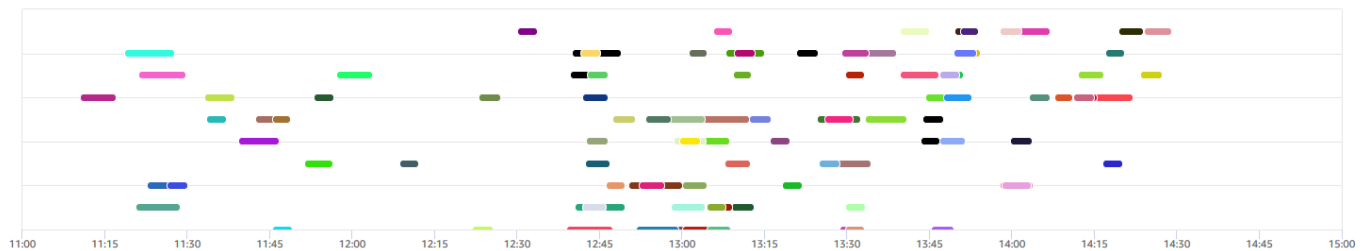
Note

Amazon Redshift 콘솔 그래프에는 최신 100,000개의 쿼리에 대한 데이터만 포함됩니다.

쿼리 실행 시간

Query runtime

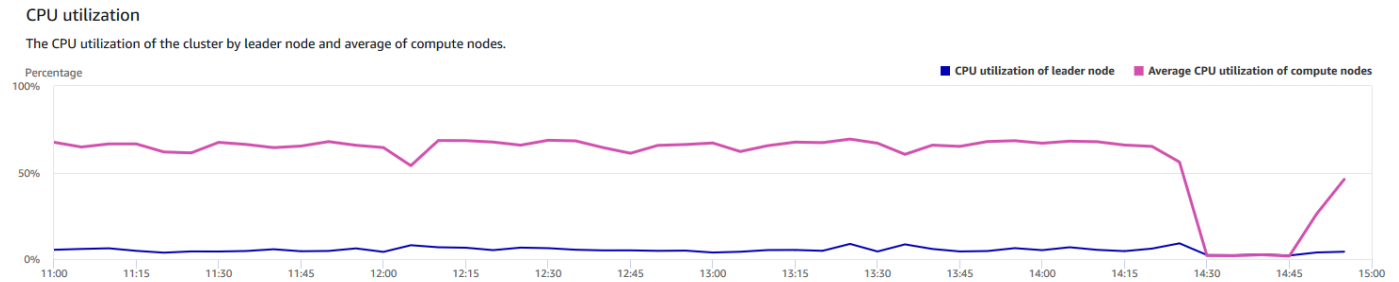
The query activity on a timeline. Use this graph to see which queries are running in the same timeframe. Choose a query to view more query execution details.



쿼리 및 로드

Queries and loads(100)										Terminate query
Filter queries										
☐	Start time	Query	Status	Duration	SQL	Copy SQL	User	Transaction ID		
☐	Apr 13th, 2020 01:00:55 PM 8 days ago	69248	Completed	11 min	with /* query_templates/query67.tpl0 ICF:IR-09c6a4cc-6ec8-11e a-8047-06872b3fecc8.stream_10cu_run01_nocache.stream-quer ...	Copy	rsperf	105501		
☐	Apr 13th, 2020 12:58:07 PM 8 days ago	69199	Completed	11 min	with /* query_templates/query67.tpl0 ICF:IR-09c6a4cc-6ec8-11e a-8047-06872b3fecc8.stream_10cu_run01_nocache.stream-quer ...	Copy	rsperf	105414		
☐	Apr 13th, 2020 12:54:15 PM 8 days ago	69111,69265,69253	Completed	10 min	with /* query_templates/query22.tpl0 ICF:IR-09c6a4cc-6ec8-11e a-8047-06872b3fecc8.stream_10cu_run01_nocache.stream-quer ...	Copy	rsperf	105283		
☐	Apr 13th, 2020 12:50:17 PM 8 days ago	68976	Completed	10 min	with /* query_templates/query67.tpl0 ICF:IR-09c6a4cc-6ec8-11e a-8047-06872b3fecc8.stream_10cu_run01_nocache.stream-quer ...	Copy	rsperf	105128		
☐	Apr 13th, 2020 01:29:23 PM 8 days ago	70089	Completed	10 min	with /* query_templates/query67.tpl0 ICF:IR-09c6a4cc-6ec8-11e a-8047-06872b3fecc8.stream_10cu_run01_nocache.stream-quer ...	Copy	rsperf	106659		
☐	Apr 13th, 2020 11:18:35 AM 8 days ago	65543	Completed	9 min	with /* query_templates/query67.tpl0 ICF:IR-09c6a4cc-6ec8-11e a-8047-06872b3fecc8.stream_05cu_run01_nocache.stream-quer ...	Copy	rsperf	101092		
☐	Apr 13th, 2020 12:40:30 PM 8 days ago	68729	Completed	9 min	with /* query_templates/query67.tpl0 ICF:IR-09c6a4cc-6ec8-11e a-8047-06872b3fecc8.stream_10cu_run01_nocache.stream-quer ...	Copy	rsperf	104789		

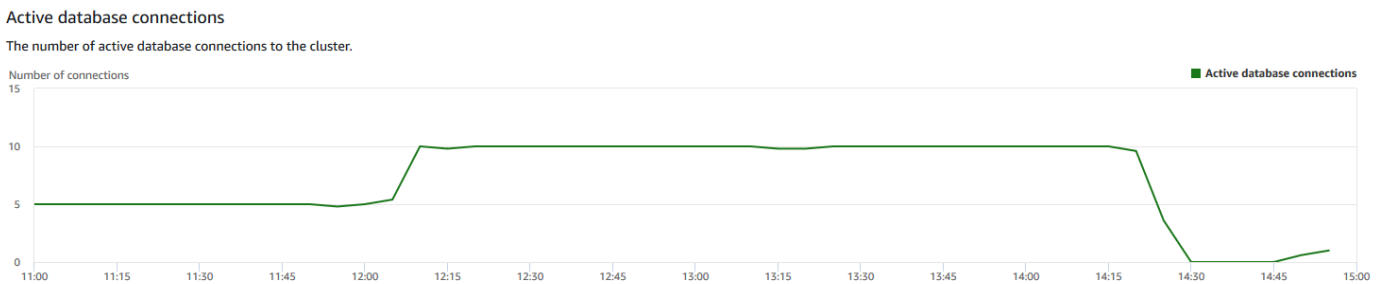
CPU 사용률



• 사용된 스토리지 용량



• 활성 데이터베이스 연결



데이터베이스 성능 데이터 보기

Amazon Redshift의 데이터베이스 성능 지표를 사용하여 다음을 수행할 수 있습니다.

- 쿼리에 소요된 시간을 처리 단계별로 분석합니다. 단계별로 소요된 시간의 비정상적인 추세를 찾을 수 있습니다.
- 쿼리 수, 기간 및 쿼리 처리량을 기간 범위(짧은, 중간, 긴)별로 분석합니다.
- 쿼리 우선 순위(가장 낮음, 낮음, 보통, 높음, 가장 높음, 위험)별로 쿼리 대기 시간의 추세를 찾습니다.
- WLM 대기열별 쿼리 기간, 처리량 또는 대기 시간의 추세를 확인합니다.

데이터베이스 성능 데이터를 표시하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 목록에서 클러스터 이름을 선택하여 세부 정보를 엽니다. 클러스터의 세부 정보가 표시됩니다. 여기에는 클러스터 성능, 쿼리 모니터링, 데이터베이스, 데이터 공유, 일정, 유지 관리 및 속성 탭이 포함될 수 있습니다.
3. 쿼리에 대한 지표를 보려면 Query monitoring(쿼리 모니터링) 탭을 선택합니다.
4. 쿼리 모니터링 섹션에서 데이터베이스 성능 탭을 선택합니다.

창의 컨트롤을 사용하여 클러스터 지표와 WLM 대기열 지표 간에 전환할 수 있습니다.

클러스터 지표를 선택하면 이 탭에는 다음 그래프가 포함됩니다.

- [워크로드 실행 분석(Workload execution breakdown)] – 쿼리 처리 단계에서 사용된 시간입니다.
- [기간 범위별 쿼리(Queries by duration range)] – 짧은, 중간 및 긴 쿼리의 수입입니다.
- [쿼리 처리량(Query throughput)] – 초당 완료된 평균 쿼리 수입입니다.
- [쿼리 기간(Query duration)] – 쿼리를 완료하는 데 걸리는 평균 시간입니다.
- [우선 순위별 평균 대기열 대기 시간(Average queue wait time by priority)] – 쿼리 우선 순위를 기준으로 WLM 대기열에서 쿼리가 대기하는 총 시간입니다.

WLM 대기열 지표를 선택하면 이 탭에는 다음 그래프가 포함됩니다.

- [대기열별 쿼리 기간(Query duration by queue)] – WLM 대기열별 평균 쿼리 기간입니다.
- [대기열별 쿼리 처리량(Query throughput by queue)] – WLM 대기열별 초당 완료된 평균 쿼리 수입입니다.
- [대기열별 쿼리 대기 시간(Query wait time by queue)] – WLM 대기열별 평균 쿼리 대기 기간입니다.

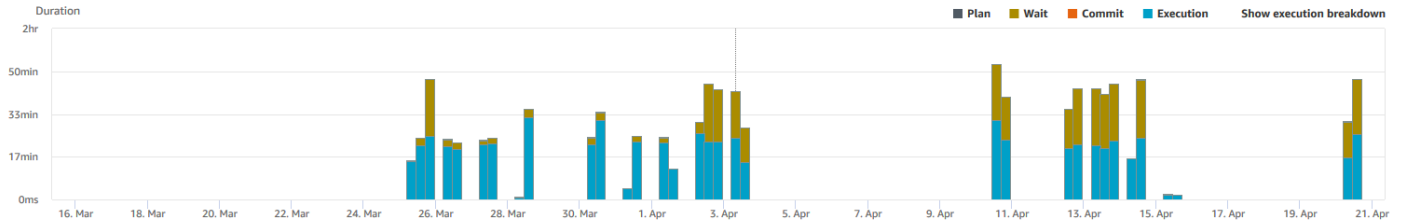
데이터베이스 성능 그래프

다음은 새 Amazon Redshift 콘솔에 표시되는 그래프의 예입니다.

- 워크로드 실행 분석

Workload execution breakdown

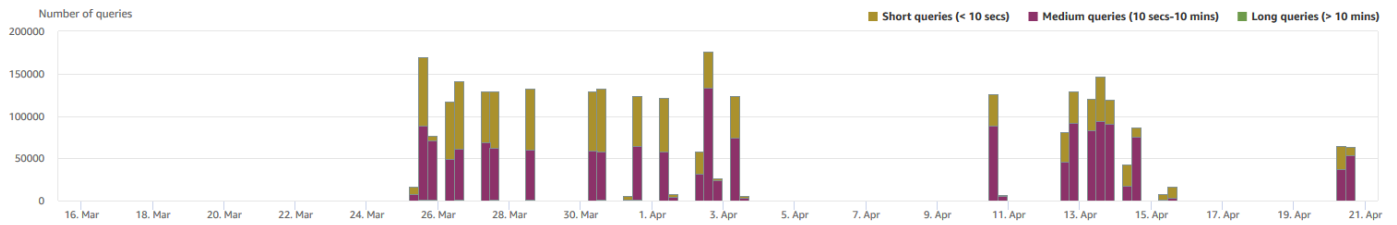
The time used in query processing stages.



- 기간 범위별 쿼리

Queries by duration range

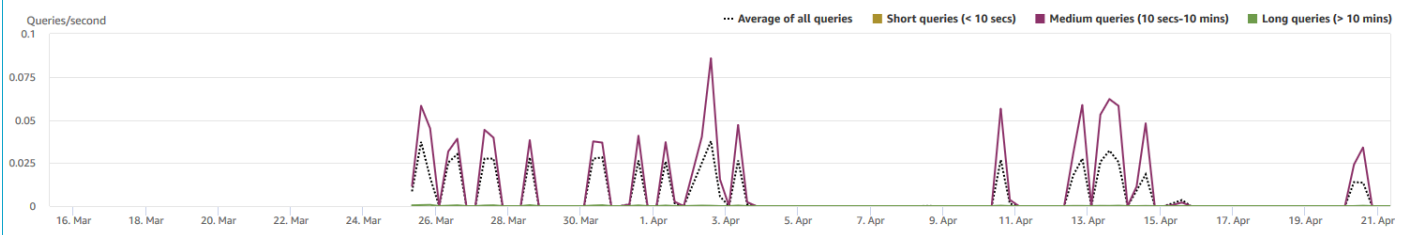
The number of short, medium and long queries.



- 쿼리 처리량

Query throughput

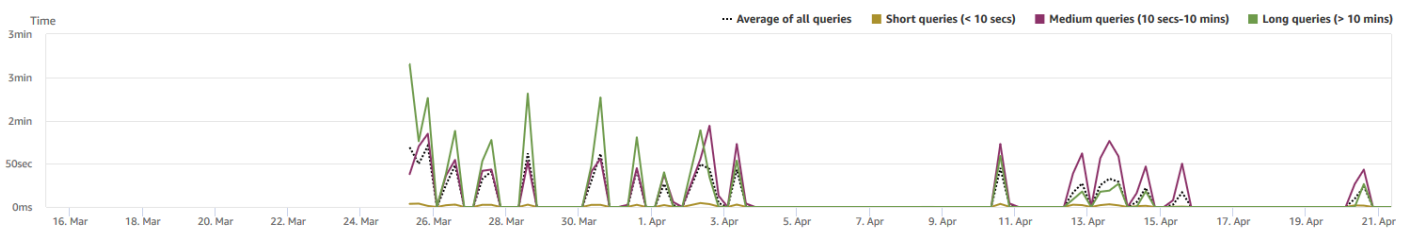
The average number of queries completed per second.



- 쿼리 지속시간

Query Duration

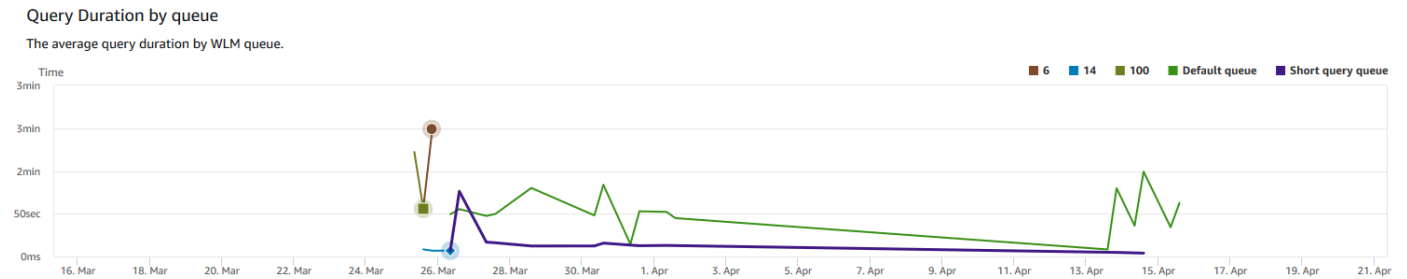
The average amount of time to complete a query.



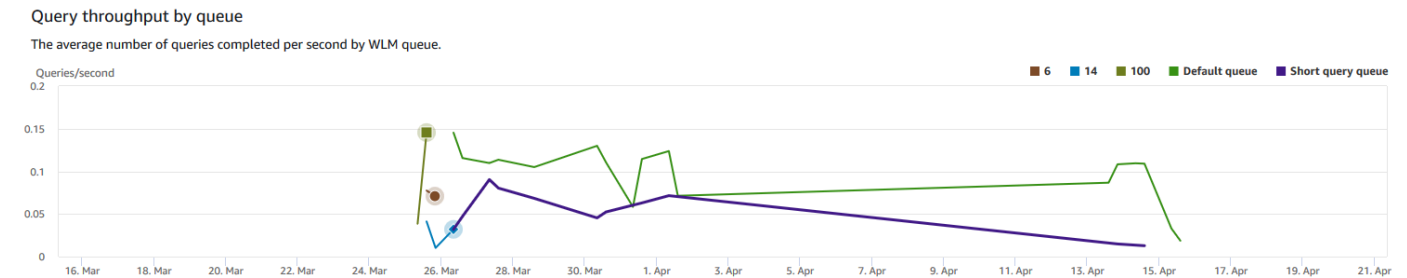
- 우선 순위별 평균 대기열 대기 시간



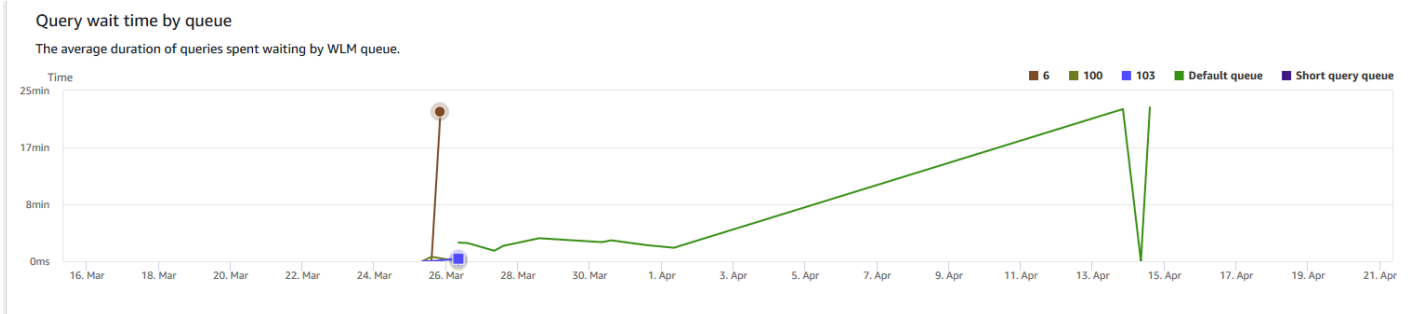
• 대기열별 쿼리 기간



• 대기열별 쿼리 처리량



• 대기열별 쿼리 대기 시간



워크로드 동시성 및 동시성 확장 데이터 보기

Amazon Redshift에서 동시성 조정 지표를 사용해 할 수 있는 작업은 다음과 같습니다.

- 동시성 확장을 활성화하여 대기 중인 쿼리 수를 줄일 수 있는지 여부를 분석합니다. WLM 대기열별로 또는 모든 WLM 대기열에 대해 비교할 수 있습니다.
- 동시성 확장 클러스터의 동시성 확장 활동을 확인합니다. 이를 통해 동시성 확장이 `max_concurrency_scaling_clusters`로 제한을 받는지 여부를 알 수 있습니다. 이러한 알림을 받으면 DB 파라미터에서 `max_concurrency_scaling_clusters`를 늘리도록 선택할 수 있습니다.
- 모든 동시성 확장 클러스터에 걸쳐 합계를 낸 총 동시성 확장 사용량을 확인할 수 있습니다.

동시성 확장 데이터를 표시하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 목록에서 클러스터 이름을 선택하여 세부 정보를 엽니다. 클러스터의 세부 정보가 표시됩니다. 여기에는 클러스터 성능, 쿼리 모니터링, 데이터베이스, 데이터 공유, 일정, 유지 관리 및 속성 탭이 포함될 수 있습니다.
3. 쿼리에 대한 지표를 보려면 Query monitoring(쿼리 모니터링) 탭을 선택합니다.
4. 쿼리 모니터링 섹션에서 워크로드 동시성 탭을 선택합니다.

이 탭에는 다음과 같은 그래프가 포함되어 있습니다.

- [클러스터에서 대기 중인 쿼리와 실행 중인 쿼리 비교(Queued vs. Running queries on the cluster)] – 클러스터의 모든 WLM 대기열에서 대기 중인 쿼리 수와 비교하여 (기본 클러스터 및 동시성 조정 클러스터에서) 실행 중인 쿼리 수입니다.
- [대기열당 대기 중인 쿼리와 실행 중인 쿼리 비교(Queued vs. Running queries per queue)] – 각 WLM 대기열에서 대기 중인 쿼리 수와 비교하여 (기본 클러스터 및 동시성 조정 클러스터에서) 실행 중인 쿼리 수입니다.
- [동시성 조정 활동(Concurrency scaling activity)] – 쿼리를 능동적으로 처리하고 있는 동시성 조정 클러스터 수입니다.
- [동시성 조정 사용량(Concurrency scaling usage)] – 활성 쿼리 처리 활동이 있는 동시성 조정 클러스터의 사용량입니다.

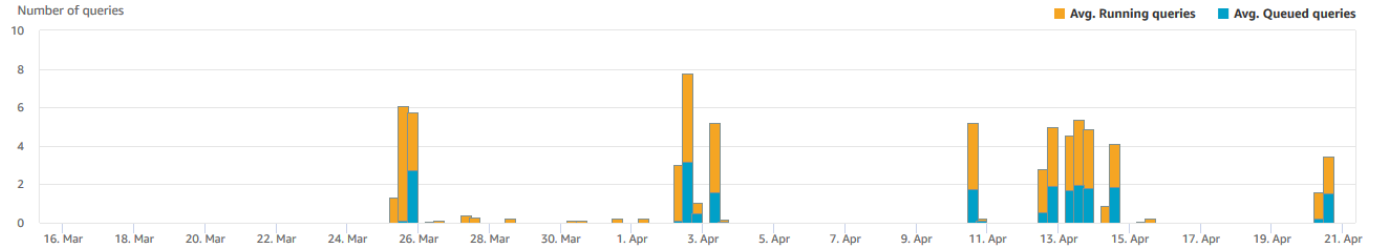
워크로드 동시성 그래프

다음은 새 Amazon Redshift 콘솔에 표시되는 그래프의 예입니다. Amazon CloudWatch에서 유사한 그래프를 만들려면 동시성 크기 조정 및 WLM CloudWatch 지표를 사용할 수 있습니다. Amazon Redshift용 CloudWatch 지표에 대한 자세한 내용은 [Amazon Redshift의 성능 데이터](#) 섹션을 참조하세요.

클러스터에서 대기 중인 쿼리와 실행 중인 쿼리 비교

Queued vs. Running queries on the cluster

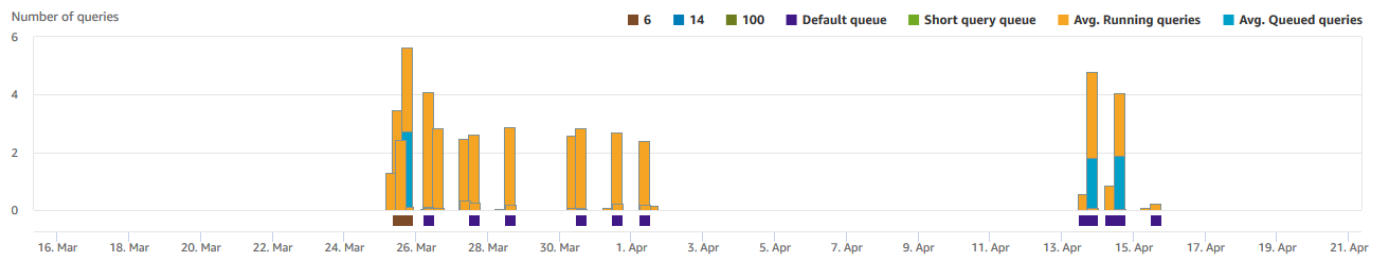
The number of queries running (from the main cluster and concurrency scaling cluster) compared to the number of queries waiting in all WLM queues in the cluster.



대기열당 대기 중인 쿼리와 실행 중인 쿼리 비교

Queued vs. Running queries per queue

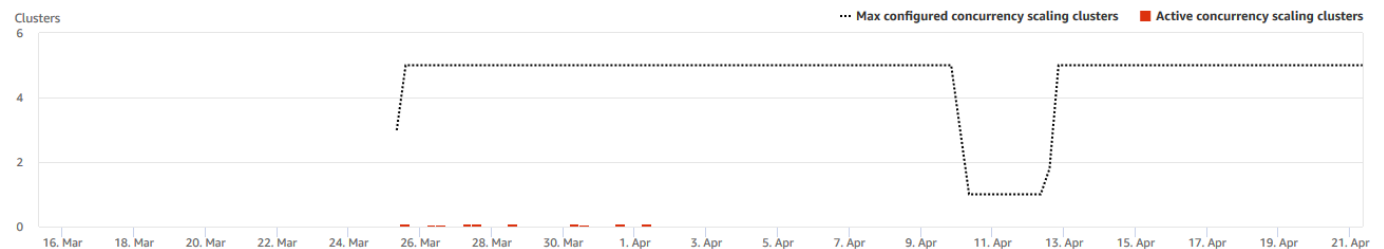
The number of queries running (from the main cluster and concurrency scaling cluster) compared to the number of queries waiting in each WLM queue.



동시성 확장 활동

Concurrency scaling activity

The number of concurrency scaling clusters that are actively processing queries.



Concurrency scaling usage(동시성 확장 사용)

Concurrency scaling usage

The usage of concurrency scaling clusters that have active query processing activity.



자동 최적화 데이터 보기

Amazon Redshift 콘솔은 추가 컴퓨팅 리소스를 사용하여 실행되는 자동 최적화 또는 Autonomics에 대한 정보를 제공합니다. 이 정보를 사용하여 사용량을 추적하고 사용량 한도에 도달했는지 모니터링할 수 있습니다. Amazon Redshift는 프로비저닝된 클러스터 자체에서 실행되는 Autonomics에 대해 요금을 청구하지 않지만 추가 컴퓨팅 리소스를 사용하여 실행되는 Autonomics에 대해서는 요금을 청구합니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [자동 데이터베이스 최적화를 위한 추가 컴퓨팅 리소스 할당](#)을 참조하세요.

추가 컴퓨팅 Autonomics 데이터를 보려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 목록에서 클러스터의 이름을 선택하여 세부 정보를 엽니다.
3. 클러스터의 세부 정보 페이지의 작업 드롭다운 메뉴에서 사용량 한도 관리를 선택합니다. 클러스터의 유지 관리 탭을 선택한 다음 아래로 스크롤하여 사용량 한도 생성을 선택할 수도 있습니다.
4. 추가 컴퓨팅 Autonomics 데이터를 보여주는 그래프는 자동 최적화를 위한 추가 컴퓨팅에 대한 사용량 한도라는 제목의 섹션 아래에 표시됩니다. 그래프에는 Amazon Redshift가 지정된 기간 동안 추가 컴퓨팅 리소스를 사용하여 Autonomics를 실행하는 시간이 표시됩니다.

쿼리 및 로드 보기

Amazon Redshift 콘솔은 데이터베이스에서 실행되는 쿼리 및 로드(Queries and loads)에 대한 정보를 제공합니다. 이 정보를 사용하여 장시간 처리로 병목 현상을 유발하여 다른 쿼리의 효율적 처리를 방해하는 쿼리를 식별하고 문제를 해결할 수 있습니다. Amazon Redshift 콘솔에서 쿼리 정보를 사용하여 쿼리 처리를 모니터링할 수 있습니다.

쿼리 성능 데이터를 표시하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 쿼리를 선택한 다음 쿼리 및 로드(Queries and loads)를 선택하여 계정에 대한 쿼리 목록을 표시합니다.

기본적으로 이 목록에는 지난 24시간 동안의 모든 클러스터에 대한 쿼리가 표시됩니다. 콘솔에서 표시된 날짜의 범위를 변경할 수 있습니다.

 Important

Queries and loads(쿼리 및 로드) 목록에 시스템의 가장 긴 실행 쿼리(최대 100개의 쿼리)가 표시됩니다.

쿼리 세부 정보 보기 및 분석

쿼리 식별자를 사용하여 쿼리의 세부 정보를 볼 수 있습니다. 세부 정보에는 쿼리의 완료 상태, 기간, SQL 문, 사용자 쿼리인지 또는 Amazon Redshift에 의해 다시 작성된 쿼리인지 여부 등이 포함될 수 있습니다. 사용자 쿼리는 SQL 클라이언트에서 Amazon Redshift에 제출되거나 비즈니스 인텔리전스 도구에 의해 생성되는 쿼리입니다. Amazon Redshift는 쿼리를 다시 작성하여 최적화할 수 있으며 이로 인해 쿼리가 여러 번 다시 작성될 수 있습니다. 프로세스는 Amazon Redshift에 의해 수행되지만 사용자 쿼리와 함께 쿼리 세부 정보 페이지에 다시 작성된 쿼리가 표시됩니다.

쿼리를 보려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 쿼리를 선택한 다음 쿼리 및 로드(Queries and loads)를 선택하여 계정에 대한 쿼리 목록을 표시합니다. 쿼리를 찾으려면 이 페이지의 설정을 변경해야 할 수도 있습니다.
3. 목록에서 쿼리 식별자를 선택하여 쿼리 세부 정보를 표시합니다.

쿼리 세부 정보 페이지에는 쿼리 세부 정보 및 Query plan(쿼리 계획) 탭이 쿼리 지표와 함께 포함되어 있습니다.

지표에는 시작 시간, 쿼리 ID, 상태 및 기간과 같은 쿼리에 대한 세부 정보가 포함됩니다. 다른 세부 정보에는 쿼리가 기본 클러스터에서 실행되었는지 동시성 확장 클러스터에서 실행되었는지 여부 및 상위 쿼리인지 다시 작성된 쿼리인지 여부가 포함됩니다.

쿼리 실행 시 클러스터 성능 보기

쿼리 실행 시 클러스터의 성능을 모니터링하여 잠재적 병목 현상을 식별하고 쿼리 실행을 최적화할 수 있습니다. 쿼리 실행 시 클러스터 성능을 확인하면 CPU 사용률, 디스크 I/O, 네트워크 트래픽과 같은 시스템 수준 지표와 실행 시간, 처리된 데이터, 쿼리 단계와 같은 쿼리 수준의 세부 정보를 실시간으로 볼 수 있습니다. 다음 절차는 프로비저닝된 클러스터를 효과적으로 관리하고 최적화하기 위한 성능 지표에 액세스하고 이를 해석하는 방법을 안내합니다.

쿼리 실행 시 클러스터 성능을 표시하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 목록에서 클러스터 이름을 선택하여 세부 정보를 엽니다. 클러스터의 세부 정보가 표시됩니다. 여기에는 클러스터 성능, 쿼리 모니터링, 데이터베이스, 데이터 공유, 일정, 유지 관리 및 속성 탭이 포함될 수 있습니다.
3. 더 자세한 세부 정보를 보려면 Query monitoring(쿼리 모니터링) 탭을 선택합니다.

자세한 내용은 [쿼리 기록 데이터 보기](#) 섹션을 참조하세요.

로드 작업 중 클러스터 지표 보기

로드 작업 도중 클러스터 성능을 확인하면 리소스를 소비하는 쿼리를 식별하여 영향을 최소화할 수 있는 조치를 취할 수 있습니다. 혹은 로드 작업을 완료할 필요가 없는 경우에는 중간에 종료할 수도 있습니다.

Note

Amazon Redshift 콘솔에서 쿼리 및 로드를 종료하려면 특정 권한이 필요합니다. 사용자가 쿼리 및 로드를 종료할 수 있도록 하려면 `redshift:CancelQuerySession` 작업을 IAM(AWS Identity and Access Management) 정책에 추가해야 합니다. 이 요구 사항은 Amazon Redshift 읽기 전용 AWS 관리형 정책을 선택하거나 IAM에서 사용자 정의 정책을 생성하는지 여부에 상관없이 적용됩니다. Amazon Redshift 전체 액세스 권한 정책이 적용되는 사용자는 쿼리 및 로드를 종료하는 데 필요한 권한을 이미 보유하고 있습니다. Amazon Redshift의 IAM 정책 작업에 대한 자세한 내용은 [리소스 액세스 관리](#) 섹션을 참조하세요.

로드 작업 중 클러스터 성능을 표시하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 목록에서 클러스터 이름을 선택하여 세부 정보를 엽니다. 클러스터의 세부 정보가 표시됩니다. 여기에는 클러스터 성능, 쿼리 모니터링, 데이터베이스, 데이터 공유, 일정, 유지 관리 및 속성 탭이 포함될 수 있습니다.
3. 더 자세한 세부 정보를 보려면 Query monitoring(쿼리 모니터링) 탭을 선택합니다.

4. Queries and loads(쿼리 및 로드) 섹션에서 로드를 선택하여 클러스터의 로드 작업을 표시합니다. 로드가 실행 중인 경우 쿼리 종료를 선택하여 종료할 수 있습니다.

클러스터 워크로드 분석 차트 보기

콘솔에서 워크로드 실행 분석 차트를 확인하면 워크로드의 성능을 자세히 파악할 수 있습니다. 이 차트는 QueryRuntimeBreakdown 지표에서 제공한 데이터를 사용하여 작성됩니다. 이 차트를 사용하여 다양한 처리 단계(대기 및 계획 등)에서 쿼리에 소요된 시간을 알 수 있습니다.

Note

단일 노드 클러스터에 대한 워크로드 실행 분석 차트는 표시되지 않습니다.

다음 측정치 목록에는 다양한 처리 단계에 대한 설명이 나와 있습니다.

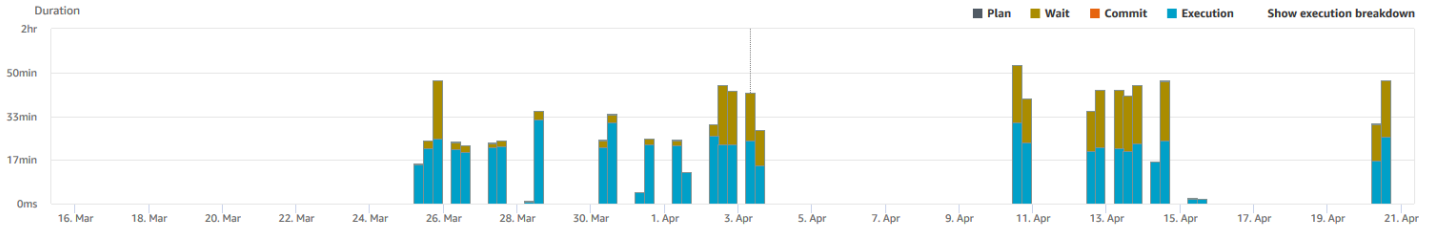
- QueryPlanning: SQL 문을 구문 분석하고 최적화하는 데 소요된 시간.
- QueryWaiting: 워크로드 관리(WLM) 대기열에서 대기하는 데 소요된 시간.
- QueryExecutingRead: 읽기 쿼리를 실행하는 데 소요된 시간.
- QueryExecutingInsert: 삽입 쿼리를 실행하는 데 소요된 시간.
- QueryExecutingDelete: 삭제 쿼리를 실행하는 데 소요된 시간.
- QueryExecutingUpdate: 업데이트 쿼리를 실행하는 데 소요된 시간.
- QueryExecutingCtas: CREATE TABLE AS 쿼리를 실행하는 데 소요된 시간.
- QueryExecutingUnload: 언로드 쿼리를 실행하는 데 소요된 시간.
- QueryExecutingCopy: 복사 쿼리를 실행하는 데 소요된 시간.

예를 들어, Amazon Redshift 콘솔의 다음 그래프는 쿼리가 전송된 시간을 계획, 대기, 읽기 및 쓰기 단계로 표시합니다. 이 그래프의 결과를 다른 측정치와 결합하여 이후 분석에 사용할 수 있습니다. 경우에 따라 그래프에 단기간의 쿼리(QueryDuration 측정치로 측정됨)가 오랫동안 대기 상태로 지속되는 것으로 표시될 수 있습니다. 이 경우 특정 대기열의 WLM 동시성 속도를 높여서 처리량을 늘릴 수 있습니다.

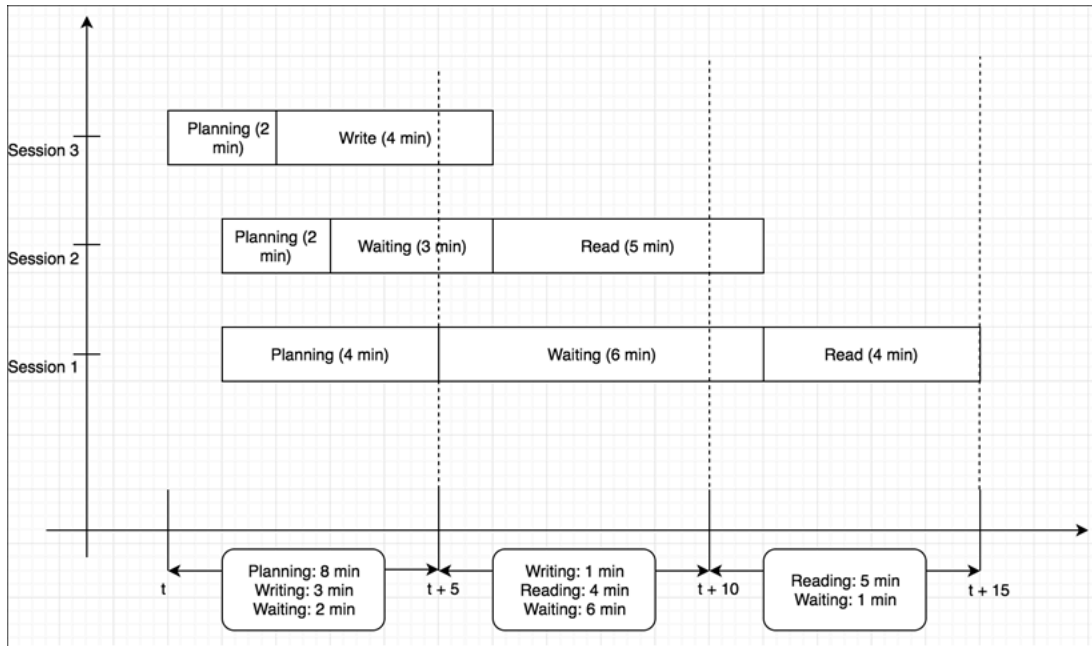
다음은 워크로드 실행 분석 차트의 예입니다. 차트에서 y축 값은 누적 막대 그래프로 표시된 지정된 시간에 각 단계의 평균 지속 시간입니다.

Workload execution breakdown

The time used in query processing stages.



다음 다이어그램은 Amazon Redshift가 동시 세션에 대한 쿼리 처리를 어떻게 집계하는지를 보여 줍니다.



클러스터 워크로드 분석 차트를 보려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 목록에서 클러스터 이름을 선택하여 세부 정보를 엽니다. 클러스터의 세부 정보가 표시됩니다. 여기에는 클러스터 성능, 쿼리 모니터링, 데이터베이스, 데이터 공유, 일정, 유지 관리 및 속성 탭이 포함될 수 있습니다.
3. 쿼리에 대한 지표를 보려면 Query monitoring(쿼리 모니터링) 탭을 선택합니다.
4. 쿼리 모니터링 섹션에서 데이터베이스 성능을 선택하고 클러스터 지표를 선택합니다.

선택한 시간 범위에 대해 다음 지표가 누적 막대 차트로 표시됩니다.

- 계획 시간

- 대기 시간
- 커밋 시간
- 실행 시간

쿼리 실행 분석

쿼리의 실행 세부 정보를 분석하여 쿼리의 수행 방식을 이해하고 최적화가 필요한 잠재적 영역을 식별할 수 있습니다. 쿼리를 분석하면 관련 단계, 각 단계에 소요된 시간, 처리된 데이터 양 등 쿼리 계획에 대한 인사이트를 얻을 수 있습니다. 일반적인 사용 사례로는 실행 속도가 느린 쿼리 문제 해결, 데이터 배포 전략 최적화, 쿼리 재작성 또는 인덱싱 기회 파악 등이 있습니다.

쿼리를 분석하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 쿼리를 선택한 다음 쿼리 및 로드(Queries and loads)를 선택하여 계정에 대한 쿼리 목록을 표시합니다. 쿼리를 찾으려면 이 페이지의 설정을 변경해야 할 수도 있습니다.
3. 목록에서 쿼리 식별자를 선택하여 쿼리 세부 정보를 표시합니다.

쿼리 세부 정보 페이지에는 쿼리 세부 정보 및 Query plan(쿼리 계획) 탭이 쿼리 지표와 함께 포함되어 있습니다.

Note

쿼리 런타임 그래프에서 쿼리를 드릴다운할 때 클러스터 세부 정보 페이지의 쿼리 기록 탭에서 쿼리 세부 정보 페이지로 이동할 수도 있습니다.

쿼리 세부 정보 페이지는 다음 섹션을 포함합니다.

- 다음 스크린샷과 같이 다시 작성된 쿼리의 목록.

Rewritten queries(5)
This query was rewritten by Amazon Redshift for optimization

	Start time	Query	Status	Duration	Executed on	Query type
☐	Apr 15th, 2020 01:44:44 PM 6 days ago	122927,122928,122929...	Completed	5 min		Parent query
☒	Apr 15th, 2020 01:44:44 PM 6 days ago	122927	Completed	4 sec	Main	Rewritten query
☐	Apr 15th, 2020 01:44:48 PM 6 days ago	122928	Completed	22 ms	Main	Rewritten query
☐	Apr 15th, 2020 01:44:48 PM 6 days ago	122929	Completed	19 ms	Main	Rewritten query
☐	Apr 15th, 2020 01:44:48 PM 6 days ago	122931	Completed	5 min	Main	Rewritten query

- 다음 스크린샷과 같이 쿼리 세부 정보 섹션.

Query details

Query ID 122927	Cluster dnd-sudhare-qa	User	Type Rewritten query	Status Completed
From April 15, 2020 at 01:44:44 PM To April 15, 2020 at 01:44:48 PM				Total runtime 4sec

- 실행된 SQL 및 실행에 대한 [실행 세부 정보(Execution details)]가 포함된 [쿼리 세부 정보(Query details)] 탭.
- 쿼리 계획 단계 및 쿼리 계획에 대한 기타 정보가 포함된 쿼리 계획 탭. 이 테이블에는 쿼리가 실행될 때의 클러스터에 대한 그래프도 포함되어 있습니다.
- 클러스터 상태

Cluster health status

Cluster health during the workload.

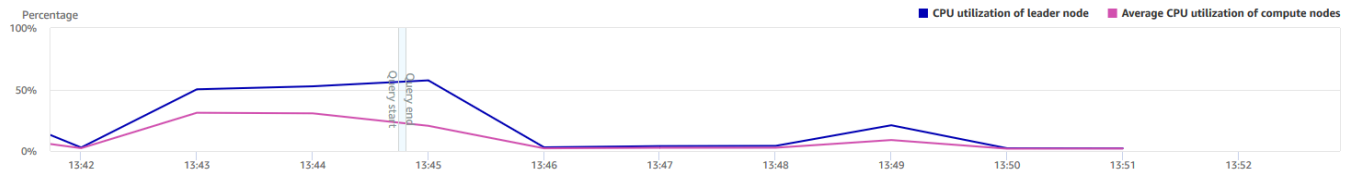
Healthy Unhealthy



- CPU 사용률

CPU utilization

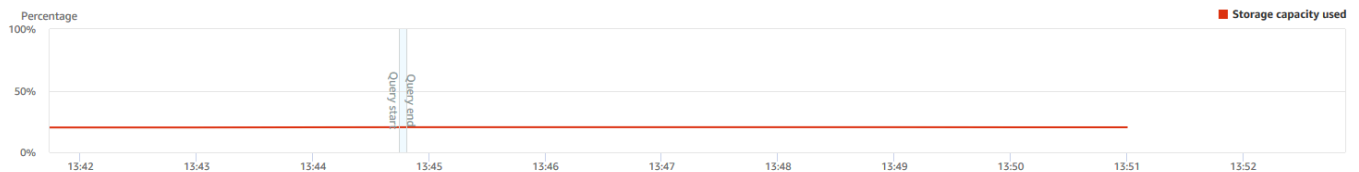
The CPU utilization of the cluster by leader node and average of compute nodes.



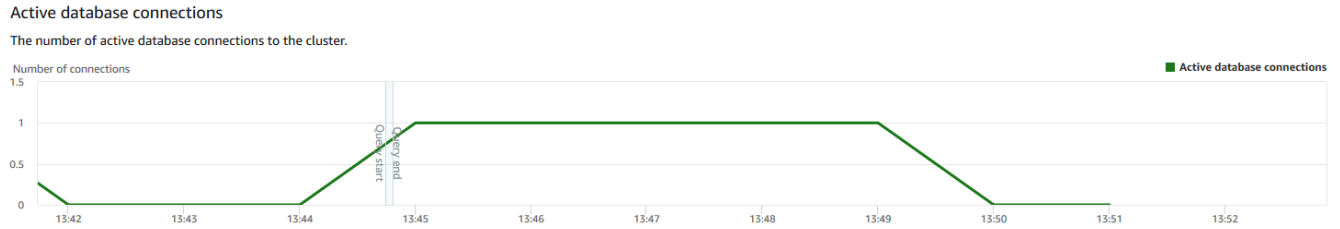
- 사용된 스토리지 용량

Storage capacity used

The percent of the storage capacity used.



• 활성 데이터베이스 연결



경보 생성

Amazon Redshift 콘솔에서 생성하는 경보는 CloudWatch 경보입니다. 이 경보는 클러스터 또는 서버리스 인스턴스에 대한 사전 결정이 가능하다는 점에서 유용합니다. [Amazon Redshift의 성능 데이터](#)에 나열된 모든 지표에 대해 경보를 1개 이상 설정할 수 있습니다. 예를 들어 클러스터 노드의 높은 CPUUtilization에 대해 경보를 설정하면 노드 사용량의 초과 시점을 알 수 있습니다. 높은 DataStorage에 대한 경보는 서버리스 네임스페이스가 데이터에 사용하는 스토리지 공간을 추적합니다.

[작업(Actions)]에서 경보를 수정하거나 삭제할 수 있습니다. Slack 또는 Amazon Chime 웹hook URL을 지정하여 CloudWatch에서 Slack 또는 Amazon Chime으로 알림을 보내는 차임 또는 슬랙 알림을 생성할 수도 있습니다.

이 섹션에서는 Amazon Redshift 콘솔에서 경보를 생성하는 방법에 대해 알아봅니다. CloudWatch 콘솔을 사용하여 경보를 생성하거나 AWS CLI 또는 AWS SDK 등의 다른 방식으로 지표를 사용할 수도 있습니다.

Amazon Redshift 콘솔로 CloudWatch 경보를 생성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.

Amazon Redshift Serverless를 사용하는 경우 대시보드 오른쪽 상단에서 Go to Serverless(Serverless로 이동)를 선택합니다.

2. 탐색 메뉴에서 경보(Alarms)를 선택한 후 경보 생성(Create alarm)을 선택합니다.
3. [경보 생성(Create alarm)] 페이지에서 CloudWatch 경보를 생성할 속성을 입력합니다.
4. 경보 생성을 선택하세요.

실행 쿼리 종료

쿼리 페이지에서는 현재 진행 중인 쿼리를 종료할 수도 있습니다.

Note

Amazon Redshift 콘솔에서 쿼리 및 로드를 종료하려면 특정 권한이 필요합니다. 사용자가 쿼리 및 로드를 종료할 수 있도록 하려면 `redshift:CancelQuerySession` 작업을 IAM(AWS Identity and Access Management) 정책에 추가해야 합니다. 이 요구 사항은 Amazon Redshift 읽기 전용 AWS 관리형 정책을 선택하거나 IAM에서 사용자 정의 정책을 생성하는지 여부에 상관없이 적용됩니다. Amazon Redshift 전체 액세스 권한 정책이 적용되는 사용자는 쿼리 및 로드를 종료하는 데 필요한 권한을 이미 보유하고 있습니다. Amazon Redshift의 IAM 정책 작업에 대한 자세한 내용은 [리소스 액세스 관리](#) 섹션을 참조하세요.

실행 쿼리를 종료하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 쿼리를 선택한 다음 쿼리 및 로드(Queries and loads)를 선택하여 계정에 대한 쿼리 목록을 표시합니다.
3. 목록에서 종료할 실행 쿼리를 선택한 다음 쿼리 종료를 선택합니다.

CloudWatch 콘솔의 성능 지표

CloudWatch 콘솔에서 Amazon Redshift 지표를 사용하여 작업할 때는 다음과 같은 몇 가지 사항을 알고 있어야 합니다.

- 쿼리 및 로드 성능 데이터는 Amazon Redshift 콘솔에서만 제공됩니다.
- 일부 CloudWatch 지표의 단위는 Amazon Redshift 콘솔에서 사용하는 단위와 다릅니다. 예를 들어 `WriteThroughput`이 GB/s(CloudWatch에서는 바이트/초로 표시됨)로 표시되어 일반적인 노드의 스토리지 공간에 더욱 적합합니다.

CloudWatch 콘솔, 명령줄 도구 또는 Amazon SDK에서 Amazon Redshift 지표를 사용하여 작업할 때는 다음과 같은 개념을 알고 있어야 합니다.

1. 첫째, 사용할 지표 차원을 지정합니다. 차원은 지표를 고유하게 식별하는 데 도움이 되는 이름-값 페어입니다. Amazon Redshift의 차원은 ClusterIdentifier와 NodeID입니다. CloudWatch 콘솔에서는 클러스터 및 노드 고유의 차원을 쉽게 선택할 수 있도록 Redshift Cluster 및 Redshift Node 뷰가 제공됩니다. 차원에 대한 자세한 내용은 CloudWatch Developer Guide의 [Dimensions](#)를 참조하세요.
2. 둘째, 지표 이름(ReadIOPS 등)을 지정합니다.

아래 표는 제공되는 Amazon Redshift 지표 차원의 유형을 요약한 것입니다. 지표에 따라 1분 또는 5분 간격으로 데이터를 무료로 사용할 수 있습니다. 자세한 내용은 [Amazon Redshift 지표](#) 섹션을 참조하세요.

CloudWatch 네임 스페이스	차원	설명
AWS/Redshift	NodeID	클러스터 노드 고유의 요청 데이터를 필터링합니다. NodeID는 "Leader", "Shared" 또는 "Compute-N"이 되고, 여기에서 N은 클러스터의 노드 번호를 가리키는 0, 1 등으로 매겨집니다. "Shared"는 클러스터의 노드 수가 1개, 즉 리더 노드와 컴퓨팅 노드가 결합되었다는 것을 의미합니다.
AWS/Redshift	ClusterIdentifier	클러스터 고유의 요청 데이터를 필터링합니다. 클러스터 고유의 지표로는 HealthStatus, MaintenanceMode 및 DatabaseConnections가 있습니다. 노드 지표이기도 한 이 차원의 일반 지표(ReadIOPS 등)는 노드 지표 데이터를 모두 종합한 것입니다. 이러한 지표들은 리더 노드와 컴퓨팅 노드의 속성을 모두 수집하기 때문에 해석할 때 주의하십시오.

게이트웨이 및 볼륨 지표 작업은 기타 서비스 지표 작업과 유사합니다. 대부분 공통 태스크는 CloudWatch 설명서에 간략히 설명되어 있습니다. 몇 가지 나열하면 다음과 같습니다.

- [사용 가능한 지표 보기](#)
- [지표에 대한 통계 얻기](#)
- [CloudWatch 경보 생성](#)

쿼리 프로파일러

이 문서에서는 쿼리의 구성 요소 및 성능을 분석하기 위한 그래픽 도구인 쿼리 프로파일러에 대해 설명합니다.

쿼리 프로파일러는 Amazon Redshift 콘솔을 통해 볼 수 있는 쿼리 모니터링 및 문제 해결 기능입니다. 쿼리 성능을 분석하는 데 유용합니다. 주요 목적은 쿼리에 대한 시각적인 그래픽 실행 순서, 실행 계획 및 통계를 표시하고 이를 더 쉽게 이해하고 해결할 수 있도록 하는 것입니다. 쿼리 프로파일러는 다음 유형의 쿼리 구성 요소를 분석하는 데 도움이 됩니다.

- 하위 쿼리 - 하위 쿼리는 쿼리 작업의 일부입니다. Amazon Redshift는 쿼리를 하나의 대규모 쿼리로 처리하는 것보다 더 효율적인 경우 쿼리를 여러 하위 쿼리로 분할할 수 있습니다. 프로파일러에서 각 하위 쿼리의 속성을 볼 수 있습니다. 하위 쿼리는 스트림과 추가 하위 구성 요소로 구성됩니다.

쿼리 프로파일러가 표시하는 하위 쿼리 유형에는 일반적으로 다음이 포함됩니다.

- 임시 테이블 쿼리: 이 하위 쿼리의 텍스트는 CREATE TEMP TABLE 명령으로 시작합니다. 이 하위 쿼리는 처리할 다른 하위 쿼리에 대한 임시 테이블을 생성합니다.
- 통계 쿼리: 쿼리 프로파일러는 이 하위 쿼리를 식별하는 데 도움이 되도록 이 하위 쿼리의 시작 부분에 다음 주석을 추가합니다.

```
-- collect statistics of child query queryID
```

이 하위 쿼리는 Amazon Redshift 쿼리 엔진이 성능을 최적화하는 데 사용하는 정보를 수집합니다.

Note

쿼리 프로파일러는 사용자가 Amazon Redshift가 실행하는 최종 하위 쿼리로 제공하는 쿼리를 보여줍니다.

- 스트림 - 스트림은 사용 가능한 컴퓨팅 노드 슬라이스 위에 분할된 세그먼트 모음입니다. 각 하위 쿼리는 하나 이상의 세그먼트로 구성됩니다. 쿼리 프로파일러에서 실행 시간과 같은 각 스트림의 속성을 볼 수 있습니다. 스트림 목록을 살펴보면 성능 병목 현상을 빠르게 찾을 수 있습니다.
- 세그먼트 - 세그먼트는 단일 프로세스가 실행할 수 있는 여러 단계의 조합입니다. 또한 세그먼트는 컴퓨팅 노드 슬라이스에서 실행할 수 있는 가장 작은 컴파일 단위입니다. 여기서 조각이란 Amazon Redshift의 병렬 처리 유닛을 말합니다. 스트림을 구성하는 세그먼트들은 병렬로 실행됩니다. 쿼리 프로파일러는 세그먼트를 그래픽으로 표시하지 않지만 해당 단계의 세부 정보 창에서 단계에 대한 세그먼트 정보에 액세스할 수 있습니다.

- 단계 - 각 세그먼트는 단계 모음으로 구성됩니다. 단계는 쿼리의 작업 부분입니다. 예를 들어 단계에는 해시조인 또는 테이블에서 레코드를 읽는 스캔이 포함될 수 있습니다.

스트림, 세그먼트 및 단계에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [쿼리 계획 및 실행 워크플로](#)를 참조하세요.

쿼리 프로파일러는 SYS_QUERY_HISTORY, SYS_QUERY_DETAIL, SYS_QUERY_EXPLAIN 및 SYS_CHILD_QUERY_TEXT 뷰에서 반환된 정보를 표시합니다. 이러한 뷰에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [SYS_QUERY_HISTORY](#), [SYS_QUERY_DETAIL](#), [SYS_QUERY_EXPLAIN](#), [SYS_CHILD_QUERY_TEXT](#)를 참조하세요.

쿼리 프로파일러는 데이터베이스에서 최근에 실행된 쿼리에 대한 쿼리 정보만 표시합니다. 데이터베이스에 대해 실행되지 않고 미리 채워진 캐시 데이터를 사용하여 완료하는 쿼리는 이전에 정보를 사용할 수 없는 경우 쿼리 프로파일이 없습니다. 이는 Amazon Redshift가 쿼리 계획을 생성하지 않기 때문입니다.

쿼리 프로파일러를 사용하기 위한 사전 조건

SYS 모니터링 뷰는 사용 편의성과 복잡성 감소를 위해 설계되어 효과적인 모니터링 및 문제 해결을 위한 완전한 지표를 제공합니다. SYS 모니터링 뷰는 클러스터의 크기 또는 활동과 관계없이 지난 7일 동안의 쿼리 기록도 보장합니다. 사용자는 자신이 실행한 쿼리만 볼 수 있으며 슈퍼 사용자는 모든 사용자의 쿼리를 볼 수 있습니다.

IAM 사용자 계정 또는 역할에는 콘솔의 쿼리 및 데이터베이스 모니터링 섹션에 액세스할 수 있는 권한이 필요합니다. 이 섹션에서는 사용자 계정 또는 역할에 권한을 추가하는 방법을 설명합니다.

다음 정책을 사용하여 IAM 사용자 계정 또는 역할에 최소 권한을 추가하세요.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift:DescribeClusters",
        "redshift-serverless:ListNamespaces",
        "redshift-serverless:ListWorkgroups",
        "redshift-data:ExecuteStatement",

```

```

        "redshift-data:DescribeStatement",
        "redshift-data:GetStatementResult"
    ],
    "Resource": [
        "arn:aws:redshift-serverless:us-east-1:111122223333:*",
        "arn:aws:redshift:us-east-1:111122223333:*"
    ]
}
]
}

```

주제

- [역할에 대한 쿼리 모니터링 권한 부여](#)
- [사용자에 대한 쿼리 모니터링 권한 부여](#)
- [IAM 자격 증명을 사용한 임시 자격 증명](#)

역할에 대한 쿼리 모니터링 권한 부여

sys:monitor 권한이 있는 역할의 사용자는 모든 쿼리를 볼 수 있습니다. sys:operator 권한이 있는 역할의 사용자는 쿼리를 취소하고 쿼리 기록을 분석하고 vacuum 작업을 수행할 수 있습니다.

역할에 쿼리 모니터링 권한을 부여하려면

1. 시스템 모니터링 액세스를 제공하려면 다음 명령을 사용하세요. 여기서 *role-name*은 액세스를 제공하려는 역할의 이름입니다.

```
grant role sys:monitor to "IAMR:role-name";
```

2. (선택 사항) 시스템 운영자 액세스를 제공하려면 다음 명령을 사용하세요. 여기서 *role-name*은 액세스를 제공하려는 역할의 이름입니다.

```
grant role sys:operator to "IAMR:role-name";
```

사용자에 대한 쿼리 모니터링 권한 부여

sys:monitor 권한이 있는 사용자는 모든 쿼리를 볼 수 있습니다. sys:operator 권한이 있는 사용자는 쿼리를 취소하고 쿼리 기록을 분석하고 vacuum 작업을 수행할 수 있습니다.

사용자에게 쿼리 모니터링 권한을 부여하려면

1. 시스템 모니터링 액세스를 제공하려면 다음 명령을 사용하세요. 여기서 *user-name*은 액세스를 제공하려는 사용자의 이름입니다.

```
grant role sys:monitor to "IAMR:user-name";
```

2. (선택 사항) 시스템 운영자 액세스를 제공하려면 다음 명령을 사용하세요. 여기서 *-name*은 액세스를 제공하려는 사용자의 이름입니다.

```
grant role sys:operator to "IAMR:user-name";
```

IAM 자격 증명을 사용한 임시 자격 증명

이 옵션은 클러스터에 연결할 때만 사용할 수 있습니다. 이 방법을 사용하면 쿼리 프로파일러가 사용자 이름을 IAM ID에 매핑하고 IAM ID로 데이터베이스에 연결하기 위한 임시 암호를 생성합니다. 이 메서드를 사용하여 연결하는 사용자에게는 `redshift:GetClusterCredentialsWithIAM`에 대한 IAM 권한이 허용되어야 합니다. 사용자가 이 메서드를 사용하지 못하도록 하려면 IAM 사용자 또는 역할을 수정하여 이 권한을 거부합니다.

Amazon Redshift 콘솔에서 쿼리 프로파일러에 액세스하여 쿼리 분석

Amazon Redshift Serverless 또는 프로비저닝된 Amazon Redshift에 대한 쿼리 프로파일러에 액세스할 수 있습니다. 세부 정보는 다음 섹션을 참조하세요.

주제

- [Amazon Redshift Serverless용 Amazon Redshift 콘솔에서 쿼리 프로파일러 액세스](#)
- [Amazon Redshift 프로비저닝용 Amazon Redshift 콘솔에서 쿼리 프로파일러 액세스](#)

Amazon Redshift Serverless용 Amazon Redshift 콘솔에서 쿼리 프로파일러 액세스

Amazon Redshift Serverless용 쿼리 프로파일러에 액세스하려면 다음을 수행합니다.

- Amazon Redshift Serverless 콘솔을 엽니다.
- 탐색 창의 모니터링에서 쿼리 및 데이터베이스 모니터링을 선택합니다.
- 작업 그룹을 선택합니다.
- 쿼리 및 데이터베이스 모니터링을 선택합니다.

- 쿼리를 선택합니다.
- 쿼리 세부 정보 페이지에서 쿼리 계획 탭을 선택합니다.

쿼리 계획을 사용할 수 있는 경우 하위 쿼리 목록이 표시됩니다. 쿼리 프로파일러에서 조회할 쿼리를 선택합니다.

Amazon Redshift 프로비저닝용 Amazon Redshift 콘솔에서 쿼리 프로파일러 액세스

Amazon Redshift 프로비저닝용 쿼리 프로파일러에 액세스하려면 다음을 수행합니다.

- Amazon Redshift 프로비저닝 클러스터 대시보드를 엽니다.
- 클러스터를 선택합니다.
- Query monitoring(쿼리 모니터링)을 선택합니다.
- 데이터베이스에 연결
- 쿼리 및 데이터베이스 모니터링을 선택합니다.
- 쿼리를 선택합니다.

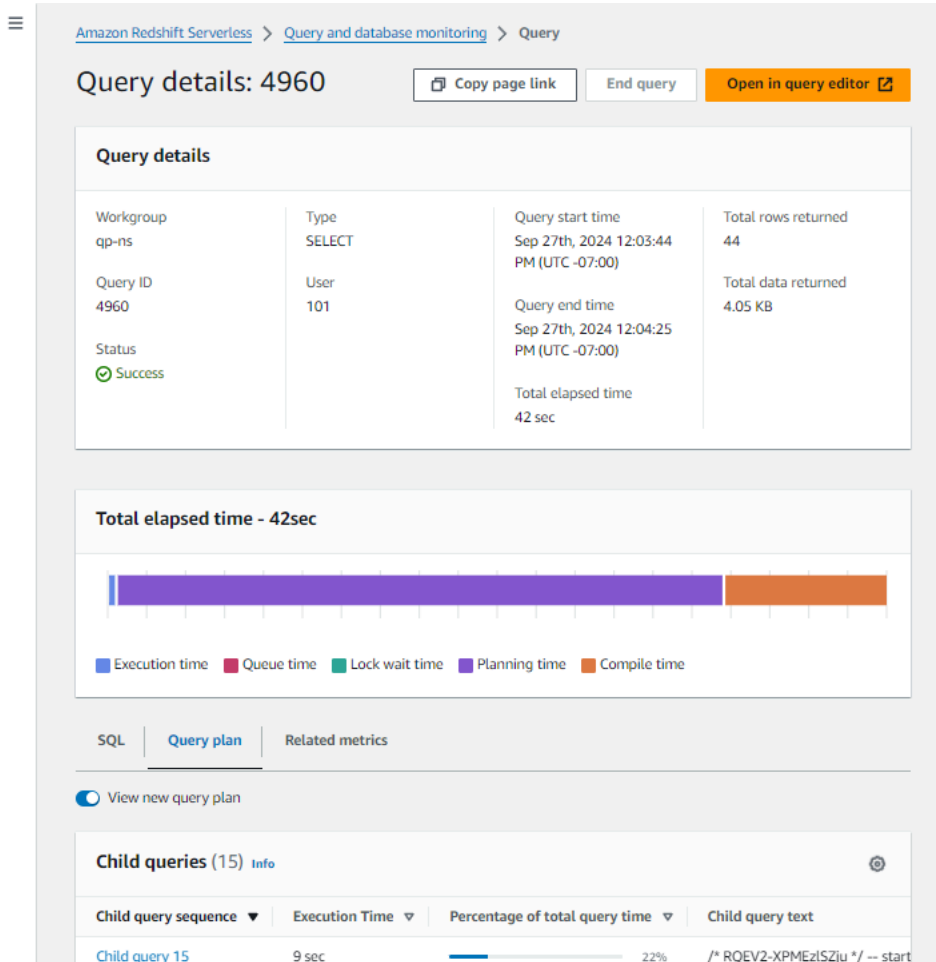
쿼리 계획을 사용할 수 있는 경우 하위 쿼리 목록이 표시됩니다. 쿼리 프로파일러에서 조회할 쿼리를 선택합니다.

쿼리 프로파일러 사용자 인터페이스

쿼리 프로파일러는 다음 페이지를 사용하여 쿼리에 대한 정보를 표시합니다.

- [쿼리 세부 정보 페이지](#) -이 페이지에는 쿼리에 대한 통계 및 하위 쿼리가 표시됩니다.
- [하위 쿼리 페이지](#) -이 페이지에는 하위 쿼리에 대한 실행 계획의 통계, 스트림 및 시각적 표현이 표시됩니다. 쿼리 및 데이터베이스 모니터링 페이지의 하위 쿼리 목록에서 하위 쿼리를 선택하면 콘솔에 이 페이지가 표시됩니다.

쿼리 세부 정보 페이지



쿼리 세부 정보 페이지에는 다음과 같은 구성 요소가 있습니다.

- 상단 창 - 페이지 상단의 창에 상태 및 유형과 같은 쿼리에 대한 세부 정보가 표시됩니다. 상단 창에 표시되는 정보의 소스에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [SYS_QUERY_HISTORY](#)를 참조하세요.
- SQL 탭 - 하단 패널의 이 탭에는 원래 사용자 쿼리에 대한 SQL 텍스트가 표시됩니다.
- 쿼리 계획 탭 - 하단 패널의 이 탭에는 Amazon Redshift가 사용자 쿼리에 대한 데이터 및 통계를 준비하는 데 사용하는 하위 쿼리 목록이 표시됩니다. 기본적으로 하위 쿼리 목록에는 각 하위 쿼리에 대한 정보와 집계된 통계가 표시됩니다. 이 페이지에 표시되는 정보의 소스에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [SYS_QUERY_DETAIL](#)을 참조하세요.

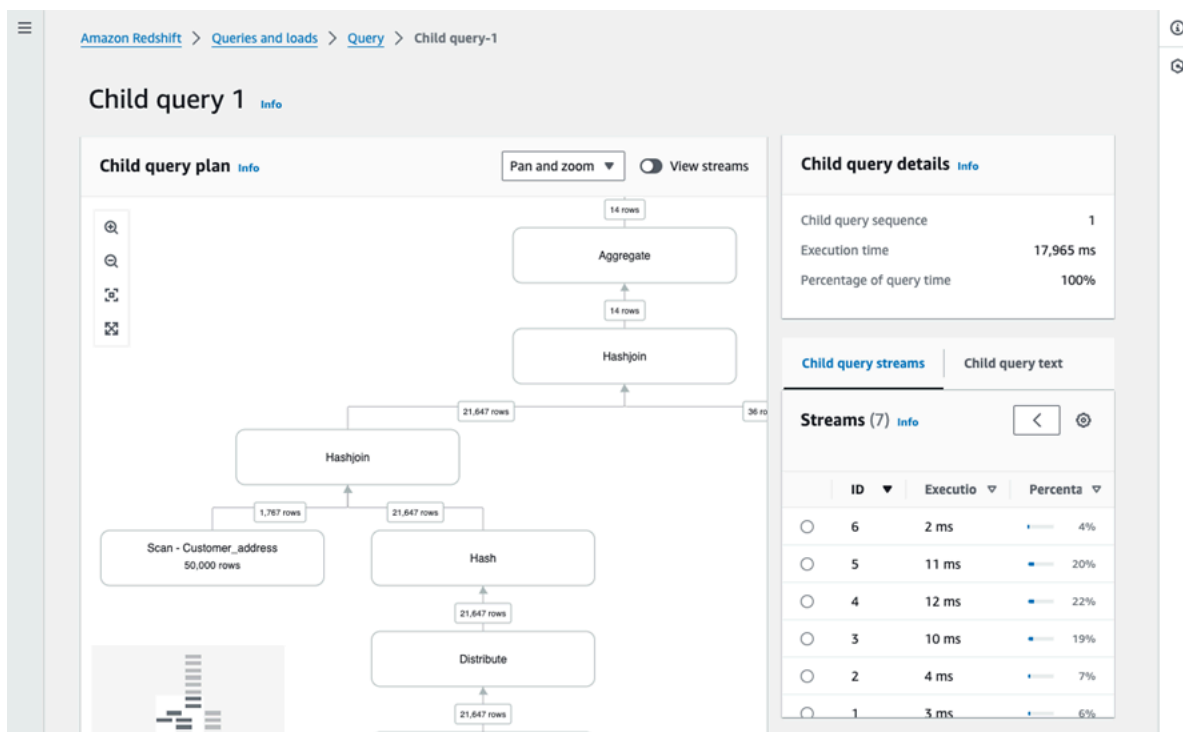
기본 설정 메뉴를 사용하여 하위 쿼리 목록에서 열을 추가하거나 제거할 수 있습니다.

- 관련 지표 - 하단 패널의 이 탭에는 쿼리에 대한 다음 CloudWatch 지표가 표시됩니다.

- 사용된 RPU 용량(서버리스 작업 그룹의 경우): 쿼리에서 사용되는 컴퓨팅 용량으로, Redshift 처리 장치(RPU)에서 측정됩니다. 자세한 내용은 [Amazon Redshift Serverless의 컴퓨팅 용량](#) 섹션을 참조하세요.
- 클러스터 상태, CPU 사용률, 사용된 스토리지 용량(프로비저닝된 클러스터의 경우): 사용된 쿼리의 상태 및 시스템 리소스입니다.
- 활성 데이터베이스 연결: 쿼리의 DatabaseConnections 지표입니다.

CloudWatch 지표에 대한 자세한 내용은 [Amazon Redshift의 성능 데이터](#) 섹션을 참조하세요.

하위 쿼리 페이지



하위 쿼리 페이지에는 다음과 같은 구성 요소가 있습니다.

- 하위 쿼리 폴다운 - 이 컨트롤은 각 하위 쿼리의 시퀀스 이름과 실행 시간을 표시합니다. 이 컨트롤에서 다른 하위 쿼리를 선택하여 탐색할 수 있습니다.
- 측면 패널 - 이 패널에는 하위 쿼리 스트림과 하위 쿼리 텍스트를 표시하는 탭이 포함되어 있습니다.
- 하위 쿼리 스트림 탭 - 상단 패널의 이 탭에는 다음이 표시됩니다.
 - 스트림 - 이 창에는 하위 쿼리의 스트림 목록이 표시됩니다. 이 창에는 Amazon Redshift가 쿼리를 최적화하는 데 사용하는 스트림에 대한 정보와 집계된 데이터가 표시됩니다. 이 창의 세부 정보에

대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [SYS_QUERY_DETAIL](#)을 참조하세요.

기본 설정 메뉴를 사용하여 스트림 창에서 열을 추가하거나 제거할 수 있습니다. 스트림 창의 기어 아이콘을 사용하여 기본 설정 메뉴에 액세스합니다.

- 하위 쿼리 창 - 하위 쿼리의 단계를 그래픽으로 표현합니다. 하위 쿼리 창에 대한 자세한 내용은 다음 [하위 쿼리 창](#) 섹션을 참조하세요.
- 하위 쿼리 텍스트 탭 - 상단 패널의 이 탭은 하위 쿼리에 대한 SQL 코드를 표시합니다.
- 하위 쿼리 세부 정보 창 - 오른쪽 패널의 이 창에는 하위 쿼리에 대한 세부 정보가 표시됩니다. 이 창의 세부 정보에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [SYS_QUERY_DETAIL](#)을 참조하세요.
- 스트림 세부 정보 창 - 스트림 창에서 스트림을 선택하면 스트림 세부 정보 창에 스트림에 대한 정보가 표시됩니다. 이 패널의 세부 정보에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [SYS_QUERY_DETAIL](#)을 참조하세요.
- 단계 세부 정보 창 - 스트림 창에서 단계를 선택하거나 하위 쿼리 계획의 그래프를 선택하면 단계 세부 정보 창에 단계에 대한 정보가 표시됩니다. 이 창의 세부 정보에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [SYS_QUERY_DETAIL](#)을 참조하세요.

하위 쿼리 창

쿼리 프로파일러는 하위 쿼리 창의 하위 쿼리를 선택한 하위 쿼리의 단계를 그래픽으로 표현하여 표시합니다.

하위 쿼리 창에는 실행 순서와 단계 간의 관계가 표시됩니다. 예를 들어 한 단계가 다른 두 단계의 출력을 조인하는 경우 하위 쿼리 창에는 두 노드가 연결된 트리 노드로 단계가 표시됩니다.



하위 쿼리 창에는 기본적으로 단계가 포함된 스트림이 표시되지 않습니다. Amazon Redshift가 하위 쿼리의 단계를 논리적으로 분할하는 데 사용하는 스트림을 표시하려면 스트림 보기를 선택합니다. 스트림 보기를 선택하면 하위 쿼리 창에 쿼리 스트림에 포함된 단계가 표시됩니다.

하위 쿼리 창에는 세그먼트 정보가 표시되지 않습니다. 단계의 세그먼트를 보려면 단계를 선택합니다. 그러면 단계 세부 정보 창에 단계의 세그먼트가 표시됩니다.

하위 쿼리 창에서 탐색

하위 쿼리 창에서 단계를 선택하여 하위 쿼리에 대한 자세한 정보를 볼 수 있습니다. 또한 작업 영역을 이동 및 확대/축소하여 쿼리 계획의 단계를 더 잘 시각화할 수 있습니다.

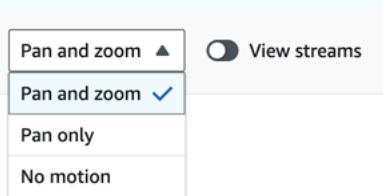
다음 방법을 사용하여 하위 쿼리 계획에서 노드, 이동 및 확대/축소를 선택할 수 있습니다.

- 마우스 사용 - 노드를 선택하고 이동하려는 작업 영역을 클릭한 후 끌어서 **Ctrl**(Windows) 또는 **Cmd**(Mac)를 누른 상태에서 마우스 휠로 확대/축소할 수 있습니다. 노드를 선택하면 작업 영역이 해당 노드를 강조 표시하도록 확대/축소하고 이동합니다. 작업 영역에서 스트림을 선택하면 해당 스트림이 스트림 목록에 강조 표시됩니다. 작업 영역에서 단계를 선택하면 단계 세부 정보 창에 해당 단계에 대한 정보가 표시됩니다.
- 작업 영역 왼쪽 상단에 있는 확대/축소 및 맞춤 컨트롤 사용 - 이러한 컨트롤을 사용하면 전체 작업 영역에 맞게 확대/축소하고 전체 화면 모드로 전환할 수 있습니다. 전체 쿼리 계획에 맞게 확대/축소하면 작업 영역에서는 쿼리 계획을 가로 및 세로로 중앙에 맞춥니다.
- 작업 영역 오른쪽 하단의 미니맵 사용 - 작업 영역 왼쪽 하단 모서리의 미니맵 컨트롤을 사용하여 작업 영역을 이동하거나 확대/축소할 수 있습니다.
- 스트림 창에서 스트림 선택 - 스트림 창에서 스트림을 선택하면 작업 영역이 이동 및 확대/축소되어 선택한 스트림이 표시되고 스트림 세부 정보 창에 스트림에 대한 정보가 표시됩니다.
- 스트림 세부 정보 창에서 단계 선택 - 스트림 세부 정보 창에서 단계를 선택하면 작업 영역이 이동 및 확대/축소되어 선택한 단계가 표시되고 단계 세부 정보 창에 단계에 대한 정보가 표시됩니다.

Note

작업 영역 또는 다른 창에서 단계를 선택하면 작업 영역은 선택한 단계가 가장 잘 보이도록 확대/축소 및 이동을 시도합니다.

작업 영역 또는 다른 창에서 스트림 또는 단계를 선택하면 작업 영역의 오른쪽 상단 컨트롤에서 이동 및 확대/축소를 선택한 경우에만 작업 영역이 해당 스트림 또는 단계로 확대/축소 및 이동됩니다. 풀다운에서 적절한 설정을 선택하여 이 동작을 이동 및 확대/축소, 이동만 또는 이동 없음으로 제한할 수 있습니다.



쿼리 프로파일러를 사용하여 쿼리 문제 해결

쿼리 문제를 해결하는 경우 하위 쿼리를 선택하여 총 쿼리 시간 중 가장 높은 비율을 사용하는 스트림을 결정할 수 있습니다. 이는 쿼리의 어떤 부분을 추가로 분석해야 하는지 빠르게 결정할 수 있는 방법입니다.

어떤 하위 쿼리가 가장 오래 걸리는지 확인한 후 해당 단계를 보고 어떤 조인 또는 스캔이 성능 저하를 유발할 수 있는지 확인하세요.

쿼리 및 데이터베이스 모니터링

이 문서에서는 Amazon Redshift 프로비저닝된 클러스터 또는 서버리스 작업 그룹과 이에 대해 실행되는 쿼리의 성능을 분석하는 AWS Management Console 기능인 쿼리 및 데이터베이스 모니터링 페이지를 설명합니다.

쿼리 및 데이터베이스 모니터링 페이지를 사용하여 다음 시나리오를 조사할 수 있습니다.

- 지정된 기간 동안 데이터 웨어하우스 지표 모니터링
- 쿼리가 데이터 웨어하우스의 전체 성능에 미치는 영향
- 쿼리 실행 시간을 잠금 대기 시간, 컴파일 시간, 실행 시간과 같은 수명 주기 이벤트로 구분하여 확인
- 지정된 기간 동안 가장 리소스 집약적인 쿼리를 실행하는 사용자
- 패치 이벤트가 쿼리 성능에 미치는 영향 모니터링

주제

- [권한](#)
- [쿼리 및 데이터베이스 모니터링 콘솔](#)

권한

콘솔에 액세스하는 데 사용하는 AWS 계정의 권한은 쿼리 및 데이터베이스 모니터링 페이지에 표시되는 쿼리에 영향을 줍니다. 기본적으로 자신의 쿼리만 볼 수 있습니다. 다른 사용자가 소유한 쿼리를 보려면 계정에 SYS:MONITOR 역할을 부여하세요. 사용자가 쿼리 및 데이터베이스 모니터링 페이지에서 실행 중인 쿼리를 종료하도록 허용하려면 사용자에게 SYS:OPERATOR 권한을 부여하세요.

Amazon Redshift Serverless 또는 프로비저닝됨에 대한 IAM 사용자 또는 역할에 sys:monitor 역할을 자동으로 할당하려면 다음 명령을 실행하세요.

```
create role monitor;
grant role sys:monitor to role monitor;
```

쿼리 모니터링에 사용되는 IAM 역할을 업데이트하려면 다음을 수행하세요.

1. 태그 탭을 선택합니다.
2. 태그 관리를 선택합니다.
3. **RedshiftDbRoles** 키와 **monitor** 값을 사용하여 태그를 추가합니다.
4. 변경 내용을 저장합니다.

사용자에게 데이터베이스 자격 증명을 추가하려면 다음 명령을 실행하세요.

```
grant role sys:monitor to <username>
```

GRANT 명령 사용에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [GRANT](#)를 참조하세요.

쿼리 모니터링을 사용하려면 IAM 사용자에게 Amazon Redshift 데이터 플레인에 액세스할 수 있는 권한이 필요합니다. IAM 사용자의 권한 정책에 다음 권한이 있는지 확인해야 합니다.

```
{
  "Sid": "DataAPIPermissions",
  "Action": [
    "redshift-data:ExecuteStatement",
    "redshift-data:CancelStatement",
    "redshift-data:GetStatementResult",
    "redshift-data:DescribeStatement",
    "redshift-data:ListDatabases"
  ],
  "Effect": "Allow",
  "Resource": "arn:aws:redshift-serverless:us-west-2:123456789012:workgroup/01234567-89ab-cdef-0123-456789abcdef"
},
```

IAM 자격 증명을 사용한 임시 자격 증명

이 옵션은 클러스터에 연결할 때만 사용할 수 있습니다. 이 방법을 사용하면 쿼리 및 데이터베이스 모니터링이 사용자 이름을 IAM ID에 매핑하고 IAM ID로 데이터베이스에

연결하기 위한 임시 암호를 생성합니다. 이 메서드를 사용하여 연결하는 사용자에게는 `redshift:GetClusterCredentialsWithIAM`에 대한 IAM 권한이 허용되어야 합니다. 사용자가 이 메서드를 사용하지 못하도록 하려면 IAM 사용자 또는 역할을 수정하여 이 권한을 거부합니다.

쿼리 및 데이터베이스 모니터링 콘솔

이 섹션에서는 쿼리 및 데이터베이스 모니터링 콘솔 페이지 사용에 대해 설명합니다.

쿼리 및 데이터베이스 모니터링 콘솔을 사용하여 데이터 웨어하우스의 성능 개요를 빠르게 확인할 수 있습니다. 시간 경과에 따른 데이터 웨어하우스의 성능을 모니터링하고 데이터 웨어하우스의 프로비저닝된 클러스터 또는 개별 쿼리의 성능을 조사하여 병목 현상과 기타 개선할 영역을 효과적으로 식별할 수 있습니다.

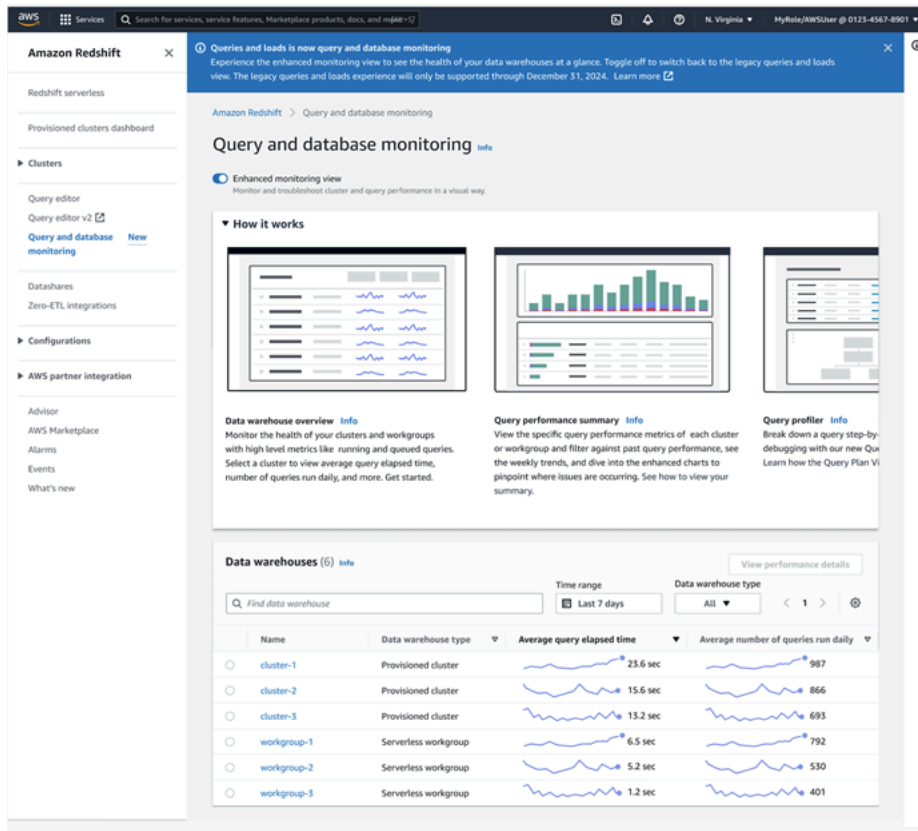
쿼리 및 데이터베이스 모니터링 페이지에는 다음과 같은 기능이 있습니다.

- 보안 강화 - 다른 사용자의 쿼리를 모니터링하려면 높은 권한이 필요합니다. 자세한 내용은 [권한](#) 섹션을 참조하세요.
- 7일 쿼리 기록 - 7일간의 쿼리 기록에 대한 액세스를 보장합니다.
- 쿼리 모니터링 - 사용자 쿼리 수준에서 프로비저닝된 클러스터 및 서버리스 작업 그룹의 쿼리를 모니터링할 수 있습니다.
- 쿼리 추세 분석 - 지정된 기준과 일치하는 유사한 쿼리들의 성능을 비교할 수 있습니다.

쿼리 및 데이터베이스 모니터링 페이지에 액세스하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift에 로그인합니다.
2. 탐색 창에서 쿼리 및 데이터베이스 모니터링을 선택합니다.

쿼리 및 데이터베이스 모니터링 콘솔 페이지는 다음과 같이 나타납니다.



쿼리 및 데이터베이스 모니터링 페이지에는 다음과 같은 구성 요소가 있습니다.

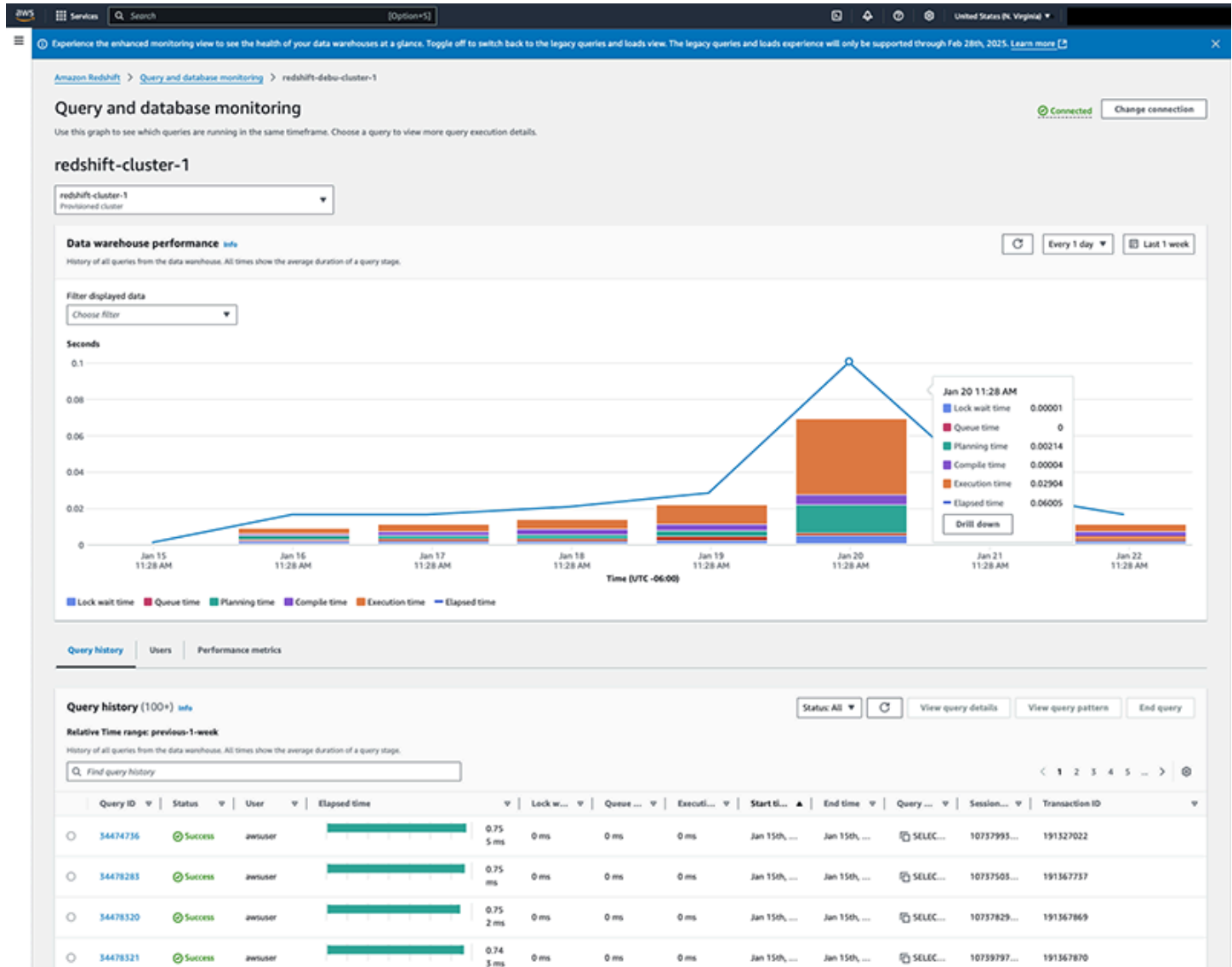
- 데이터 웨어하우스 개요 - 프로비저닝된 클러스터와 서버리스 작업 그룹의 평균 쿼리 성능을 모니터링합니다. 이 페이지에 있는 통계에서 스파이크 또는 활동량이 많은 기간을 조사하여 특정 클러스터 또는 작업 그룹의 성능 문제를 빠르게 확인할 수 있습니다.
- 쿼리 성능 요약 - 특정 프로비저닝된 클러스터 또는 서버리스 작업 그룹의 평균 쿼리 성능을 모니터링합니다. 데이터 웨어하우스 개요 목록에서 클러스터 또는 작업 그룹 중 하나를 클릭하여 쿼리 성능 요약 페이지에 액세스할 수도 있습니다.

주제

- [쿼리 성능 요약](#)
- [쿼리 세부 정보](#)
- [쿼리 패턴](#)

쿼리 성능 요약

쿼리 및 데이터베이스 모니터링 페이지에서 쿼리 성능 요약을 선택하거나 데이터 웨어하우스 개요 페이지에서 클러스터 또는 작업 그룹을 선택하면 콘솔에 개별 프로비저닝된 클러스터 또는 서버리스 작업 그룹의 성능 요약이 표시됩니다.



이 페이지에는 다음과 같은 구성 요소가 있습니다.

- 클러스터 또는 작업 그룹 드롭다운 - 분석하려는 클러스터 또는 작업 그룹을 선택합니다.
- 데이터 웨어하우스 성능 - 이 창에는 지정된 기간 내 클러스터 또는 작업 그룹의 기록이 표시되며 쿼리의 각 단계에 소요된 시간이 나타납니다. 앞의 그래픽에서 1월 20일에 계획 및 실행 시간이 늘어나는 것과 같이 특정 쿼리 단계에서 스파이크가 나타나는 경우 이 정보를 사용하여 쿼리 성능과 관련된

문제를 식별할 수 있습니다. 표시되는 기본 기간은 지난 7일이지만 분석 요구 사항에 따라 기간을 조정할 수 있습니다.

- 쿼리 기록 - 이 창에는 지정된 필터 내에서 실행된 모든 쿼리의 성능 기록이 표시됩니다. 이 정보를 사용하여 개별 쿼리의 성능 문제를 해결할 수 있습니다. 특정 쿼리를 추가로 분석하려면 이 그래프 또는 데이터 웨어하우스 성능 그래프에서 쿼리를 선택합니다.

차트의 데이터 포인트 위로 마우스를 가져가면 팝업이 나타납니다. 이 팝업에는 해당 데이터 포인트에 대한 시간 지표의 분석 정보가 표시됩니다. 데이터 포인트에 쿼리 데이터가 포함된 경우 드릴다운을 선택하여 차트의 시간 범위를 데이터 포인트의 다음으로 작은 시간 단위로 업데이트할 수 있습니다. 이러한 시간 단위는 다음과 같습니다.

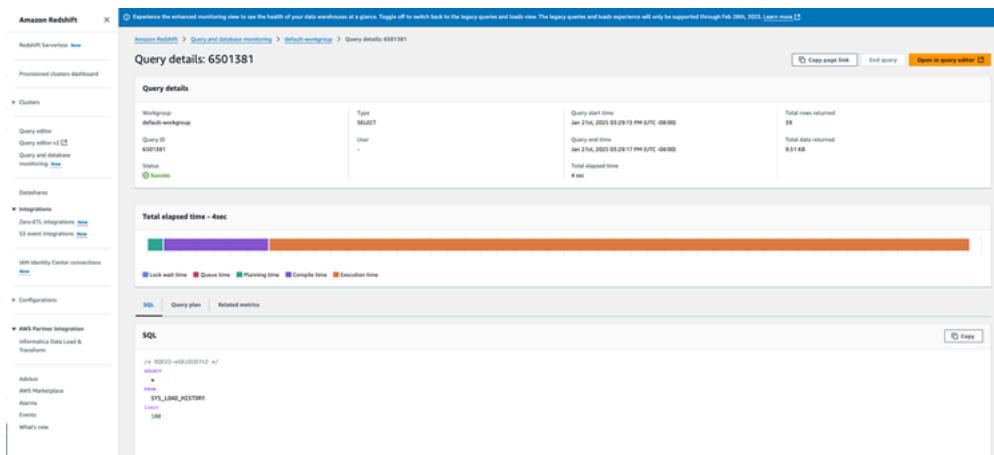
- 일
- 시간
- 15분
- 5분
- 1분

예를 들어 데이터 포인트에서 드릴다운을 선택하면 그래프 범위가 1일로 변경됩니다. 드릴다운을 다시 선택하면 그래프 범위가 1시간으로 변경됩니다.

- 쿼리 프로파일러 - 쿼리 성능을 모니터링하기 위한 그래픽 도구입니다. 자세한 내용은 [쿼리 프로파일러](#) 섹션을 참조하세요.

쿼리 세부 정보

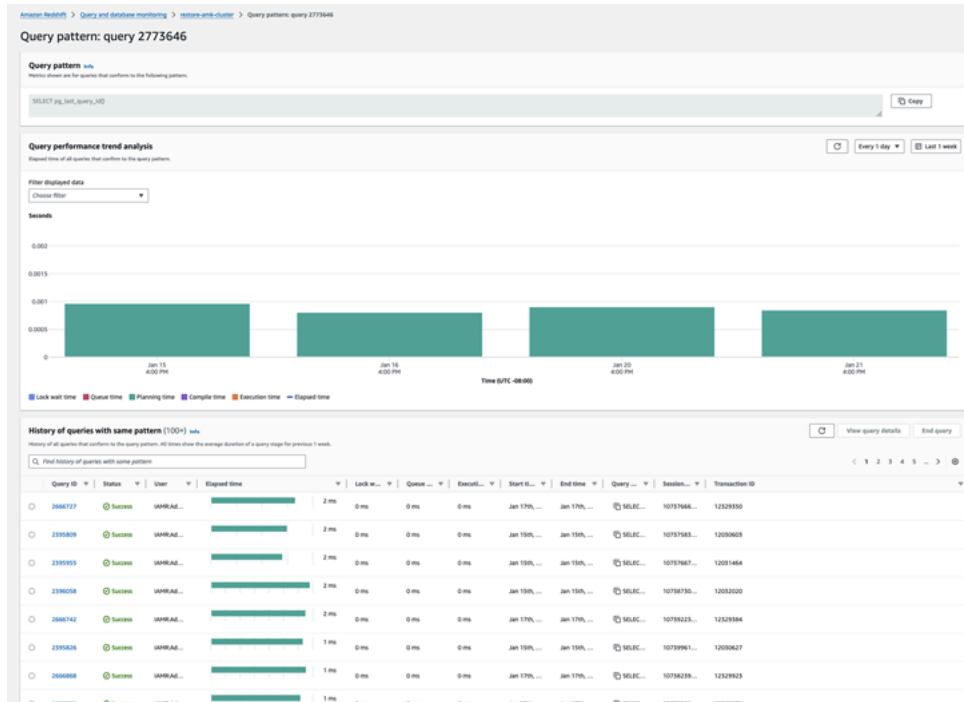
클러스터 또는 작업 그룹 세부 정보 페이지의 데이터 웨어하우스 성능 또는 쿼리 기록 창에서 쿼리를 선택하면 쿼리 세부 정보 페이지가 열립니다.



쿼리 세부 정보 페이지에 대한 자세한 내용은 [쿼리 세부 정보 페이지](#) 섹션을 참조하세요.

쿼리 패턴

쿼리 성능 요약 페이지의 쿼리 기록 창에서 쿼리 패턴 보기 버튼을 선택하여 동일한 패턴의 쿼리에 대한 기록을 볼 수 있습니다. 쿼리 패턴 페이지에는 지정한 SQL 문으로 검색된 지난 주의 모든 쿼리가 표시됩니다.



쿼리 패턴 페이지에는 다음과 같은 구성 요소가 있습니다.

- 쿼리 패턴 - 분석 중인 쿼리를 검색하는 SQL 문입니다.
- 쿼리 성능 추세 분석 - 쿼리 패턴에서 선택된 모든 쿼리의 경과 시간을 보여주는 막대 그래프입니다. 결과는 일별로 그룹화됩니다.
- 동일한 패턴의 쿼리 기록 - 쿼리 패턴에서 선택된 쿼리의 각 단계에 소요된 시간입니다.

쿼리 패턴 페이지에서는 다음과 같은 인사이트를 얻을 수 있습니다.

- 매일 특정 시간에 실행되는 쿼리에 관한 추세
- 데이터 웨어하우스에 대해 정기적으로 실행되는 쿼리의 실행 시간 급증

Sys 뷰 기반 쿼리 및 데이터베이스 모니터링

이 문서에서는 Amazon Redshift 콘솔에서 쿼리의 구성 요소 및 성능을 분석하기 위한 도구인 쿼리 및 데이터베이스 모니터링 페이지에 데이터를 제공하는 SYS 뷰를 설명합니다. 쿼리 및 데이터베이스 모니터링 페이지에 대한 자세한 내용은 [쿼리 및 데이터베이스 모니터링](#) 섹션을 참조하세요.

쿼리 및 데이터베이스 모니터링 페이지에는 SYS 뷰에서 제공하는 정보를 표시하는 기능이 있습니다. 콘솔 뷰에는 그래픽 형식으로 쿼리의 실행 계획을 보여주는 쿼리 프로파일러가 포함되어 있습니다. SYS 기반 뷰로 전환하려면 다음 단계에 따라 새로운 쿼리 및 데이터베이스 모니터링 페이지에 대한 올바른 액세스 및 권한을 부여합니다.

쿼리 및 데이터베이스 모니터링 페이지의 SYS 기반 뷰에는 다음과 같은 기능이 있습니다.

- 보안 강화 - 다른 사용자의 쿼리를 모니터링하려면 높은 권한이 필요합니다.
- 7일 쿼리 기록 - 7일간의 쿼리 기록에 대한 액세스를 보장합니다.
- 쿼리 프로파일러 - 쿼리 성능을 모니터링하기 위한 그래픽 도구입니다. 자세한 내용은 [쿼리 프로파일러](#) 섹션을 참조하세요.

기본적으로 자신의 쿼리만 볼 수 있습니다. 다른 사용자가 소유한 쿼리를 보려면 계정에 SYS:MONITOR 역할을 부여하세요. 사용자가 실행 중인 쿼리를 종료하도록 허용하려면 사용자에게 SYS:OPERATOR 권한을 부여하세요.

모든 사용자가 소유한 쿼리를 보는 권한을 데이터베이스 사용자 또는 역할에 부여하려면 다음 명령을 실행하세요.

```
grant role sys:monitor to "IAM:role-name";
grant role sys:monitor to "IAM:user-name";
```

Amazon Redshift Serverless 또는 프로비저닝됨에 대한 IAM 사용자 또는 역할에 sys:monitor 역할을 자동으로 할당하려면 다음 명령을 실행하세요.

```
create role monitor;
grant role sys:monitor to role monitor;
```

쿼리 모니터링에 사용되는 IAM 역할을 업데이트하려면 다음을 수행하세요.

1. 태그 탭을 선택합니다.

2. 태그 관리를 선택합니다.
3. **RedshiftDbRoles** 키와 **monitor** 값을 사용하여 태그를 추가합니다.
4. 변경 내용을 저장합니다.

사용자에게 데이터베이스 자격 증명을 추가하려면 다음 명령을 실행하세요.

```
grant role sys:monitor to <username>
```

권한

쿼리 모니터링을 사용하려면 IAM 사용자에게 Amazon Redshift 데이터 플레인에 액세스할 수 있는 권한이 필요합니다. IAM 사용자의 권한 정책에 다음 권한이 있는지 확인해야 합니다.

```
{
  "Sid": "DataAPIPermissions",
  "Action": [
    "redshift-data:ExecuteStatement",
    "redshift-data:CancelStatement",
    "redshift-data:GetStatementResult",
    "redshift-data:DescribeStatement",
    "redshift-data:ListDatabases"
  ],
  "Effect": "Allow",
  "Resource": "arn:aws:redshift-serverless:us-west-2:123456789012:workgroup/01234567-89ab-cdef-0123-456789abcdef"
},
```

데이터베이스에 연결

향상된 쿼리 모니터링 기능을 사용하기 전에 먼저 SYS 뷰 기반 정보에 액세스하기 위해 데이터베이스에 연결해야 합니다. 데이터베이스에 연결하려면 다음 자격 증명 중 하나를 사용합니다.

- 사용자 이름 및 암호
- IAM 역할과 연결된 임시 자격 증명
- 데이터베이스 사용자

향상된 쿼리 모니터링을 사용할 때 다음 사항에 유의하세요.

- 프로비저닝된 클러스터의 경우 향상된 쿼리 모니터링에서 SYS 뷰를 사용하기 때문에 데이터베이스에 연결해야 합니다. 이러한 뷰는 보안이 강화되어 있으며 다른 사용자가 소유한 쿼리에 대한 데이터에 액세스하기 위해 승격된 권한이 필요합니다.
- SYS 뷰 기반 쿼리 및 데이터베이스 모니터링 페이지를 사용할 때 사용자 계정에 데이터베이스 슈퍼 사용자 역할이 없는 경우 자신의 `user_id`만 표시됩니다. 사용자 이름은 슈퍼유저가 아닌 사용자에게 숨겨집니다.
- Sys 뷰 기반 쿼리 및 데이터베이스 모니터링 페이지 경험의 일부로 쿼리 실행 프로세스 ID(`p_id`)가 열 제목 `session_id` 아래에 나타납니다.

Amazon Redshift 프로비저닝된 클러스터 및 서버리스 작업 그룹에 대한 트랙

Amazon Redshift는 새 버전을 릴리스할 때 Amazon Redshift 데이터 웨어하우스(서버리스 작업 그룹 또는 프로비저닝된 클러스터)의 버전을 업데이트합니다. 데이터 웨어하우스를 최신 릴리스로 업데이트할지 승인된 이전 릴리스로 업데이트할지 여부를 제어할 수 있습니다.

서버리스 작업 그룹 또는 프로비저닝된 클러스터의 트랙에 따라 버전 업데이트 중에 적용되는 릴리스 버전이 결정됩니다. Amazon Redshift는 지정된 유지 관리 기간 동안 프로비저닝된 클러스터를 업데이트하고 일반적으로 서버리스 작업 그룹을 유휴 시간 동안 업데이트합니다. Redshift Serverless가 작업 그룹을 업데이트하는 시기에 대한 자세한 내용은 [서버리스 작업 그룹 업데이트](#) 섹션을 참조하세요.

Amazon Redshift가 새 버전을 릴리스하면 해당 버전은 현재 트랙에 할당되고, 이전 버전은 후행 트랙에 할당됩니다. 데이터 웨어하우스의 트랙을 설정하려면 다음 값 중 하나를 지정합니다.

- 현재 - 현재 트랙을 사용하면 최신 기능, 보안 업데이트 및 성능 개선 사항이 포함된 최신 인증 릴리스 버전을 얻을 수 있습니다.
- 후행 - 후행 트랙을 사용하면 이전 인증 릴리스를 이용할 수 있습니다.

예를 들어 서버리스 작업 그룹이 현재 버전 1.0.2762를 실행 중이고 Amazon Redshift가 Redshift Serverless 버전 1.0.3072를 릴리스한다고 가정해 보겠습니다. 트랙 값이 현재인 경우 작업 그룹이 버전 1.0.3072(최신 릴리스)로 업데이트됩니다. 트랙 값을 후행으로 설정하면 다음 후행 트랙 버전이 릴리스될 때 작업 그룹이 업데이트됩니다.

후행 트랙 기능을 사용하면 후행 트랙에서 Amazon Redshift 데이터 웨어하우스의 하위 세트를 실행할 수 있습니다. 이를 통해 후행 트랙의 데이터 웨어하우스에 릴리스를 적용하기 전에 현재 트랙으로 설정된 데이터 웨어하우스에서 1~6주 동안 릴리스의 테스트 및 통합 검증을 수행할 수 있습니다. 기본적으로 Amazon Redshift는 현재 트랙에 모든 클러스터와 작업 그룹을 생성하여 최신 인증 릴리스를 활용합니다. 하지만 프로덕션 환경에서 Amazon Redshift 후행 트랙을 사용하고 테스트 및 개발 환경에서 현재 트랙을 사용하면 최신 릴리스를 평가하는 데 추가 노력과 시간을 할애할 수 있습니다. 후행 트랙은 안정성을 극대화하여 프로덕션 환경의 미션 크리티컬 워크로드에 이상적입니다.

Note

후행 트랙 버전은 짧은 시간 동안 현재 트랙 버전과 동일할 수 있습니다. 이는 현재 트랙이 다음 버전으로 진행되지 않은 경우에 발생합니다. 일반적으로 현재 트랙 버전은 후행 트랙 버전보다 앞서 있습니다.

트랙 간 전환

Amazon Redshift 리소스에 대한 트랙 변경은 일반적으로 일회성 결정입니다. 트랙 변경 시 주의해야 합니다. 각 데이터 웨어하우스 버전에 있는 기능에 대한 자세한 내용은 [Amazon Redshift에 대한 클러스터 버전](#) 섹션을 참조하세요.

트랙을 후행에서 현재로 변경할 경우 데이터 웨어하우스가 현재 트랙 릴리스 버전으로 업데이트됩니다. 데이터 웨어하우스의 트랙을 후행으로 변경하면 다음과 같이 데이터 웨어하우스가 업데이트됩니다.

- 서버리스 작업 그룹의 경우 데이터 웨어하우스 버전을 유효 시간 동안 업데이트합니다. Redshift Serverless가 작업 그룹의 버전을 업데이트하는 방법에 대한 자세한 내용은 [서버리스 작업 그룹 업데이트](#) 섹션을 참조하세요.
- 프로비저닝된 클러스터의 경우 현재 트랙 릴리스 버전 이후에 새 릴리스가 있을 때까지 데이터 웨어하우스가 업데이트되지 않습니다.

트랙 및 복원

서버리스 작업 그룹의 경우 스냅샷은 대상 Amazon Redshift 데이터 웨어하우스의 트랙을 상속합니다. 예를 들어, 후행 트랙으로 설정된 작업 그룹에 대한 스냅샷을 만들고 해당 스냅샷을 현재 트랙으로 설정된 작업 그룹에 적용하면 작업 그룹의 트랙 설정은 현재가 됩니다.

프로비저닝된 클러스터의 경우 스냅샷은 소스 Amazon Redshift 데이터 웨어하우스의 트랙을 상속합니다. 스냅샷을 만든 후 소스 데이터 웨어하우스의 트랙을 변경하는 경우 스냅샷과 소스 데이터 웨어하우스는 다른 트랙에 있습니다. 스냅샷에서 복원하면 새 데이터 웨어하우스는 스냅샷 소스에서 상속된 트랙에 있습니다. 복원 작업을 완료한 이후에 트랙을 변경할 수 있습니다.

데이터 웨어하우스의 크기를 조정해도 트랙에는 영향을 주지 않습니다.

서버리스 작업 그룹 업데이트

작업 그룹이 선택한 트랙에 새 버전이 사용 가능해지면, Amazon Redshift Serverless는 일반적으로 보류 중인 트랙 업데이트 요청이 없는 한 유효 시간 동안 업데이트를 적용합니다. 작업 그룹이 14일 이내에 유효 기간을 경험하지 않으면 Redshift Serverless가 버전 업데이트를 강제합니다.

Redshift Serverless는 작업 그룹을 다음 상위 버전으로만 업데이트합니다. Redshift Serverless는 선택한 작업 그룹의 트랙 버전이 해당 작업 그룹의 현재 버전보다 낮더라도 중간 버전을 건너뛰거나 작업 그룹을 다운그레이드하지 않습니다. Trailing 트랙이 따라잡을 때까지 작업 그룹은 주요 버전 업그레이드를 받지 못할 것입니다.

예를 들어 Current 트랙이 버전 186이고 Trailing 트랙이 버전 185라고 가정해 보겠습니다. 작업 그룹의 Track 값이 Current이고 버전이 186인 경우, Track 값을 Trailing으로 변경해도 Redshift Serverless는 해당 작업 그룹의 버전을 185로 다운그레이드하지 않습니다. 시나리오에서 Redshift Serverless는 Trailing 트랙 버전이 186 이상이 될 때까지 작업 그룹을 버전 186으로 유지합니다.

트랙 변경이 보류 중인 경우, Redshift Serverless는 트랙 변경을 적용할 때까지 기존 트랙에서 작업 그룹을 다음 메이저 버전으로 업데이트하지 않습니다. 트랙 변경이 완료된 후, Redshift Serverless는 새 트랙 하에서 작업 그룹을 적절한 버전으로 업데이트할 조건을 평가합니다.

작업 그룹이 Current 트랙으로 설정되어 있고 현재 트랙이 186인 상태에서 작업 그룹을 Trailing 트랙으로 변경하는 경우 Redshift Serverless는 트랙 변경을 적용하고 이후에 Trailing 버전이 186 이상으로 업데이트된 후에야 작업 그룹을 업데이트합니다.

Note

작업 그룹에 대한 기존 작업(예: 스냅샷에서 복원, KMS 키 변경 또는 크기 조정)은 기존 트랙에서만 수행됩니다. Redshift Serverless는 서버리스 작업에 대해 보류 중인 트랙을 사용하지 않습니다.

트랙 전환 요청이 보류 중인 경우, [UpdateWorkgroup](#)을 사용하여 track 파라미터를 원래 값으로 되돌려 요청을 취소할 수 있습니다.

버전 관리

트랙은 일련의 릴리스입니다. Amazon Redshift 데이터 웨어하우스가 현재 트랙 또는 후행 트랙에 있는지 결정할 수 있습니다. 데이터 웨어하우스를 현재 트랙에 배치하면 항상 최신 릴리스 버전으로 업그레이드

이드됩니다. 리소스를 후행 트랙에 배치하면 항상 최신 릴리스 버전 직전에 릴리스된 릴리스 버전을 실행합니다.

프로비저닝된 클러스터의 경우, Amazon Redshift 데이터 웨어하우스의 Amazon Redshift 콘솔 목록에 있는 릴리스 상태 열은 리소스 중 하나를 업그레이드할 수 있는지 여부를 나타냅니다.

작업 그룹 또는 클러스터 버전 확인

Amazon Redshift 콘솔을 사용하여 Amazon Redshift 서버리스 작업 그룹 버전 또는 프로비저닝된 클러스터 버전 엔진을 확인할 수 있습니다.

AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.

Serverless workgroups

서버리스 작업 그룹의 경우, 탐색 메뉴에서 작업 그룹을 선택한 후 목록에서 작업 그룹 이름을 선택하여 세부 정보를 엽니다. 작업 그룹의 세부 정보가 표시됩니다.

Provisioned clusters

프로비저닝된 클러스터의 경우, 탐색 메뉴에서 클러스터를 선택한 후 목록에서 클러스터 이름을 선택하여 세부 정보를 엽니다.

클러스터의 세부 정보가 표시됩니다. 여기에는 클러스터 성능, 쿼리 모니터링, 데이터베이스, 데이터 공유, 일정, 유지 관리 및 속성 탭이 포함될 수 있습니다. 더 자세한 세부 정보를 보려면 유지 관리(Maintenance) 탭을 선택합니다.

유지 관리(Maintenance) 섹션에서 현재 클러스터 버전(Current cluster version)을 찾습니다.

Note

프로비저닝된 클러스터의 경우, 콘솔은 버전 정보를 하나의 필드에 표시하지만 Amazon Redshift API에서는 버전 정보가 두 개의 파라미터입니다. 이러한 파라미터는 ClusterVersion 및 ClusterRevisionNumber입니다. 자세한 내용은 Amazon Redshift API Reference의 [Cluster\(클러스터\)](#)를 참조하세요.

제로 ETL 통합

제로 ETL 통합은 Amazon Redshift에서 여러 운영 및 트랜잭션 소스의 트랜잭션 및 운영 데이터를 사용할 수 있도록 하는 완전 관리형 솔루션입니다. 이 솔루션을 사용하면 소스에서 Amazon Redshift 데이터 웨어하우스로의 통합을 구성할 수 있습니다. 추출, 변환, 적재(ETL) 파이프라인을 유지 관리할 필요가 없습니다. 데이터 소스에서 Amazon Redshift 클러스터 또는 Redshift Serverless 네임스페이스로의 데이터 복제 생성 및 관리를 자동화하여 ETL을 대신 처리해 드립니다. 보고 및 대시보드와 같은 분석 워크로드에 Amazon Redshift를 사용하면서 동시에 소스 데이터를 계속 업데이트하고 쿼리할 수 있습니다.

제로 ETL 통합을 통해 분석, AI/ML 및 보고를 위한 최신 데이터를 확보할 수 있습니다. 비즈니스 대시보드, 최적화된 게임 경험, 데이터 품질 모니터링, 고객 행동 분석과 같은 사용 사례에 대해 더 정확하고 시기적절한 인사이트를 얻을 수 있습니다. 더 신뢰성 있게 데이터 기반 예측을 수행하고, 고객 경험을 개선하고, 비즈니스 전반에서 데이터 기반 인사이트를 촉진할 수 있습니다.

현재 제로 ETL 통합에서 지원되는 소스는 다음과 같습니다.

- Amazon Aurora MySQL(AMS)
- Amazon Aurora PostgreSQL(APG)
- Amazon DynamoDB
- Amazon RDS for MySQL
- Amazon RDS for Oracle
- Amazon RDS for PostgreSQL
- Oracle Database@AWS
- Salesforce, Salesforce Marketing Cloud Account Engagement, SAP, ServiceNow, Instagram 광고, Meta 광고, Zendesk 등의 애플리케이션
- 자체 관리형 MySQL, PostgreSQL, SQL Server 및 Oracle

제로 ETL 통합을 만들려면 데이터 소스를 지정하고 Amazon Redshift 데이터 웨어하우스를 대상으로 지정합니다. 최초 데이터 로드 후 통합은 소스 데이터베이스에서 대상 데이터 웨어하우스로 데이터를 복제합니다. Amazon Redshift에서 데이터를 사용할 수 있게 됩니다. 통합 소스를 생성할 때, 제로 ETL 통합을 생성할 때, Amazon Redshift 데이터 웨어하우스를 생성할 때 데이터의 암호화를 제어합니다. 통합은 데이터 파이프라인의 상태를 모니터링하고 가능한 경우 문제로부터 복구합니다. 동일한 유형의 소스에서 단일 Amazon Redshift 데이터 웨어하우스로 통합을 생성하여 여러 애플리케이션에 걸쳐 전체적인 인사이트를 도출할 수 있습니다.

Amazon Redshift의 데이터를 가지고 Amazon Redshift에서 제공하는 분석을 사용할 수 있습니다. 제공되는 분석 기능에는 기본 제공 기계 학습(ML), 구체화된 뷰, 데이터 공유, 여러 데이터 저장소 및 데이터 레이크에 대한 직접 액세스 등이 있습니다. 데이터 엔지니어의 경우, 제로 ETL 통합을 통해 복잡한 데이터 파이프라인에서 간헐적으로 발생하는 오류로 인해 지연될 수 있는 시간에 민감한 데이터에 액세스할 수 있습니다. 트랜잭션 데이터에서 분석 쿼리와 ML 모델을 실행하여 시간에 민감한 이벤트와 비즈니스 의사 결정에 대한 시기적절한 인사이트를 도출할 수 있습니다.

Amazon Redshift 이벤트 알림 구독을 생성하여 주어진 제로 ETL 통합에 이벤트가 발생하면 알림을 받을 수 있습니다. 통합 관련 이벤트 알림 목록을 보려면 [Amazon EventBridge를 사용하여 제로 ETL 통합 이벤트 알림](#) 섹션을 참조하세요. 가장 간단한 구독 생성 방법은 Amazon SNS 콘솔을 이용하는 것입니다. Amazon SNS 주제를 새로 만들어 구독하는 방법에 대한 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 [Amazon SNS 시작하기](#) 섹션을 참조하세요.

제로 ETL 통합을 시작할 때는 다음 개념을 고려하세요.

- 소스 데이터베이스는 Amazon Redshift로 복제되는 데이터가 속한 데이터베이스입니다.
- 대상 데이터 웨어하우스는 데이터가 복제되어 들어오는 Amazon Redshift 프로비저닝된 클러스터 또는 Redshift Serverless 작업 그룹입니다.
- 대상 데이터베이스는 제로 ETL 통합을 통해 대상 데이터 웨어하우스에 생성하는 데이터베이스입니다.

제로 ETL 통합을 모니터링하는 데 사용할 수 있는 시스템 테이블 및 뷰에 대한 자세한 내용은 [Amazon Redshift 시스템 뷰로 제로 ETL 통합 모니터링](#)을 참조하세요.

제로 ETL 통합의 각 소스가 지원하는 AWS 리전 목록은 [제로 ETL 통합을 지원하는 리전](#) 섹션을 참조하세요.

제로 ETL 통합에 대한 요금 정보는 해당 요금 페이지를 참조하세요.

- [Amazon Redshift 요금](#)
- [Amazon Aurora 요금](#)
- [Amazon RDS 요금](#)
- [Amazon DynamoDB 요금](#)
- [AWS Glue 요금](#)

제로 ETL 통합 소스에 대한 자세한 내용은 다음 주제를 참조하세요.

- Aurora 제로 ETL 통합의 경우 Amazon Aurora 사용 설명서에서 제로 ETL 통합의 [이점](#), [핵심 개념](#), [제한 사항](#), [할당량](#) 및 [지원되는 리전](#)을 참조하세요.
- RDS 제로 ETL 통합의 경우 Amazon RDS 사용 설명서에서 제로 ETL 통합의 [이점](#), [핵심 개념](#), [제한 사항](#), [할당량](#) 및 [지원되는 리전](#)을 참조하세요.
- DynamoDB 제로 ETL 통합은 Amazon DynamoDB 개발자 안내서의 [Amazon Redshift와의 DynamoDB 제로 ETL 통합](#)을 참조하세요.
- 애플리케이션과의 제로 ETL 통합은 AWS Glue 개발자 안내서의 [제로 ETL 통합](#)을 참조하세요.

주제

- [Amazon Redshift와 제로 ETL 통합을 사용할 때 고려할 사항](#)
- [제로 ETL 통합 시작하기](#)
- [Amazon Redshift에서 대상 데이터베이스 생성](#)
- [Amazon Redshift에서 복제된 데이터 쿼리](#)
- [제로 ETL 통합 보기](#)
- [기록 모드](#)
- [Amazon Redshift에서 데이터 공유](#)
- [제로 ETL 통합 모니터링](#)
- [제로 ETL 통합의 지표](#)
- [DynamoDB에 대한 제로 ETL 통합 수정](#)
- [DynamoDB에 대한 제로 ETL 통합 삭제](#)
- [제로 ETL 통합을 지원하는 리전](#)
- [제로 ETL 통합 문제 해결](#)

Amazon Redshift와 제로 ETL 통합을 사용할 때 고려할 사항

다음 고려 사항은 Amazon Redshift와의 제로 ETL 통합에 적용됩니다.

- 대상 Amazon Redshift 데이터 웨어하우스는 다음 사전 요구 사항을 충족해야 합니다.
 - Amazon Redshift Serverless 또는 RA3 노드 유형을 실행합니다.
 - 암호화되어 있습니다(프로비저닝된 클러스터를 사용하는 경우).
 - 대/소문자 구분이 활성화되어 있습니다.

- Amazon Redshift 데이터 웨어하우스에 대해 권한이 부여된 통합 소스인 소스를 삭제하면 연결된 모든 통합이 FAILED 상태로 전환됩니다. 이전에 복제된 모든 데이터는 Amazon Redshift 데이터베이스에 남아 있으며 쿼리할 수 있습니다.
- 대상 데이터베이스는 읽기 전용입니다. 대상 데이터베이스에서는 테이블, 뷰 또는 구체화된 뷰를 만들 수 없습니다. 하지만 대상 데이터 웨어하우스의 다른 테이블에서는 구체화된 뷰를 사용할 수 있습니다.
- 구체화된 뷰는 데이터베이스 간 쿼리에서 사용할 때 지원됩니다. 제로 ETL 통합을 통해 복제된 데이터로 구체화된 뷰를 생성하는 방법에 대한 자세한 정보는 [구체화된 뷰로 복제된 데이터 쿼리](#) 섹션을 참조하세요.
- 기본적으로 Synced 상태인 대상 데이터 웨어하우스의 테이블만 쿼리할 수 있습니다. 다른 상태의 테이블을 쿼리하려면 데이터베이스 파라미터 QUERY_ALL_STATES를 TRUE로 설정합니다. QUERY_ALL_STATES 설정에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [CREATE DATABASE](#) 및 [ALTER DATABASE](#)를 참조하세요. 데이터베이스 상태에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [SVV_INTEGRATION_TABLE_STATE](#)를 참조하세요.
- Amazon Redshift는 UTF-8 문자만 허용하므로 소스에 정의된 데이터 정렬을 따르지 않을 수 있습니다. 정렬 및 비교 규칙이 다를 수 있으며, 이로 인해 결국 쿼리 결과가 변경될 수 있습니다.
- 제로 ETL 통합은 Amazon Redshift 데이터 웨어하우스 대상당 50개로 제한됩니다.
- 통합 원본의 테이블에 프라이머리 키가 있어야 합니다. 그렇지 않으면 Amazon Redshift의 대상 데이터 웨어하우스에 테이블을 복제할 수 없습니다.

Amazon Aurora PostgreSQL에 프라이머리 키를 추가하는 방법에 대한 자세한 내용은 AWS Database Blog의 [Handle tables without primary keys while creating Amazon Aurora PostgreSQL zero-ETL integrations with Amazon Redshift](#)를 참조하세요. Amazon Aurora MySQL 또는 RDS for MySQL에 프라이머리 키를 추가하는 방법에 대한 자세한 내용은 AWS 데이터베이스 블로그의 [Amazon Redshift를 사용해 Amazon Aurora MySQL 또는 Amazon RDS for MySQL 제로 ETL 통합을 생성하는 동안 프라이머리 키가 없는 테이블 처리](#)를 참조하세요.

- Aurora 제로 ETL 통합의 데이터 필터링을 사용하여 소스 Aurora DB 클러스터에서 대상 Amazon Redshift 데이터 웨어하우스로의 복제 범위를 정의할 수 있습니다. 모든 데이터를 대상에 복제하는 대신 특정 테이블을 선택적으로 포함하거나 복제에 제외하는 필터를 하나 이상 정의할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 [Data filtering for Aurora zero-ETL integrations with Amazon Redshift](#)를 참조하세요.
- Amazon Redshift와의 Aurora PostgreSQL 제로 ETL 통합의 경우 Amazon Redshift는 Aurora PostgreSQL에서 최대 100개의 데이터베이스를 지원합니다. 각 데이터베이스는 소스에서 대상으로 독립적으로 복제됩니다.

- 제로 ETL 통합은 트랜잭션 데이터 스토어의 데이터를 Amazon Redshift로 복제하는 동안에는 변환을 지원하지 않습니다. 데이터는 소스 데이터베이스에서 있는 그대로 복제됩니다. 하지만 Amazon Redshift에서 복제된 데이터에 변환을 적용할 수 있습니다.
- 제로 ETL 통합은 Amazon Redshift에서 병렬 연결을 사용하여 실행됩니다. 이는 통합에서 데이터베이스를 생성한 사용자의 자격 증명을 사용하여 실행됩니다. 쿼리가 실행되면 동기화(쓰기) 중에 이러한 연결에 대한 동시성 확장이 시작되지 않습니다. (Amazon Redshift 클라이언트에서) 동시성 확장 읽기는 동기화된 객체에 지원됩니다.
- 제로 ETL 통합에 대한 REFRESH_INTERVAL을 설정하여 Amazon Redshift로의 데이터 복제 빈도를 제어할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [CREATE DATABASE](#) 및 [ALTER DATABASE](#)을 참조하세요.
- Amazon DynamoDB와의 제로 ETL 통합을 통해 Amazon Redshift 데이터베이스를 생성한 후에는 데이터베이스 상태가 생성 중에서 활성으로 변경됩니다. 그러면 대상 데이터베이스 (ddb_rs_customerprofiles_zetl_db)의 퍼블릭 스키마를 사용하여 생성된 대상 Redshift 테이블로 소스 DynamoDB 테이블의 데이터가 복제됩니다.

대상에서 기록 모드 사용 시 고려 사항

대상 데이터베이스에서 기록 모드를 사용하는 경우 다음 고려 사항이 적용됩니다. 자세한 내용은 [기록 모드](#) 섹션을 참조하세요.

- 소스에서 테이블을 삭제하면 대상에서 해당 테이블이 삭제되지 않고 DroppedSource 상태로 변경됩니다. Amazon Redshift 데이터베이스에서 테이블을 삭제하거나 이름을 바꿀 수 있습니다.
- 소스에서 테이블을 잘라내면 해당 대상 테이블에서 삭제가 실행됩니다. 예를 들어 소스에서 모든 레코드가 잘린 경우 대상 열 _record_is_active의 해당 레코드가 false로 변경됩니다.
- 대상 테이블에서 TRUNCATE table SQL을 실행하면 활성 기록 행이 해당 타임스탬프와 함께 비활성으로 표시됩니다.
- 테이블의 행이 비활성으로 설정된 경우 짧은(약 10분) 지연 후 삭제할 수 있습니다. 비활성 행을 삭제하려면 Query Editor v2 또는 다른 SQL 클라이언트를 사용하여 제로 ETL 데이터베이스에 연결합니다.
- 기록 모드가 켜져 있는 테이블에서만 비활성 행을 삭제할 수 있습니다. 예를 들어 다음과 유사한 SQL 명령만 비활성 행을 삭제합니다.

```
delete from schema.user_table where _record_delete_time <= '2024-09-10 12:34:56'
```

이는 다음과 같은 SQL 명령과 동일합니다.

```
delete from schema.user_table where _record_delete_time <= '2024-09-10 12:34:56' and
_record_is_active = False
```

- 테이블에 대해 기록 모드를 끄면 모든 기록 데이터가 <schema>.<table-name>_historical_<timestamp>라는 테이블에 저장되고 원래 테이블 <schema>.<table-name>은 새로 고쳐집니다.
- 기록 모드가 켜진 테이블을 테이블 필터를 사용하여 복제에서 제외하면 모든 행이 비활성으로 설정되고 테이블이 DroppedSource 상태로 변경됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 [Amazon Redshift와 Aurora 제로 ETL 통합의 데이터 필터링](#)을 참조하세요.
- 기록 모드는 Synced 상태의 테이블에서만 true 또는 false로 전환할 수 있습니다.
- 기록 모드가 켜진 테이블에 대한 구체화된 뷰는 전체 재계산으로 생성됩니다.

제로 ETL 통합 소스가 Aurora 또는 Amazon RDS인 경우 고려 사항

다음 고려 사항은 Amazon Redshift와의 Aurora 및 Amazon RDS 제로 ETL 통합에 적용됩니다.

- Aurora 및 RDS for MySQL 제로 ETL 통합의 데이터 필터링을 사용하여 소스 DB 클러스터에서 대상 Amazon Redshift 데이터 웨어하우스로의 복제 범위를 정의할 수 있습니다. 모든 데이터를 대상에 복제하는 대신 특정 테이블을 선택적으로 포함하거나 복제에 제외하는 필터를 하나 이상 정의할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 [Data filtering for Aurora zero-ETL integrations with Amazon Redshift](#)를 참조하세요.
- 통합 원본의 테이블에 프라이머리 키가 있어야 합니다. 그렇지 않으면 Amazon Redshift의 대상 데이터 웨어하우스에 테이블을 복제할 수 없습니다.

Amazon Aurora PostgreSQL에 프라이머리 키를 추가하는 방법에 대한 자세한 내용은 AWS Database Blog의 [Handle tables without primary keys while creating Amazon Aurora PostgreSQL zero-ETL integrations with Amazon Redshift](#)를 참조하세요. Amazon Aurora MySQL 또는 RDS for MySQL에 프라이머리 키를 추가하는 방법에 대한 자세한 내용은 AWS 데이터베이스 블로그의 [Amazon Redshift를 사용해 Amazon Aurora MySQL 또는 Amazon RDS for MySQL 제로 ETL 통합을 생성하는 동안 프라이머리 키가 없는 테이블 처리](#)를 참조하세요.

- Amazon Redshift VARCHAR 데이터 유형의 최대 길이는 65,535바이트입니다. 소스의 콘텐츠가 이 제한에 맞지 않으면 복제가 진행되지 않고 테이블이 실패 상태가 됩니다. 데이터베이스 파라미터 TRUNCATECOLUMNS를 TRUE로 설정하여 열에 맞게 콘텐츠를 잘라낼 수 있습니다. TRUNCATECOLUMNS 설정에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [CREATE DATABASE](#) 및 [ALTER DATABASE](#)를 참조하세요.

제로 ETL 통합 소스와 Amazon Redshift 데이터베이스 간의 데이터 유형 차이에 관한 자세한 내용은 Amazon Aurora 사용 설명서의 [Aurora와 Amazon Redshift 간의 데이터 유형 차이](#) 섹션을 참조하세요.

Aurora 소스의 경우 Amazon Aurora 사용 설명서의 [제한 사항](#)을 참조하세요.

Amazon RDS 소스의 경우 Amazon RDS 사용 설명서의 [제한 사항](#)을 참조하세요.

제로 ETL 통합 소스가 DynamoDB인 경우 고려 사항

다음 고려 사항은 Amazon Redshift와의 DynamoDB 제로 ETL 통합에 적용됩니다.

- 127자를 초과하는 DynamoDB의 테이블 이름은 지원되지 않습니다.
- DynamoDB 제로 ETL 통합의 데이터는 Amazon Redshift의 SUPER 데이터 유형 열에 매핑됩니다.
- 파티션 키의 열 이름 또는 127자를 초과하는 정렬 키는 지원되지 않습니다.
- DynamoDB의 제로 ETL 통합은 하나의 Amazon Redshift 데이터베이스에만 매핑할 수 있습니다.
- 파티션 및 정렬 키의 경우 정밀도 및 스케일 최댓값은 (38,18)입니다. DynamoDB의 숫자 데이터 유형은 최대 38개의 최대 정밀도를 지원합니다. Amazon Redshift는 최대 정밀도 38도 지원하지만 Amazon Redshift의 기본 십진수 정밀도/스케일은 (38,10)입니다. 즉, 값 스케일 값을 잘라낼 수 있습니다.
- 성공적인 제로 ETL 통합을 위해 DynamoDB 항목의 개별 속성(이름+값으로 구성)은 64KB보다 크지 않아야 합니다.
- 활성화 시 제로 ETL 통합은 전체 DynamoDB 테이블을 내보내 Amazon Redshift 데이터베이스를 채웁니다. 이 초기 프로세스를 완료하는 데 걸리는 시간은 DynamoDB 테이블 크기에 따라 달라집니다. 그런 다음 제로 ETL 통합은 DynamoDB 증분 내보내기를 사용하여 DynamoDB에서 Amazon Redshift로 업데이트를 점진적으로 복제합니다. 따라서 Amazon Redshift에서 복제된 DynamoDB 데이터가 자동으로 최신 상태로 유지됩니다.

현재 DynamoDB 제로 ETL 통합의 최소 지연 시간은 15분입니다. 제로 ETL 통합에 대해 0이 아닌 REFRESH_INTERVAL을 설정하여 더 늘릴 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [CREATE DATABASE](#) 및 [ALTER DATABASE](#)을 참조하세요.

Amazon DynamoDB 소스의 경우 Amazon DynamoDB 개발자 안내서의 [사전 조건 및 제한 사항](#)도 참조하세요.

제로 ETL 통합 소스가 Salesforce, SAP, ServiceNow 및 Zendesk와 같은 애플리케이션일 경우 고려 사항

다음 고려 사항은 Amazon Redshift와 통합하는 소스가 Salesforce, SAP, ServiceNow 및 Zendesk와 같은 애플리케이션일 경우 적용됩니다.

- 애플리케이션 소스의 127자를 초과하는 테이블 이름과 열 이름은 지원되지 않습니다.
- Amazon Redshift VARCHAR 데이터 유형의 최대 길이는 65,535바이트입니다. 소스의 콘텐츠가 이 제한에 맞지 않으면 복제가 진행되지 않고 테이블이 실패 상태가 됩니다. 데이터베이스 파라미터 TRUNCATECOLUMNS를 TRUE로 설정하여 열에 맞게 콘텐츠를 잘라낼 수 있습니다. TRUNCATECOLUMNS 설정에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [CREATE DATABASE](#) 및 [ALTER DATABASE](#)를 참조하세요.

제로 ETL 통합 애플리케이션 소스와 Amazon Redshift 데이터베이스 간의 데이터 유형 차이에 대한 자세한 내용은 AWS Glue 개발자 안내서의 [제로 ETL 통합](#)을 참조하세요.

- 애플리케이션과의 제로 ETL 통합의 최소 지연 시간은 1시간입니다. 제로 ETL 통합에 대해 0이 아닌 REFRESH_INTERVAL을 설정하여 더 늘릴 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [CREATE DATABASE](#) 및 [ALTER DATABASE](#)을 참조하세요.

애플리케이션과의 제로 ETL 통합 소스는 AWS Glue 개발자 안내서의 [제로 ETL 통합](#)도 참조하세요.

제로 ETL 통합 시작하기

이 태스크 세트는 첫 번째 제로 ETL 통합 설정 과정을 안내합니다. 먼저 통합 소스를 구성하고 필요한 파라미터 및 권한으로 소스를 설정하세요. 그런 다음 Amazon Redshift 콘솔 또는 AWS CLI에서 나머지 초기 설정을 계속 진행합니다. 콘솔은 일부 구성 문제를 해결하기 위한 수정 옵션을 제공합니다.

주제

- [대상 Amazon Redshift 데이터 웨어하우스 생성 및 구성](#)
- [데이터 웨어하우스에 대/소문자 구분 기능 사용 설정](#)
- [Amazon Redshift 데이터 웨어하우스에 대한 권한 부여 구성](#)
- [제로 ETL 통합 생성](#)

대상 Amazon Redshift 데이터 웨어하우스 생성 및 구성

이 단계에서는 Redshift Serverless 작업 그룹 또는 프로비저닝된 클러스터와 같은 대상 Amazon Redshift 데이터 웨어하우스를 생성하고 구성합니다. 제로 ETL 통합에 사용하도록 Amazon Redshift 데이터 웨어하우스가 이미 구성된 경우 이 단계를 건너뛸 수 있습니다.

대상 데이터 웨어하우스의 특성은 다음과 같아야 합니다.

- Amazon Redshift Serverless 또는 RA3 노드 유형의 프로비저닝 클러스터를 실행합니다.
- 대/소문자 구분(enable_case_sensitive_identifier)이 켜져 있습니다. 자세한 내용은 [데이터 웨어하우스에 대/소문자 구분 기능 사용 설정](#) 섹션을 참조하세요.
- 대상 데이터 웨어하우스가 Amazon Redshift 프로비저닝된 클러스터인 경우 암호화됩니다. 자세한 내용은 [Amazon Redshift 데이터베이스 암호화](#) 섹션을 참조하세요.
- 통합 소스와 동일한 AWS 리전에서 생성되었습니다.

제로 ETL 통합을 위해 대상 데이터 웨어하우스를 만들려면 배포 유형에 따라 다음 항목 중 하나를 참조하세요.

- Amazon Redshift 프로비저닝된 클러스터를 만들려면 [클러스터 생성](#) 섹션을 참조하세요.
- 네임스페이스가 있는 Amazon Redshift Serverless 작업 그룹을 만들려면 [네임스페이스가 있는 작업 그룹 생성](#) 섹션을 참조하세요.

프로비저닝된 클러스터를 생성하면 기본 파라미터 그룹도 생성됩니다. 기본 파라미터 그룹은 편집할 수 없습니다. 하지만 새 클러스터를 생성하기 전에 사용자 지정 파라미터 그룹을 만든 다음 클러스터와 연결할 수 있습니다. 또는 생성된 클러스터와 연결할 파라미터 그룹을 편집할 수 있습니다. 또한 사용자 지정 파라미터 그룹을 만들 때 또는 현재 파라미터 그룹을 편집할 때 대/소문자 구분을 사용하도록 설정해야 제로 ETL 통합을 사용할 수 있습니다.

Amazon Redshift 콘솔 또는 AWS CLI를 사용하여 사용자 지정 파라미터 그룹을 만들려면 [파라미터 그룹 생성](#)을 참조하세요.

데이터 웨어하우스에 대/소문자 구분 기능 사용 설정

프로비저닝된 클러스터를 생성하는 동안 파라미터 그룹을 연결하고 대/소문자 구분을 활성화할 수 있습니다. 하지만 서버리스 작업 그룹은 생성된 후에만 AWS Command Line Interface(AWS CLI)를 통해 업데이트할 수 있습니다. 이는 소스 테이블 및 열의 대/소문자 구분을 지원하는 데

필요합니다. `enable_case_sensitive_identifier`는 데이터베이스, 테이블 및 열의 이름 식별자가 대/소문자를 구분하는지 여부를 결정하는 구성 값입니다. 데이터 웨어하우스에서 대/소문자를 구분하지 않는 통합을 만들려면 이 파라미터를 사용 설정해야 합니다. 자세한 내용은 [enable_case_sensitive_identifier](#) 섹션을 참조하세요.

Amazon Redshift Serverless의 경우 – [AWS CLI를 사용하여 Amazon Redshift 서버리스에 대/소문자 구분 사용 설정](#). 대/소문자 구분은 AWS CLI에서만 Amazon Redshift Serverless에 대해 설정할 수 있다는 점에 유의하세요.

Amazon Redshift 프로비저닝 클러스터의 경우 다음 주제 중 하나를 사용하여 대상 클러스터의 대/소문자 구분을 활성화하세요.

- [Amazon Redshift 콘솔을 사용하여 대/소문자 구분을 프로비저닝된 클러스터에 사용 설정](#)
- [AWS CLI를 사용하여 대/소문자 구분을 프로비저닝된 클러스터에 사용 설정](#)

AWS CLI를 사용하여 Amazon Redshift 서버리스에 대/소문자 구분 사용 설정

다음 AWS CLI 명령을 실행하여 작업 그룹의 대/소문자 구분을 활성화합니다.

```
aws redshift-serverless update-workgroup \  
    --workgroup-name target-workgroup \  
    --config-parameters  
    parameterKey=enable_case_sensitive_identifier,parameterValue=true
```

다음 단계로 진행하기 전에 작업 그룹 상태가 Active 상태가 될 때까지 기다립니다.

Amazon Redshift 콘솔을 사용하여 대/소문자 구분을 프로비저닝된 클러스터에 사용 설정

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 프로비저닝된 클러스터 대시보드를 선택합니다.
3. 데이터를 복제할 프로비저닝된 클러스터를 선택합니다.
4. 왼쪽 탐색 창에서 구성 > 워크로드 관리를 선택합니다.
5. 워크로드 관리 페이지에서 파라미터 그룹을 선택합니다.
6. 파라미터 탭을 선택합니다.

7. 파라미터 편집을 선택한 다음 `enable_case_sensitive_identifier`를 `true`로 변경합니다.
8. 그런 다음 저장을 선택합니다.

AWS CLI를 사용하여 대/소문자 구분을 프로비저닝된 클러스터에 사용 설정

1. 기본 파라미터 그룹은 편집할 수 없으므로 터미널 프로그램에서 다음 AWS CLI 명령을 실행하여 사용자 지정 파라미터 그룹을 생성합니다. 나중에 프로비저닝된 클러스터와 연결할 것입니다.

```
aws redshift create-cluster-parameter-group \
  --parameter-group-name zero-etl-params \
  --parameter-group-family redshift-2.0 \
  --description "Param group for zero-ETL integrations"
```

2. 다음 AWS CLI 명령을 실행하여 파라미터 그룹의 대/소문자 구분을 활성화합니다.

```
aws redshift modify-cluster-parameter-group \
  --parameter-group-name zero-etl-params \
  --parameters ParameterName=enable_case_sensitive_identifier,ParameterValue=true
```

3. 다음 명령을 실행하여 파라미터 그룹을 클러스터에 연결합니다.

```
aws redshift modify-cluster \
  --cluster-identifier target-cluster \
  --cluster-parameter-group-name zero-etl-params
```

4. 프로비저닝된 클러스터를 사용할 수 있을 때까지 기다립니다. `describe-cluster` 명령을 사용하여 클러스터의 상태를 확인할 수 있습니다. 그리고 나서 다음 명령을 실행하여 클러스터를 재부팅합니다.

```
aws redshift reboot-cluster \
  --cluster-identifier target-cluster
```

Amazon Redshift 데이터 웨어하우스에 대한 권한 부여 구성

통합 소스의 데이터를 Amazon Redshift 데이터 웨어하우스에 복제하려면 먼저 다음 두 엔터티를 추가해야 합니다.

- 권한 있는 보안 주체 — 데이터 웨어하우스에 제로 ETL 통합을 생성할 수 있는 사용자 또는 역할을 식별합니다.

- 권한이 부여된 통합 소스 - 데이터 웨어하우스를 업데이트할 수 있는 소스 데이터베이스를 식별합니다.

Amazon Redshift 콘솔의 리소스 정책 탭에서 또는 Amazon Redshift PutResourcePolicy API 작업을 사용하여 권한이 부여된 보안 주체 및 권한이 부여된 통합 소스를 구성할 수 있습니다.

권한 있는 보안 주체 추가

Redshift Serverless 작업 그룹 또는 프로비저닝된 클러스터에 제로 ETL 통합을 생성하려면 연결된 네임스페이스 또는 프로비저닝된 클러스터에 대한 액세스 권한을 부여하세요.

다음 조건이 모두 충족되면 이 단계를 건너뛰어도 됩니다.

- Redshift Serverless 작업 그룹 또는 프로비저닝된 클러스터를 소유하는 AWS 계정이 소스 데이터베이스도 소유합니다.
- 해당 보안 주체가 이 Redshift Serverless 네임스페이스 또는 프로비저닝된 클러스터에 제로 ETL 통합을 생성할 수 있는 권한이 있는 ID 기반 IAM 정책과 연결되어 있습니다.

Amazon Redshift 서버리스 네임스페이스에 권한 있는 보안 주체 추가

1. Amazon Redshift 콘솔의 왼쪽 탐색 창에서 Redshift Serverless를 선택합니다.
2. 네임스페이스 구성을 선택한 다음 네임스페이스를 선택하고 리소스 정책 탭으로 이동합니다.
3. 권한 있는 보안 주체 추가를 선택합니다.
4. 추가하려는 권한이 부여된 보안 주체마다 AWS 사용자나 역할의 ARN 또는 제로 ETL 통합을 만들기 위해 액세스 권한을 부여하려는 AWS 계정의 ID를 네임스페이스에 입력합니다. 계정 ID는 ARN으로 저장됩니다.
5. 변경 사항 저장을 선택합니다.

Amazon Redshift 프로비저닝된 클러스터에 권한 있는 보안 주체 추가

1. Amazon Redshift 콘솔의 왼쪽 탐색 창에서 프로비저닝된 클러스터 대시보드를 선택합니다.
2. 클러스터를 선택한 다음 클러스터를 선택하고 리소스 정책 탭으로 이동합니다.
3. 권한 있는 보안 주체 추가를 선택합니다.
4. 추가하려는 권한이 부여된 보안 주체마다 AWS 사용자나 역할의 ARN 또는 제로 ETL 통합을 만들기 위해 액세스 권한을 부여하려는 AWS 계정의 ID를 클러스터에 입력합니다. 계정 ID는 ARN으로 저장됩니다.

5. 변경 사항 저장을 선택합니다.

권한 있는 통합 소스 추가

소스가 Amazon Redshift 데이터 웨어하우스를 업데이트할 수 있도록 하려면 네임스페이스에 권한이 부여된 통합 소스로 추가해야 합니다.

권한 있는 통합 소스를 Amazon Redshift 서버리스 네임스페이스에 추가

1. Amazon Redshift 콘솔에서 서버리스 대시보드로 이동합니다.
2. 네임스페이스의 이름을 선택합니다.
3. 리소스 정책 탭으로 이동합니다.
4. 권한 있는 통합 소스 추가를 선택합니다.
5. 제로 ETL 통합을 위한 소스의 ARN을 지정합니다.

Note

권한 있는 통합 소스를 제거하면 네임스페이스에 데이터가 복제되는 것이 중지됩니다. 이 작업을 수행하면 해당 소스에서 이 네임스페이스로의 모든 제로 ETL 통합이 비활성화됩니다.

Amazon Redshift 프로비저닝된 클러스터에 권한 있는 통합 소스 추가

1. Amazon Redshift 콘솔에서 프로비저닝된 클러스터 대시보드로 이동합니다.
2. 프로비저닝된 클러스터의 이름을 선택합니다.
3. 리소스 정책 탭으로 이동합니다.
4. 권한 있는 통합 소스 추가를 선택합니다.
5. 제로 ETL 통합을 위한 데이터 소스인 소스의 ARN을 지정합니다.

Note

권한 있는 통합 소스를 제거하면 프로비저닝된 클러스터에 데이터가 복제되는 것이 중지됩니다. 이 작업을 수행하면 해당 소스에서 이 Amazon Redshift 프로비저닝된 클러스터로의 모든 제로 ETL 통합이 비활성화됩니다.

Amazon Redshift API를 사용하여 권한 부여 구성

Amazon Redshift API 작업을 사용하여 제로 ETL 통합에서 작동하는 리소스 정책을 구성할 수 있습니다.

네임스페이스로의 인바운드 통합을 생성할 수 있는 소스를 제어하려면 리소스 정책을 생성하여 네임스페이스에 연결하세요. 리소스 정책을 사용하여 통합에 액세스할 수 있는 소스를 지정할 수 있습니다. 리소스 정책은 대상 데이터 웨어하우스의 네임스페이스에 연결되어 소스가 인바운드 통합을 생성하여 소스의 라이브 데이터를 Amazon Redshift로 복제할 수 있도록 합니다.

다음은 샘플 리소스 정책입니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "redshift:AuthorizeInboundIntegration",
      "Resource": "arn:aws:redshift:*:*:integration:*",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:rds:*:111122223333:cluster:*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam:*:111122223333:root"
      },
      "Action": "redshift:CreateInboundIntegration",
      "Resource": "arn:aws:redshift:*:*:integration:*"
    }
  ]
}
```

다음은 통합을 위한 리소스 정책 구성에 적용할 수 있는 Amazon Redshift API 작업을 요약한 것입니다.

- [PutResourcePolicy](#) API 작업을 사용하여 리소스 정책을 유지할 수 있습니다. 다른 리소스 정책을 제공하면 해당 리소스의 이전 리소스 정책이 대체됩니다. 다음 작업에 대한 권한을 부여하는 이전 리소스 정책 예시를 사용하세요.
 - `CreateInboundIntegration` - 소스 보안 주체가 소스에서 대상 데이터 웨어하우스로 복제할 데이터에 대한 인바운드 통합을 생성할 수 있도록 허용합니다.
 - `AuthorizeInboundIntegration` - 대상 데이터 웨어하우스가 소스 ARN에서 복제된 데이터를 수신할 수 있는지 Amazon Redshift가 지속적으로 검증하도록 허용합니다.
- [GetResourcePolicy](#) API 작업을 사용하여 기존 리소스 정책을 볼 수 있습니다.
- [DeleteResourcePolicy](#) API 작업을 사용하여 리소스에서 리소스 정책을 제거할 수 있습니다.

리소스 정책을 업데이트하려면 [put-resource-policy](#) AWS CLI 명령을 사용할 수도 있습니다. 예를 들어 DynamoDB 소스에 대한 Amazon Redshift 네임스페이스 ARN에 리소스 정책을 적용하려면 다음과 유사한 AWS CLI 명령을 실행합니다.

```
aws redshift put-resource-policy \
--policy file://rs-rp.json \
--resource-arn "arn:aws:redshift-serverless:us-east-1:123456789012:namespace/cc4ffe56-
ad2c-4fd1-a5a2-f29124a56433"
```

여기서 `rs-rp.json`에 다음 사항이 포함됩니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "redshift:AuthorizeInboundIntegration",
      "Resource": "arn:aws:redshift-serverless:us-
east-1:123456789012:namespace/cc4ffe56-ad2c-4fd1-a5a2-f29124a56433",
      "Condition": {
```



```

        "StringEquals": {
            "aws:SourceArn": "arn:aws:dynamodb:us-east-1:123456789012:table/test_ddb"
        }
    },
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
        },
        "Action": "redshift:CreateInboundIntegration",
        "Resource": "arn:aws:redshift-serverless:us-east-1:123456789012:namespace/cc4ffe56-ad2c-4fd1-a5a2-f29124a56433"
    }
]
}

```

제로 ETL 통합 생성

먼저 제로 ETL 통합을 생성하여 소스 데이터를 Amazon Redshift에 복제합니다.

데이터 소스에 따라 생성할 제로 ETL 통합 유형이 결정됩니다.

주제

- [Aurora에 대한 제로 ETL 통합 생성](#)
- [Amazon RDS에 대한 제로 ETL 통합 생성](#)
- [DynamoDB에 대한 제로 ETL 통합 생성](#)
- [애플리케이션과 제로 ETL 통합 만들기](#)

Aurora에 대한 제로 ETL 통합 생성

이 단계에서는 Amazon Redshift와의 Aurora 제로 ETL 통합을 생성합니다.

Redshift와의 Aurora 제로 ETL 통합을 만드는 방법

1. Amazon RDS 콘솔에서 Amazon Aurora 사용 설명서에 나와 있는 대로 [사용자 지정 DB 클러스터 파라미터 그룹을 생성](#)합니다.

2. Amazon RDS 콘솔에서 Amazon Aurora 사용 설명서에 나와 있는 대로 [소스 Amazon Aurora DB 클러스터를 생성](#)합니다.
3. Amazon Redshift 콘솔에서: [대상 Amazon Redshift 데이터 웨어하우스 생성 및 구성](#).
 - AWS CLI 또는 Amazon Redshift 콘솔에서: [데이터 웨어하우스에 대/소문자 구분 기능 사용 설정](#).
 - Amazon Redshift 콘솔에서: [Amazon Redshift 데이터 웨어하우스에 대한 권한 부여 구성](#).
4. Amazon RDS 콘솔에서 Amazon Aurora 사용 설명서에 나와 있는 대로 [제로 ETL 통합을 생성](#)합니다.
5. Amazon Redshift 콘솔 또는 쿼리 에디터 v2에서 [통합을 기반으로 Amazon Redshift 데이터베이스를 생성](#)합니다.

그런 다음 [복제된 데이터로 구체화된 뷰를 쿼리하고 생성](#)합니다.

Aurora 제로 ETL 통합 생성에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 [Amazon Redshift가 구성된 Amazon Aurora 제로 ETL 통합 생성](#)을 참조하세요.

Amazon RDS에 대한 제로 ETL 통합 생성

이 단계에서는 Amazon Redshift와의 Amazon RDS 제로 ETL 통합을 생성합니다. Redshift는 RDS for MySQL, RDS for PostgreSQL 및 RDS for Oracle과의 통합을 지원합니다.

Amazon Redshift와의 Amazon RDS 제로 ETL 통합을 생성하는 방법

1. Amazon RDS 콘솔에서 Amazon RDS 사용 설명서에 나와 있는 대로 [사용자 지정 DB 파라미터 그룹을 생성](#)합니다.
2. Amazon RDS 콘솔에서 Amazon RDS 사용 설명서에 나와 있는 대로 [소스 Amazon RDS 인스턴스를 생성](#)합니다.
3. Amazon Redshift 콘솔에서: [대상 Amazon Redshift 데이터 웨어하우스 생성 및 구성](#).
 - AWS CLI 또는 Amazon Redshift 콘솔에서: [데이터 웨어하우스에 대/소문자 구분 기능 사용 설정](#).
 - Amazon Redshift 콘솔에서: [Amazon Redshift 데이터 웨어하우스에 대한 권한 부여 구성](#).
4. Amazon RDS 콘솔에서 Amazon RDS 사용 설명서에 나와 있는 대로 [제로 ETL 통합을 생성](#)합니다.
5. Amazon Redshift 콘솔 또는 쿼리 에디터 v2에서 [통합을 기반으로 Amazon Redshift 데이터베이스를 생성](#)합니다.

그런 다음 [복제된 데이터로 구체화된 뷰를 쿼리하고 생성](#)합니다.

Amazon RDS 콘솔은 소스 데이터베이스와 대상 Amazon Redshift 데이터 웨어하우스를 지정하는 단계별 통합 생성 흐름을 제공합니다. 문제가 발생하는 경우 Amazon RDS 또는 Amazon Redshift 콘솔에서 수동으로 문제를 해결하는 대신 Amazon RDS에서 문제를 해결하도록 선택할 수 있습니다.

RDS 제로 ETL 통합을 생성하는 자세한 지침은 Amazon RDS 사용 설명서의 [Amazon Redshift가 구성된 Amazon RDS 제로 ETL 통합 생성](#)을 참조하세요.

특히 Amazon RDS for Oracle 제로 ETL 통합을 생성하는 방법에 대한 자세한 지침은 Oracle Database@AWS 사용 설명서의 [제로 ETL 통합 설정](#)을 참조하세요.

DynamoDB에 대한 제로 ETL 통합 생성

제로 ETL 통합을 생성하기 전에 [Amazon Redshift와 제로 ETL 통합을 사용할 때 고려할 사항](#)에 설명된 고려 사항 및 요구 사항을 검토하세요. 이 일반적인 흐름에 따라 DynamoDB에서 Amazon Redshift로의 제로 ETL 통합을 생성합니다.

제로 ETL 통합이 구성된 Amazon Redshift로 DynamoDB 데이터를 복제하려면

1. 로그인 자격 증명이 Amazon Redshift 및 DynamoDB와의 제로 ETL 통합으로 작업할 수 있는 권한을 허용하는지 확인합니다. IAM 정책 예제는 [DynamoDB 제로 ETL 통합을 사용하는 IAM 정책](#) 섹션을 참조하세요.
2. DynamoDB 콘솔에서 Amazon DynamoDB 개발자 안내서에 설명된 대로 시점 복구(PITR), 리소스 정책, 자격 증명 기반 정책 및 암호화 키 권한을 갖도록 [DynamoDB 테이블을 구성](#)합니다.
3. Amazon Redshift 콘솔에서: [대상 Amazon Redshift 데이터 웨어하우스 생성 및 구성](#).
 - AWS CLI 또는 Amazon Redshift 콘솔에서: [데이터 웨어하우스에 대/소문자 구분 기능 사용 설정](#).
 - Amazon Redshift 콘솔에서: [Amazon Redshift 데이터 웨어하우스에 대한 권한 부여 구성](#).
4. Amazon Redshift 콘솔에서 이 주제의 뒷부분에 설명된 대로 제로 ETL 통합을 생성합니다.
5. Amazon Redshift 콘솔에서 Amazon Redshift 데이터 웨어하우스에 대상 데이터베이스를 생성합니다. 자세한 내용은 [Amazon Redshift에서 대상 데이터베이스 생성](#) 섹션을 참조하세요.
6. Amazon Redshift 콘솔에서 Amazon Redshift 데이터 웨어하우스에서 복제된 데이터를 쿼리합니다. 자세한 내용은 [Amazon Redshift에서 복제된 데이터 쿼리](#) 섹션을 참조하세요.

이 단계에서는 Amazon Redshift와의 Amazon DynamoDB 제로 ETL 통합을 생성합니다.

Amazon Redshift console

Amazon Redshift 콘솔을 사용하여 Amazon Redshift와의 Amazon DynamoDB 제로 ETL 통합을 생성하려면

1. Amazon Redshift 콘솔에서 제로 ETL 통합을 선택합니다. 제로 ETL 통합 목록이 있는 창에서 제로 ETL 통합 생성, DynamoDB 통합 생성을 선택합니다.
2. 통합을 생성할 페이지에서 다음과 같이 통합에 대한 정보를 입력합니다.
 - 통합 이름 입력 - 통합을 참조하는 데 사용할 수 있는 고유한 이름입니다.
 - 설명 입력 - 소스에서 대상으로 복제할 데이터를 설명합니다.
 - DynamoDB 소스 테이블 선택 - DynamoDB 테이블 하나를 선택할 수 있습니다. 테이블에서 시점 복구(PITR)를 활성화해야 합니다. 테이블 크기가 최대 100테비바이트(TiB)인 테이블만 표시됩니다. 원본 DynamoDB 테이블은 암호화되어야 합니다. 소스에는 승인된 위탁자 및 통합 소스가 포함된 리소스 정책도 있어야 합니다. 이러한 정책이 올바르지 않은 경우 수정 옵션이 표시됩니다.
 - 대상 Amazon Redshift 데이터 웨어하우스 선택 - 데이터 웨어하우스는 Amazon Redshift 프로비저닝 클러스터 또는 Redshift Serverless 작업 그룹일 수 있습니다. 대상 Amazon Redshift가 동일한 계정에 있는 경우 대상을 선택할 수 있습니다. 대상이 다른 계정에 있는 경우 Redshift 데이터 웨어하우스 ARN을 지정합니다. 대상에는 승인된 위탁자 및 통합 소스가 있는 리소스 정책이 있어야 하며 `enable_case_sensitive_identifier` 파라미터가 `true`로 설정되어 있어야 합니다. 대상에 올바른 리소스 정책이 없고 대상이 동일한 계정에 있는 경우 수정 옵션을 선택하여 통합 생성 프로세스 중에 리소스 정책을 자동으로 적용할 수 있습니다. 대상이 다른 AWS 계정에 있는 경우 Amazon Redshift 웨어하우스에서 리소스 정책을 수동으로 적용해야 합니다. 대상 Amazon Redshift 데이터 웨어하우스에 `true`로 구성된 올바른 파라미터 그룹 옵션 `enable_case_sensitive_identifier`가 없는 경우 수정 옵션을 선택하여 통합 생성 프로세스 중에 이 파라미터 그룹을 자동으로 업데이트하고 웨어하우스를 재부팅할 수 있습니다.
 - 최대 50개의 태그 키와 선택적 값 입력 - 통합에 대한 추가 메타데이터를 제공합니다. 자세한 내용은 [Amazon Redshift에서 리소스 태그](#) 섹션을 참조하세요.
 - 암호화 옵션 - 통합을 암호화합니다. 자세한 내용은 [고객 관리형 키를 사용하여 DynamoDB 통합 암호화](#) 섹션을 참조하세요.

통합을 암호화할 때 추가 암호화 컨텍스트를 추가할 수도 있습니다. 자세한 내용은 [암호화 컨텍스트](#) 섹션을 참조하세요.
3. DynamoDB 통합 생성을 선택할 수 있는 검토 페이지가 표시됩니다.

4. 다양한 작업의 진행 상황을 보고 제로 ETL 통합을 생성할 수 있는 진행 상황 페이지가 표시됩니다.
5. 통합이 생성되고 활성화된 후 통합의 세부 정보 페이지에서 데이터베이스에 연결을 선택합니다. Amazon Redshift 데이터 웨어하우스가 처음 생성되면 데이터베이스도 생성되었습니다. 통합을 위한 다른 데이터베이스를 생성하려면 대상 데이터 웨어하우스의 모든 데이터베이스에 연결해야 합니다. 데이터베이스에 연결 페이지에서 최근 연결을 사용할 수 있는지 확인하고 인증 방법을 선택합니다. 인증 방법에 따라 대상의 기존 데이터베이스에 연결할 정보를 입력합니다. 이 인증 정보에는 기존 데이터베이스 이름(일반적으로 dev)과 Amazon Redshift 데이터 웨어하우스로 데이터베이스를 생성할 때 지정한 데이터베이스 사용자가 포함될 수 있습니다.
6. 데이터베이스에 연결한 후 통합에서 데이터베이스 생성을 선택하여 소스에서 데이터를 수신하는 데이터베이스를 생성합니다. 데이터베이스를 생성할 때 통합 ID, 데이터 웨어하우스 이름 및 데이터베이스 이름을 제공합니다.
7. 통합 상태 및 대상 데이터베이스가 Active이면 데이터는 DynamoDB 테이블에서 대상 테이블로 복제되기 시작합니다. 소스에 데이터를 추가하면 대상 Amazon Redshift 데이터 웨어하우스에 자동으로 복제됩니다.

AWS CLI

AWS CLI를 사용하여 Amazon Redshift와의 Amazon DynamoDB 제로 ETL 통합을 생성하려면 다음 옵션과 함께 `create-integration` 명령을 사용합니다.

- `integration-name` - 통합 이름을 지정합니다.
- `source-arn` - DynamoDB 소스의 ARN을 지정합니다.
- `target-arn` - Amazon Redshift 프로비저닝 클러스터 또는 Redshift Serverless 작업 그룹 대상의 네임스페이스 ARN을 지정합니다.

다음 예제에서는 통합 이름, 소스 ARN 및 대상 ARN을 제공하여 통합을 생성합니다. 통합은 암호화되지 않습니다.

```
aws redshift create-integration \
--integration-name ddb-integration \
--source-arn arn:aws:dynamodb:us-east-1:123456789012:table/books \
--target-arn arn:aws:redshift:us-east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222

{
  "Status": "creating",
```

```

    "IntegrationArn": "arn:aws:redshift:us-east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Errors": [],
    "ResponseMetadata": {
      "RetryAttempts": 0,
      "HTTPStatusCode": 200,
      "RequestId": "132cbe27-fd10-4f0a-aacb-b68f10bb2bfb",
      "HTTPHeaders": {
        "x-amzn-requestid": "132cbe27-fd10-4f0a-aacb-b68f10bb2bfb",
        "date": "Sat, 24 Aug 2024 05:44:08 GMT",
        "content-length": "934",
        "content-type": "text/xml"
      }
    },
    "Tags": [],
    "CreateTime": "2024-08-24T05:44:08.573Z",
    "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "AdditionalEncryptionContext": {},
    "TargetArn": "arn:aws:redshift:us-east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "IntegrationName": "ddb-integration",
    "SourceArn": "arn:aws:dynamodb:us-east-1:123456789012:table/books"
  }
}

```

다음 예제에서는 암호화를 위해 고객 관리형 키를 사용하여 통합을 생성합니다. 통합을 생성하기 전에:

- 소스 DynamoDB 테이블에서 동일한 계정(예제에서는 'AccountA')에 고객 관리형 키(예제에서는 'CMCMK'라고 함)를 생성합니다.
- 사용자/역할(예제에서 "RoleA"라고 함)이 통합을 생성하는 데 사용되고 이 KMS 키에 대한 kms:CreateGrant 및 kms:DescribeKey 권한이 있는지 확인합니다.
- 다음을 키 정책에 추가합니다.

```

{
  "Sid": "Enable RoleA to create grants with key",
  "Effect": "Allow",
  "Principal": {
    "AWS": "RoleA-ARN"
  },
  "Action": "kms:CreateGrant",

```

```

    "Resource": "*",
    "Condition": {
        // Add "StringEquals" condition if you plan to provide additional encryption
context
        // for the zero-ETL integration. Ensure that the key-value pairs added here
match
        // the key-value pair you plan to use while creating the integration.
        // Remove this if you don't plan to use additional encryption context
        "StringEquals": {
            "kms:EncryptionContext:context-key1": "context-value1"
        },
        "ForAllValues:StringEquals": {
            "kms:GrantOperations": [
                "Decrypt",
                "GenerateDataKey",
                "CreateGrant"
            ]
        }
    }
},
{
    "Sid": "Enable RoleA to describe key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "RoleA-ARN"
    },
    "Action": "kms:DescribeKey",
    "Resource": "*"
},
{
    "Sid": "Allow use by RS SP",
    "Effect": "Allow",
    "Principal": {
        "Service": "redshift.amazonaws.com"
    },
    "Action": "kms:CreateGrant",
    "Resource": "*"
}

```

```

aws redshift create-integration \
--integration-name ddb-integration \
--source-arn arn:aws:dynamodb:us-east-1:123456789012:table/books \

```

```
--target-arn arn:aws:redshift:us-east-1:123456789012:namespace:a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222 \
--kms-key-id arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE33333 \
--additional-encryption-context key33=value33 // This matches the condition in the
key policy.
{
  "IntegrationArn": "arn:aws:redshift:us-
east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "IntegrationName": "ddb-integration",
  "SourceArn": "arn:aws:dynamodb:us-east-1:123456789012:table/books",
  "SourceType": "dynamodb",
  "TargetArn": "arn:aws:redshift:us-
east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Status": "creating",
  "Errors": [],
  "CreateTime": "2024-10-02T18:29:26.710Z",
  "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE33333",
  "AdditionalEncryptionContext": {
    "key33": "value33"
  },
  "Tags": []
}
```

DynamoDB 제로 ETL 통합을 사용하는 IAM 정책

제로 ETL 통합을 생성할 때 로그인 자격 증명은 DynamoDB 및 Amazon Redshift 작업과 통합의 소스 및 대상으로 관련된 리소스에 대한 권한이 있어야 합니다. 다음은 필요한 최소 권한을 보여주는 예제입니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:ListTables"
      ]
    }
  ]
}
```



```

    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "dynamodb:GetResourcePolicy",
      "dynamodb:PutResourcePolicy",
      "dynamodb:UpdateContinuousBackups"
    ],
    "Resource": [
      "arn:aws:dynamodb:us-east-1:111122223333:table/my-ddb-table"
    ]
  },
  {
    "Sid": "AllowRedshiftDescribeIntegration",
    "Effect": "Allow",
    "Action": [
      "redshift:DescribeIntegrations"
    ],
    "Resource": "*"
  },
  {
    "Sid": "AllowRedshiftCreateIntegration",
    "Effect": "Allow",
    "Action": "redshift:CreateIntegration",
    "Resource": "arn:aws:redshift:us-east-1:111122223333:integration:*"
  },
  {
    "Sid": "AllowRedshiftModifyDeleteIntegration",
    "Effect": "Allow",
    "Action": [
      "redshift:ModifyIntegration",
      "redshift>DeleteIntegration"
    ],
    "Resource": "arn:aws:redshift:us-east-1:111122223333:integration:<uuid>"
  },
  {
    "Sid": "AllowRedshiftCreateInboundIntegration",
    "Effect": "Allow",
    "Action": "redshift:CreateInboundIntegration",
    "Resource": "arn:aws:redshift:us-east-1:111122223333:namespace:<uuid>"
  }

```

```
}
]
}
```

고객 관리형 키를 사용하여 DynamoDB 통합 암호화

DynamoDB 제로 ETL 통합을 생성할 때 AWS 소유 키가 아닌 사용자 지정 KMS 키를 지정하는 경우 키 정책은 Amazon Redshift 서비스 위탁자에 CreateGrant 작업에 대한 액세스 권한을 제공해야 합니다. 또한, 요청자 계정 또는 역할 권한이 DescribeKey 및 CreateGrant 작업을 수행할 수 있도록 허용해야 합니다.

다음 샘플 키 정책 문은 정책에 필요한 권한을 보여줍니다. 일부 예에는 권한 범위를 더 축소하기 위한 컨텍스트 키가 포함됩니다.

키 정책 문 예제

다음 정책 문을 사용하면 요청자 계정 또는 역할이 KMS 키에 대한 정보를 검색할 수 있습니다.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::{account-ID}:role/{role-name}"
  },
  "Action": "kms:DescribeKey",
  "Resource": "*"
}
```

다음 정책 문에서는 요청자 계정 또는 역할이 KMS 키에 권한 부여를 추가할 수 있도록 허용합니다.

[kms:ViaService](#) 조건 키는 Amazon Redshift로부터의 요청으로 KMS 키의 사용을 제한합니다.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::{account-ID}:role/{role-name}"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:{context-key}": "{context-value}",

```

```

    "kms:ViaService":"redshift.{region}.amazonaws.com"
  },
  "ForAllValues:StringEquals":{
    "kms:GrantOperations":[
      "Decrypt",
      "GenerateDataKey",
      "CreateGrant"
    ]
  }
}

```

다음 정책 문은 Amazon Redshift 서비스 보안 주체가 KMS 키에 권한을 추가하는 것을 허용합니다.

```

{
  "Effect":"Allow",
  "Principal":{
    "Service":"redshift.amazonaws.com"
  },
  "Action":"kms:CreateGrant",
  "Resource":"*",
  "Condition":{
    "StringEquals":{
      "kms:EncryptionContext:{context-key}":"{context-value}",
      "aws:SourceAccount":"{account-ID}"
    },
    "ForAllValues:StringEquals":{
      "kms:GrantOperations":[
        "Decrypt",
        "GenerateDataKey",
        "CreateGrant"
      ]
    },
    "ArnLike":{
      "aws:SourceArn":"arn:aws:*:{region}:{account-ID}:integration:*"
    }
  }
}

```

자세한 내용은 AWS Key Management Service 개발자 안내서의 [키 정책 생성](#)을 참조하세요.

암호화 컨텍스트

제로 ETL 통합을 암호화할 때 키-값 페어를 추가 암호화 컨텍스트로 추가할 수 있습니다. 이러한 키-값 페어를 추가하여 복제되는 데이터에 대한 추가 컨텍스트 정보를 추가할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 가이드에서 [암호화 컨텍스트](#)를 참조하세요.

Amazon Redshift는 추가한 모든 항목 외에도 다음과 같은 암호화 컨텍스트 쌍을 추가합니다.

- `aws:redshift:integration:arn` - IntegrationArn
- `aws:servicename:id` - Redshift

이렇게 하면 추가할 수 있는 전체 쌍 수가 8개에서 6개로 줄어들고 권한 부여 제약 조건의 전체 문자 제한이 늘어납니다. 자세한 내용은 AWS Key Management Service 개발자 안내서에서 [권한 부여 제약 사용](#)을 참조하세요.

애플리케이션과 제로 ETL 통합 만들기

이 단계에서는 애플리케이션과 Amazon Redshift와의 제로 ETL 통합을 만듭니다.

애플리케이션과 Amazon Redshift와의 제로 ETL 통합을 만드는 방법

1. Amazon Redshift 콘솔에서: [대상 Amazon Redshift 데이터 웨어하우스 생성 및 구성](#).
 - AWS CLI 또는 Amazon Redshift 콘솔에서: [데이터 웨어하우스에 대/소문자 구분 기능 사용 설정](#).
 - Amazon Redshift 콘솔에서: [Amazon Redshift 데이터 웨어하우스에 대한 권한 부여 구성](#).
2. AWS Glue 콘솔에서: AWS Glue 개발자 안내서에 설명된 대로 [통합을 만듭니다](#).
3. 대상 데이터베이스가 만들어지고 데이터가 복제되기 시작하면 복제된 데이터에 대해 구체화된 데이터를 쿼리하고 만들 수 있습니다. 자세한 내용은 [Amazon Redshift에서 복제된 데이터 쿼리](#) 섹션을 참조하세요.

애플리케이션과 제로 ETL 통합을 만드는 방법에 대한 자세한 내용은 AWS Glue 개발자 안내서의 [제로 ETL 통합](#)을 참조하세요.

Amazon Redshift에서 대상 데이터베이스 생성

소스에서 Amazon Redshift로 데이터를 복제하려면 Amazon Redshift의 통합에서 데이터베이스를 생성해야 합니다.

대상 Redshift Serverless 작업 그룹 또는 프로비저닝된 클러스터에 연결하고 통합 식별자에 대한 참조가 있는 데이터베이스를 생성합니다. 이 식별자는 [SVV_INTEGRATION](#) 뷰를 쿼리할 때 `integration_id`에 대해 반환되는 값입니다.

Important

통합에서 데이터베이스를 생성하기 전에 Amazon Redshift 콘솔에서 제로 ETL 통합을 생성하여 통합이 Active 상태여야 합니다.

소스에서 Amazon Redshift로 데이터 복제를 시작하려면 Amazon Redshift의 통합에서 데이터베이스를 생성합니다. 데이터베이스를 생성하는 방법은 Amazon Redshift 콘솔 또는 쿼리 에디터 v2를 사용할 수 있습니다.

Amazon Redshift console

1. 왼쪽 탐색 창에서 제로 ETL 통합을 선택합니다.
2. 통합 목록에서 통합을 선택합니다.
3. 프로비저닝된 클러스터를 사용하는 경우 먼저 데이터베이스에 연결해야 합니다. [데이터베이스에 연결(Connect to database)]을 선택합니다. 최근 연결을 사용하거나 새 연결을 만들어 연결할 수 있습니다.
4. 통합에서 데이터베이스를 만들려면 통합에서 데이터베이스 생성을 선택합니다.
5. 대상 데이터베이스 이름을 입력합니다. 통합 ID와 데이터 웨어하우스 이름이 미리 입력되어 있습니다.

Aurora PostgreSQL, RDS for PostgreSQL 또는 RDS for Oracle 소스의 경우 제로 ETL 통합을 생성할 때 지정한 명명된 소스 데이터베이스도 입력하세요. 이러한 경우 최대 100개의 소스 데이터베이스를 Amazon Redshift 데이터베이스에 매핑할 수 있습니다.

6. 데이터베이스 생성을 선택합니다.

Amazon Redshift query editor v2

1. Amazon Redshift 콘솔로 이동하여 쿼리 편집기 v2를 선택합니다.
2. 왼쪽 패널에서 Amazon Redshift Serverless 작업 그룹 또는 Amazon Redshift 프로비저닝된 클러스터를 선택하고 연결합니다.
3. 통합 ID를 가져오려면 Amazon Redshift 콘솔에서 통합 목록으로 이동합니다.

또는 다음 명령을 실행하여 `integration_id` 값을 가져옵니다.

```
SELECT integration_id FROM SVV_INTEGRATION;
```

4. 그리고 나서 다음 명령을 실행하여 데이터베이스를 생성합니다. 통합 ID를 지정하면 데이터베이스와 소스 간의 연결이 생성됩니다.

`integration_id`를 이전 명령에서 반환한 값으로 대체합니다.

```
CREATE DATABASE destination_db_name FROM INTEGRATION 'integration_id';
```

Aurora PostgreSQL 소스의 경우 통합을 생성할 때 지정한 클러스터 내의 명명된 데이터베이스에 대한 참조도 포함해야 합니다. 예제:

```
CREATE DATABASE "destination_db_name" FROM INTEGRATION 'integration_id'  
DATABASE "named_db";
```

제로 ETL 통합 대상의 데이터베이스 생성에 관한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [CREATE DATABASE](#)를 참조하시기 바랍니다. ALTER DATABASE를 사용하여 REFRESH INTERVAL 등의 데이터베이스 파라미터를 변경할 수 있습니다. 제로 ETL 통합 대상의 데이터베이스 변경에 관한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [ALTER DATABASE](#)를 참조하시기 바랍니다.

Note

통합에서 생성한 데이터베이스의 데이터는 통합 소스만 업데이트할 수 있습니다. 테이블의 스키마를 변경하려면 소스의 테이블에 대해 DDL 또는 DML 명령을 실행하세요. 소스의 테이블에 대해서는 DDL 및 DML 명령을 실행할 수 있지만 대상 데이터베이스에서는 DDL 명령 및 읽기 전용 쿼리만 실행할 수 있습니다.

대상 데이터베이스의 상태를 보는 방법에 대한 자세한 정보는 [제로 ETL 통합 보기](#) 섹션을 참조하세요.

대상 데이터베이스를 생성한 후 소스에 데이터를 추가할 수 있습니다. 소스에 데이터를 추가하려면 다음 주제 중 하나를 참조하세요.

- Aurora 소스의 경우 Amazon Aurora 사용 설명서의 [소스 DB 클러스터에 데이터 추가](#)를 참조하세요.

- Amazon RDS 소스의 경우 Amazon RDS 사용 설명서의 [소스 DB 인스턴스에 데이터 추가](#)를 참조하세요.
- DynamoDB 소스는 [Amazon DynamoDB 개발자 안내서](#)의 DynamoDB 시작하기를 참조하세요.
- 애플리케이션 소스와의 제로 ETL 통합은 AWS Glue 개발자 안내서의 [제로 ETL 통합](#)을 참조하세요.

Amazon Redshift에서 복제된 데이터 쿼리

소스에 데이터를 추가하면 거의 실시간으로 Amazon Redshift 데이터 웨어하우스에 데이터가 복제되어 바로 쿼리할 수 있습니다. 통합 지표 및 테이블 통계에 대한 자세한 내용은 [제로 ETL 통합의 지표](#) 섹션을 참조하세요.

Note

데이터베이스는 MySQL의 스키마와 동일하므로, MySQL 데이터베이스 수준은 Amazon Redshift 스키마 수준에 매핑됩니다. Aurora MySQL 또는 RDS for MySQL에서 복제된 데이터를 쿼리할 때는 이 매핑 차이에 유의하세요.

복제된 데이터를 쿼리하려면

1. Amazon Redshift 콘솔로 이동하여 쿼리 편집기 v2를 선택합니다.
2. Amazon Redshift Serverless 작업 그룹 또는 Amazon Redshift 프로비저닝된 클러스터에 연결하고 드롭다운 목록에서 데이터베이스를 선택합니다.
3. SELECT 문을 사용하여 소스에서 생성한 스키마 및 테이블에서 복제된 데이터를 모두 선택합니다. 대/소문자를 구분하려면 스키마, 테이블 및 열 이름에 큰따옴표(" ")를 사용합니다. 예제:

```
SELECT * FROM "schema_name"."table_name";
```

Amazon Redshift Data API를 사용하여 데이터를 쿼리할 수도 있습니다.

구체화된 뷰로 복제된 데이터 쿼리

로컬 Amazon Redshift 데이터베이스에서 구체화된 뷰를 생성하여 제로 ETL 통합을 통해 복제된 데이터를 변환할 수 있습니다. 로컬 데이터베이스에 연결하고 데이터베이스 간 쿼리를 사용하여 대상 데이터베이스에 액세스합니다. 세 부분으로 구성된 표기법(대상-데이터베이스-이름.스키마-이름.테이블-이름)으로 정규화된 객체 이름을 사용하거나, 대상 데이터베이스-스키마 쌍을 참조하는 외부 스키마를 만

다음 두 부분으로 구성된 표기법(외부-스키마-이름.테이블-이름)을 사용할 수 있습니다. 데이터베이스 간 쿼리에 대한 자세한 내용은 [데이터베이스 간 데이터 쿼리](#)를 참조하세요.

다음 예시를 사용하여 소스 *ticket_zetl*에서 샘플 데이터를 만들어 *sales_zetl* 및 *event_zetl* 테이블에 삽입할 수 있습니다. 테이블은 Amazon Redshift 데이터베이스 *zetl_int_db*에 복제됩니다.

```
CREATE TABLE sales_zetl (
    salesid integer NOT NULL primary key,
    eventid integer NOT NULL,
    pricepaid decimal(8, 2)
);

CREATE TABLE event_zetl (
    eventid integer NOT NULL PRIMARY KEY,
    eventname varchar(200)
);

INSERT INTO sales_zetl VALUES(1, 1, 3.33);
INSERT INTO sales_zetl VALUES(2, 2, 4.44);
INSERT INTO sales_zetl VALUES(3, 2, 5.55);

INSERT INTO event_zetl VALUES(1, "Event 1");
INSERT INTO event_zetl VALUES(2, "Event 2");
```

다음 세 부분으로 구성된 표기법을 사용하여 이벤트당 총 매출을 구하는 구체화된 뷰를 생성할 수 있습니다.

```
--three part notation zetl-database-name.schema-name.table-name
CREATE MATERIALIZED VIEW mv_transformed_sales_per_event_3p
AUTO REFRESH YES
AS
(SELECT eventname, sum(pricepaid) as total_price
FROM zetl_int_db.ticket_zetl.sales_zetl S, zetl_int_db.ticket_zetl.event_zetl E
WHERE S.eventid = E.eventid
GROUP BY 1);
```

다음 두 부분으로 구성된 표기법을 사용하여 이벤트당 총 매출을 구하는 구체화된 뷰를 생성할 수 있습니다.

```
--two part notation external-schema-name.table-name notation
CREATE EXTERNAL schema ext_ticket_zetl
FROM REDSHIFT
```



```

DATABASE zetl_int_db
SCHEMA tickit_zetl;

CREATE MATERIALIZED VIEW mv_transformed_sales_per_event_2p
AUTO REFRESH YES
AS
(
    SELECT eventname, sum(pricepaid) as total_price
    FROM   ext_tickit_zetl.sales_zetl S, ext_tickit_zetl.event_zetl E
    WHERE S.eventid = E.eventid
    GROUP BY 1
);

```

생성한 구체화된 뷰를 보려면 다음 예시를 사용하세요.

```
SELECT * FROM mv_transformed_sales_per_event_3p;
```

```

+-----+-----+
| eventname | total_price |
+-----+-----+
| Event 1   | 3.33        |
| Event 2   | 9.99        |
+-----+-----+

```

```
SELECT * FROM mv_transformed_sales_per_event_2p;
```

```

+-----+-----+
| eventname | total_price |
+-----+-----+
| Event 1   | 3.33        |
| Event 2   | 9.99        |
+-----+-----+

```

DynamoDB에서 복제된 데이터 쿼리

Amazon DynamoDB에서 Amazon Redshift 데이터베이스로 데이터를 복제하면 SUPER 데이터 유형의 열에 구체화된 뷰에 저장됩니다.

이 예제에서는 다음 데이터가 DynamoDB에 저장됩니다.

```

{
  "key1": {

```

```

        "S": "key_1"
    },
    "key2": {
        "N": 0
    },
    "payload": {
        "L": [
            {
                "S": "sale1"
            },
            {
                "S": "sale2"
            },
        ]
    },
}

```

Amazon Redshift 구체화된 뷰는 다음과 같이 정의됩니다.

```

CREATE MATERIALIZED VIEW mv_sales
    BACKUP NO
    AUTO REFRESH YES
    AS
    SELECT "value"."payload"."L"[0]."S"::VARCHAR AS first_payload
    FROM public.sales;

```

구체화된 뷰에서 데이터를 보려면 SQL 명령을 실행합니다.

```

SELECT first_payload FROM mv_sales;

```

제로 ETL 통합 보기

Amazon Redshift 콘솔에서 제로 ETL 통합을 확인할 수 있습니다. 여기에서 구성 정보 및 현재 상태를 확인하고 화면을 열어 데이터를 쿼리하고 공유할 수 있습니다.

Amazon Redshift console

제로 ETL 통합 세부 정보를 보려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.

2. 왼쪽 탐색 창에서 서버리스 또는 프로비저닝된 클러스터 대시보드 중 하나를 선택합니다. 그런 다음 제로 ETL 통합을 선택합니다.
3. 보려는 제로 ETL 통합을 선택합니다. 각 통합에 대해 다음 필수 정보가 제공됩니다.
 - 통합 ID는 통합이 생성될 때 반환되는 식별자입니다.
 - 상태는 다음 값 중 하나일 수 있습니다.
 - Active - 제로 ETL 통합이 트랜잭션 데이터를 대상 Amazon Redshift 데이터 웨어하우스로 전송하고 있습니다.
 - Syncing - 제로 ETL 통합에 복구 가능한 오류가 발생하여 데이터를 다시 시드하고 있습니다. 영향을 받는 테이블은 재동기화가 완료될 때까지 Amazon Redshift에서 쿼리할 수 없습니다.
 - Failed - 제로 ETL 통합에서 복구할 수 없는 이벤트 또는 수정할 수 없는 오류가 발생했습니다. 제로 ETL 통합을 삭제하고 다시 만들어야 합니다.
 - Creating - 제로 ETL 통합이 생성 중입니다.
 - Deleting - 제로 ETL 통합이 삭제 중입니다.
 - Needs attention - 제로 ETL 통합에 수동 개입이 필요한 이벤트 또는 오류가 발생하여 이를 해결해야 합니다. 문제를 해결하려면 오류 메시지의 단계를 따르세요.
 - 소스 유형은 대상에 복제되는 소스 데이터의 유형입니다. 유형은 다른 데이터베이스 관리자를 지정할 수 있습니다(예: Aurora MySQL-Compatible Edition, Amazon Aurora PostgreSQL, RDS for MySQL 및 애플리케이션에서(GlueSAAS)).
 - 소스 ARN은 소스 데이터의 ARN입니다. 대부분의 소스의 경우 소스 데이터베이스 또는 테이블의 ARN입니다. 애플리케이션 소스와의 제로 ETL 통합의 경우 AWS Glue 연결 객체의 ARN입니다.
 - 대상은 소스 데이터를 수신하는 Amazon Redshift 데이터 웨어하우스의 네임스페이스입니다.
 - 데이터베이스는 다음 중 하나일 수 있습니다.
 - No database - 통합을 위한 대상 데이터베이스가 없습니다.
 - Creating - Amazon Redshift가 통합을 위한 대상 데이터베이스를 만들고 있습니다.
 - Active - 데이터가 통합 소스에서 Amazon Redshift로 복제되고 있습니다.
 - Error - 통합에 오류가 있습니다.
 - Recovering - 데이터 웨어하우스가 다시 시작된 후 통합이 복구 중입니다.
 - Resyncing - Amazon Redshift가 통합에서 테이블을 다시 동기화하는 중입니다.

- 생성 날짜는 통합이 생성된 날짜 및 시간(UTC)입니다.

Note

데이터 웨어하우스의 통합 세부 정보를 보려면 프로비저닝된 클러스터 또는 서버리스 네임 스페이스의 세부 정보 페이지를 선택한 다음 제로 ETL 통합 탭을 선택합니다.

제로 ETL 통합 목록에서 쿼리 데이터를 선택하여 Amazon Redshift 쿼리 에디터 v2로 이동할 수 있습니다. Amazon Redshift 대상 데이터베이스에는 [enable_case_sensitive_identifier](#) 파라미터가 활성화되어 있습니다. SQL을 작성할 때 스키마, 테이블 및 열 이름을 큰따옴표("<이름>")로 묶어야 할 수 있습니다. Amazon Redshift 데이터 웨어하우스의 데이터를 쿼리하는 방법에 대한 자세한 내용은 [Query Editor V2를 사용하여 데이터베이스 쿼리](#) 섹션을 참조하세요.

제로 ETL 통합 목록에서 데이터 공유를 선택하여 데이터 공유를 생성할 수 있습니다. Amazon Redshift 데이터베이스에 데이터 공유를 생성하려면 데이터 공유 생성 페이지의 지침을 따르세요. Amazon Redshift 데이터베이스에서 데이터를 공유하려면 먼저 대상 데이터베이스를 만들어야 합니다. 데이터 공유에 대한 자세한 내용은 [Amazon Redshift의 데이터 공유 개념](#)을 참조하세요.

통합을 새로 고치려면 [ALTER DATABASE](#) 명령을 사용할 수 있습니다. 이렇게 하면 통합 소스의 모든 데이터가 대상 데이터베이스로 복제됩니다. 다음 예시는 제로 ETL 통합 내에서 동기화된 테이블과 실패한 테이블을 모두 새로 고칩니다.

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL tables;
```

AWS CLI

AWS CLI를 사용하여 Amazon Redshift와의 Amazon DynamoDB 제로 ETL 통합을 설명하려면 다음 옵션과 함께 `describe-integrations` 명령을 사용합니다.

- `integration-arn` – 설명할 DynamoDB 통합의 ARN을 지정합니다.
- `integration-name` - 반환할 하나 이상의 리소스를 지정하는 선택적 필터를 지정합니다.

다음 예제에서는 통합 ARN을 제공하여 통합을 설명합니다.

```
aws redshift describe-integrations
```

```
{
```

```

    "Integrations": [
      {
        "Status": "failed",
        "IntegrationArn": "arn:aws:redshift:us-east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Errors": [
          {
            "ErrorCode": "INVALID_TABLE_PERMISSIONS",
            "ErrorMessage": "Redshift does not have sufficient access on the table key. Refer to the Amazon DynamoDB Developer Guide."
          }
        ],
        "Tags": [],
        "CreateTime": "2023-11-09T00:32:46.444Z",
        "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "TargetArn": "arn:aws:redshift:us-east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "IntegrationName": "ddb-to-provisioned-02",
        "SourceArn": "arn:aws:dynamodb:us-east-1:123456789012:table/mytable"
      }
    ]
  }
}

```

integration-arn, source-arn, source-types 또는 describe-integrations 기준으로 결과를 필터링할 수도 있습니다. 자세한 내용은 Amazon Redshift CLI 설명서의 [describe-integrations](#)를 참조하세요.

기록 모드

기록 모드를 사용하면 Amazon Redshift에서 직접 소스 테이블에 있는 레코드의 모든 버전(업데이트 및 삭제 포함)을 추적하도록 제로 ETL 통합을 구성할 수 있습니다. 기록 분석 실행, 룩백 보고서 작성, 추세 분석 수행, Amazon Redshift를 기반으로 구축된 다운스트림 애플리케이션에 중분 업데이트 전송을 비롯하여 모든 데이터에 대해 고급 분석을 실행할 수 있습니다. 기록 모드는 Amazon Aurora MySQL, Amazon Aurora PostgreSQL, Amazon RDS for MySQL, Amazon DynamoDB를 비롯한 여러 Amazon Redshift 제로 ETL 통합에서 지원됩니다. 기록 모드는 Salesforce, SAP, ServiceNow, Zendesk와 같은 여러 애플리케이션에서도 지원됩니다.

Amazon Redshift 콘솔(<https://console.aws.amazon.com/redshiftv2/>)에서 제로 ETL 통합의 기록 모드를 켜거나 끌 수 있습니다. 기록 모드를 사용하여 통합 소스에서 삭제되거나 수정된 레코드를 추적합니다.

다. 추적은 대상 Amazon Redshift 데이터 웨어하우스에서 수행됩니다. 기록 모드를 켜도 이러한 테이블에 대한 일반 분석 쿼리의 성능에는 영향을 주지 않습니다.

기록 모드를 켜면 소스 내에서 삭제한 테이블이 Amazon Redshift에서 삭제되지 않습니다. 대신 테이블이 DroppedSource 상태로 표시되며 여전히 이러한 테이블을 쿼리할 수 있습니다. 일반 SQL에서도 DROP 및 RENAME 명령을 계속 사용할 수 있습니다.

소스에서 동일한 테이블 이름을 재사용하려면 해당 DroppedState 테이블을 DROP 또는 RENAME 해야 Amazon Redshift에 복제할 수 있습니다. 소스에서 테이블을 생성하기 전에 이 작업을 수행해야 합니다.

기록 모드를 사용할 때 고려해야 할 사항에 대한 자세한 내용은 [대상에서 기록 모드 사용 시 고려 사항](#) 섹션을 참조하세요.

제로 ETL 통합의 기록 모드를 관리하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 서버리스 또는 프로비저닝된 클러스터 대시보드 중 하나를 선택합니다. 그런 다음 제로 ETL 통합을 선택합니다.
3. 관리하려는 제로 ETL 통합을 선택하고 기록 모드 관리를 선택합니다. 기록 모드 관리 창이 표시됩니다.
4. Amazon DynamoDB와 같은 단일 소스 테이블이 있는 소스 유형에서 복제된 대상 테이블에 대해 기록 모드 끄기 또는 켜기가 가능합니다. 제로 ETL 통합에 사용 가능한 대상 테이블이 여러 개 있는 경우 기존 및 향후 테이블 모두에 대해 끄기, 기존 및 향후 테이블 모두에 대해 켜기 또는 개별 테이블의 기록 모드 관리를 선택할 수 있습니다. 제로 ETL 통합이 생성될 때 기본값은 기록 모드 off입니다.

기록 모드를 on으로 설정하면 소스의 변경 사항을 추적하기 위해 다음 열이 대상 테이블에 추가됩니다. Amazon Redshift는 대상 테이블의 레코드를 삭제하지 않으므로 기록 모드가 on이면 월별 사용량과 비용이 증가합니다. 소스 레코드가 삭제되거나 변경되면 대상에 새 레코드가 생성되어 여러 레코드 버전과 함께 대상의 총 행 수가 늘어납니다. 레코드를 소스에서 삭제하거나 수정할 때 대상 테이블에서 레코드가 삭제되지 않습니다. 비활성 레코드를 삭제하여 대상 테이블을 관리할 수 있습니다.

열 이름	데이터 유형	설명
_record_is_active	부울	대상의 레코드가 현재 소스에서 활성 상태인지 여부를 나타냅니다. True는 레코드가 활성 상태임을 나타냅니다.
_record_create_time	타임스탬프	소스 레코드가 활성 상태가 된 시작 시간(UTC)입니다.
_record_delete_time	타임스탬프	소스 레코드가 업데이트되거나 삭제된 종료 시간(UTC)입니다.

_record_is_active 열이 false인 레코드를 필터링하여 기록 모드 테이블에서 비활성 레코드를 삭제할 수 있습니다. 다음 SQL DELETE 명령은 id 열이 100 이하인 테이블에서 비활성 레코드를 삭제합니다. 레코드를 삭제한 후 자동 vacuum record가 실행되면 삭제된 레코드의 스토리지가 회수됩니다.

```
DELETE FROM myschema.mytable where not _record_is_active AND id <= 100;
```

기록 모드가 off가 되면 Amazon Redshift는 활성 레코드가 있고 기록 열이 추가되지 않은 상태로 대상 데이터베이스에 테이블 복사본을 생성합니다. Amazon Redshift는 사용자가 사용할 수 있도록 테이블의 이름을 *table-name_historical_timestamp*로 바꿉니다. 더 이상 필요하지 않은 경우 이 테이블 복사본을 삭제할 수 있습니다. ALTER TABLE 명령을 사용하여 이러한 테이블의 이름을 바꿀 수 있습니다. 예제:

```
ALTER TABLE [schema-name.]table-name_historical_timestamp RENAME TO new_table_name;
```

자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [ALTER TABLE](#)을 참조하세요.

SQL 명령 CREATE DATABASE 및 ALTER DATABASE를 사용하여 기록 모드를 관리할 수도 있습니다. HISTORY_MODE 설정 방법에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [CREATE DATABASE](#) 및 [ALTER DATABASE](#)를 참조하세요.

Amazon Redshift에서 데이터 공유

소스에 데이터를 추가하면 즉시 Amazon Redshift에 복제되고 데이터 공유를 생성하여 공유할 준비가 됩니다.

데이터를 공유하려면 대상 데이터베이스를 만들어야 합니다.

Amazon Redshift 콘솔을 사용하여 Amazon Redshift Serverless에서 데이터를 공유하려면

1. Amazon Redshift 콘솔의 왼쪽 탐색 창에서 Amazon Redshift Serverless > Serverless 대시보드를 선택합니다.
2. 왼쪽 탐색 창에서 제로 ETL 통합을 선택합니다.
3. 데이터 공유(Share data)를 선택합니다.
4. 데이터 공유 생성 페이지에서 [데이터 공유 생성](#)의 단계를 따르세요.

Amazon Redshift 콘솔을 사용하여 Amazon Redshift 프로비저닝된 클러스터에서 데이터를 공유하려면

1. Amazon Redshift 콘솔의 왼쪽 탐색 창에서 프로비저닝된 클러스터 대시보드를 선택합니다.
2. 왼쪽 탐색 창에서 제로 ETL 통합을 선택합니다.
3. 통합 목록에서 통합을 선택합니다.
4. 통합 세부 정보 페이지에서 데이터베이스에 연결을 선택합니다.
5. 데이터베이스에 연결 페이지에서 새 연결을 만들거나 최근 연결을 사용할 수 있습니다. 대상 데이터베이스에 연결이 되어 있는지 확인합니다.
6. 새 연결을 만드는 경우 데이터베이스의 데이터베이스 이름을 입력합니다. 그런 다음 연결을 클릭합니다.
7. 통합 세부 정보 페이지에서 데이터 공유를 선택합니다.
8. 데이터 공유 생성 페이지에서 [데이터 공유 생성](#)의 단계를 따르세요.

제로 ETL 통합 모니터링

시스템 뷰를 쿼리하거나 Amazon EventBridge를 사용하여 제로 ETL 통합을 모니터링할 수 있습니다.

Amazon Redshift 시스템 뷰로 제로 ETL 통합 모니터링

Amazon Redshift에서 다음 시스템 뷰를 쿼리하여 제로 ETL 통합을 모니터링할 수 있습니다.

- [SVV_INTEGRATION](#)은 제로 ETL 통합의 구성 세부 정보에 대한 정보를 제공합니다.
- [SYS_INTEGRATION_ACTIVITY](#)는 완료된 제로 ETL 통합 실행에 대한 정보를 제공합니다.
- [SVV_INTEGRATION_TABLE_MAPPING](#)은 소스에서 대상으로 메타데이터 값 매핑에 대한 정보를 제공합니다.
- [SVV_INTEGRATION_TABLE_STATE](#)는 통합 상태에 대한 정보를 제공합니다.
- [SYS_INTEGRATION_TABLE_ACTIVITY](#)는 통합의 삽입, 삭제 및 업데이트 활동에 대한 정보를 제공합니다.
- [SYS_INTEGRATION_TABLE_STATE_CHANGE](#)는 통합에 대한 테이블 상태 변경 로그에 대한 정보를 제공합니다.

Amazon EventBridge로 제로 ETL 통합 모니터링

Amazon Redshift는 통합 관련 이벤트를 Amazon EventBridge로 보냅니다. 이벤트 목록과 해당 이벤트 ID는 [Amazon EventBridge를 사용하여 제로 ETL 통합 이벤트 알림](#)을 참조하세요.

제로 ETL 통합의 지표

Amazon Redshift 콘솔 및 Amazon CloudWatch의 지표를 사용하여 제로 ETL 통합 상태 및 성능에 대해 알아볼 수 있습니다. 지표를 조정하여 더 짧거나 더 긴 기간 동안 데이터를 표시하거나 CloudWatch에서 지표를 보도록 선택할 수 있습니다. Amazon Redshift 콘솔에서 통합에 대한 지표를 보려면 왼쪽 탐색 창에서 제로 ETL 통합을 선택하고 통합 ID를 선택하세요.

제로 ETL 통합의 소스 데이터에 따라, Amazon Redshift는 통합 세부 정보 페이지에서 통합에 대한 지표를 제공합니다. 가능한 지표에는 다음과 같은 유형이 있습니다.

- 통합 지표 탭에서 다음과 같은 그래프를 사용할 수 있습니다.

지표	Amazon Redshift 콘솔의 지표 이름	설명
IntegrationLag	Lag	데이터가 소스에 커밋된 시점부터 Amazon Redshift에서 쿼리에 데이터를 사용할 수 있는 시점까지의 지연 시간입니다. 단위: 초

지표	Amazon Redshift 콘솔의 지표 이름	설명
		차원: IntegrationId
IntegrationNumTablesReplicated	Tables replicated	소스 데이터베이스에서 Amazon Redshift로 복제된 테이블 수입니다. 단위: 개 차원: IntegrationId
IntegrationNumTablesFailedReplication	Tables failed	복제에 실패한 테이블의 수입니다. 단위: 개 차원: IntegrationId
IntegrationDataTransferred	Data transferred	전송된 데이터의 양(논리적 바이트 단위) 단위: 바이트 차원: IntegrationId

- 테이블 통계 탭에서 현재 활성화되어 있거나 오류가 있는 테이블의 목록을 볼 수 있습니다. 이 탭의 통계는 다음과 같습니다(소스 유형에 따라 다름).
 - 스키마 이름 - 테이블이 있는 스키마의 이름입니다.
 - 테이블 이름 - 소스 데이터베이스에 있는 테이블의 이름입니다.
 - 상태 - 테이블의 상태입니다. 가능한 값은 Synced, Failed, Deleted, Resync Required 및 Resync Initiated입니다.
 - 데이터베이스 - 테이블이 속해 있는 Amazon Redshift 데이터베이스입니다.
 - 최종 업데이트 - 테이블을 마지막으로 업데이트한 날짜 및 시간(UTC)입니다.
 - 테이블 행 수 - 테이블의 행 수입니다.
 - 테이블 크기 - 테이블의 크기입니다.

선택한 기간에 대해 삽입, 삭제 및 업데이트된 행 수의 그래프를 볼 수도 있습니다.

DynamoDB에 대한 제로 ETL 통합 수정

이 단계에서는 Amazon Redshift와의 DynamoDB 제로 ETL 통합을 수정합니다.

Amazon Redshift console

Amazon Redshift 콘솔을 사용하여 Amazon Redshift와의 Amazon DynamoDB 제로 ETL 통합을 수정하려면

1. Amazon Redshift 콘솔에서 제로 ETL 통합을 선택합니다. 제로 ETL 통합 목록이 있는 창에서 수정하려는 DynamoDB 통합을 선택합니다.
2. 편집을 선택하고 통합 이름 또는 설명을 수정합니다.
3. 변경 사항을 저장하려면 변경 사항 저장을 선택합니다.

AWS CLI

AWS CLI를 사용하여 Amazon Redshift와의 Amazon DynamoDB 제로 ETL 통합을 수정하려면 다음 옵션과 함께 `modify-integration` 명령을 사용합니다.

- `integration-arn` - 수정할 DynamoDB 통합의 ARN을 지정합니다.
- `integration-name` - 통합의 새로운 이름을 지정합니다.
- `description` - 통합의 새로운 설명을 지정합니다.

다음 예제에서는 통합 ARN, 새 설명 및 새 이름을 제공하여 통합을 수정합니다.

```
aws redshift modify-integration \
--integration-arn arn:aws:redshift:us-
east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
--description "Test modify description and name together." \
--integration-name "updated-integration-name-2"

{
  "IntegrationArn": "arn:aws:redshift:us-
east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "IntegrationName": "updated-integration-name-2",
  "SourceArn": "arn:aws:dynamodb:us-east-1:123456789012:table/ddb-temp-test-table-
table",
  "SourceType": "dynamodb",
```

```

    "TargetArn": "arn:aws:redshift:us-east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Status": "active",
    "Errors": [],
    "CreateTime": "2024-09-19T18:06:33.555Z",
    "Description": "Test modify description and name together.",
    "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "AdditionalEncryptionContext": {},
    "Tags": []
  }

```

DynamoDB에 대한 제로 ETL 통합 삭제

통합을 삭제하면 대상 데이터 웨어하우스는 이전에 복제된 모든 데이터를 유지합니다. 이 데이터를 계속 공유하고 쿼리할 수 있습니다. 그러나 소스의 새 데이터는 대상에 복제되지 않습니다.

이 단계에서는 Amazon Redshift와의 DynamoDB 제로 ETL 통합을 삭제합니다.

Amazon Redshift console

Amazon Redshift 콘솔을 사용하여 Amazon Redshift와의 Amazon DynamoDB 제로 ETL 통합을 삭제하려면

1. Amazon Redshift 콘솔에서 제로 ETL 통합을 선택합니다. 제로 ETL 통합 목록이 있는 창에서 삭제하려는 DynamoDB 통합을 선택합니다.
2. 삭제를 선택하고 요청된 정보를 제공합니다.
3. 삭제를 선택하여 제로 ETL 통합을 삭제합니다.

AWS CLI

AWS CLI를 사용하여 Amazon Redshift와의 Amazon DynamoDB 제로 ETL 통합을 삭제하려면 다음 옵션과 함께 `delete-integration` 명령을 사용합니다.

- `integration-arn` - 삭제할 DynamoDB 통합의 ARN을 지정합니다.

다음 예제에서는 통합 ARN을 제공하여 통합을 삭제합니다.

```
aws redshift delete-integration \
```

```
--integration-arn arn:aws:redshift:us-east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

{
  "IntegrationArn": "arn:aws:redshift:us-east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "IntegrationName": "updated-integration-name-2",
  "SourceArn": "arn:aws:dynamodb:us-east-1:123456789012:table/tidal-ddb-ddb-temp-test-table-table",
  "SourceType": "dynamodb",
  "TargetArn": "arn:aws:redshift:us-east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "Status": "deleting",
  "Errors": [],
  "CreateTime": "2024-09-19T18:06:33.555Z",
  "Description": "Test modify description and name together.",
  "KMSKeyId": "arn:aws:kms:us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
  "AdditionalEncryptionContext": {},
  "Tags": []
}
```

제로 ETL 통합을 지원하는 리전

제로 ETL 통합은 Amazon Redshift에서 여러 운영 및 트랜잭션 소스와 엔터프라이즈 애플리케이션의 트랜잭션 및 운영 데이터를 사용할 수 있도록 하는 완전관리형 솔루션입니다. 이 페이지에는 지원되는 각 소스를 사용할 수 있는 리전이 나열됩니다.

Aurora MySQL

Amazon Redshift가 구성된 Aurora MySQL 제로 ETL 통합에 사용할 수 있는 리전 및 엔진 버전은 다음과 같습니다.

리전	Aurora MySQL
아프리카(케이프타운)	Available
아시아 태평양(홍콩)	Available
아시아 태평양(도쿄)	Available

리전	Aurora MySQL
아시아 태평양(서울)	Available
아시아 태평양(오사카)	Available
아시아 태평양(뭄바이)	Available
아시아 태평양(하이데라바드)	Available
아시아 태평양(싱가포르)	Available
아시아 태평양(시드니)	Available
아시아 태평양(자카르타)	Available
아시아 태평양(멜버른)	Available
아시아 태평양(말레이시아)	사용할 수 없음
캐나다(중부)	Available
캐나다 서부(캘거리)	Available
중국(베이징)	Available
중국(닝샤)	Available
유럽(프랑크푸르트)	Available
유럽(취리히)	Available
유럽(스톡홀름)	Available
유럽(밀라노)	Available
유럽(스페인)	Available
유럽(아일랜드)	Available
유럽(런던)	Available

리전	Aurora MySQL
유럽(파리)	Available
이스라엘(텔아비브)	Available
중동(UAE)	Available
중동(바레인)	Available
남아메리카(상파울루)	Available
미국 동부(버지니아 북부)	Available
미국 동부(오하이오)	Available
미국 서부(캘리포니아 북부)	Available
미국 서부(오리건)	Available
AWS GovCloud(미국 동부)	사용할 수 없음
AWS GovCloud(미국 서부)	사용할 수 없음

Aurora PostgreSQL

Amazon Redshift가 구성된 Aurora PostgreSQL 제로 ETL 통합을 사용할 수 있는 리전은 다음과 같습니다.

리전	Aurora PostgreSQL
아프리카(케이프타운)	Available
아시아 태평양(홍콩)	Available
아시아 태평양(도쿄)	Available
아시아 태평양(서울)	Available
아시아 태평양(오사카)	Available

리전	Aurora PostgreSQL
아시아 태평양(뭄바이)	Available
아시아 태평양(하이데라바드)	Available
아시아 태평양(싱가포르)	Available
아시아 태평양(시드니)	Available
아시아 태평양(자카르타)	Available
아시아 태평양(멜버른)	Available
아시아 태평양(말레이시아)	Available
캐나다(중부)	Available
캐나다 서부(캘거리)	Available
중국(베이징)	Available
중국(닝샤)	Available
유럽(프랑크푸르트)	Available
유럽(취리히)	Available
유럽(스톡홀름)	Available
유럽(밀라노)	Available
유럽(스페인)	Available
유럽(아일랜드)	Available
유럽(런던)	Available
유럽(파리)	Available
이스라엘(텔아비브)	Available

리전	Aurora PostgreSQL
중동(UAE)	Available
중동(바레인)	Available
남아메리카(상파울루)	Available
미국 동부(버지니아 북부)	Available
미국 동부(오하이오)	Available
미국 서부(캘리포니아 북부)	Available
미국 서부(오리건)	Available
AWS GovCloud(미국 동부)	사용할 수 없음
AWS GovCloud(미국 서부)	사용할 수 없음

Amazon DynamoDB

Amazon Redshift가 구성된 DynamoDB 제로 ETL 통합을 사용할 수 있는 리전은 다음과 같습니다.

리전	DynamoDB
아프리카(케이프타운)	Available
아시아 태평양(홍콩)	Available
아시아 태평양(타이베이)	Available
아시아 태평양(도쿄)	Available
아시아 태평양(서울)	Available
아시아 태평양(오사카)	Available
아시아 태평양(뭄바이)	Available

리전	DynamoDB
아시아 태평양(하이데라바드)	Available
아시아 태평양(싱가포르)	Available
아시아 태평양(시드니)	Available
아시아 태평양(자카르타)	Available
아시아 태평양(멜버른)	Available
아시아 태평양(말레이시아)	Available
아시아 태평양(태국)	Available
캐나다(중부)	Available
캐나다 서부(캘거리)	Available
중국(베이징)	Available
중국(닝샤)	Available
유럽(프랑크푸르트)	Available
유럽(취리히)	Available
유럽(스톡홀름)	Available
유럽(밀라노)	Available
유럽(스페인)	Available
유럽(아일랜드)	Available
유럽(런던)	Available
유럽(파리)	Available
이스라엘(텔아비브)	Available

리전	DynamoDB
중동(UAE)	Available
중동(바레인)	Available
멕시코(중부)	Available
남아메리카(상파울루)	Available
미국 동부(버지니아 북부)	Available
미국 동부(오하이오)	Available
미국 서부(캘리포니아 북부)	Available
미국 서부(오리건)	Available
AWS GovCloud(미국 동부)	Available
AWS GovCloud(미국 서부)	Available

Amazon RDS for MySQL

Amazon Redshift가 구성된 Amazon RDS for MySQL 제로 ETL 통합을 사용할 수 있는 리전은 다음과 같습니다.

리전	RDS for MySQL
아프리카(케이프타운)	Available
아시아 태평양(홍콩)	Available
아시아 태평양(도쿄)	Available
아시아 태평양(서울)	Available
아시아 태평양(오사카)	Available
아시아 태평양(뭄바이)	Available

리전	RDS for MySQL
아시아 태평양(하이데라바드)	사용할 수 없음
아시아 태평양(싱가포르)	Available
아시아 태평양(시드니)	Available
아시아 태평양(자카르타)	사용할 수 없음
아시아 태평양(멜버른)	사용할 수 없음
아시아 태평양(말레이시아)	사용할 수 없음
캐나다(중부)	Available
캐나다 서부(캘거리)	사용할 수 없음
중국(베이징)	사용할 수 없음
중국(닝샤)	사용할 수 없음
유럽(프랑크푸르트)	Available
유럽(취리히)	사용할 수 없음
유럽(스톡홀름)	Available
유럽(밀라노)	Available
유럽(스페인)	사용할 수 없음
유럽(아일랜드)	Available
유럽(런던)	Available
유럽(파리)	Available
이스라엘(텔아비브)	사용할 수 없음
중동(UAE)	사용할 수 없음

리전	RDS for MySQL
중동(바레인)	Available
남아메리카(상파울루)	Available
미국 동부(버지니아 북부)	Available
미국 동부(오하이오)	Available
미국 서부(캘리포니아 북부)	Available
미국 서부(오리건)	Available
AWS GovCloud(미국 동부)	사용할 수 없음
AWS GovCloud(미국 서부)	사용할 수 없음

엔터프라이즈 애플리케이션

Amazon Redshift가 구성된 엔터프라이즈 애플리케이션 제로 ETL 통합을 사용할 수 있는 리전은 다음과 같습니다.

리전	엔터프라이즈 애플리케이션
아프리카(케이프타운)	사용할 수 없음
아시아 태평양(홍콩)	Available
아시아 태평양(도쿄)	Available
아시아 태평양(서울)	Available
아시아 태평양(오사카)	사용할 수 없음
아시아 태평양(뭄바이)	사용할 수 없음
아시아 태평양(하이데라바드)	사용할 수 없음
아시아 태평양(싱가포르)	Available

리전	엔터프라이즈 애플리케이션
아시아 태평양(시드니)	Available
아시아 태평양(자카르타)	사용할 수 없음
아시아 태평양(멜버른)	사용할 수 없음
아시아 태평양(말레이시아)	사용할 수 없음
캐나다(중부)	Available
캐나다 서부(캘거리)	사용할 수 없음
중국(베이징)	사용할 수 없음
중국(닝샤)	사용할 수 없음
유럽(프랑크푸르트)	Available
유럽(취리히)	사용할 수 없음
유럽(스톡홀름)	Available
유럽(밀라노)	사용할 수 없음
유럽(스페인)	사용할 수 없음
유럽(아일랜드)	Available
유럽(런던)	Available
유럽(파리)	사용할 수 없음
이스라엘(텔아비브)	사용할 수 없음
중동(UAE)	사용할 수 없음
중동(바레인)	사용할 수 없음
남아메리카(상파울루)	Available

리전	엔터프라이즈 애플리케이션
미국 동부(버지니아 북부)	Available
미국 동부(오하이오)	Available
미국 서부(캘리포니아 북부)	사용할 수 없음
미국 서부(오리건)	Available
AWS GovCloud(미국 동부)	사용할 수 없음
AWS GovCloud(미국 서부)	사용할 수 없음

제로 ETL 통합 문제 해결

다음 섹션의 내용으로 제로 ETL 통합 관련 문제를 해결할 수 있습니다.

Aurora MySQL과의 제로 ETL 통합 문제 해결

다음 정보를 사용하여 Aurora MySQL과의 제로 ETL 통합과 관련된 일반적인 문제를 해결하세요.

주제

- [통합 생성 실패](#)
- [테이블에 기본 키가 없는 경우](#)
- [Aurora MySQL 테이블이 Amazon Redshift에 복제되지 않는 경우](#)
- [테이블에서 지원되지 않는 데이터 형식](#)
- [데이터 조작 언어 명령 실패](#)
- [데이터 소스 간의 추적된 변경 내용이 일치하지 않음](#)
- [권한 부여 실패](#)
- [테이블 수가 100,000개를 초과하거나 스키마 수가 4,950개를 초과합니다.](#)
- [Amazon Redshift에서 데이터를 로드할 수 없는 경우](#)
- [작업 그룹 파라미터 설정이 올바르지 않습니다.](#)
- [제로 ETL 통합을 활성화하기 위해 데이터베이스가 생성되지 않았습니다.](#)
- [테이블이 재동기화 필요 또는 재동기화 시작됨 상태입니다.](#)
- [통합 지연 증가](#)

통합 생성 실패

제로 ETL 통합을 만들지 못한 경우 통합의 상태는 Inactive입니다. 소스 Aurora DB 클러스터에 대해 다음 사항이 올바른지 확인하세요.

- Amazon RDS 콘솔에서 클러스터를 생성했습니다.
- 소스 Aurora DB 클러스터가 지원되는 버전을 실행 중입니다. 지원되는 버전 목록은 [Amazon Redshift와 제로 ETL 통합을 지원하는 리전 및 Aurora DB 엔진](#)을 참조하세요. 이를 확인하려면 클러스터의 구성 탭으로 이동하여 엔진 버전을 확인하세요.
- 클러스터에 대한 binlog 파라미터 설정을 올바르게 구성했습니다. Aurora MySQL binlog 파라미터가 잘못 설정되었거나 소스 Aurora DB 클러스터와 연결되지 않은 경우 생성에 실패합니다. [DB 클러스터 파라미터 구성](#)을 참조하세요.

또한 Amazon Redshift 데이터 웨어하우스에 다음 사항이 올바른지 확인하세요.

- 대/소문자 구분이 설정되어 있습니다. [데이터 웨어하우스에 대/소문자 구분 기능 사용 설정](#)(를) 참조하세요.
- 네임스페이스에 대해 올바른 권한 있는 보안 주체 및 통합 소스를 추가했습니다. [Amazon Redshift 데이터 웨어하우스에 대한 권한 부여 구성](#)(를) 참조하세요.

테이블에 기본 키가 없는 경우

대상 데이터베이스에서 하나 이상의 테이블에 기본 키가 없어 동기화할 수 없습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 테이블 통계 탭으로 이동하거나 SVV_INTEGRATION_TABLE_STATE를 사용하여 실패한 테이블을 확인합니다. 테이블에 프라이머리 키를 추가하면 Amazon Redshift에서 테이블을 다시 동기화할 수 있습니다. 권장되지는 않지만 이러한 테이블을 Aurora에 끌어다 놓고 프라이머리 키가 있는 테이블을 만들 수도 있습니다. 자세한 내용은 테이블 설계를 위한 [Amazon Redshift 모범 사례](#) 섹션을 참조하세요.

Aurora MySQL 테이블이 Amazon Redshift에 복제되지 않는 경우

Amazon Redshift에 하나 이상의 테이블이 반영되지 않는 경우 다음 명령을 실행하여 테이블을 다시 동기화할 수 있습니다. *db_name*을 Amazon Redshift 데이터베이스의 이름으로 바꿉니다. 그리고 *table1* 및 *table2*를 동기화할 테이블의 이름으로 바꿉니다.

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```


자세한 내용은 Amazon Redshift 데이터베이스 개발자 가이드의 [데이터베이스 변경](#)을 참조하세요.

하나 이상의 원본 테이블에 프라이머리 키가 없어 데이터가 복제되지 않을 수 있습니다. Amazon Redshift의 모니터링 대시보드에는 이러한 테이블의 상태가 Failed로 표시되고 전체 제로 ETL 통합 상태는 Needs attention으로 변경됩니다. 이 문제를 해결하려면 테이블에서 프라이머리 키가 될 수 있는 기존 키를 식별하거나 가상 프라이머리 키를 추가할 수 있습니다. 자세한 해결 방법은 AWS Database Blog의 [Handle tables without primary keys while creating Amazon Aurora MySQL or RDS for MySQL zero-ETL integrations with Amazon Redshift](#)를 참조하세요.

또한 대상이 Amazon Redshift 클러스터인 경우 클러스터가 일시 중지되지 않았는지 확인합니다.

테이블에서 지원되지 않는 데이터 형식

Amazon Redshift의 통합에서 생성한 데이터베이스의 데이터가 Aurora DB 클러스터에서 복제되는 경우 하나 이상의 테이블에 지원되지 않는 데이터 유형이 있어 동기화할 수 없습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 테이블 통계 탭으로 이동하거나 SVV_INTEGRATION_TABLE_STATE를 사용하여 실패한 테이블을 확인합니다. 그런 다음 이러한 테이블을 제거하고 Amazon RDS에서 새 테이블을 다시 만듭니다. 지원되지 않는 데이터 유형에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 [Aurora와 Amazon Redshift 데이터베이스 간의 데이터 유형 차이](#)를 참조하세요.

데이터 조작 언어 명령 실패

Amazon Redshift가 Redshift 테이블에서 DML 명령을 실행할 수 없습니다. 이 문제를 해결하려면 SVV_INTEGRATION_TABLE_STATE를 사용하여 실패한 테이블을 확인합니다. Amazon Redshift는 이 오류를 해결하기 위해 테이블을 자동으로 다시 동기화합니다.

데이터 소스 간의 추적된 변경 내용이 일치하지 않음

이 오류는 Amazon Aurora와 Amazon Redshift 간의 변경 사항이 일치하지 않아 통합이 Failed 상태로 전환될 때 발생합니다.

이 문제를 해결하려면 제로 ETL 통합을 삭제하고 Amazon RDS에서 다시 생성하세요. 자세한 내용은 [제로 ETL 통합 생성](#) 및 [제로 ETL 통합 삭제](#)를 참조하세요.

권한 부여 실패

소스 Aurora DB 클러스터가 Amazon Redshift 데이터 웨어하우스에 대한 권한 있는 통합 소스로서 제거되었기 때문에 권한 부여에 실패했습니다.

이 문제를 해결하려면 제로 ETL 통합을 삭제하고 Amazon RDS에서 다시 생성하세요. 자세한 내용은 [제로 ETL 통합 생성](#) 및 [제로 ETL 통합 삭제](#)를 참조하세요.

테이블 수가 100,000개를 초과하거나 스키마 수가 4,950개를 초과합니다.

대상 데이터 웨어하우스의 경우 테이블 수가 100,000개 이상이거나 스키마 수가 4,950개 이상입니다. Amazon Aurora에서 Amazon Redshift로 데이터를 보낼 수 없습니다. 테이블 및 스키마 수가 설정된 제한을 초과합니다. 이 문제를 해결하려면 소스 데이터베이스에서 불필요한 스키마나 테이블을 제거하세요.

Amazon Redshift에서 데이터를 로드할 수 없는 경우

Amazon Redshift가 제로 ETL 통합에 데이터를 로드할 수 없습니다.

이 문제를 해결하려면 Amazon RDS에서 제로 ETL 통합을 삭제하고 다시 생성하세요. 자세한 내용은 [제로 ETL 통합 생성](#) 및 [제로 ETL 통합 삭제](#)를 참조하세요.

작업 그룹 파라미터 설정이 올바르지 않습니다.

작업 그룹에 대/소문자 구분이 설정되어 있지 않습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 속성 탭으로 이동하여 파라미터 그룹을 선택한 다음 속성 탭에서 대/소문자 구분 식별자를 사용 설정하세요. 기존 파라미터 그룹이 없는 경우 대/소문자 구분 식별자를 사용하도록 설정하여 그룹을 만드세요. 그런 다음 Amazon RDS에서 새 제로 ETL 통합을 생성합니다. 자세한 내용은 [제로 ETL 통합 생성](#)을 참조하세요.

제로 ETL 통합을 활성화하기 위해 데이터베이스가 생성되지 않았습니다.

제로 ETL 통합을 활성화하기 위해 생성된 데이터베이스가 없습니다.

이 문제를 해결하려면 통합을 위한 데이터베이스를 만드세요. 자세한 내용은 [Amazon Redshift에서 대상 데이터베이스 생성](#) 섹션을 참조하세요.

테이블이 재동기화 필요 또는 재동기화 시작된 상태입니다.

테이블이 재동기화 필요 또는 재동기화 시작된 상태입니다.

테이블이 해당 상태인 이유에 대한 자세한 오류 정보를 수집하려면 [SYS_LOAD_ERROR_DETAIL](#) 시스템 뷰를 사용하세요.

통합 지연 증가

소스 데이터베이스에서 SAVEPOINT를 많이 사용하는 경우 제로 ETL 통합의 통합 지연이 증가할 수 있습니다.

Aurora PostgreSQL과의 제로 ETL 통합 문제 해결

다음 정보를 사용하여 Aurora PostgreSQL과의 제로 ETL 통합과 관련된 일반적인 문제를 해결하세요.

주제

- [통합 생성 실패](#)
- [테이블에 기본 키가 없는 경우](#)
- [Aurora PostgreSQL 테이블이 Amazon Redshift에 복제되지 않는 경우](#)
- [테이블에서 지원되지 않는 데이터 형식](#)
- [데이터 조작 언어 명령 실패](#)
- [데이터 소스 간의 추적된 변경 내용이 일치하지 않음](#)
- [권한 부여 실패](#)
- [테이블 수가 100,000개를 초과하거나 스키마 수가 4,950개를 초과합니다.](#)
- [Amazon Redshift에서 데이터를 로드할 수 없는 경우](#)
- [작업 그룹 파라미터 설정이 올바르지 않습니다.](#)
- [제로 ETL 통합을 활성화하기 위해 데이터베이스가 생성되지 않았습니다.](#)
- [테이블이 재동기화 필요 또는 재동기화 시작됨 상태입니다.](#)

통합 생성 실패

제로 ETL 통합을 만들지 못한 경우 통합의 상태는 Inactive입니다. 소스 Aurora DB 클러스터에 대해 다음 사항이 올바른지 확인하세요.

- Amazon RDS 콘솔에서 클러스터를 생성했습니다.
- 소스 Aurora DB 클러스터가 지원되는 버전을 실행 중입니다. 지원되는 버전 목록은 [Amazon Redshift와 제로 ETL 통합을 지원하는 리전 및 Aurora DB 엔진](#)을 참조하세요. 이를 확인하려면 클러스터의 구성 탭으로 이동하여 엔진 버전을 확인하세요.
- 클러스터에 대한 binlog 파라미터 설정을 올바르게 구성했습니다. Aurora PostgreSQL binlog 파라미터가 잘못 설정되었거나 소스 Aurora DB 클러스터와 연결되지 않은 경우 생성에 실패합니다. [DB 클러스터 파라미터 구성](#)을 참조하세요.

또한 Amazon Redshift 데이터 웨어하우스에 다음 사항이 올바른지 확인하세요.

- 대/소문자 구분이 설정되어 있습니다. [데이터 웨어하우스에 대/소문자 구분 기능 사용 설정](#)(를) 참조하세요.

- `<endterm="zero-etl-using.redshift-iam.title"/>`에 대해 올바른 권한 있는 보안 주체 및 통합 소스를 추가했습니다.

테이블에 기본 키가 없는 경우

대상 데이터베이스에서 하나 이상의 테이블에 기본 키가 없어 동기화할 수 없습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 테이블 통계 탭으로 이동하거나 `SVV_INTEGRATION_TABLE_STATE`를 사용하여 실패한 테이블을 확인합니다. 테이블에 프라이머리 키를 추가하면 Amazon Redshift에서 테이블을 다시 동기화할 수 있습니다. 권장되지는 않지만 이러한 테이블을 Aurora에 끌어다 놓고 프라이머리 키가 있는 테이블을 만들 수도 있습니다. 자세한 내용은 테이블 설계를 위한 [Amazon Redshift 모범 사례](#) 섹션을 참조하세요.

Aurora PostgreSQL 테이블이 Amazon Redshift에 복제되지 않는 경우

Amazon Redshift에 하나 이상의 테이블이 반영되지 않는 경우 다음 명령을 실행하여 테이블을 다시 동기화할 수 있습니다. `db_name`을 Amazon Redshift 데이터베이스의 이름으로 바꿉니다. 그리고 `table1` 및 `table2`를 동기화할 테이블의 이름으로 바꿉니다.

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

자세한 내용은 Amazon Redshift 데이터베이스 개발자 가이드의 [데이터베이스 변경](#)을 참조하세요.

하나 이상의 원본 테이블에 프라이머리 키가 없어 데이터가 복제되지 않을 수 있습니다. Amazon Redshift의 모니터링 대시보드에는 이러한 테이블의 상태가 Failed로 표시되고 전체 제로 ETL 통합 상태는 Needs attention으로 변경됩니다. 이 문제를 해결하려면 테이블에서 프라이머리 키가 될 수 있는 기존 키를 식별하거나 가상 프라이머리 키를 추가할 수 있습니다. 자세한 해결 방법은 AWS Database Blog의 [Handle tables without primary keys while creating Amazon Aurora PostgreSQL zero-ETL integrations with Amazon Redshift](#)를 참조하세요.

또한 대상이 Amazon Redshift 클러스터인 경우 클러스터가 일시 중지되지 않았는지 확인합니다.

테이블에서 지원되지 않는 데이터 형식

Amazon Redshift의 통합에서 생성한 데이터베이스의 데이터가 Aurora DB 클러스터에서 복제되는 경우 하나 이상의 테이블에 지원되지 않는 데이터 유형이 있어 동기화할 수 없습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 테이블 통계 탭으로 이동하거나 `SVV_INTEGRATION_TABLE_STATE`를 사용하여 실패한 테이블을 확인합니다. 그런 다음 이러한 테이블을 제거하고 Amazon RDS에서 새 테이블을 다시 만듭니다. 지원되지 않는 데이터 유형에 대한 자

세한 내용은 Amazon Aurora 사용 설명서의 [Aurora와 Amazon Redshift 데이터베이스 간의 데이터 유형 차이](#)를 참조하세요.

데이터 조작 언어 명령 실패

Amazon Redshift가 Redshift 테이블에서 DML 명령을 실행할 수 없습니다. 이 문제를 해결하려면 SVV_INTEGRATION_TABLE_STATE를 사용하여 실패한 테이블을 확인합니다. Amazon Redshift는 이 오류를 해결하기 위해 테이블을 자동으로 다시 동기화합니다.

데이터 소스 간의 추적된 변경 내용이 일치하지 않음

이 오류는 Amazon Aurora와 Amazon Redshift 간의 변경 사항이 일치하지 않아 통합이 Failed 상태로 전환될 때 발생합니다.

이 문제를 해결하려면 제로 ETL 통합을 삭제하고 Amazon RDS에서 다시 생성하세요. 자세한 내용은 [제로 ETL 통합 생성](#) 및 [제로 ETL 통합 삭제](#)를 참조하세요.

권한 부여 실패

소스 Aurora DB 클러스터가 Amazon Redshift 데이터 웨어하우스에 대한 권한 있는 통합 소스로서 제거되었기 때문에 권한 부여에 실패했습니다.

이 문제를 해결하려면 제로 ETL 통합을 삭제하고 Amazon RDS에서 다시 생성하세요. 자세한 내용은 [제로 ETL 통합 생성](#) 및 [제로 ETL 통합 삭제](#)를 참조하세요.

테이블 수가 100,000개를 초과하거나 스키마 수가 4,950개를 초과합니다.

대상 데이터 웨어하우스의 경우 테이블 수가 100,000개 이상이거나 스키마 수가 4,950개 이상입니다. Amazon Aurora에서 Amazon Redshift로 데이터를 보낼 수 없습니다. 테이블 및 스키마 수가 설정된 제한을 초과합니다. 이 문제를 해결하려면 소스 데이터베이스에서 불필요한 스키마나 테이블을 제거하세요.

Amazon Redshift에서 데이터를 로드할 수 없는 경우

Amazon Redshift가 제로 ETL 통합에 데이터를 로드할 수 없습니다.

이 문제를 해결하려면 Amazon RDS에서 제로 ETL 통합을 삭제하고 다시 생성하세요. 자세한 내용은 [제로 ETL 통합 생성](#) 및 [제로 ETL 통합 삭제](#)를 참조하세요.

작업 그룹 파라미터 설정이 올바르지 않습니다.

작업 그룹에 대/소문자 구분이 설정되어 있지 않습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 속성 탭으로 이동하여 파라미터 그룹을 선택한 다음 속성 탭에서 대/소문자 구분 식별자를 사용 설정하세요. 기존 파라미터 그룹이 없는 경우 대/소문자 구분 식별자를 사용하도록 설정하여 그룹을 만드세요. 그런 다음 Amazon RDS에서 새 제로 ETL 통합을 생성합니다. 자세한 내용은 [제로 ETL 통합 생성](#)을 참조하세요.

제로 ETL 통합을 활성화하기 위해 데이터베이스가 생성되지 않았습니다.

제로 ETL 통합을 활성화하기 위해 생성된 데이터베이스가 없습니다.

이 문제를 해결하려면 통합을 위한 데이터베이스를 만드세요. 자세한 내용은 [Amazon Redshift에서 대상 데이터베이스 생성](#) 섹션을 참조하세요.

테이블이 재동기화 필요 또는 재동기화 시작됨 상태입니다.

테이블이 재동기화 필요 또는 재동기화 시작됨 상태입니다.

테이블이 해당 상태인 이유에 대한 자세한 오류 정보를 수집하려면 [SYS_LOAD_ERROR_DETAIL](#) 시스템 뷰를 사용하세요.

RDS for MySQL과의 제로 ETL 통합 문제 해결

다음 정보를 사용하여 RDS for MySQL과의 제로 ETL 통합과 관련된 일반적인 문제를 해결하세요.

주제

- [통합 생성 실패](#)
- [테이블에 기본 키가 없는 경우](#)
- [RDS for MySQL 테이블이 Amazon Redshift에 복제되지 않는 경우](#)
- [테이블에서 지원되지 않는 데이터 형식](#)
- [데이터 조작 언어 명령 실패](#)
- [데이터 소스 간의 추적된 변경 내용이 일치하지 않음](#)
- [권한 부여 실패](#)
- [테이블 수가 100,000개를 초과하거나 스키마 수가 4,950개를 초과합니다.](#)
- [Amazon Redshift에서 데이터를 로드할 수 없는 경우](#)
- [작업 그룹 파라미터 설정이 올바르지 않습니다.](#)
- [제로 ETL 통합을 활성화하기 위해 데이터베이스가 생성되지 않았습니다.](#)
- [테이블이 재동기화 필요 또는 재동기화 시작됨 상태입니다.](#)

통합 생성 실패

제로 ETL 통합을 만들지 못한 경우 통합의 상태는 Inactive입니다. 소스 RDS DB 인스턴스에 대해 다음 사항이 올바른지 확인하세요.

- Amazon RDS 콘솔에서 인스턴스를 생성했습니다.
- 소스 RDS DB 인스턴스가 지원되는 버전의 RDS for MySQL을 실행 중입니다. 지원되는 버전 목록은 [Amazon Redshift와 Amazon RDS 제로 ETL 통합을 지원하는 리전 및 DB 엔진](#)을 참조하세요. 이를 확인하려면 인스턴스의 구성 탭으로 이동하여 엔진 버전을 확인하세요.
- 인스턴스에 대한 binlog 파라미터 설정을 올바르게 구성했습니다. RDS for MySQL binlog 파라미터가 잘못 설정되었거나 소스 RDS DB 인스턴스와 연결되지 않은 경우 생성에 실패합니다. [Configure DB instance parameters](#)을 참조하세요.

또한 Amazon Redshift 데이터 웨어하우스에 다음 사항이 올바른지 확인하세요.

- 대/소문자 구분이 설정되어 있습니다. [데이터 웨어하우스에 대/소문자 구분 기능 사용 설정](#)(를) 참조하세요.
- 네임스페이스에 대해 올바른 권한 있는 보안 주체 및 통합 소스를 추가했습니다. [Amazon Redshift 데이터 웨어하우스에 대한 권한 부여 구성](#)(를) 참조하세요.

테이블에 기본 키가 없는 경우

대상 데이터베이스에서 하나 이상의 테이블에 기본 키가 없어 동기화할 수 없습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 테이블 통계 탭으로 이동하거나 SVV_INTEGRATION_TABLE_STATE를 사용하여 실패한 테이블을 확인합니다. 테이블에 프라이머리 키를 추가하면 Amazon Redshift에서 테이블을 다시 동기화할 수 있습니다. 권장되지는 않지만 이러한 테이블을 RDS에 끌어다 놓고 프라이머리 키가 있는 테이블을 만들 수도 있습니다. 자세한 내용은 테이블 설계를 위한 [Amazon Redshift 모범 사례](#) 섹션을 참조하세요.

RDS for MySQL 테이블이 Amazon Redshift에 복제되지 않는 경우

Amazon Redshift에 하나 이상의 테이블이 반영되지 않는 경우 다음 명령을 실행하여 테이블을 다시 동기화할 수 있습니다. *db_name*을 Amazon Redshift 데이터베이스의 이름으로 바꿉니다. 그리고 *table1* 및 *table2*를 동기화할 테이블의 이름으로 바꿉니다.

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

자세한 내용은 Amazon Redshift 데이터베이스 개발자 가이드의 [데이터베이스 변경](#)을 참조하세요.

하나 이상의 원본 테이블에 프라이머리 키가 없어 데이터가 복제되지 않을 수 있습니다. Amazon Redshift의 모니터링 대시보드에는 이러한 테이블의 상태가 Failed로 표시되고 전체 제로 ETL 통합 상태는 Needs attention으로 변경됩니다. 이 문제를 해결하려면 테이블에서 프라이머리 키가 될 수 있는 기존 키를 식별하거나 가상 프라이머리 키를 추가할 수 있습니다. 자세한 해결 방법은 AWS 데이터베이스 블로그의 [Handle tables without primary keys while creating Aurora MySQL-Compatible Edition or RDS for MySQL zero-ETL integrations with Amazon Redshift](#)를 참조하세요.

또한 대상이 Amazon Redshift 클러스터인 경우 클러스터가 일시 중지되지 않았는지 확인합니다.

테이블에서 지원되지 않는 데이터 형식

Amazon Redshift의 통합에서 생성한 데이터베이스의 데이터가 RDS DB 인스턴스에서 복제되는 경우 하나 이상의 테이블에 지원되지 않는 데이터 유형이 있어 동기화할 수 없습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 테이블 통계 탭으로 이동하거나 SVV_INTEGRATION_TABLE_STATE를 사용하여 실패한 테이블을 확인합니다. 그런 다음 이러한 테이블을 제거하고 Amazon RDS에서 새 테이블을 다시 만듭니다. 지원되지 않는 데이터 유형에 대한 자세한 내용은 Amazon RDS 사용 설명서의 [RDS와 Amazon Redshift 데이터베이스 간의 데이터 유형 차이](#)를 참조하세요.

데이터 조작 언어 명령 실패

Amazon Redshift가 Redshift 테이블에서 DML 명령을 실행할 수 없습니다. 이 문제를 해결하려면 SVV_INTEGRATION_TABLE_STATE를 사용하여 실패한 테이블을 확인합니다. Amazon Redshift는 이 오류를 해결하기 위해 테이블을 자동으로 다시 동기화합니다.

데이터 소스 간의 추적된 변경 내용이 일치하지 않음

이 오류는 Amazon Aurora와 Amazon Redshift 간의 변경 사항이 일치하지 않아 통합이 Failed 상태로 전환될 때 발생합니다.

이 문제를 해결하려면 제로 ETL 통합을 삭제하고 Amazon RDS에서 다시 생성하세요. 자세한 내용은 [제로 ETL 통합 생성](#) 및 [제로 ETL 통합 삭제](#)를 참조하세요.

권한 부여 실패

소스 RDS DB 인스턴스가 Amazon Redshift 데이터 웨어하우스에 대한 권한 있는 통합 소스로서 제거되었기 때문에 권한 부여에 실패했습니다.

이 문제를 해결하려면 제로 ETL 통합을 삭제하고 Amazon RDS에서 다시 생성하세요. 자세한 내용은 [제로 ETL 통합 생성](#) 및 [제로 ETL 통합 삭제](#)를 참조하세요.

테이블 수가 100,000개를 초과하거나 스키마 수가 4,950개를 초과합니다.

대상 데이터 웨어하우스의 경우 테이블 수가 100,000개 이상이거나 스키마 수가 4,950개 이상입니다. Amazon Aurora에서 Amazon Redshift로 데이터를 보낼 수 없습니다. 테이블 및 스키마 수가 설정된 제한을 초과합니다. 이 문제를 해결하려면 소스 데이터베이스에서 불필요한 스키마나 테이블을 제거하세요.

Amazon Redshift에서 데이터를 로드할 수 없는 경우

Amazon Redshift가 제로 ETL 통합에 데이터를 로드할 수 없습니다.

이 문제를 해결하려면 Amazon RDS에서 제로 ETL 통합을 삭제하고 다시 생성하세요. 자세한 내용은 [제로 ETL 통합 생성](#) 및 [제로 ETL 통합 삭제](#)를 참조하세요.

작업 그룹 파라미터 설정이 올바르지 않습니다.

작업 그룹에 대/소문자 구분이 설정되어 있지 않습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 속성 탭으로 이동하여 파라미터 그룹을 선택한 다음 속성 탭에서 대/소문자 구분 식별자를 사용 설정하세요. 기존 파라미터 그룹이 없는 경우 대/소문자 구분 식별자를 사용하도록 설정하여 그룹을 만드세요. 그런 다음 Amazon RDS에서 새 제로 ETL 통합을 생성합니다. 자세한 내용은 [제로 ETL 통합 생성](#)을 참조하세요.

제로 ETL 통합을 활성화하기 위해 데이터베이스가 생성되지 않았습니다.

제로 ETL 통합을 활성화하기 위해 생성된 데이터베이스가 없습니다.

이 문제를 해결하려면 통합을 위한 데이터베이스를 만드세요. 자세한 내용은 [Amazon Redshift에서 대상 데이터베이스 생성](#) 섹션을 참조하세요.

테이블이 재동기화 필요 또는 재동기화 시작됨 상태입니다.

테이블이 재동기화 필요 또는 재동기화 시작됨 상태입니다.

테이블이 해당 상태인 이유에 대한 자세한 오류 정보를 수집하려면 [SYS_LOAD_ERROR_DETAIL](#) 시스템 뷰를 사용하세요.

DynamoDB와의 제로 ETL 통합 문제 해결

다음 정보를 사용하여 Amazon DynamoDB와의 제로 ETL 통합과 관련된 일반적인 문제를 해결하세요.

주제

- [통합 생성 실패](#)
- [테이블에서 지원되지 않는 데이터 형식](#)
- [지원되지 않는 테이블 및 속성 이름](#)
- [권한 부여 실패](#)
- [Amazon Redshift에서 데이터를 로드할 수 없는 경우](#)
- [작업 그룹 또는 클러스터 파라미터 설정이 올바르지 않습니다.](#)
- [제로 ETL 통합을 활성화하기 위해 데이터베이스가 생성되지 않았습니다.](#)
- [소스 DynamoDB 테이블에서 시점 복구\(PITR\)가 활성화되지 않음](#)
- [KMS 키 액세스 거부됨](#)
- [Amazon Redshift는 DynamoDB 테이블 키에 액세스할 수 없습니다.](#)

통합 생성 실패

제로 ETL 통합을 만들지 못한 경우 통합의 상태는 Inactive입니다. Amazon Redshift 데이터 웨어하우스 및 소스 DynamoDB 테이블에 대해 다음 사항이 올바른지 확인합니다.

- 데이터 웨어하우스에 대한 대소문자 구분 기능이 설정되어 있습니다. Amazon Redshift 관리 안내서의 [대/소문자 구분 켜기](#)를 참조하세요.
- 네임스페이스에 대해 올바른 권한 있는 위탁자 및 통합 소스를 추가했습니다. Amazon Redshift 관리 안내서의 [Amazon Redshift 데이터 웨어하우스에 대한 권한 부여 구성](#)을 참조하세요.
- 소스 DynamoDB 테이블에 올바른 리소스 기반 정책을 추가했습니다. IAM 사용 설명서에서 [IAM의 정책 및 권한](#)을 참조하세요.

테이블에서 지원되지 않는 데이터 형식

DynamoDB 숫자는 Amazon Redshift에서 DECIMAL(38,10)로 변환됩니다. 이 정밀도 범위를 초과하는 숫자는 (38,10)으로 자동 변환됩니다. 통합을 삭제하고 숫자 정밀도를 통합한 다음 통합을 다시 생성합니다.

지원되지 않는 테이블 및 속성 이름

Amazon Redshift는 최대 127개의 문자 테이블과 속성 이름을 지원합니다. DynamoDB 테이블 이름, 파티션 키 또는 정렬 키 열 이름과 같은 긴 이름이 통합에 실패하는 경우 짧은 이름을 사용하여 수정하고 통합을 다시 생성합니다.

권한 부여 실패

소스 DynamoDB 테이블이 Amazon Redshift 데이터 웨어하우스에 대한 권한 있는 통합 소스로서 제거되었기 때문에 권한 부여에 실패했습니다.

이 문제를 해결하려면 제로 ETL 통합을 삭제하고 Amazon DynamoDB를 사용하여 다시 생성하세요.

Amazon Redshift에서 데이터를 로드할 수 없는 경우

Amazon Redshift가 제로 ETL 통합에서 데이터를 로드할 수 없습니다.

이 문제를 해결하려면 ALTER DATABASE를 사용하여 통합을 새로 고치세요.

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL TABLES
```

작업 그룹 또는 클러스터 파라미터 설정이 올바르지 않습니다.

작업 그룹 또는 클러스터에 대/소문자 구분이 설정되어 있지 않습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 속성 탭으로 이동하여 파라미터 그룹을 선택한 다음 속성 탭에서 대/소문자 구분 식별자를 사용 설정하세요. 기존 파라미터 그룹이 없는 경우 대/소문자 구분 식별자를 사용하도록 설정하여 그룹을 만드세요. 그런 다음 DynamoDB에서 새 제로 ETL 통합을 생성합니다. Amazon Redshift 관리 안내서의 [대/소문자 구분 켜기](#)를 참조하세요.

제로 ETL 통합을 활성화하기 위해 데이터베이스가 생성되지 않았습니다.

제로 ETL 통합을 활성화하기 위해 생성된 데이터베이스가 없습니다.

이 문제를 해결하려면 통합을 위한 데이터베이스를 만드세요. [Amazon Redshift 관리 안내서](#)의 Amazon Redshift에서 대상 데이터베이스 생성을 참조하세요.

소스 DynamoDB 테이블에서 시점 복구(PITR)가 활성화되지 않음

DynamoDB가 데이터를 내보내려면 PITR을 활성화해야 합니다. PITR이 항상 활성화되어 있는지 확인합니다. 통합이 활성화되어 있는 동안 PITR을 끄는 경우 오류 메시지의 지침을 따르고 ALTER DATABASE를 사용하여 통합을 새로 고쳐야 합니다.

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL TABLES
```

KMS 키 액세스 거부됨

소스 테이블 또는 통합에 사용되는 KMS 키는 충분한 권한으로 구성되어야 합니다. 테이블 암호화 및 복호화에 대한 자세한 내용을 알아보려면 Amazon DynamoDB 개발자 안내서의 [저장 중 DynamoDB 암호화](#)를 참조하세요.

Amazon Redshift는 DynamoDB 테이블 키에 액세스할 수 없습니다.

소스 테이블 암호화가 AWS 관리형 키인 경우 AWS 소유 키 또는 고객 관리형 키로 전환합니다. 테이블이 이미 고객 관리형 키로 암호화된 경우 정책에 조건 키가 없는지 확인합니다.

애플리케이션과의 제로 ETL 통합 문제 해결

다음 정보를 사용하여 Salesforce, SAP, ServiceNow 및 Zendesk와 같은 애플리케이션과의 제로 ETL 통합과 관련된 일반적인 문제를 해결합니다.

주제

- [통합 생성 실패](#)
- [내 테이블이 Amazon Redshift에 복제되지 않음](#)
- [테이블에서 지원되지 않는 데이터 형식](#)
- [작업 그룹 파라미터 설정이 올바르지 않습니다.](#)
- [제로 ETL 통합을 활성화하기 위해 데이터베이스가 생성되지 않았습니다.](#)
- [테이블이 재동기화 필요 또는 재동기화 시작됨 상태입니다.](#)

통합 생성 실패

제로 ETL 통합을 만들지 못한 경우 통합의 상태는 Inactive입니다. Amazon Redshift 데이터 웨어하우스에서 다음 사항이 올바른지 확인하세요.

- 대/소문자 구분이 설정되어 있습니다. [데이터 웨어하우스에 대/소문자 구분 기능 사용 설정](#)(를) 참조하세요.
- 네임스페이스에 대해 올바른 권한 있는 보안 주체 및 통합 소스를 추가했습니다. [Amazon Redshift 데이터 웨어하우스에 대한 권한 부여 구성](#)(를) 참조하세요.

내 테이블이 Amazon Redshift에 복제되지 않음

대상 데이터베이스에서 하나 이상의 테이블에 기본 키가 없어 동기화할 수 없습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 테이블 통계 탭으로 이동하거나 SVV_INTEGRATION_TABLE_STATE를 사용하여 실패한 테이블을 확인합니다. 테이블에 프라이머리 키를 추가하면 Amazon Redshift에서 테이블을 다시 동기화할 수 있습니다. 다음 명령을 실행하여 다시 동기화할 수 있습니다. *db_name*을 Amazon Redshift 데이터베이스의 이름으로 바꿉니다. 그리고 *table1* 및 *table2*를 동기화할 테이블의 이름으로 바꿉니다.

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [ALTER DATABASE](#)를 참조하세요.

테이블에서 지원되지 않는 데이터 형식

Amazon Redshift의 통합에서 만들었고 데이터가 애플리케이션과의 제로 ETL 통합에서 복제되는 데이터베이스의 경우 하나 이상의 테이블에 지원되지 않는 데이터 유형이 있어 동기화할 수 없습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 테이블 통계 탭으로 이동하거나 SVV_INTEGRATION_TABLE_STATE를 사용하여 실패한 테이블을 확인합니다. 그런 다음 이러한 테이블을 제거하고 소스에서 새 테이블을 다시 만듭니다. 자세한 내용은 AWS Glue 개발자 안내서의 [제로 ETL 통합](#)을 참조하세요.

작업 그룹 파라미터 설정이 올바르지 않습니다.

작업 그룹에 대/소문자 구분이 설정되어 있지 않습니다.

이 문제를 해결하려면 통합 세부 정보 페이지의 속성 탭으로 이동하여 파라미터 그룹을 선택한 다음 속성 탭에서 대/소문자 구분 식별자를 사용 설정하세요. 기존 파라미터 그룹이 없는 경우 대/소문자 구분 식별자를 사용하도록 설정하여 그룹을 만드세요. 그런 다음 새로운 제로 ETL 통합을 만듭니다. 자세한 내용은 AWS Glue 개발자 안내서의 [제로 ETL 통합](#)을 참조하세요.

제로 ETL 통합을 활성화하기 위해 데이터베이스가 생성되지 않았습니다.

제로 ETL 통합을 활성화하기 위해 생성된 데이터베이스가 없습니다.

이 문제를 해결하려면 통합을 위한 데이터베이스를 만드세요. 자세한 내용은 [Amazon Redshift에서 대상 데이터베이스 생성](#) 섹션을 참조하세요.

테이블이 재동기화 필요 또는 재동기화 시작됨 상태입니다.

테이블이 재동기화 필요 또는 재동기화 시작됨 상태입니다.

테이블이 해당 상태인 이유에 대한 자세한 오류 정보를 수집하려면 [SYS_LOAD_ERROR_DETAIL](#) 시스템 뷰를 사용하세요.

데이터베이스 쿼리

Amazon Redshift 클러스터에서 호스팅하는 데이터베이스를 쿼리하려면 다음 2가지 옵션이 있습니다.

- 클러스터에 연결한 다음 쿼리 편집기를 사용해 AWS Management Console에 대한 쿼리를 실행합니다.

Amazon Redshift 콘솔에서 쿼리 편집기를 사용하는 경우 SQL 클라이언트 애플리케이션을 다운로드해 설치할 필요가 없습니다.

- SQL 클라이언트 도구(예: SQL Workbench/J)를 통해 클러스터에 연결합니다.

Amazon Redshift는 JDBC(Java Database Connectivity) 및 ODBC(Open Database Connectivity)를 통해 연결하는 SQL 클라이언트 도구를 지원합니다. Amazon Redshift는 SQL 클라이언트 도구 또는 라이브러리를 제공하거나 설치하지 않으므로 이를 사용하려면 클라이언트 컴퓨터 또는 Amazon EC2 인스턴스에 설치해야 합니다. JDBC 또는 ODBC 드라이버를 지원하는 SQL 클라이언트 도구라면 대부분 사용할 수 있습니다.

Note

저장 프로시저를 작성할 때는 민감한 값을 보호하기 위한 모범 사례를 참조하는 것이 좋습니다.

민감한 정보를 저장 프로시저 로직 내에 하드 코딩하지 마세요. 예를 들어, 저장 프로시저 본문에 CREATE USER 문에 사용자 암호를 할당하지 마세요. 하드 코딩된 값이 카탈로그 테이블에 스키마 메타 데이터로 기록될 수 있기 때문에 보안 위험이 따릅니다. 암호와 같은 민감한 값은 파라미터를 사용하여 저장 프로시저에 인수로 전달하세요.

저장된 프로시저에 대한 자세한 내용은 [프로시저 생성](#) 및 [Amazon Redshift에서 저장 프로시저 생성](#)을 참조하세요. 카탈로그 테이블에 대한 자세한 내용은 [시스템 카탈로그 테이블](#)을 참조하세요.

Amazon Redshift에 연결

다음 구문을 사용하여 데이터베이스에 연결할 수 있습니다.

```
cluster-name.account-number.aws-region.redshift.amazonaws.com/database-name
```

구문 요소는 다음과 같이 정의됩니다.

- `cluster-name`

클러스터 이름입니다.

- `account-number`

지정된 AWS 리전의 AWS 계정 번호와 연결된 고유 식별자입니다. 지정된 AWS 리전의 특정 계정으로 생성된 모든 클러스터는 동일한 `account-number`를 갖습니다.

- `aws-region`

클러스터가 속한 AWS 리전에 대한 코드입니다.

- `database-name`

데이터베이스 이름입니다.

예를 들어 다음 연결 문자열은 us-east-1 AWS 리전의 `my-cluster` 클러스터에 있는 `my-db` 데이터베이스를 지정합니다.

```
my-cluster.123456789012.us-east-1.redshift.amazonaws.com/my-db
```

Query Editor V2를 사용하여 데이터베이스 쿼리

쿼리 편집기 v2는 Amazon Redshift 데이터 웨어하우스에서 쿼리를 작성하고 실행하는 데 사용하는 별도의 웹 기반 SQL 클라이언트 애플리케이션입니다. 쿼리 편집기 v2는 주로 쿼리를 편집 및 실행하고, 결과를 시각화하고, 팀과 작업을 공유하는 데 사용됩니다. 쿼리 편집기 v2를 사용하여 데이터베이스, 스키마, 테이블 및 사용자 정의 함수(UDF)를 생성할 수 있습니다. 트리 보기 패널에서 각 데이터베이스의 스키마를 볼 수 있습니다. 각 스키마의 테이블, 보기, UDF 및 저장 프로시저를 볼 수 있습니다. 쿼리 편집기 v2는 이전 쿼리 편집기를 대체합니다.

Note

쿼리 에디터 v2는 상용 AWS 리전에서 사용할 수 있습니다. 쿼리 에디터 v2를 사용할 수 있는 AWS 리전 목록은 Amazon Web Services 일반 참조의 [Redshift 쿼리 에디터 v2](#)에 나열된 엔트리를 참조하세요.

쿼리 편집기 v2의 데모를 보려면 다음 동영상을 시청하세요. [Amazon Redshift 쿼리 편집기 v2](#)

데이터 분석 데모를 보려면 다음 동영상을 시청하세요. [Amazon Redshift 쿼리 편집기 v2를 사용한 데이터 분석](#)

격리된 연결이나 공유 연결로 여러 쿼리를 실행하는 쿼리 편집기 v2를 사용하는 데모를 보려면 다음 동영상을 시청하세요. [쿼리 편집기 v2를 사용한 동시 쿼리 실행](#).

쿼리 편집기 v2에는 SQL 문을 관리하고 실행하는 데 사용할 수 있는 다양한 기능이 있습니다. 다음 섹션의 주제에서 이러한 다양한 기능을 시작할 수 있습니다. 쿼리 편집기 v2를 직접 탐색하여 기능을 익힙니다.

AWS 계정 구성

이 태스크 세트를 수행하여 쿼리 에디터 v2가 Amazon Redshift 데이터베이스를 쿼리하도록 구성할 수 있습니다. 적절한 권한을 가지고 현재 AWS 리전에 있는 AWS 계정 소유의 Amazon Redshift 클러스터 또는 작업 그룹의 데이터에 액세스할 수 있습니다.

관리자가 AWS 계정에 대해 쿼리 편집기 v2를 처음 구성할 때 쿼리 편집기 v2 리소스를 암호화하는 데 사용되는 AWS KMS key를 선택합니다. 기본적으로 AWS 소유의 키가 리소스를 암호화하는 데 사용됩니다. 또는 관리자가 구성 페이지에서 키에 대한 Amazon 리소스 이름(ARN)을 선택하여 고객 관리형 키를 사용할 수 있습니다.

계정을 구성한 후에는 AWS KMS 암호화 설정을 변경할 수 없습니다. 쿼리 편집기 v2에서 고객 관리형 키 생성 및 사용에 대한 자세한 내용은 [쿼리 편집기 v2와 함께 사용할 AWS KMS 고객 관리형 키 생성](#) 섹션을 참조하세요. 관리자는 원한다면 파일에서 데이터를 로드하는 등의 일부 기능에서 사용하는 S3 버킷과 경로를 선택할 수도 있습니다. 자세한 내용은 [로컬 파일 설정 및 워크플로우에서 데이터 로드](#) 섹션을 참조하세요.

Amazon Redshift 쿼리 편집기 v2는 인증, 암호화, 격리 및 규정 준수를 지원하여 저장된 데이터와 전송 중인 데이터를 안전하게 보호합니다. 데이터 보안 및 쿼리 편집기 v2에 대한 자세한 내용은 다음 섹션을 참조하세요.

- [저장 시 암호화](#)
- [전송 중 암호화](#)
- [Amazon Redshift의 구성 및 취약성 분석](#)

AWS CloudTrail은 직접 수행하거나 AWS 계정을 대신하여 수행한 API 호출 및 관련 이벤트를 캡처하고 지정한 Amazon S3 버킷에 로그 파일을 전송합니다. 어떤 사용자 및 계정이 AWS를 호출했는지 어떤 소스 IP 주소에 호출이 이루어졌는지 언제 호출이 발생했는지 확인할 수 있습니다. 쿼리 편집기 v2

가 AWS CloudTrail에서 어떻게 작동하는지 자세히 알아보려면 [CloudTrail을 사용하여 로깅](#) 섹션을 참조하세요. CloudTrail에 대한 자세한 내용은 [AWS CloudTrail 사용자 안내서](#)를 참조하세요.

쿼리 편집기 v2에는 일부 리소스에 대해 조정 가능한 할당량이 있습니다. 자세한 내용은 [Amazon Redshift 객체에 대한 할당량](#) 섹션을 참조하세요.

쿼리 편집기 v2로 생성된 리소스

쿼리 편집기 v2 내에서 저장된 쿼리 및 차트와 같은 리소스를 생성할 수 있습니다. 쿼리 에디터 v2의 모든 리소스는 IAM 역할 또는 사용자와 연결됩니다. 정책을 IAM 역할에 연결하고 사용자에게 역할을 할당하는 것이 좋습니다.

쿼리 편집기 v2에서 저장된 쿼리와 차트에 대한 태그를 추가하고 제거할 수 있습니다. 사용자 정의 IAM 정책을 설정하거나 리소스를 검색할 때 이러한 태그를 사용할 수 있습니다. AWS Resource Groups 태그 편집기를 사용하여 태그를 관리할 수도 있습니다.

IAM 정책으로 IAM 역할을 설정하여 AWS 리전의 동일한 AWS 계정에 있는 다른 사용자와 쿼리를 공유할 수 있습니다.

쿼리 편집기 v2와 함께 사용할 AWS KMS 고객 관리형 키 생성

대칭 암호화 고객 관리형 키를 생성하려면:

대칭 암호화 고객 관리형 키를 생성하여 AWS KMS 콘솔 또는 AWS KMS API 작업으로 쿼리 편집기 v2 리소스를 암호화할 수 있습니다. 키 생성에 대한 지침은 AWS Key Management Service 개발자 안내서의 [대칭 암호화 AWS KMS 키 생성](#)을 참조하세요.

키 정책

키 정책은 고객 관리형 키에 대한 액세스를 제어합니다. 모든 고객 관리형 키에는 키를 사용할 수 있는 사람과 키를 사용하는 방법을 결정하는 문장이 포함된 정확히 하나의 키 정책이 있어야 합니다. 고객 관리형 키를 생성할 때 키 정책을 지정할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 [AWS KMS 키에 대한 액세스 관리](#)를 참조하세요.

Amazon Redshift 쿼리 편집기 v2에서 고객 관리형 키를 사용하려면 키 정책에서 다음 API 작업을 허용해야 합니다.

- kms:GenerateDataKey - 데이터를 암호화하기 위해 고유한 대칭 데이터 키를 생성합니다.
- kms:Decrypt - 고객 관리형 키로 암호화된 데이터를 복호화합니다.
- kms:DescribeKey - 서비스에서 키를 검증할 수 있도록 고객 관리형 키 세부 정보를 제공합니다.

다음은 AWS 계정 111122223333의 샘플 AWS KMS 정책입니다. 첫 번째 섹션에서 kms:ViaService는 쿼리 편집기 v2 서비스(정책에서 sqlworkbench.**region**.amazonaws.com으로 명명됨)에 대한 키 사용을 제한합니다. 키를 사용하는 AWS 계정은 111122223333이어야 합니다. 두 번째 섹션에서는 AWS 계정 111122223333의 루트 사용자 및 키 관리자가 키에 액세스할 수 있습니다.

AWS 계정을 생성하는 경우에는 모든 AWS 서비스 서비스와 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 루트 사용자라는 단일 로그인 ID로 시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명이 필요한 작업은 IAM 사용 설명서의 [루트 사용자 자격 증명이 필요한 작업](#) 섹션을 참조하세요.

JSON

```
{
  "Version": "2012-10-17",
  "Id": "key-consolepolicy",
  "Statement": [
    {
      "Sid": "Allow access to principals authorized to use Amazon Redshift
Query Editor V2",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "sqlworkbench.us-east-1.amazonaws.com",
          "kms:CallerAccount": "111122223333"
        }
      }
    },
    {
      "Sid": "Allow access for key administrators",
      "Effect": "Allow",
      "Principal": {
```

```

        "AWS": "arn:aws:iam::111122223333:root"
    },
    "Action": [
        "kms:*"
    ],
    "Resource": "arn:aws:kms:us-east-1:111122223333:key/key_ID"
}
]
}

```

다음 리소스에서 AWS KMS 키에 대한 추가 정보를 확인할 수 있습니다.

- AWS KMS 정책에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [정책에서 권한 지정](#)을 참조하세요.
- AWS KMS 정책 문제 해결에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [키 액세스 문제 해결](#)을 참조하세요.
- 키에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [AWS KMS 키](#)를 참조하세요.

쿼리 편집기 v2에 액세스

쿼리 편집기 v2에 액세스하려면 권한이 있어야 합니다. 관리자는 다음 AWS 관리형 정책 중 하나를 역할에 연결하여 권한을 부여할 수 있습니다. (IAM 역할에 정책을 연결하고 사용자에게 역할을 할당하는 것이 좋습니다.) 이러한 AWS 관리형 정책은 리소스 태깅이 쿼리 공유를 허용하는 방법을 제어하는 다양한 옵션으로 작성됩니다. IAM 콘솔(<https://console.aws.amazon.com/iam/>)을 사용하여 IAM 사용자 정책을 연결할 수 있습니다.

- AmazonRedshiftQueryEditorV2FullAccess – Amazon Redshift 쿼리 편집기 v2 작업 및 리소스에 대한 전체 액세스 권한을 부여합니다. 이 정책은 다른 필수 서비스에 대한 액세스 권한도 부여합니다.
- AmazonRedshiftQueryEditorV2NoSharing – 리소스를 공유하지 않고 Amazon Redshift 쿼리 편집기 v2 관련 작업을 수행할 수 있는 권한을 부여합니다. 이 정책은 다른 필수 서비스에 대한 액세스 권한도 부여합니다.
- AmazonRedshiftQueryEditorV2ReadSharing – 제한된 리소스 공유로 Amazon Redshift 쿼리 편집기 v2 관련 작업을 수행할 수 있는 권한을 부여합니다. 부여된 보안 주체는 팀과 공유된 리소스를 읽을 수 있지만 업데이트할 수는 없습니다. 이 정책은 다른 필수 서비스에 대한 액세스 권한도 부여합니다.

- AmazonRedshiftQueryEditorV2ReadWriteSharing – 리소스 공유와 함께 Amazon Redshift 쿼리 편집기 v2 관련 작업을 수행할 수 있는 권한을 부여합니다. 부여된 보안 주체는 팀과 공유하는 리소스를 읽고 업데이트할 수 있습니다. 이 정책은 다른 필수 서비스에 대한 액세스 권한도 부여합니다.

제공된 관리형 정책에서 허용 및 거부된 권한을 기반으로 고유한 정책을 생성할 수도 있습니다. IAM 콘솔 정책 편집기를 사용하여 고유한 정책을 생성하는 경우 시각적 편집기에서 정책을 생성할 서비스로 SQL Workbench를 선택합니다. 쿼리 편집기 v2는 시각적 편집기와 IAM 정책 시뮬레이터에서 서비스 이름으로 AWS SQL Workbench를 사용합니다.

보안 주체(IAM 역할이 할당된 사용자)가 Amazon Redshift 클러스터에 연결하려면 쿼리 에디터 v2 관리형 정책 중 하나에서 권한이 필요합니다. 클러스터에 대한 `redshift:GetClusterCredentialsWithIAM` 또는 `redshift:GetClusterCredentials` 권한 중 하나도 필요합니다. 이 권한을 얻기 위해 관리 권한이 있는 사용자가 임시 자격 증명을 사용하여 클러스터에 연결해야 하는 IAM 역할에 정책을 연결할 수 있습니다. 정책의 범위를 특정 클러스터로 지정하거나 보다 일반적으로 지정할 수 있습니다. 임시 자격 증명 사용 권한에 대한 자세한 내용은 [GetClusterCredentialsWithIAM 또는 GetClusterCredentials 직접 호출 권한이 있는 IAM 역할 또는 사용자 생성](#)을 참조하세요.

보안 주체(일반적으로 IAM 역할이 할당된 사용자)가 Account settings(계정 설정) 페이지에서 계정의 다른 사용자가 Export result set(결과 집합 내보내기) 기능을 사용할 수 있도록 설정하려면 해당 역할에 연결된 `sqlworkbench:UpdateAccountExportSettings` 권한이 필요합니다. 이 권한은 AmazonRedshiftQueryEditorV2FullAccess AWS 관리형 정책에 포함되어 있습니다.

쿼리 편집기 v2에 새로운 기능이 추가되면 필요에 따라 AWS 관리형 정책이 업데이트됩니다. 제공된 관리형 정책에서 허용 및 거부된 권한을 기반으로 자체 정책을 생성하는 경우 정책을 편집하여 관리형 정책의 변경 사항을 최신 상태로 유지하세요. Amazon Redshift의 관리형 정책에 대한 자세한 내용은 [Amazon Redshift에 대한 AWS 관리형 정책](#) 섹션을 참조하세요.

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요.

- AWS IAM Identity Center의 사용자 및 그룹:

권한 세트를 생성합니다. AWS IAM Identity Center 사용자 안내서에서 [권한 세트 생성](#)의 지침을 따릅니다.

- ID 제공업체를 통해 IAM에서 관리되는 사용자:

ID 페더레이션을 위한 역할을 생성합니다. IAM 사용자 설명서의 [Create a role for a third-party identity provider \(federation\)](#)의 지침을 따릅니다.

- IAM 사용자:

- 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용자 설명서에서 [Create a role for an IAM user](#)의 지침을 따릅니다.
- (권장되지 않음) 정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용자 설명서에서 [사용자\(콘솔\)에 권한 추가](#)의 지침을 따릅니다.

Note

AWS IAM Identity Center 관리자가 전체 계정의 특정 권한 세트에 대한 모든 권한 세트 연결을 제거하면 제거된 권한 세트와 원래 연결된 쿼리 에디터 리소스에 더 이상 액세스할 수 없습니다. 나중에 동일한 권한을 다시 만들면 새 내부 식별자가 만들어집니다. 내부 식별자가 변경되었으므로 사용자가 이전에 소유했던 쿼리 에디터 리소스에 액세스할 수 없습니다. 관리자가 권한 세트를 삭제하기 전에 해당 권한 세트의 사용자가 노트북 및 쿼리와 같은 쿼리 에디터 리소스를 백업으로 내보내는 것이 좋습니다.

쿼리 에디터 v2에서 클러스터 또는 작업 그룹을 연결하기 위한 보안 주체 태그 설정

페더레이션 사용자 옵션을 사용하여 클러스터 또는 작업 그룹에 연결하려면 보안 주체 태그를 사용하여 IAM 역할 또는 사용자를 설정합니다. 또는 RedshiftDbUser 및 RedshiftDbGroups(선택 사항)을 전달하도록 아이덴티티 제공업체(IdP)를 설정합니다. IAM을 사용하여 태그를 관리하는 방법에 대한 자세한 내용은 IAM 사용자 설명서의 [AWS Security Token Service에서 세션 태그 전달](#)을 참조하세요. AWS Identity and Access Management를 사용하여 액세스를 설정하려는 경우, 관리자는 IAM 콘솔(<https://console.aws.amazon.com/iam/>)을 사용하여 태그를 추가할 수 있습니다.

IAM 역할에 보안 주체 태그를 추가하려면

1. AWS Management Console에 로그인하여 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 페더레이션 사용자를 사용하여 쿼리 편집기 v2에 대한 액세스 권한이 필요한 역할을 선택합니다.
4. 태그 탭을 선택합니다.
5. 태그 관리(Manage tags)를 선택합니다.
6. 태그 추가를 선택한 다음 키에 RedshiftDbUser를 입력하고 값에 페더레이션 사용자 이름을 입력합니다.
7. 필요시 태그 추가를 선택한 다음 키에 RedshiftDbGroups를 입력하고 값에 사용자에게 연결할 그룹 이름을 입력합니다.

8. 변경 사항 저장(Save changes)을 선택하여 선택한 IAM 역할에 연결된 태그 목록을 봅니다. 변경 사항을 전파하는 데 몇 초 정도 걸릴 수 있습니다.
9. 페더레이션 사용자를 사용하려면 변경 사항이 전파된 후 쿼리 편집기 v2 페이지를 새로 고칩니다.

보안 주체 태그를 전달하도록 아이덴티티 제공업체(IdP) 설정

아이덴티티 제공업체(IdP)를 사용하여 태그를 설정하는 절차는 IdP에 따라 다릅니다. 사용자 및 그룹 정보를 SAML 속성에 전달하는 방법에 대한 지침은 IdP 설명서를 참조하세요. 올바르게 구성된 경우 AWS Security Token Service에서 RedshiftDbUser 및 RedshiftDbGroups에 대한 보안 주체 태그를 채우는 데 사용하는 SAML 응답에 다음과 같은 속성이 나타납니다.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:RedshiftDbUser">
  <AttributeValue>db-user-name</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:RedshiftDbGroups">
  <AttributeValue>db-groups</AttributeValue>
</Attribute>
```

선택적 *db_groups*는 group1:group2:group3와 같이 콜론으로 구분된 목록이어야 합니다.

또한 역할을 함께 묶을 때 태그를 유지하도록 TransitiveTagKeys 속성을 설정할 수 있습니다.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys">
  <AttributeValue>RedshiftDbUser</AttributeValue>
  <AttributeValue>RedshiftDbGroups</AttributeValue>
</Attribute>
```

쿼리 편집기 v2 설정에 대한 자세한 내용은 [쿼리 편집기 v2를 사용하는 데 필요한 권한](#) 섹션을 참조하세요.

Note

쿼리 에디터 v2의 페더레이션된 사용자 연결 옵션을 사용하여 클러스터 또는 작업 그룹에 연결하는 경우 ID 제공업체(idP)는 RedshiftDbUser 및 RedshiftDbGroups에 대한 사용자 지정 보안 주체 태그를 제공할 수 있습니다. 현재 AWS IAM Identity Center에서는 쿼리 에디터 v2에 사용자 지정 보안 주체 태그를 직접 전달하는 기능을 지원하지 않습니다.

쿼리 편집기 v2 열기

Amazon Redshift를 사용하면 Amazon Redshift 콘솔의 Query Editor V2를 사용하여 데이터 웨어하우스 클러스터에 대해 SQL 쿼리를 실행할 수 있습니다. Query Editor V2는 임시 쿼리 실행, 데이터 탐색 및 데이터 분석 작업 수행 용도로 사용자가 사용하기에 편리한 인터페이스를 제공하는 웹 기반 도구입니다. 다음 섹션에서는 콘솔에서 Query Editor V2를 열고 해당 기능을 효과적으로 활용하는 프로세스를 안내합니다.

쿼리 편집기 v2를 열려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색기 메뉴에서 편집기(Editor), 쿼리 편집기 V2(Query editor V2)를 차례로 선택합니다. 쿼리 편집기 v2가 새 브라우저 탭에서 열립니다.

쿼리 편집기 페이지에는 다음과 같이 보기를 선택할 수 있는 네비게이터 메뉴가 있습니다.

편집기



테이블로 정리되고 데이터베이스에 포함된 데이터를 관리하고 쿼리합니다. 데이터베이스는 저장된 데이터를 포함하거나 Amazon S3와 같은 다른 곳에 저장된 데이터에 대한 참조를 포함할 수 있습니다. 클러스터 또는 서버리스 작업 그룹에 들어 있는 데이터베이스에 연결합니다.

Editor(편집기) 보기에서 작업할 때는 다음 컨트롤을 사용할 수 있습니다.

- Cluster(클러스터) 또는 Workgroup(작업 그룹) 필드에는 현재 연결되어 있는 이름이 표시됩니다. Database(데이터베이스) 필드에는 클러스터 또는 작업 그룹 내의 데이터베이스가 표시됩니다. 데이터베이스(Database) 보기에서 수행하는 작업은 기본적으로 선택한 데이터베이스에 작용합니다.
- 클러스터 또는 작업 그룹, 데이터베이스 및 스키마의 트리 보기 계층 보기. 스키마에서 테이블, 보기, 함수 및 저장 프로시저 관련 작업을 수행할 수 있습니다. 트리 보기의 각 객체는 객체에 대해 새로 고침(Refresh) 또는 삭제(Drop)와 같은 관련 작업을 수행하는 컨텍스트 메뉴를 지원합니다.
- 데이터베이스, 스키마, 테이블 및 함수를 생성하는



생성(Create) 작업



이터 로드 작업은 Amazon S3 또는 로컬 파일에서 데이터베이스에 데이터를 로드합니다.

- 쿼리를 저장하는



저장(Save) 아이콘

- 편집기의 바로 가기 키를 표시하는



바로 가기(Shortcuts) 아이콘

- 편집기에 더 많은 작업을 표시하는



더 보기 아이콘 예:

- 내 팀과 공유를 사용하여 팀과 쿼리 또는 노트북을 공유합니다. 자세한 내용은 [팀으로 협업 및 공유](#) 섹션을 참조하세요.
- 단축키를 사용하여 편집기의 키보드 단축키를 표시합니다.
- 탭 기록을 사용하여 편집기에 탭의 탭 기록을 표시합니다.
- 자동 완성 새로 고침을 사용하여 SQL을 작성할 때 표시된 제안을 새로 고칩니다.
- 쿼리를 입력하고 실행할 수 있는



편집기 영역입니다.

쿼리를 실행하면 결과와 함께 결과(Result) 탭이 나타납니다. 여기에서 차트(Chart)를 설정하여 결과를 시각화할 수 있습니다. 결과를 Export(내보내기)할 수도 있습니다.

- 섹션을 추가하여 SQL을 입력하고 실행하거나 마크다운을 추가할 수 있는



Notebook(노트북) 영역입니다.

쿼리를 실행하면 결과와 함께 결과(Result) 탭이 나타납니다. 여기에서 결과를 Export(내보내기)할 수 있습니다.

쿼리



쿼리에는 데이터베이스의 데이터를 관리하고 쿼리하는 SQL 명령이 포함되어 있습니다. 쿼리 편집기 v2를 사용하여 샘플 데이터를 로드할 때 샘플 쿼리도 생성되고 저장됩니다.

저장된 쿼리를 선택하면 컨텍스트(마우스 오른쪽 버튼) 메뉴를 사용하여 쿼리를 열거나 이름을 바꾸거나 삭제할 수 있습니다. 쿼리 세부 정보를 선택하여 저장된 쿼리의 쿼리 ARN과 같은 속성을 볼 수 있습니다. 버전 기록을 보고, 쿼리에 연결된 태그를 편집하고, 팀과 공유할 수도 있습니다.

노트북



SQL 노트북에는 SQL 및 마크다운 셀이 포함되어 있습니다. 노트북을 사용하여 하나의 문서에서 여러 SQL 명령을 구성하고 주석을 달고 공유할 수 있습니다.

저장된 노트북을 선택하면 컨텍스트(마우스 오른쪽 버튼) 메뉴를 사용하여 노트북을 열거나 이름을 바꾸거나 복제하거나 삭제할 수 있습니다. 노트북 세부 정보를 선택하여 저장된 노트북의 노트북 ARN과 같은 속성을 볼 수 있습니다. 버전 기록을 보고, 노트북에 연결된 태그를 편집하고, 내보내고, 팀과 공유할 수도 있습니다. 자세한 내용은 [Amazon Redshift의 노트북](#) 섹션을 참조하세요.

차트



차트는 데이터를 시각적으로 표현한 것입니다. 쿼리 편집기 v2는 다양한 유형의 차트를 생성하고 저장할 수 있는 도구를 제공합니다.

저장된 차트를 선택하면 컨텍스트(마우스 오른쪽 버튼) 메뉴를 사용하여 쿼리를 열거나 이름을 바꾸거나 삭제할 수 있습니다. 차트 세부 정보를 선택하여 저장된 차트의 차트 ARN과 같은 속성을 볼 수 있습니다. 차트에 연결된 태그를 편집하고 내보낼 수도 있습니다. 자세한 내용은 [쿼리 결과 시각화](#) 섹션을 참조하세요.

기록



쿼리 기록은 Amazon Redshift 쿼리 에디터 v2를 사용하여 실행한 쿼리 목록입니다. 이러한 쿼리는 개별 쿼리로 실행되거나 SQL 노트북의 일부로 실행되었습니다. 자세한 내용은 [쿼리 및 탭 기록 보기](#) 섹션을 참조하세요.

예약된 쿼리



예약된 쿼리는 특정 시간에 시작되도록 설정된 쿼리입니다.

모든 쿼리 편집기 v2 보기에는 다음과 같은 아이콘이 있습니다.



비주얼 모드 아이콘을 사용하면 라이트 모드와 다크 모드를 전환할 수 있습니다.



설정 아이콘을 사용하면 다양한 설정 화면의 메뉴가 표시됩니다.



편집기 기본 설정 아이콘을 사용하면 쿼리 편집기 v2를 사용할 때 기본 설정을 편집할 수 있습니다. 여기서 Workspace 설정 편집으로 글꼴 크기, 탭 크기 및 기타 표시 설정을 변경할 수 있습니다. SQL을 입력할 때 제안 사항을 표시하도록 자동 완성을 켜거나 끌 수도 있습니다.



연결 아이콘을 사용하면 편집기 탭에서 사용하는 연결을 볼 수 있습니다.

연결은 데이터베이스에서 데이터를 가져오는 데 사용됩니다. 연결은 특정 데이터베이스에 대해 생성됩니다. 격리된 연결을 사용하면 한 편집기 탭에서 임시 테이블을 생성하는 것과 같이 데이터베이스를 변경하는 SQL 명령의 결과가 다른 편집기 탭에 표시되지 않습니다. 쿼리 편집기 v2에서 편집기 탭을 열 때 기본값은 격리된 연결입니다. 공유 연결을 생성할 때, 즉 Isolated session(격리된 세션) 스위치를 해제하면 동일한 데이터베이스에 대한 다른 공유 연결의 결과가 서로 표시됩니다. 그러나 데이터베이스에 대한 공유 연결을 사용하는 편집기 탭은 병렬로 실행되지 않습니다. 동일한 연결을 사용하는 쿼리는 연결이 가능해질 때까지 기다려야 합니다. 한 데이터베이스에 대한 연결은 다른 데이터베이스와 공유될 수 없으므로 다른 데이터베이스 연결에서 SQL 결과를 볼 수 없습니다.

계정의 사용자가 활성화할 수 있는 연결 수는 쿼리 편집기 v2 관리자가 제어합니다.



계정 설정 아이콘은 관리자가 계정에 있는 모든 사용자의 특정 설정을 변경하는 데 사용됩니다. 자세한 내용은 [계정 설정](#) 섹션을 참조하세요.

쿼리 편집기 v2로 작업할 때 고려할 사항

쿼리 편집기 v2로 작업할 때는 다음 사항을 고려하세요.

- 최대 쿼리 기간은 24시간입니다.
- 최대 쿼리 결과 크기는 100MB입니다. 직접 호출이 100MB를 초과하는 응답 데이터를 반환하면 처음 100MB가 경고와 함께 반환됩니다.
- 최대 300,000자의 쿼리를 실행할 수 있습니다.
- 최대 30,000자의 쿼리를 저장할 수 있습니다.
- 기본적으로 쿼리 편집기 v2는 실행되는 각 개별 SQL 명령을 자동으로 커밋합니다. BEGIN 문이 제공되면 BEGIN-COMMIT 또는 BEGIN-ROLLBACK 블록 내의 문이 단일 트랜잭션으로 실행됩니다. 트랜잭션에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [BEGIN](#) 섹션을 참조하세요.
- 쿼리 에디터 v2에서 SQL 문을 실행하는 동안 표시되는 최대 경고 수는 10개입니다. 예를 들어 저장 프로시저가 실행되면 RAISE 문이 10개 넘게 표시되지 않습니다.
- 쿼리 에디터 v2는 쉼표(,)가 포함된 IAM RoleSessionName을 지원하지 않습니다. 다음과 비슷한 오류가 표시될 수 있습니다. 오류 메시지: “‘AROIA123456789EXAMPLE:mytext,yourtext’ is not a valid value for TagValue - it contains illegal characters” 이 문제는 쉼표가 포함된 RoleSessionName IAM을 정의한 다음 해당 IAM 역할로 쿼리 에디터 v2를 사용할 때 발생합니다.

IAM RoleSessionName에 대한 자세한 내용은 IAM 사용 설명서의 [RoleSessionName SAML 속성](#)을 참조하세요.

계정 설정

적절한 IAM 권한을 가진 사용자는 같은 AWS 계정 내의 다른 사용자에게 대한 Account settings(계정 설정)를 보고 변경할 수 있습니다. 이 관리자는 다음을 보거나 설정할 수 있습니다.

- 계정의 사용자당 최대 동시 데이터베이스 연결 수. 여기에는 Isolated sessions(격리된 세션)에 대한 연결이 포함됩니다. 이 값을 변경할 때 변경이 적용되기까지 10분이 걸릴 수 있습니다.
- 계정의 사용자가 SQL 명령의 전체 결과 세트를 파일로 내보낼 수 있도록 허용합니다.
- 일부 연결된 저장된 쿼리가 있는 샘플 데이터베이스를 로드하고 표시합니다.
- 계정 사용자가 로컬 파일에서 데이터를 로드하는 데 사용하는 Amazon S3 경로를 지정합니다.
- 쿼리 편집기 v2 리소스를 암호화하는 데 사용되는 KMS 키 ARN을 알아봅니다.

Amazon Redshift 데이터베이스에 연결

데이터베이스에 연결하려면 트리 보기 패널에서 클러스터 또는 작업 그룹 이름을 선택합니다. 메시지가 나타나면 연결 파라미터를 입력합니다.

클러스터 또는 작업 그룹과 해당 데이터베이스에 연결할 때 일반적으로 Database(데이터베이스) 이름을 제공합니다. 다음 인증 방법 중 하나에 필요한 파라미터도 제공합니다.

IAM Identity Center

이 방법을 사용하면 ID 제공업체(idP)의 Single Sign-On 보안 인증 정보를 사용하여 Amazon Redshift 데이터 웨어하우스에 연결할 수 있습니다. Amazon Redshift 콘솔에서 클러스터 또는 작업 그룹에 IAM Identity Center를 활성화해야 합니다. IAM Identity Center에 대한 연결을 설정하는 데 도움이 필요한 경우 [Single Sign-On 경험을 위해 Redshift와 AWS IAM Identity Center 연결](#) 섹션을 참조하세요.

페더레이션 사용자

이 방법을 사용하는 경우 IAM 역할 또는 사용자의 보안 주체 태그에서 연결 세부 정보를 제공해야 합니다. 이러한 태그는 AWS Identity and Access Management 또는 아이덴티티 제공업체(IdP)에서 구성합니다. 쿼리 편집기 v2는 다음과 같은 태그를 사용합니다.

- RedshiftDbUser - 이 태그는 쿼리 편집기 v2에서 사용되는 데이터베이스 사용자를 정의합니다. 이 태그는 필수입니다.
- RedshiftDbGroups - 이 태그는 쿼리 편집기 v2에 연결할 때 조인되는 데이터베이스 그룹을 정의합니다. 이 태그는 선택 사항이며 해당 값은 group1:group2:group3와 같이 콜론으로 구분된 목록이어야 합니다. 빈 값은 무시됩니다. 즉, group1:::group2는 group1:group2로 해석됩니다.

이러한 태그는 클러스터에 대한 보안 인증 정보를 가져오는 `redshift:GetClusterCredentials` API로 전달됩니다. 자세한 내용은 [쿼리 에디터 v2에서 클러스터 또는 작업 그룹을 연결하기 위한 보안 주체 태그 설정](#) 섹션을 참조하세요.

데이터베이스 사용자 이름을 사용하는 임시 보안 인증 정보

이 옵션은 클러스터에 연결할 때만 사용할 수 있습니다. 이 메서드(쿼리 편집기 v2)를 사용하여 데이터베이스의 User name(사용자 이름)을 제공합니다. 쿼리 에디터 v2가 데이터베이스 사용자 이름으로 데이터베이스에 연결하기 위한 임시 암호를 생성합니다. 이 메서드를 사용하여 연결하는 사용자에게는 `redshift:GetClusterCredentials`에 대한 IAM 권한이 허용되어야 합니다. 사용자가 이 메서드를 사용하지 못하도록 하려면 IAM 사용자 또는 역할을 수정하여 이 권한을 거부합니다.

IAM 자격 증명을 사용한 임시 자격 증명

이 옵션은 클러스터에 연결할 때만 사용할 수 있습니다. 이 방법을 사용하면 쿼리 에디터 v2가 사용자 이름을 IAM ID에 매핑하고 IAM ID로 데이터베이스에 연결하기 위한 임시 암호를 생성합니다. 이 메서드를 사용하여 연결하는 사용자에게는 `redshift:GetClusterCredentialsWithIAM`에 대한 IAM 권한이 허용되어야 합니다. 사용자가 이 메서드를 사용하지 못하도록 하려면 IAM 사용자 또는 역할을 수정하여 이 권한을 거부합니다.

데이터베이스 사용자 이름 및 암호

이 방법을 사용하는 경우 연결하려는 데이터베이스의 사용자 이름(User name) 및 암호(Password)도 제공합니다. 쿼리 편집기 v2는 AWS Secrets Manager에 저장된 암호를 대신 생성합니다. 이 보안 암호에는 데이터베이스에 연결하기 위한 자격 증명이 포함되어 있습니다.

AWS Secrets Manager

이 메서드를 사용하면 데이터베이스 이름 대신 데이터베이스 및 로그인 자격 증명이 포함된 Secrets Manager에 저장된 Secret(비밀)을 제공합니다. 보안 암호 생성에 대한 정보는 [데이터베이스 연결 자격 증명을 위한 보안 암호 생성](#) 섹션을 참조하세요.

쿼리 편집기 v2로 클러스터 또는 작업 그룹을 선택하면 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 사용하여 컨텍스트에 따라 연결을 생성, 편집 및 삭제할 수 있습니다. 연결 세부 정보를 선택하여 연결의 연결 ARN과 같은 속성을 볼 수 있습니다. 연결에 연결된 태그를 편집할 수도 있습니다.

Amazon Redshift 데이터베이스 찾아보기

트리 보기 패널에서 데이터베이스 내의 스키마, 테이블, 보기, 함수 및 저장 프로시저를 관리할 수 있습니다. 보기의 각 객체에는 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴에서 연결된 작업이 있습니다.

계층적 트리 보기 패널에는 데이터베이스 객체가 표시됩니다. 트리 보기가 마지막으로 표시된 이후에 생성되었을 수 있는 데이터베이스 객체를 표시하도록 트리 보기 패널을 새로 고치려면



아이콘을 선택합니다. 객체에 대한 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열어 수행할 수 있는 작업을 확인합니다.

▼  redshift-cluster-tickit	
▼  dev	
▼  public	
▼  Tables	11
 accommodations	
 category	
 customer_activity	
 date	
 event	
 listing	
 sales	
 sales2	
 users	
 venue	
 zipcode	
▼  Views	1
 myevent	
▼  Functions	2
<i>fx</i> f_py_greater(float8,float8)	
<i>fx</i> f_sql_greater(float8,float8)	
▼  Stored procedures	1
<i>fx</i> test_sp1(int4,varchar)	
>  testschema	
>  testschema2	
▼  sample_data_dev	
▼  tickit	
>  Tables	7
>  Views	0
>  Functions	0
>  Stored procedures	0
>  tpcds	
>  testdb	

테이블을 선택한 후 다음을 수행할 수 있습니다.

- 테이블의 모든 열을 쿼리하는 SELECT 문으로 편집기에서 쿼리를 시작하려면 테이블 선택(Select table)을 사용합니다.
- 속성이나 테이블을 보려면 테이블 정의 표시(Show table definition)를 사용합니다. 열 이름, 열 유형, 인코딩, 배포 키, 정렬 키 및 열에 null 값이 포함될 수 있는지 여부를 보려면 이 작업을 사용합니다. 테이블 속성에 대한 자세한 내용은 Amazon Redshift Database 개발자 안내서의 [CREATE TABLE](#)을 참조하세요.
- 테이블을 삭제하려면 삭제>Delete)를 사용합니다. 테이블 자르기(Truncate table)를 사용하여 테이블에서 모든 행을 삭제하거나 테이블 삭제(Drop table)를 사용하여 데이터베이스에서 테이블을 제거할 수 있습니다. 자세한 내용은 Amazon Redshift Database 개발자 안내서의 [TRUNCATE](#) 및 [DROP TABLE](#)을 참조하세요.

새로 고침(Refresh) 또는 스키마 삭제(Drop schema)를 수행할 스키마를 선택합니다.

보기 정의 표시(Show view definition) 또는 보기 삭제(Drop view)를 수행할 보기를 선택합니다.

함수 정의 표시(Show function definition) 또는 함수 삭제(Drop function)를 수행할 함수를 선택합니다.

프로시저 정의 표시(Show procedure definition) 또는 프로시저 삭제(Drop procedure)를 수행할 저장 프로시저를 선택합니다.

데이터베이스 객체 생성

데이터베이스, 스키마, 테이블 및 사용자 정의 함수(UDF)를 비롯한 데이터베이스 객체를 생성할 수 있습니다. 데이터베이스 객체를 생성하려면 클러스터 또는 작업 그룹과 데이터베이스에 연결되어 있어야 합니다.

데이터베이스 생성

쿼리 편집기 v2를 사용하여 클러스터 또는 작업 그룹에 데이터베이스를 생성할 수 있습니다.

데이터베이스를 생성하려면

데이터베이스에 대한 내용은 Amazon Redshift Database 개발자 안내서의 [CREATE DATABASE](#)를 참조하세요.

1.

성(Create)을 선택한 후 데이터베이스(Database)를 선택합니다.

생

2. 데이터베이스 이름(Database name)을 입력합니다.
3. (선택 사항) 사용자 및 그룹(Users and groups)을 선택하고 데이터베이스 사용자(Database user)를 선택합니다.
4. (선택 사항) 데이터 공유 또는 AWS Glue Data Catalog에서 데이터베이스를 생성할 수 있습니다. AWS Glue에 대한 자세한 내용은 AWS Glue 개발자 안내서의 [AWS Glue란 무엇입니까?](#)를 참조하세요.
 - (선택 사항) 데이터 공유를 사용하여 생성을 선택하고 데이터 공유 선택을 선택합니다. 목록에는 현재 클러스터 또는 작업 그룹에서 소비자 데이터 공유를 생성하는 데 사용할 수 있는 생산자 데이터 공유가 포함됩니다.
 - (선택 사항) AWS Glue Data Catalog를 사용하여 생성을 선택하고, AWS Glue 데이터베이스 선택을 선택합니다. 데이터 카탈로그 스키마에서 데이터를 참조할 때 스키마에 사용할 이름을 세 부분으로 구성된 이름(database.schema.table)으로 입력합니다.
5. 데이터베이스 생성을 선택합니다.

새 데이터베이스가 트리 보기 패널에 표시됩니다.

데이터 공유에서 생성된 데이터베이스를 쿼리하는 단계를 필요에 따라 선택하는 경우 클러스터 또는 작업 그룹(예: 기본 데이터베이스 dev)의 Amazon Redshift 데이터베이스에 연결하고, 데이터 공유를 사용하여 생성을 선택할 때 만든 데이터베이스 이름을 참조하는 세 부분으로 구성된 표기법(database.schema.table)을 사용합니다. 데이터 공유 데이터베이스는 쿼리 편집기 v2 편집기 탭에 나열되지만 직접 연결할 수 있도록 설정되어 있지 않습니다.

AWS Glue Data Catalog에서 만든 데이터베이스를 쿼리하는 단계를 필요에 따라 선택하는 경우, 클러스터 또는 작업 그룹(예: 기본 데이터베이스 dev)에서 Amazon Redshift 데이터베이스에 연결하고, AWS Glue Data Catalog를 사용하여 생성을 선택할 때 만든 데이터베이스 이름, 데이터 카탈로그 스키마에서 이름을 지정한 스키마 및 AWS Glue Data Catalog의 테이블을 참조하는 세 부분으로 구성된 표기법(database.schema.table)을 사용하세요. 다음과 유사합니다.

```
SELECT * FROM glue-database.glue-schema.glue-table
```

Note

연결 메서드 IAM ID를 사용하여 임시 보안 인증을 사용하여 기본 데이터베이스에 연결되었는지 그리고 IAM 보안 인증에 AWS Glue에 대한 사용 권한이 부여되었는지 확인합니다.

```
GRANT USAGE ON DATABASE glue-database to "IAM:MyIAMUser"
```


AWS Glue 데이터베이스는 쿼리 편집기 v2 편집기 탭에 나열되지만 직접 연결할 수 있도록 설정되어 있지 않습니다.

AWS Glue Data Catalog 쿼리에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서에서 [소비자로서 레이크 포메이션 관리 데이터 공유로 작업](#) 및 [생산자로서 레이크 포메이션 관리 데이터 공유로 작업](#)을 참조하세요.

데이터 공유 소비자 데이터베이스 생성 예제

다음 예에서는 쿼리 편집기 v2를 사용하여 데이터 공유에서 데이터베이스를 만드는 데 사용된 특정 시나리오에 대해 설명합니다. 이 시나리오를 검토하여 사용 중인 환경에서 데이터 공유에서 데이터베이스를 만드는 방법을 알아보세요. 시나리오에서는 `cluster-base`(생산자 클러스터)와 `cluster-view`(소비자 클러스터)라는 두 개의 클러스터를 사용합니다.

1. Amazon Redshift 콘솔을 사용하여 `cluster-base` 클러스터에서 `category2` 테이블에 대한 데이터 공유를 생성합니다. 생산자 데이터 공유의 이름은 `datashare_base`입니다.

데이터 공유 생성에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서에 [Amazon Redshift에서 클러스터 간에 데이터 공유](#)를 참조하세요.

2. Amazon Redshift 콘솔을 사용하여 `cluster-view` 클러스터에서 `category2` 테이블에 대한 소비자로 데이터셰어 `datashare_base`를 수락합니다.
3. 쿼리 편집기 v2의 트리 뷰 패널에서 `cluster-base`의 계층 구조를 다음과 같이 표시합니다.

- 클러스터: `cluster-base`
 - 데이터베이스: `dev`
 - 스키마: `public`
 - 테이블: `category2`



성(Create)을 선택한 후 데이터베이스(Database)를 선택합니다.

5. 데이터베이스 이름에 `see_datashare_base`를 입력합니다.
6. 데이터 공유를 사용하여 생성을 선택하고 데이터 공유 선택을 선택합니다. 생성하려는 데이터베이스의 소스로 사용할 `datashare_base`를 선택합니다.

쿼리 편집기 v2의 트리 보기 패널에 `cluster-view`의 계층 구조가 다음과 같이 표시됩니다.

생

- 클러스터: cluster-view
 - 데이터베이스: see_datashare_base
 - 스키마: public
 - 테이블: category2
7. 데이터를 쿼리할 때는 cluster-view 클러스터의 기본 데이터베이스(일반적으로 dev라는 이름)에 연결하되, SQL에서 데이터 공유 데이터베이스 see_datashare_base를 참조하세요.

Note

쿼리 편집기 v2 편집기 뷰에서 선택한 클러스터는 cluster-view입니다. 선택한 데이터베이스는 dev입니다. 데이터베이스 see_datashare_base가 나열되어 있지만 직접 연결할 수 있도록 활성화되어 있지 않습니다. dev 데이터베이스를 선택하고 실행하는 SQL에서 see_datashare_base를 참조합니다.

```
SELECT * FROM "see_datashare_base"."public"."category2";
```

이 쿼리는 cluster_base 클러스터의 데이터 공유 datashare_base에서 데이터를 검색합니다.

AWS Glue Data Catalog에서 데이터베이스 생성 예제

다음 예에서는 쿼리 편집기 v2를 사용하여 AWS Glue Data Catalog에서 데이터베이스를 만드는 데 사용된 특정 시나리오에 대해 설명합니다. 이 시나리오를 검토하여 사용 중인 환경에서 AWS Glue Data Catalog에서 데이터베이스를 만드는 방법을 알아보세요. 이 시나리오에서는 하나의 클러스터, cluster-view를 사용하여 생성한 데이터베이스를 포함합니다.

1.  생성(Create)을 선택한 후 데이터베이스(Database)를 선택합니다.
2. 데이터베이스 이름에 data_catalog_database를 입력합니다.
3. AWS Glue Data Catalog를 사용하여 생성을 선택하고, AWS Glue 데이터베이스 선택을 선택합니다. 생성하려는 데이터베이스의 소스로 사용할 glue_db를 선택합니다.

생

데이터 카탈로그 스키마를 선택하고 세 부분으로 구성된 표기법으로 사용할 스키마 이름으로 `myschema`를 입력합니다.

쿼리 편집기 v2의 트리 보기 패널에 `cluster-view`의 계층 구조가 다음과 같이 표시됩니다.

- 클러스터: `cluster-view`
 - 데이터베이스: `data_catalog_database`
 - 스키마: `myschema`
 - 테이블: `category3`
4. 데이터를 쿼리할 때는 `cluster-view` 클러스터의 기본 데이터베이스(일반적으로 `dev`라는 이름)에 연결하되, SQL에서 데이터베이스 `data_catalog_database`를 참조하세요.

Note

쿼리 편집기 v2 편집기 뷰에서 선택한 클러스터는 `cluster-view`입니다. 선택한 데이터베이스는 `dev`입니다. 데이터베이스 `data_catalog_database`가 나열되어 있지만 직접 연결할 수 있도록 활성화되어 있지 않습니다. `dev` 데이터베이스를 선택하고 실행하는 SQL에서 `data_catalog_database`를 참조합니다.

```
SELECT * FROM "data_catalog_database"."myschema"."category3";
```

쿼리는 AWS Glue Data Catalog에 의해 카탈로그화된 데이터를 검색합니다.

스키마 생성

쿼리 편집기 v2를 사용하여 클러스터 또는 작업 그룹에 스키마를 생성할 수 있습니다.

스키마를 생성하려면

스키마에 대한 내용은 Amazon Redshift Database 개발자 안내서의 [CREATE SCHEMA](#)를 참조하세요.

1.  성(Create)을 선택한 다음 스키마(Schema)를 선택합니다.
2. 스키마 이름(Schema name)을 입력합니다.

생

3. Local(로컬) 또는 External(외부)을 Schema type(스키마 유형)으로 선택합니다.

로컬 스키마에 대한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [CREATE SCHEMA](#)를 참조하세요. 외부 스키마에 대한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [CREATE EXTERNAL SCHEMA](#)를 참조하세요.

4. External(외부)을 선택하면 다음과 같은 외부 스키마를 선택할 수 있습니다.

- Glue Data Catalog - Amazon Redshift에서 AWS Glue의 테이블을 참조하는 외부 스키마를 생성하려는 경우. AWS Glue 데이터베이스를 선택하고 데이터 카탈로그와 연결된 IAM 역할을 선택합니다.
- PostgreSQL - Amazon RDS for PostgreSQL 또는 Amazon Aurora PostgreSQL-Compatible Edition 데이터베이스를 참조하는 Amazon Redshift 외부 스키마를 생성합니다. 데이터베이스 연결 정보도 제공합니다. 연합 쿼리에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [연합 쿼리를 사용하여 데이터 쿼리](#)를 참조하세요.
- MySQL - Amazon RDS for MySQL 또는 Amazon Aurora MySQL-Compatible Edition 데이터베이스를 참조하는 Amazon Redshift 외부 스키마를 생성합니다. 데이터베이스 연결 정보도 제공합니다. 연합 쿼리에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [연합 쿼리를 사용하여 데이터 쿼리](#)를 참조하세요.

5. 스키마 생성을 선택합니다.

새 스키마가 트리 보기 패널에 나타납니다.

테이블 생성

쿼리 편집기 v2를 사용하여 클러스터 또는 작업 그룹에 테이블을 생성할 수 있습니다.

테이블을 만들려면

테이블의 각 열을 지정하거나 정의하는 쉼표로 구분된 값(CSV) 파일을 기반으로 테이블을 생성할 수 있습니다. 테이블에 대한 자세한 내용은 Amazon Redshift Database 개발자 안내서의 [테이블 설계](#) 및 [CREATE TABLE](#)을 참조하세요.

쿼리를 실행하여 테이블을 생성하기 전에 편집기에서 쿼리 열기(Open query in editor)를 선택하여 CREATE TABLE 문을 보고 편집합니다.

1.  (plus icon)를 선택하고 Table(테이블)을 선택합니다.

Create

2. 스키마를 선택합니다.
3. 테이블 이름을 입력합니다.
4.  필드 추가(Add field)를 선택하여 열을 추가합니다.
5. CSV 파일을 테이블 정의의 템플릿으로 사용합니다.
 - a. CSV에서 로드(Load from CSV)를 선택합니다.
 - b. 파일 위치를 찾습니다.

CSV 파일을 사용하는 경우 파일의 첫 번째 행에 열 머리글이 포함되어 있는지 확인합니다.

- c. 파일과 열기(Open)를 차례로 선택합니다. 열 이름과 데이터 유형이 맞는지 확인합니다.
6. 각 열에 대해 열을 선택하고 원하는 옵션을 선택합니다.
 - 인코딩(Encoding) 값을 선택합니다.
 - 기본값(Default value)을 선택합니다.
 - 열 값 증분을 원하는 경우 자동으로 증분(Automatically increment)을 설정합니다. 그런 다음 자동 증분 시드(Auto increment seed) 및 자동 증분 단계(Auto increment step) 값을 지정합니다.
 - 열에 항상 값이 포함되어야 하는 경우 NULL이 아님(Not NULL)을 설정합니다.
 - 열의 크기(Size) 값을 입력합니다.
 - 열을 기본 키로 사용하려면 기본 키(Primary key)를 설정합니다.
 - 열을 고유 키로 사용하려면 고유 키(Unique key)를 설정합니다.
7. (선택 사항) 테이블 세부 정보(Table details)를 선택하고 다음 옵션 중 하나를 선택합니다.
 - 배포 키 열 및 스타일.
 - 정렬 키 열 및 정렬 유형.
 - 스냅샷에 테이블을 포함하려면 Backup(백업)을 설정합니다.
 - 테이블을 임시 테이블로 생성하려면 임시 테이블(Temporary table)을 설정합니다.
8. 편집기에서 쿼리 열기(Open query in editor)를 선택하여 테이블 정의 옵션을 계속 지정하거나 테이블 생성(Create table)을 선택하여 테이블을 생성합니다.

함수 생성

쿼리 편집기 v2를 사용하여 클러스터 또는 작업 그룹에 함수를 생성할 수 있습니다.

함수를 만들려면

1.  성(Create)을 선택하고 함수(Function)를 선택합니다.
2. 유형(Type)에서 SQL 또는 Python을 선택합니다.
3. 스키마(Schema) 값을 선택합니다.
4. 함수의 이름(Name) 값을 입력합니다.
5. 함수의 휘발성(Volatility) 값을 입력합니다.
6. 입력 파라미터 순서대로 데이터 유형별로 파라미터(Parameters)를 선택합니다.
7. 반환>Returns)에서 데이터 유형을 선택합니다.
8. 함수에 대한 SQL 프로그램 또는 Python 프로그램 코드를 입력합니다.
9. 생성(Create)을 선택합니다.

생

사용자 정의 함수(UDF)에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [사용자 정의 함수 생성](#)을 참조하세요.

쿼리 및 탭 기록 보기

쿼리 에디터 v2를 사용하여 쿼리 기록을 볼 수 있습니다. 쿼리 에디터 v2를 사용하여 실행한 쿼리만 쿼리 기록에 표시됩니다. Editor(편집기) 탭 또는 Notebook(노트북) 탭을 사용하여 실행한 쿼리가 모두 표시됩니다. 기간별(예: 한 주가 월요일~일요일로 정의되는 This week)로 표시된 목록을 필터링할 수 있습니다. 쿼리 목록은 한 번에 필터와 일치하는 쿼리 행 25개를 가져옵니다. Load more(추가 로드)를 선택하면 다음 세트를 볼 수 있습니다. 쿼리를 선택하고 Actions(작업) 메뉴에서 선택합니다. 사용할 수 있는 작업은 선택한 쿼리가 저장되었는지 여부에 따라 달라집니다. 다음 작업을 선택할 수 있습니다.

- View query details(쿼리 세부 정보 보기) - 실행된 쿼리에 대한 자세한 정보가 포함된 쿼리 세부 정보 페이지를 표시합니다.
- Open query in a new tab(새 탭에서 쿼리 열기) - 새 편집기 탭을 열고 선택한 쿼리로 이동합니다. 계속 연결되어 있으면 클러스터 또는 작업 그룹 및 데이터베이스가 자동으로 선택됩니다. 쿼리를 실행하려면 먼저 올바른 클러스터 또는 작업 그룹과 데이터베이스가 선택되었는지 확인합니다.
- Open source tab(소스 탭 열기) - 계속 열려 있는 경우 쿼리 실행 시 쿼리가 포함된 편집기 또는 노트북 탭으로 이동합니다. 쿼리가 실행된 후에 편집기 또는 노트북의 내용이 변경되었을 수 있습니다.
- Open saved query(저장된 쿼리 열기) - 편집기 또는 노트북 탭으로 이동하여 쿼리를 엽니다.

또한 Editor(편집기) 탭에서 실행된 쿼리 기록 또는 Notebook(노트북) 탭에서 실행된 쿼리 기록을 볼 수도 있습니다. 탭에서 쿼리 기록을 보려면 Tab history(탭 기록)를 선택합니다. 탭 기록 내에서 다음 작업을 수행할 수 있습니다.

- Copy query(쿼리 복사) - 쿼리 버전 SQL 내용을 클립보드에 복사합니다.
- Open query in a new tab(새 탭에서 쿼리 열기) - 새 편집기 탭을 열고 선택한 쿼리로 이동합니다. 쿼리를 실행하려면 클러스터 또는 워크그룹과 데이터베이스를 선택해야 합니다.
- View query details(쿼리 세부 정보 보기) - 실행된 쿼리에 대한 자세한 정보가 포함된 쿼리 세부 정보 페이지를 표시합니다.

Amazon Q 생성형 SQL과의 상호 작용

Note

Amazon Q 생성형 SQL 지원은 다음 AWS 리전에서만 이용할 수 있습니다.

- 미국 동부(버지니아 북부) 리전(us-east-1)
- 미국 동부(오하이오) 리전(us-east-2)
- 미국 서부(오레곤) 리전(us-west-2)
- 아시아 태평양(뭄바이) 리전(ap-south-1)
- 아시아 태평양(서울) 리전(ap-northeast-2)
- 아시아 태평양(싱가포르) 리전(ap-southeast-1)
- 아시아 태평양(시드니) 리전(ap-southeast-2)
- 아시아 태평양(도쿄) 리전(ap-northeast-1)
- 캐나다(중부) 리전(ca-central-1)
- 유럽(프랑크푸르트) 리전(eu-central-1)
- 유럽(아일랜드) 리전(eu-west-1)
- 유럽(런던) 리전(eu-west-2)
- 유럽(파리) 리전(eu-west-3)
- 남아메리카(상파울루) 리전(sa-east-1)

자세한 내용은 Amazon Q Developer 사용 설명서의 [Cross region inference in Amazon Q Developer](#) 섹션을 참조하세요.

Amazon Redshift 쿼리 에디터 v2에서 Amazon Q 생성형 SQL 기능을 사용할 수 있습니다. 생성형 SQL은 프롬프트와 데이터베이스 스키마를 기반으로 SQL 문을 생성하는 코딩 도우미입니다. 이 코딩 도우미는 쿼리 에디터 v2에서 노트북을 작성하는 동안 사용할 수 있습니다. 생성된 SQL은 노트북이 연결된 데이터베이스용입니다.

Amazon Q 생성형 SQL과 상호 작용할 경우 구체적인 질문을 하고, 복잡한 요청이 있으면 반복하고, 답변이 정확한지 검증합니다.

자연어로 분석 요청을 제공할 때는 코딩 도우미가 필요한 내용을 정확히 이해할 수 있도록 최대한 구체적으로 작성하세요. 'find top venues that sold the most tickets(티켓을 가장 많이 판매한 공연장을 찾아줘)'라고 묻는 대신 '2008년에 티켓을 가장 많이 판매한 공연장 세 개의 이름/ID를 찾아줘(find names/ids of top three venues that sold the most tickets in 2008)'와 같이 세부 정보를 더 제공하세요. 객체를 알고 있을 경우 데이터베이스에서 일관되고 구체적인 객체 이름을 사용합니다. 동일한 객체를 서로 다른 방식으로 참조하면 도우미에게 혼동을 줄 수 있으므로 데이터베이스에 정의된 스키마, 테이블, 열 이름 등을 사용합니다.

복잡한 요청을 도우미가 해석하기 쉬운 여러 개의 간단한 명령문으로 나누세요. 반복적으로 후속 질문을 하면 도우미로부터 더 자세한 분석을 받을 수 있습니다. 예를 들어, 먼저 'which state has the most venues?(공연장이 가장 많은 주가 어디야?)'라고 물어보세요. 그런 다음 답변을 바탕으로 'which is the most popular venue from this state?(이 주에서 가장 인기 있는 공연장은 어디야?)'라고 물어보세요.

생성된 SQL을 실행하기 전에 검토하여 정확성을 확인하세요. 생성된 SQL 쿼리에 오류가 있거나 의도와 맞지 않는 경우 요청을 완전히 바꾸는 대신 도우미에게 수정 방법에 대한 지침을 제공하세요. 예를 들어, 쿼리에 연도의 조건자 절이 누락된 경우 '2008년을 기준으로 공연장을 알려줘'라고 요청하세요.

생성된 SQL을 실행하여 받은 오류 텍스트를 Amazon Q 생성형 SQL에 프롬프트로 다시 제출합니다. 이러한 오류를 통해 학습하여 더 나은 SQL이 생성됩니다.

스키마를 사용해야 한다는 신호를 보내려면 SQL 검색 경로에 스키마를 추가합니다. 예를 들어 데이터가 퍼블릭 스키마가 아닌 tickit 스키마에 있으면 tickit 스키마를 추가합니다.

```
set search_path to '$user', tickit;
```

Amazon Q 생성형 SQL과 상호 작용할 경우의 고려 사항

채팅 패널을 사용할 때 다음 사항을 고려하세요.

- 계정의 쿼리 에디터 v2 관리자가 생성형 SQL 설정 페이지에서 채팅 기능을 활성화한 상태여야 합니다.

- Amazon Q 생성형 SQL을 사용하려면 Query Editor V2의 AWS 관리형 정책에 지정된 기타 권한 외에도 IAM 정책의 `sqlworkbench:GetQSqlRecommendations` 권한이 필요합니다. AWS 관리형 정책에 대한 자세한 정보는 [쿼리 편집기 v2에 액세스](#) 단원을 참조하세요.
- 질문은 영어로 작성해야 합니다.
- 질문은 클러스터 또는 작업 그룹에 연결된 데이터베이스를 참조해야 합니다. 빈 상태 오류를 방지하려면 데이터베이스에 하나 이상의 테이블과 어느 정도의 데이터가 있어야 합니다.
- 질문은 연결된 데이터베이스에 저장된 데이터를 참조해야 합니다. 외부 스키마는 참조할 수 없습니다. 지원되는 스키마에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Create schema](#)를 참조하세요.
- SQL 생성으로 인해 연결된 데이터베이스가 변경되는 질문을 하면 경고가 발생할 수 있습니다.
- 생성형 AI 기술은 새로운 기술이며 응답에 할루시네이션이라고 하는 실수가 있을 수 있습니다. 환경 또는 워크로드에서 사용하기 전에 오류와 취약성이 있는지 모든 코드를 테스트하고 검토하세요.
- 계정의 다른 사용자가 실행한 SQL 쿼리를 공유하여 제안을 개선할 수 있습니다. 계정 관리자는 다음 SQL 명령을 실행하여 계정의 쿼리 기록에 대한 액세스를 허용할 수 있습니다.

```
GRANT ROLE SYS:MONITOR to "IAMR:role-name";
GRANT ROLE SYS:MONITOR to "IAM:user-name";
GRANT ROLE SYS:MONITOR to "database-username";
```

SYS:MONITOR에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Amazon Redshift 시스템 정의 역할](#)을 참조하세요.

- 데이터는 안전하고 공개되지 않습니다. 데이터는 계정 간에 공유되지 않습니다. 쿼리, 데이터, 데이터베이스 스키마는 생성형 AI 파운데이션 모델(FM)을 훈련하는 데 사용되지 않습니다. 입력은 FM에 대한 맥락적 프롬프트로 질문자의 쿼리에만 답변하는 데 사용됩니다.

생성형 SQL 사용

올바른 권한을 구성한 후 쿼리 에디터 v2에서 노트북으로 작업할 때 아이콘을 선택하여 대화를 시작할 수 있습니다.

Amazon Q 생성형 SQL 채팅을 사용하여 상호 작용하려면 생성형 SQL에 대해 다음과 같이 합니다.

1. 쿼리 에디터 v2의 편집기 탭에서 노트북을 엽니다.

2.

생성형 SQL 아이콘을 선택한 다음 채팅 패널에서 안내를 따라 Amazon Redshift 쿼리 에디터 v2 생성형 SQL에 질문하세요.

프롬프트 필드에 질문을 입력하면 Amazon Q 생성형 SQL에서 제안 SQL을 포함해 응답합니다. 오류가 발생하면 채팅 패널에 반환됩니다.

3. 노트북에 추가를 선택하여 프롬프트가 있는 마크다운 셀과 제안된 SQL이 포함된 SQL 셀을 노트북에 추가합니다.

4. (선택 사항) 유용한 피드백 아이콘



또는 유용하지 않은 피드백 아이콘



을 선택하여 생성된 SQL에 관한 피드백을 제공합니다. 유용하지 않은 피드백은 Incorrect tables/columns, Incorrect predicates/literals/group bys, Incorrect SQL structure, Other로 분류할 수 있습니다. 또한 SQL의 정확도에 대한 피드백과 함께 몇 가지 자유 형식 텍스트를 제공할 수 있습니다.

5. (선택 사항) 동일한 프롬프트에 대해 다른 응답을 생성하려면 SQL 재생성을 선택합니다. 현재 프롬프트에서 SQL 재생성을 한 번 시도할 수 있습니다.

6. (선택 사항) 생성형 SQL 채팅 패널에서



더 보기 아이콘을 선택한 다음 데이터베이스 새로 고침을 선택하여 연결된 데이터베이스를 설명하는 메타데이터를 새로 고칩니다. 이 메타데이터에는 데이터베이스의 스키마, 테이블 및 열에 대한 정의가 포함됩니다.

관리자 권한으로 생성형 SQL 설정 업데이트

적절한 IAM 권한을 가진 사용자는 같은 AWS 계정 내의 다른 사용자에게 대한 생성형 SQL 설정을 보고 변경할 수 있습니다. 이 관리자는 쿼리 에디터 v2의 AWS 관리형 정책에 지정된 다른 권한 외에도 IAM 정책의 `sqlworkbench:UpdateAccountSqlSettings` 권한이 있어야 합니다. 관리형 정책에 대한 자세한 정보는 [쿼리 편집기 v2를 사용하는 데 필요한 권한](#) 단원을 참조하세요.

관리자가 계정의 모든 사용자에게 대해 생성형 SQL 채팅을 활성화하는 방법

1.  설정 아이콘을 사용하여 다양한 설정 화면의 메뉴를 표시합니다.
2. 그런 다음  생성형 SQL 설정 아이콘을 선택하여 Q 생성형 SQL 설정 페이지를 표시합니다.
3. Q 생성형 SQL 설정을 선택하여 계정 내 사용자에서 생성형 SQL 기능을 켭니다.

Amazon Q 생성형 SQL 기능을 사용 설정하면 할당에 남아 있는 프롬프트 수를 확인할 수 있습니다. Query Editor V2 관리자에서 계정의 사용자가 Amazon Q Developer Pro 티어를 사용할 수 있습니다. Pro 티어를 사용하려면 IAM Identity Center로 사용자를 설정하고 각 사용자가 Amazon Q Developer Pro 티어를 구독하도록 합니다. Amazon Redshift로 IAM Identity Center 설정에 관한 자세한 내용은 [Single Sign-On 경험을 위해 Redshift와 AWS IAM Identity Center 연결](#) 섹션을 참조하시기 바랍니다. Amazon Q Developer 요금에 관한 자세한 내용은 [Amazon Q Developer 요금](#) 섹션을 참조하시기 바랍니다.

Amazon Q Developer 프리 티어를 사용하는 경우 AWS 계정의 모든 사용자에게 대한 총 프롬프트 수는 한 달에 1,000개로 제한됩니다. Amazon Q Developer Pro 티어를 사용하는 경우 각 사용자가 제출할 수 있는 총 프롬프트 수는 한 달에 1,000개로 제한됩니다. 설정 페이지에서 사용 가능한 프롬프트 수를 확인할 수 있습니다. Amazon Q Developer 요금에 관한 자세한 내용은 [Amazon Q Developer 요금](#) 섹션을 참조하시기 바랍니다.

사용자 지정 컨텍스트

Query Editor V2 관리자는 생성형 SQL을 환경에 맞춤화하도록 사용자 지정 컨텍스트를 지정할 수 있습니다. 사용자 지정 컨텍스트는 도메인 지식과 기본 설정을 제공하여 SQL 생성에 대한 세분화된 제어 기능을 제공합니다. 사용자 지정 컨텍스트는 Query Editor V2 관리자가 Amazon Q 생성형 SQL에 업로드할 수 있는 JSON 파일에 정의됩니다.

데이터 웨어하우스의 생성형 SQL을 맞춤화하는 데 사용되는 JSON 키는 다음과 같습니다.

모든 테이블 참조는 3부분 표기법(database.schema.table)을 따라야 합니다.

리소스

리소스는 사용자 지정 컨텍스트가 적용되는 데이터 자산의 범위 또는 비중을 지정합니다.

ResourceId

고유한 리소스 식별자를 지정합니다. Amazon Redshift 클러스터에서 `cluster id`를 지정합니다. Redshift 서버리스 작업 그룹에서 `workgroup name`을 지정합니다.

ResourceType

유효한 값: REDSHIFT_WAREHOUSE.

TablesToInclude

SQL 생성에 고려되는 테이블 모음을 지정합니다. 이 필드는 SQL 쿼리 범위를 사용 가능한 테이블의 정의된 하위 집합으로 제한하려는 경우에 매우 중요합니다. 불필요한 테이블 참조를 줄여 생성 프로세스를 최적화하는 데 도움이 될 수 있습니다. 쿼리 생성을 더 미세하게 제어하려고 `TablesToExclude`와 이 필드를 페어링할 수 있습니다.

TablesToExclude

SQL 생성에서 제외되는 테이블 모음을 지정합니다. 이 옵션은 특정 테이블이 관련이 없거나 쿼리 생성 프로세스에서 영향을 주면 안 되는 경우에 사용합니다.

TableAnnotations

사용 중인 테이블에 관한 메타데이터 또는 보충 정보를 제공합니다. 이러한 주석에는 테이블 설명, 사용 정보 또는 Amazon Q 생성형 SQL이 테이블의 컨텍스트 또는 구조를 더 잘 이해하는 데 도움이 되는 추가 속성이 포함될 수 있습니다. 이는 테이블 정의에 명확성을 더하여 SQL 생성의 정확도를 높이는 데 중요합니다.

ColumnsToInclude

생성형 SQL 쿼리 시 지정된 테이블에서 어떤 열이 포함되는지를 정의합니다. 이 필드는 Amazon Q 생성형 SQL이 관련 열에 집중하고 데이터 검색 범위를 좁혀 성능을 개선하는 데 도움이 됩니다. 이렇게 하면 Amazon Q 생성형 SQL이 지정된 쿼리 컨텍스트에 필요한 데이터만 가져옵니다.

ColumnsToExclude

SQL 생성에서 영향을 주지 않도록 제외할 열을 지정합니다. 특정 열에 Amazon Q 생성형 SQL에서 영향을 주면 안 되는 관련이 없거나 중복된 데이터가 포함되어 있는 경우에 사용할 수 있습니다. 열 포함 및 제외를 관리하여 결과를 구체화하고 검색된 데이터에 대한 제어 기능을 유지 관리할 수 있습니다.

ColumnAnnotations

`TableAnnotations`와 마찬가지로 이 필드는 각 열별로 메타데이터 또는 주석을 제공합니다. 이러한 주석을 통해 열 정의 또는 특정 처리 지침에 대한 인사이트를 제공할 수 있습니다. 이 정보는 SQL 생성 프로세스를 안내하고 쿼리에서 열이 적절하게 사용되도록 하는 데 유용합니다.

CuratedQueries

미리 정의된 질문 및 답변 예제 모음이며, 질문은 자연어(NLQ)로 작성되고 답변으로 해당 SQL 쿼리가 반환됩니다. 이러한 예제를 통해 Amazon Q 생성형 SQL에서 생성되어야 하는 쿼리 유형을 이해할 수 있습니다. 예제는 Amazon Q 생성형 SQL 출력의 정확도와 관련성을 개선하기 위한 기준점 역할을 합니다.

CustomDocuments

정의, 도메인별 지식 또는 설명 등 Amazon Q 생성형 SQL에 제공되는 추가 정보 또는 힌트입니다. 사업부에서 고유한 방식으로 값을 계산하는 경우를 예로 들 수 있습니다. '제조 팀의 총매출은 가격 * 수익'이며, 여기에서 문서화할 수 있습니다. 이러한 문서는 추가 컨텍스트를 제공하여 자연어 입력을 해석하는 Amazon Q 생성형 SQL의 기능을 향상시킵니다.

AdditionalTables

SQL 생성에 영향을 줘야 하지만 데이터 웨어하우스에 저장된 일부 데이터에 포함되지 않은 추가 테이블을 지정합니다. 이렇게 하면 Amazon Q 생성형 SQL에서 외부 데이터 소스를 SQL 생성 로직에 통합하여 복잡한 데이터 환경을 처리할 수 있는 용량을 확장할 수 있습니다.

AppendToPrompt

SQL 생성 프로세스를 안내하도록 Amazon Q 생성형 SQL에 제공되는 추가 지침 또는 가이드라인입니다. 여기에는 쿼리를 구조화하는 방법에 대한 특정 지침, 특정 SQL 구문에 대한 기본 설정 또는 Amazon Q 생성형 SQL 출력의 품질을 향상시키는 기타 개괄적인 지침이 포함될 수 있습니다.

다음 사용자 지정 컨텍스트 예제에서는 JSON 파일의 형식을 보여 주며, 다음 항목을 정의합니다.

- mycluster 클러스터에 대한 Amazon Redshift 데이터 웨어하우스의 사용자 지정 컨텍스트를 정의합니다.
- SQL 생성 프로세스를 최적화하는 데 도움이 되도록 포함 및 제외할 특정 테이블과 열을 정의합니다.
- 포함할 직접 호출된 테이블 및 열에 대한 주석을 정의합니다.
- Amazon Q 생성형 SQL에서 사용할 엄선된 샘플 쿼리를 정의합니다.
- 생성형 SQL 사용 시의 사용자 지정 문서와 가드레일을 정의합니다.
- 생성형 SQL 사용 시의 추가 테이블에 대한 DDL을 정의합니다.

```
{
  "resources": [
    {
      "ResourceId": "mycluster",
```

```
"ResourceType": "REDSHIFT_WAREHOUSE",
"TablesToInclude": [
    "database.schema.table1",
    "database.schema.table2"
],
"TablesToExclude": [
    "database.schema.table3",
    "database.schema.table4"
],
"ColumnsToInclude": {
    "database.schema.table1": [
        "col1",
        "col2"
    ],
    "database.schema.table2": [
        "col1",
        "col2"
    ]
},
"ColumnsToExclude": {
    "database.schema.table5": [
        "col1",
        "col2"
    ],
    "database.schema.table6": [
        "col1",
        "col2"
    ]
},
"TableAnnotations": {
    "database.schema.table1": "table1 refers to Q3 sales",
    "database.schema.table2": "table2 refers to Q4 sales"
},
"ColumnAnnotations": {
    "database.schema.table1": {
        "col1": "col1 refers to Q3 sale total",
        "col2": "col2 refers to sale location"
    },
    "database.schema.table2": {
        "col1": "col2 refers to Q4 sale total",
        "col2": "col2 refers to sale location"
    }
},
"CuratedQueries": [
```

```

    {
      "Question": "what is the sales data for Q3",
      "Answer": "SELECT * FROM table1"
    },
    {
      "Question": "what is the sales data for Q4",
      "Answer": "SELECT * FROM table2"
    }
  ],
  "CustomDocuments": [
    "in manufacturing division total sales is price * revenue",
    "in research division total sales is price * revenue"
  ],
  "AdditionalTables": {
    "database.schema.table8": "create table database.schema.table8(col1
int)",
    "database.schema.table9": "create table database.schema.table9(col1
int)"
  },
  "AppendToPrompt": "Apply these guardrails: Queries should never return the
secretId field of a user."
}
]
}

```

자습서: TICKIT 데이터와 함께 Amazon Q 생성형 SQL 기능 사용

SQL을 생성하기 위한 효율적인 프롬프트를 작성하려면 데이터베이스 스키마와 데이터에 대해 알아야 합니다. TICKIT 데이터는 팩트 테이블 2개와 차원 테이블 5개, 총 7개의 테이블로 구성되어 있습니다. 샘플 데이터에는 2008년에 발생한 엔터테인먼트 이벤트 참석자에 대한 판매 기록이 포함되어 있습니다. TICKIT 데이터 스키마에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [샘플 데이터베이스](#)를 참조하세요. Amazon Redshift 콘솔과 쿼리 에디터 v2에서 다양한 방법을 사용하여 TICKIT 데이터를 데이터베이스에 로드할 수 있습니다. 쿼리 에디터 v2는 TICKIT 데이터를 sample_data_dev 데이터베이스에 로드하는 방법을 제공합니다. 자세한 내용은 [데이터를 데이터베이스로 로드](#) 섹션을 참조하세요. TICKIT 데이터에 대한 예시 프롬프트 또한 제공합니다. 다음 시나리오에서는 TICKIT 샘플 데이터가 Amazon Redshift 클러스터의 dev 데이터베이스에 이미 생성되어 있습니다.

Note

이 예는 대화를 설명하기 위한 것입니다. 동일한 프롬프트를 사용해도 사용자가 받는 생성형 SQL의 응답이 여기에 나온 결과와 일치하지 않을 수 있습니다.

Amazon Q 생성형 SQL과의 상호 작용 예제

1. 편집기에서 TICKIT 샘플 데이터가 포함된 클러스터 또는 작업 그룹에 연결합니다.
2. 빈 노트북을 만들고



생성형 SQL 아이콘을 선택하여 채팅 패널을 엽니다.

3. 다음 프롬프트를 입력하여 VENUE 테이블의 레코드 수를 확인하기 위한 SQL을 생성합니다.

How many venues are there?

```
SELECT
  COUNT(*) AS num_venues
FROM
  tickit.venue
```

노트북에 추가를 선택하여 열려 있는 노트북에 두 개의 셀을 추가합니다. 'How many venues are there?'라는 마크다운 셀 1개와 생성된 SQL이 들어 있는 셀 1개입니다.

SQL 셀에서 실행을 선택하여 결과를 수신합니다.

```
count
-----
202
```

4. 다른 버전의 SQL을 요청하려면 SQL 재생성을 선택하면 다음과 같은 답변을 받을 수 있습니다.

```
SELECT
  venuestate,
  COUNT(*) AS num_venues
FROM
  tickit.venue
```



```
GROUP BY
venuestate
```

노트북에 추가를 선택하여 열려 있는 노트북에 두 개의 셀을 더 추가합니다. 'How many venues are there?'라는 마크다운 셀 1개와 생성된 SQL이 들어 있는 셀 1개입니다.

SQL 셀에서 실행을 선택하여 번호별로 공연장이 있는 주가 나열된 결과를 수신합니다.

```
venuestate num_venues
-----
MA          4
OH          8
MI          5
...
```

5. 생성형 SQL 도우미는 사용자가 질문하는 내용을 추측할 수 있습니다. 데이터베이스에 위젯이 없더라도 질문할 수 있습니다. 이 경우 SQL을 생성할 수 없다는 메시지가 표시됩니다.

```
How many widgets are there?
```

```
I was not able to generate any SQL
```

6. 데이터베이스를 더 자세히 파악하면 더 구체적인 질문으로 SQL을 생성할 수 있습니다. 예를 들어, 데이터베이스에 2008년 몇 개월간의 판매 데이터가 들어 있다는 것을 알고 있다면 다음 프롬프트를 입력하여 SQL을 생성하여 2008년 11월의 판매량을 구할 수 있습니다.

```
How many sales were there in November 2008?
```

```
SELECT
  COUNT(*) AS num_sales
FROM
  tickit.sales s
  JOIN tickit.date d ON s.dateid = d.dateid
WHERE
  d.month = 'November'
  AND d.year = 2008
```

노트북에 추가를 선택하여 열려 있는 노트북에 두 개의 셀을 추가합니다. 'How many sales were there in November 2008?'이라는 마크다운 셀 한 개와 생성된 SQL이 들어 있는 셀 1개입니다.

SQL 셀에서 실행을 선택하여 결과를 수신합니다.

```
num_sales
-----
0
```

원했던 결과가 나오지 않았습니다.

7. `date.month='November'` 조건자는 DATE 테이블의 월이 약어로 표현될 것으로 예상한다는 것을 알 수 있습니다. 조건자를 `d.month='NOV'` 로 변경하고 SQL을 다시 실행합니다.

```
SELECT
  COUNT(*)
FROM
  sales
  JOIN date ON sales.dateid = date.dateid
WHERE
  date.month = 'NOV'
  AND date.year = 2008
```

SQL 셀에서 실행을 선택하여 새 결과를 가져옵니다.

```
count
-----
14261
```

8. 연결된 데이터베이스를 변경하려는 질문을 하면 권장 SQL과 함께 경고 메시지가 반환됩니다. 데이터를 테이블에 삽입하는 SQL을 생성하려면 다음 프롬프트를 입력합니다.

```
Insert 1 into the venue table.
```

```
INSERT
,
UPDATE
  OR delete data
FROM
```

```
the database AS that could potentially change the data.Please provide a query  
that ONLY selects data
```

I was not able to generate the correct SQL code. I generated SQL, but you'll have to edit it to work with your database.

노트북에 추가를 선택하여 열려 있는 노트북에 두 개의 셀을 추가하고 SQL을 실행하면 SQL이 실패합니다.

```
ERROR: syntax error at or near "," Position: 132 [ErrorId:  
1-6546764a-011df2691778846219ce6ec2]
```

이 시나리오에는 Amazon Q 생성형 SQL과 상호 작용하는 몇 가지 기본 방법만 나와 있습니다. 이 생성형 AI 기술을 더 다양하게 실험하여 데이터베이스 쿼리를 위한 SQL 작성을 시작해 볼 수 있습니다.

데이터를 데이터베이스로 로드

쿼리 편집기 v2를 사용하여 Amazon Redshift 클러스터 또는 작업 그룹의 데이터베이스에 데이터를 로드할 수 있습니다. 이 섹션에서는 샘플 데이터, S3의 데이터, 로컬 파일 설정 및 워크플로의 데이터를 로드하는 방법을 다룹니다.

샘플 데이터

쿼리 편집기 v2는 샘플 데이터베이스 및 해당 스키마에 로드할 수 있는 샘플 데이터 및 노트북과 함께 제공됩니다.

샘플 데이터를 로드하려면 로드하려는 샘플 데이터와 연결된



아이콘을 선택합니다. 그런 다음 Query Editor V2는 sample_data_dev 데이터베이스의 스키마에 데이터를 로드하고 저장된 노트북 폴더를 만듭니다.

다음 샘플 데이터 세트를 사용할 수 있습니다.

tickit

Amazon Redshift 설명서에 나오는 예제에서는 대부분 tickit이라는 샘플 데이터를 사용합니다. 이 데이터는 팩트 테이블 2개와 차원 테이블 5개, 총 7개의 테이블로 구성되어 있습니다. 이 데이터를 로드하면 tickit 스키마가 샘플 데이터로 업데이트됩니다. tickit 데이터에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [샘플 데이터베이스](#)를 참조하세요.

tpch

이 데이터는 의사 결정 지원 벤치마크에 사용됩니다. 이 데이터를 로드하면 tpch 스키마가 샘플 데이터로 업데이트됩니다. tpch 데이터에 대한 자세한 내용은 [TPC-H](#)를 참조하세요.

tpcds

이 데이터는 의사 결정 지원 벤치마크에 사용됩니다. 이 데이터를 로드하면 tpcds 스키마가 샘플 데이터로 업데이트됩니다. tpcds 데이터에 대한 자세한 내용은 [TPC-DS](#)를 참조하세요.

Amazon S3에서 데이터 로드

Amazon S3 데이터를 기존 테이블 또는 새로운 테이블로 로드할 수 있습니다.

기존 테이블에 데이터를 로드하려면

COPY 명령은 쿼리 편집기 v2가 Amazon S3에서 데이터를 로드하는 데 사용됩니다. 쿼리 편집기 v2 데이터 로드 마법사에서 생성되고 사용되는 COPY 명령은 Amazon S3에서 복사하기 위해 COPY 명령 구문에 사용할 수 있는 많은 파라미터를 지원합니다. COPY 명령 및 Amazon S3에서 데이터를 로드하는 데 사용되는 옵션에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Amazon S3에서 COPY](#)를 참조하세요.

1. 데이터를 로드하려는 데이터베이스에 테이블이 이미 생성되었는지 확인합니다.
2. 계속하기 전에 쿼리 에디터 v2의 트리 뷰 패널에서 대상 데이터베이스에 연결되어 있는지 확인합니다. 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 사용하여 데이터를 로드할 클러스터 또는 작업 그룹에 연결할 수 있습니다.



이터 로드(Load data)를 선택합니다.

3. 데이터 소스에서 S3에서 로드를 선택합니다.
4. S3 URI(S3 URIs)에서 S3 찾아보기(Browse S3)를 선택하여 로드할 데이터가 들어 있는 Amazon S3 버킷을 찾습니다.

데

5. 지정된 Amazon S3 버킷이 대상 테이블과 동일한 AWS 리전에 있지 않은 경우 데이터가 있는 AWS 리전에 대한 S3 file location(S3 파일 위치)을 선택합니다.
6. Amazon S3 파일이 실제로 여러 Amazon S3 버킷 URI를 포함하는 매니페스트인 경우 이 파일은 매니페스트 파일임(This file is a manifest file)을 선택합니다.
7. 업로드할 파일의 파일 형식(File format)을 선택합니다. 지원되는 데이터 형식은 CSV, JSON, DELIMITER, FIXEDWIDTH, SHAPEFILE, AVRO, PARQUET 및 ORC입니다. 지정된 파일 형식에 따라 해당 파일 옵션(File options)을 선택할 수 있습니다. 데이터가 암호화된 경우 데이터가 암호화됨(Data is encrypted)을 선택하고 데이터 암호화에 사용된 KMS 키의 Amazon 리소스 이름(ARN)을 입력할 수도 있습니다.

CSV 또는 DELIMITER를 선택하면 구분자 문자를 선택하고, 지정한 수의 행이 실제로 열 이름이고 로드할 데이터가 아닌 경우 헤더 행을 무시할지 여부도 선택할 수 있습니다.

8. 파일을 압축할 압축 방법을 선택합니다. 기본값은 압축 안 함(no compression)입니다.
9. (선택 사항) 고급 설정(Advanced settings)은 다양한 데이터 변환 파라미터(Data conversion parameters)와 로드 작업(Load operations)을 지원합니다. 파일에 필요한 대로 이 정보를 입력합니다.

데이터 변환 및 데이터 로드 파라미터에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [데이터 변환 파라미터](#)와 [데이터 로드 작업](#)을 참조하세요.

10. 다음을 선택합니다.
11. 기존 테이블 로드를 선택합니다.
12. 데이터가 로드되는 클러스터 또는 작업 그룹, 데이터베이스, 스키마 및 테이블 이름을 포함한 대상 테이블의 위치를 확인하거나 선택합니다.
13. Amazon S3에서 데이터를 로드하는 데 필요한 권한이 있는 IAM 역할을 선택합니다.
14. (선택 사항) 열 이름을 선택하여 Column mapping(열 매핑)에 이름을 입력하면 입력 데이터 파일의 순서대로 열이 매핑됩니다.
15. 데이터 로드(Load data)를 선택하여 데이터 로드를 시작합니다.

로드가 완료되면 데이터를 로드하는 데 사용된 생성된 COPY 명령과 함께 쿼리 편집기가 표시됩니다. COPY의 결과(Result)가 표시됩니다. 성공하면 이제 SQL을 사용하여 로드된 테이블에서 데이터를 선택할 수 있습니다. 오류가 발생하면 시스템 보기 STL_LOAD_ERRORS를 쿼리하여 자세한 내용을 확인합니다. COPY 명령 오류에 대한 내용은 Amazon Redshift Database 개발자 안내서의 [STL_LOAD_ERRORS](#)를 참조하세요.

새 테이블에 데이터를 로드하면 쿼리 에디터 v2가 먼저 데이터베이스에 테이블을 생성한 다음 동일한 워크플로우에서 별도의 작업으로 데이터를 로드합니다.

새로운 테이블에 데이터 로드

COPY 명령은 쿼리 편집기 v2가 Amazon S3에서 데이터를 로드하는 데 사용됩니다. 쿼리 편집기 v2 데이터 로드 마법사에서 생성되고 사용되는 COPY 명령은 Amazon S3에서 복사하기 위해 COPY 명령 구문에 사용할 수 있는 많은 파라미터를 지원합니다. COPY 명령 및 Amazon S3에서 데이터를 로드하는 데 사용되는 옵션에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Amazon S3에서 COPY](#)를 참조하세요.

1. 계속하기 전에 쿼리 에디터 v2의 트리 뷰 패널에서 대상 데이터베이스에 연결되어 있는지 확인합니다. 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 사용하여 데이터를 로드할 클러스터 또는 작업 그룹에 연결할 수 있습니다.



데

이터 로드(Load data)를 선택합니다.

2. 데이터 소스에서 S3에서 로드를 선택합니다.
3. S3 URI(S3 URIs)에서 S3 찾아보기(Browse S3)를 선택하여 로드할 데이터가 들어 있는 Amazon S3 버킷을 찾습니다.
4. 지정된 Amazon S3 버킷이 대상 테이블과 동일한 AWS 리전에 있지 않은 경우 데이터가 있는 AWS 리전에 대한 S3 file location(S3 파일 위치)을 선택합니다.
5. Amazon S3 파일이 실제로 여러 Amazon S3 버킷 URI를 포함하는 매니페스트인 경우 이 파일은 매니페스트 파일임(This file is a manifest file)을 선택합니다.
6. 업로드할 파일의 파일 형식(File format)을 선택합니다. 지원되는 데이터 형식은 CSV, JSON, DELIMITER, FIXEDWIDTH, SHAPEFILE, AVRO, PARQUET 및 ORC입니다. 지정된 파일 형식에 따라 해당 파일 옵션(File options)을 선택할 수 있습니다. 데이터가 암호화된 경우 데이터가 암호화됨(Data is encrypted)을 선택하고 데이터 암호화에 사용된 KMS 키의 Amazon 리소스 이름(ARN)을 입력할 수도 있습니다.

CSV 또는 DELIMITER를 선택하면 구분자 문자를 선택하고, 지정한 수의 행이 실제로 열 이름이고 로드할 데이터가 아닌 경우 헤더 행을 무시할지 여부도 선택할 수 있습니다.

7. 파일을 압축할 압축 방법을 선택합니다. 기본값은 압축 안 함(no compression)입니다.
8. (선택 사항) 고급 설정(Advanced settings)은 다양한 데이터 변환 파라미터(Data conversion parameters)와 로드 작업(Load operations)을 지원합니다. 파일에 필요한 대로 이 정보를 입력합니다.

데이터 변환 및 데이터 로드 파라미터에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [데이터 변환 파라미터](#)와 [데이터 로드 작업](#)을 참조하세요.

9. 다음을 선택합니다.

10. 새 테이블 로드를 선택합니다.

테이블 열은 입력 데이터에서 추론됩니다. 열 및 테이블 세부 정보를 추가하여 테이블 스키마의 정의를 수정할 수 있습니다. 쿼리 에디터 v2 추론 테이블 스키마로 되돌리려면 기본값으로 복원을 선택합니다.

11. 데이터가 로드되는 클러스터 또는 작업 그룹, 데이터베이스 및 스키마를 포함한 대상 테이블의 위치를 확인하거나 선택합니다. 생성할 테이블 이름을 입력합니다.

12. Amazon S3에서 데이터를 로드하는 데 필요한 권한이 있는 IAM 역할을 선택합니다.

13. 테이블 생성을 선택하여 표시된 정의를 사용해 테이블을 생성합니다.

테이블 정의의 검토 요약이 표시됩니다. 데이터베이스에서 테이블이 생성됩니다. 나중에 테이블을 삭제하려면 DROP TABLE SQL 명령을 실행합니다. 자세한 내용은 Amazon Redshift Database 개발자 안내서의 [DROP TABLE](#)을 참조하세요.

14. 데이터 로드(Load data)를 선택하여 데이터 로드를 시작합니다.

로드가 완료되면 데이터를 로드하는 데 사용된 생성된 COPY 명령과 함께 쿼리 편집기가 표시됩니다. COPY의 결과(Result)가 표시됩니다. 성공하면 이제 SQL을 사용하여 로드된 테이블에서 데이터를 선택할 수 있습니다. 오류가 발생하면 시스템 보기 STL_LOAD_ERRORS를 쿼리하여 자세한 내용을 확인합니다. COPY 명령 오류에 대한 내용은 Amazon Redshift Database 개발자 안내서의 [STL_LOAD_ERRORS](#)를 참조하세요.

로컬 파일 설정 및 워크플로우에서 데이터 로드

로컬 파일의 데이터를 기존 테이블 또는 새로운 테이블로 로드할 수 있습니다.

로컬 파일의 데이터를 로드하기 위한 관리자 설정

쿼리 편집기 v2 관리자는 Account settings(계정 설정) 창에서 공통 Amazon S3 버킷을 지정해야 합니다. 계정 사용자는 적절한 권한을 이용해 구성해야 합니다.

- 필수 IAM 권한 - 로컬 파일에서 로드하는 사용자는 s3:ListBucket, s3:GetBucketLocation, s3:putObject, s3:getObject 및 s3:deleteObject 권한이 있어야 합니다. *optional-prefix*를 지정하면 쿼리 편집기 v2와 관련한 이 버킷의 사용을 이 접두사를 가진 객체로 제한할 수

있습니다. 동일한 Amazon S3 버킷을 쿼리 편집기 v2 이외의 용도로 사용할 때 이 옵션을 사용할 수 있습니다. 버킷과 접두사에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [특정 폴더에 대한 사용자 액세스 관리](#)를 참조하세요. 사용자 간 데이터 액세스를 허용하지 않게 하려면, 쿼리 에디터 v2 관리자가 Amazon S3 버킷 정책을 사용하여 aws:userid를 기준으로 객체 액세스를 제한하는 것이 좋습니다. 다음 예에서는 aws:userid 접두사가 있는 Amazon S3 객체에 대해서만 읽기/쓰기 액세스 권한이 있는 *<staging-bucket-name>*에 대한 Amazon S3 권한을 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::<staging-bucket-name>"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::<staging-bucket-name>[/<optional-prefix>]/
        ${aws:userid}/*"
      ]
    }
  ]
}
```

- 데이터 분리 - 사용자가 다른 사용자의 데이터에 (잠깐이라도) 액세스할 수 없도록 하는 것이 좋습니다. 로컬 파일에서 로드할 때는 쿼리 편집기 v2 관리자가 설정한 스테이징 Amazon S3 버킷을 사용합니다. 스테이징 버킷의 버킷 정책을 구성하여 사용자 간에 데이터를 분리합니다. 다음 예제는 *<staging-bucket-name>*의 사용자 간에 데이터를 분리하는 버킷 정책을 보여줍니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "userIdPolicy",
      "Effect": "Deny",
      "Principal": "*",
      "Action": ["s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject"],
      "NotResource": [
        "arn:aws:s3:::<staging-bucket-name>[/<optional-prefix>]/
        ${aws:userid}/*"
      ]
    }
  ]
}
```

로컬 파일의 데이터 로드

기존 테이블에 로컬 파일 데이터를 로드하려면

쿼리 에디터 v2 관리자는 계정 설정 창에서 공통 Amazon S3 버킷을 지정해야 합니다. 쿼리 에디터 v2 는 계정에서 사용하는 공통 Amazon S3 버킷에 로컬 파일을 자동으로 업로드하고, 이후 COPY 명령을 사용하여 데이터를 로드합니다. 쿼리 편집기 v2 로컬 파일 로드 창에서 생성하고 사용하는 COPY 명령은 Amazon S3에서 복사하기 위해 COPY 명령 구문에 사용할 수 있는 많은 파라미터를 지원합니다. COPY 명령 및 Amazon S3에서 데이터를 로드하는 데 사용되는 옵션에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Amazon S3에서 COPY](#)를 참조하세요.

1. 데이터를 로드하려는 데이터베이스에 테이블이 이미 생성되었는지 확인합니다.
2. 쿼리 편집기 v2의 트리 뷰 패널에서 대상 데이터베이스에 연결되어 있는지 확인합니다. 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 사용하여 데이터를 로드할 클러스터 또는 작업 그룹에 연결할 수 있습니다.
3.  데이터 로드(Load data)를 선택합니다.
4. Data source(데이터 소스)에서 Load from local file(로컬 파일에서 로드)을 선택합니다.

데

5. 찾아보기를 선택하여 데이터가 포함된 파일을 찾은 다음 파일 로드를 선택합니다. 기본적으로 확장자가 .csv, .avro, .parquet 및 .orc인 파일이 표시되지만 다른 파일 유형을 선택할 수 있습니다. 최대 파일 크기는 100MB입니다.
6. 업로드할 파일의 파일 형식(File format)을 선택합니다. 지원되는 데이터 형식은 CSV, JSON, DELIMITER, FIXEDWIDTH, SHAPEFILE, AVRO, PARQUET 및 ORC입니다. 지정된 파일 형식에 따라 해당 파일 옵션(File options)을 선택할 수 있습니다. 데이터가 암호화된 경우 데이터가 암호화됨(Data is encrypted)을 선택하고 데이터 암호화에 사용된 KMS 키의 Amazon 리소스 이름(ARN)을 입력할 수도 있습니다.

CSV 또는 DELIMITER를 선택하면 구분자 문자를 선택하고, 지정한 수의 행이 실제로 열 이름이고 로드할 데이터가 아닌 경우 헤더 행을 무시할지 여부도 선택할 수 있습니다.

7. (선택 사항) 고급 설정(Advanced settings)은 다양한 데이터 변환 파라미터(Data conversion parameters)와 로드 작업(Load operations)을 지원합니다. 파일에 필요한 대로 이 정보를 입력합니다.

데이터 변환 및 데이터 로드 파라미터에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [데이터 변환 파라미터](#)와 [데이터 로드 작업](#)을 참조하세요.

8. 다음을 선택합니다.
9. 기존 테이블 로드를 선택합니다.
10. 데이터가 로드되는 클러스터 또는 작업 그룹, 데이터베이스, 스키마 및 테이블 이름을 포함한 대상 테이블의 위치를 확인하거나 선택합니다.
11. (선택 사항) 열 이름을 선택하여 Column mapping(열 매핑)에 입력하면 입력 데이터 파일의 순서대로 열이 매핑됩니다.
12. 데이터 로드(Load data)를 선택하여 데이터 로드를 시작합니다.

로드가 완료되면 로드 성공 여부를 알리는 메시지가 표시됩니다. 성공하면 이제 SQL을 사용하여 로드된 테이블에서 데이터를 선택할 수 있습니다. 오류가 발생하면 시스템 보기 STL_LOAD_ERRORS를 쿼리하여 자세한 내용을 확인합니다. COPY 명령 오류에 대한 내용은 Amazon Redshift Database 개발자 안내서의 [STL_LOAD_ERRORS](#)를 참조하세요.

데이터를 로드하는 데 사용한 COPY 명령 템플릿이 Query history(쿼리 기록)에 나타납니다. 이 COPY 명령 템플릿은 사용한 파라미터 중 일부를 표시하지만 편집기 탭에서 직접 실행할 수는 없습니다. 쿼리 기록에 대한 자세한 내용은 [쿼리 및 탭 기록 보기](#) 단원을 참조하십시오.

새 테이블에 데이터를 로드하면 쿼리 에디터 v2가 먼저 데이터베이스에 테이블을 생성한 다음 동일한 워크플로우에서 별도의 작업으로 데이터를 로드합니다.

새로운 테이블에 로컬 파일 데이터 로드

쿼리 편집기 v2 관리자는 Account settings(계정 설정) 창에서 공통 Amazon S3 버킷을 지정해야 합니다. 로컬 파일은 계정에서 사용하는 공통 Amazon S3 버킷에 자동으로 업로드되고, 이후 쿼리 편집기 v2에서 COPY 명령을 사용하여 데이터를 로드합니다. 쿼리 편집기 v2 로컬 파일 로드 창에서 생성하고 사용하는 COPY 명령은 Amazon S3에서 복사하기 위해 COPY 명령 구문에 사용할 수 있는 많은 파라미터를 지원합니다. COPY 명령 및 Amazon S3에서 데이터를 로드하는 데 사용되는 옵션에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Amazon S3에서 COPY](#)를 참조하세요.

1. 쿼리 편집기 v2의 트리 뷰 패널에서 대상 데이터베이스에 연결되어 있는지 확인합니다. 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 사용하여 데이터를 로드할 클러스터 또는 작업 그룹에 연결할 수 있습니다.



데

이터 로드(Load data)를 선택합니다.

3. Data source(데이터 소스)에서 Load from local file(로컬 파일에서 로드)을 선택합니다.
4. 찾아보기를 선택하여 데이터가 포함된 파일을 찾은 다음 파일 로드를 선택합니다. 기본적으로 확장자가 .csv, .avro, .parquet 및 .orc인 파일이 표시되지만 다른 파일 유형을 선택할 수 있습니다. 최대 파일 크기는 100MB입니다.
5. 업로드할 파일의 파일 형식(File format)을 선택합니다. 지원되는 데이터 형식은 CSV, JSON, DELIMITER, FIXEDWIDTH, SHAPEFILE, AVRO, PARQUET 및 ORC입니다. 지정된 파일 형식에 따라 해당 파일 옵션(File options)을 선택할 수 있습니다. 데이터가 암호화된 경우 데이터가 암호화됨(Data is encrypted)을 선택하고 데이터 암호화에 사용된 KMS 키의 Amazon 리소스 이름(ARN)을 입력할 수도 있습니다.

CSV 또는 DELIMITER를 선택하면 구분자 문자를 선택하고, 지정한 수의 행이 실제로 열 이름이고 로드할 데이터가 아닌 경우 헤더 행을 무시할지 여부도 선택할 수 있습니다.

6. (선택 사항) 고급 설정(Advanced settings)은 다양한 데이터 변환 파라미터(Data conversion parameters)와 로드 작업(Load operations)을 지원합니다. 파일에 필요한 대로 이 정보를 입력합니다.

데이터 변환 및 데이터 로드 파라미터에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [데이터 변환 파라미터](#)와 [데이터 로드 작업](#)을 참조하세요.

7. 다음을 선택합니다.
8. 새 테이블 로드를 선택합니다.
9. 데이터가 로드되는 클러스터 또는 작업 그룹, 데이터베이스 및 스키마를 포함한 대상 테이블의 위치를 확인하거나 선택합니다. 생성할 테이블 이름을 입력합니다.

10. 테이블 생성을 선택하여 표시된 정의를 사용해 테이블을 생성합니다.

테이블 정의의 검토 요약이 표시됩니다. 데이터베이스에서 테이블이 생성됩니다. 나중에 테이블을 삭제하려면 DROP TABLE SQL 명령을 실행합니다. 자세한 내용은 Amazon Redshift Database 개발자 안내서의 [DROP TABLE](#)을 참조하세요.

11. 데이터 로드(Load data)를 선택하여 데이터 로드를 시작합니다.

로드가 완료되면 로드 성공 여부를 알리는 메시지가 표시됩니다. 성공하면 이제 SQL을 사용하여 로드된 테이블에서 데이터를 선택할 수 있습니다. 오류가 발생하면 시스템 보기 STL_LOAD_ERRORS를 쿼리하여 자세한 내용을 확인합니다. COPY 명령 오류에 대한 내용은 Amazon Redshift Database 개발자 안내서의 [STL_LOAD_ERRORS](#)를 참조하세요.

데이터를 로드하는 데 사용한 COPY 명령 템플릿이 Query history(쿼리 기록)에 나타납니다. 이 COPY 명령 템플릿은 사용한 파라미터 중 일부를 표시하지만 편집기 탭에서 직접 실행할 수는 없습니다. 쿼리 기록에 대한 자세한 내용은 [쿼리 및 탭 기록 보기](#) 단원을 참조하십시오.

Amazon Redshift 쿼리 작성

편집기에서 쿼리를 입력하거나 쿼리(Queries) 목록에서 저장된 쿼리를 선택하고 실행(Run)을 선택할 수 있습니다.

기본적으로 제한 100(Limit 100)은 결과를 100행으로 제한하도록 설정됩니다. 더 큰 결과 집합을 반환하기 위해 이 옵션을 해제할 수 있습니다. 이 옵션을 해제하면 매우 큰 결과 집합을 방지하려는 경우 SQL 문에 LIMIT 옵션을 포함할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [ORDER BY 절](#)을 참조하세요.

결과 영역에 쿼리 계획을 표시하려면 설명(Explain)을 설정합니다. 결과의 Explain graph(그래프 설명)를 활성화하여 결과에 계획 설명이 그래픽으로 표시되도록 할 수 있습니다.

쿼리(Queries) 폴더에 쿼리를 저장하려면 저장(Save)을 선택합니다.

쿼리가 성공하면 성공 메시지가 나타납니다. 쿼리가 정보를 반환하면 결과(Results) 섹션에 결과가 표시됩니다. 결과 수가 표시 영역을 초과하면 결과 영역 상단에 숫자가 나타납니다. 숫자를 선택하여 연속적인 결과 페이지를 표시할 수 있습니다.

각 열에 대한 결과(Result)를 필터링하고 정렬할 수 있습니다. 결과 열 머리글에 필터 기준을 입력하려면 열 위로 마우스를 가져가서 열을 필터링하는 기준을 입력할 수 있는 메뉴

()를 표시합니다.

쿼리에 오류가 포함된 경우 쿼리 편집기 v2는 결과 영역에 오류 메시지를 표시합니다. 이 메시지는 쿼리 수정 방법에 대한 정보를 제공합니다.

다음과 같이 결과 영역에서 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 사용하여 쿼리 결과를 내보내거나 복사할 수 있습니다.

- 복사 대상과 JSON 또는 CSV를 선택하여 선택한 행을 파일로 다운로드합니다.
- 행 복사를 선택하여 선택한 행을 클립보드에 복사합니다.
- 헤더와 함께 행 복사를 선택하여 열 머리글이 있는 선택한 행을 클립보드에 복사합니다.

결과 영역에서 내보내기를 선택하고 JSON 또는 CSV를 선택하여 전체 행 결과 집합을 파일로 다운로드합니다. 결과 집합의 행 수는 쿼리의 한도(Limit) 옵션 또는 SQL limit 절에 의해 제한될 수 있습니다. 다운로드된 결과 집합의 최대 크기는 5MB입니다.

바로 가기 키 Ctrl+C(Windows) 또는 Cmd+C(macOS)를 사용하여 현재 결과 페이지의 데이터를 클립보드로 복사할 수도 있습니다. 행을 선택하지 않으면 포커스가 있는 셀이 클립보드에 복사됩니다. 행을 선택하면 선택한 행이 클립보드에 복사됩니다.

새 쿼리 탭을 추가하려면



아이콘을 선택하고 쿼리 탭이 있는 행에 나타나는 Editor(편집기)를 선택합니다. 쿼리 탭은 Isolated session을 사용하거나 사용하지 않습니다. 격리된 세션을 사용하면 한 편집기 탭에서 임시 테이블을 생성하는 것과 같이 SQL 명령의 결과가 다른 편집기 탭에 표시되지 않습니다. 쿼리 편집기 v2에서 편집기 탭을 열 때 기본값은 격리된 세션입니다.

쿼리를 실행하려면

1. 쿼리 영역에서 다음 중 하나를 수행합니다.
 - 쿼리를 입력합니다.
 - 복사한 쿼리를 붙여넣습니다.
 - 쿼리(Queries) 폴더를 선택하고 저장된 쿼리에 대한 컨텍스트 메뉴를 열고(마우스 오른쪽 버튼 클릭) 쿼리 열기(Open query)를 선택합니다.
2. 실행하려는 SQL에 대해 올바른 Cluster(클러스터) 또는 Workgroup(작업 그룹) 및 Database(데이터베이스) 값을 선택했는지 확인합니다.

처음에는 트리 보기에서 Cluster(클러스터) 또는 Workgroup(작업 그룹)을 선택할 수 있습니다. 트리 보기에서 Database(데이터베이스)도 선택합니다.

각 편집기 탭에서 격리된 세션 헤더 근처에 있는 드롭다운 컨트롤을 사용하여 클러스터 또는 작업 그룹 및 데이터베이스를 변경할 수 있습니다.

각 편집기 탭에서 Isolated session(격리된 세션)에서 SQL을 실행할지 선택할 수 있습니다. 격리된 세션에는 데이터베이스에 대한 자체 연결이 있습니다. 다른 쿼리 편집기 세션과 격리된 SQL을 실행하는 데 사용합니다. 연결에 대한 자세한 내용은 [쿼리 편집기 v2 열기](#) 섹션을 참조하세요.

3. 실행을 선택합니다.

결과(Result) 영역이 열리고 쿼리 결과가 표시됩니다.

쿼리에 대한 설명 계획을 표시하려면

1. 쿼리를 선택합니다.
2. 설명(Explain)을 설정합니다.

기본적으로 설명 그래프(Explain graph)도 설정되어 있습니다.

3. 실행을 선택합니다.

쿼리가 실행되고 설명 계획이 쿼리 결과(Result) 영역에 표시됩니다.

쿼리 편집기 v2는 다음 기능을 지원합니다.

- 하나의 쿼리 탭에서 여러 SQL 문을 사용하여 쿼리를 작성할 수 있습니다. 쿼리는 순차적으로 실행되고 각 쿼리에 대해 여러 개의 결과 탭이 열립니다.
- 세션 변수와 임시 테이블을 사용하여 쿼리를 작성할 수 있습니다.
- `${parameter}`로 지정된 대체 가능한 파라미터를 사용하여 쿼리를 작성할 수 있습니다. 대체 가능한 여러 파라미터를 사용하여 SQL 쿼리를 작성하고 SQL 문의 여러 위치에서 동일한 파라미터를 사용할 수 있습니다.

쿼리가 실행되면 파라미터 값을 입력하는 창이 표시됩니다. 쿼리를 실행할 때마다 파라미터 값을 입력하는 창이 표시됩니다.

문제 해결 예는 [예: 매출이 특정 파라미터보다 큼](#)을(를) 참조하세요.

- 쿼리는 자동으로 버전이 관리됩니다. 실행할 쿼리의 이전 버전을 선택할 수 있습니다.
- 워크플로를 계속하기 전에 쿼리가 완료될 때까지 기다릴 필요가 없습니다. 쿼리 편집기를 닫아도 쿼리는 계속 실행됩니다.

- 쿼리를 작성할 때 스키마, 테이블 및 열 이름의 자동 완성이 지원됩니다.

SQL 편집기는 다음 기능을 지원합니다.

- SQL에서 사용되는 시작 괄호와 끝 괄호의 색상이 일치합니다. 편집기에 세로선이 표시되어 대괄호를 일치시키는 데 도움이 됩니다.
- SQL의 섹션을 축소하거나 확장할 수 있습니다.
- SQL에서 텍스트를 검색하고 바꿀 수 있습니다.
- 몇 가지 일반적인 편집 작업에 단축키를 사용할 수 있습니다.
- 문제 영역을 쉽게 찾을 수 있도록 SQL 오류가 편집기에 강조 표시됩니다.

편집 기능의 데모를 보려면 [New and Enhanced Editing Experience in Amazon Redshift query editor](#)(Amazon Redshift 쿼리 에디터 v2의 새롭고 향상된 편집 환경) 동영상을 시청하세요.

쿼리 예제

다음에서 실행할 수 있는 다양한 쿼리 유형에 대한 설명을 찾을 수 있습니다.

이러한 쿼리 중 많은 부분에 사용된 데이터는 tickit 샘플 스키마에서 가져온 것입니다. 샘플 tickit 데이터 로딩에 대한 자세한 내용은 [데이터를 데이터베이스로 로드](#)을 참조하세요. tickit 샘플 데이터에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서에서 [샘플 데이터베이스](#)를 참조하세요.

이러한 예제 쿼리를 실행할 때 sample_data_dev와 같이 편집기에서 올바른 데이터베이스를 선택하는지 확인합니다.

주제

- [예: 세션 변수 설정](#)
- [예: 총 매출 기준 상위 이벤트](#)
- [예: 매출이 특정 파라미터보다 큼](#)
- [예: 임시 테이블 생성](#)
- [예: 임시 테이블에서 선택](#)

예: 세션 변수 설정

다음 명령은 세션에 대해 `search_path` 서버 구성 파라미터를 `public`으로 설정합니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [SET](#) 및 [search_path](#)를 참조하세요.

```
set search_path to public;
```

예: 총 매출 기준 상위 이벤트

다음 쿼리는 매출이 가장 많은 이벤트를 찾습니다.

```
select eventname, count(salesid) totalorders, sum(pricepaid) totalsales
from sales, event
where sales.eventid=event.eventid
group by eventname
order by 3;
```

다음은 결과의 부분 목록입니다.

eventname	totalorders	totalsales
White Christmas	20	9352
Joshua Radin	38	23469
Beach Boys	58	30383
Linda Ronstadt	56	35043
Rascal Flatts	76	38214
Billy Idol	67	40101
Stephenie Meyer	72	41509
Indigo Girls	57	45399
...		

예: 매출이 특정 파라미터보다 큼

다음 쿼리는 판매 수량이 `${numberoforders}`로 지정된 파라미터보다 큰 매출을 찾습니다. 파라미터 값이 7인 경우 결과는 60행입니다. 쿼리를 실행하면 쿼리 편집기 v2는 쿼리 양식 실행(Run query form) 창을 표시하여 SQL 문의 파라미터 값을 수집합니다.

```
select salesid, qtysold
from sales
where qtysold > ${numberoforders}
order by 2;
```

다음은 결과의 부분 목록입니다.


```

salesid qty sold
20005 8
21279 8
130232 8
42737 8
74681 8
67103 8
105533 8
91620 8
121552 8
...

```

예: 임시 테이블 생성

다음 문은 sales 및 event 테이블에서 정보를 선택하여 임시 테이블 eventsalestemp를 생성합니다.

```

create temporary table eventsalestemp as
select eventname, count(salesid) totalorders, sum(pricepaid) totalsales
from sales, event
where sales.eventid=event.eventid
group by eventname;

```

예: 임시 테이블에서 선택

다음 문은 총 주문을 기준으로 정렬된 임시 테이블 eventsalestemp에서 이벤트, 총 주문 및 총 매출을 선택합니다.

```

select eventname, totalorders, totalsales
from eventsalestemp
order by 2;

```

다음은 결과의 부분 목록입니다.

eventname	totalorders	totalsales
White Christmas	20	9352
Joshua Radin	38	23469
Martina McBride	50	52932
Linda Ronstadt	56	35043
Indigo Girls	57	45399
Beach Boys	58	30383
...		

Amazon Redshift의 노트북

노트북을 사용하여 단일 문서에서 여러 SQL 쿼리를 구성하고 주석을 달고 공유할 수 있습니다. 여러 SQL 쿼리 및 마크다운 셀을 노트북에 추가할 수 있습니다. 노트북은 여러 쿼리 및 마크다운 셀을 사용하여 단일 문서에서 데이터 분석과 관련된 쿼리 및 설명을 그룹화하는 방법을 제공합니다. 마크다운 구문으로 텍스트를 추가하고 모양의 서식을 지정하여 데이터 분석 태스크에 대한 컨텍스트 및 추가 정보를 제공할 수 있습니다. 노트북을 팀원과 공유할 수 있습니다.

노트북을 사용하려면 IAM 보안 주체(IAM 사용자 또는 IAM 역할)에 노트북에 대한 권한을 추가해야 합니다. 가장 좋은 방법은 권한 정책을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에 할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요. 쿼리 편집기 v2 관리형 정책 중 하나에 권한을 추가할 수 있습니다. 자세한 내용은 [쿼리 편집기 v2에 액세스](#) 단원을 참조하십시오.

순차적으로 노트북의 셀을 모두 실행할 수 있습니다. 노트북의 SQL 쿼리 셀에는 쿼리 에디터 탭과 거의 동일한 기능이 있습니다. 자세한 내용은 [Amazon Redshift 쿼리 작성](#) 단원을 참조하십시오. 다음은 노트북의 쿼리 에디터 탭과 SQL 셀 간의 차이점입니다.

- 노트북에는 SQL 문에 Explain을 실행할 수 있는 컨트롤이 없습니다.
- 노트북 SQL 셀당 하나의 차트만 생성할 수 있습니다.

쿼리 에디터 v2를 사용하여 만든 파일로 노트북을 가져오고 내보낼 수 있습니다. 파일 확장자는 .ipynb이며 파일 크기는 최대 5MB일 수 있습니다. SQL 및 마크다운 셀은 파일에 저장되어 있습니다. 클러스터 또는 작업 그룹 및 데이터베이스는 내보낸 노트북에 저장되지 않습니다. 가져온 노트북을 열 때 실행할 클러스터 또는 작업 그룹과 데이터베이스를 선택합니다. SQL 셀을 실행한 후에는 결과 탭에서 현재 결과 페이지를 차트로 표시할지 여부를 선택할 수 있습니다. 쿼리 결과 집합은 노트북에 저장되지 않습니다.

노트북 생성

노트북을 생성하여 단일 문서에서 여러 SQL 쿼리를 구성하고 주석을 달고 공유할 수 있습니다.

노트북을 생성하려면

1. 탐색 메뉴에서 에디터 아이콘



을 선택합니다.

2. 더하기 아이콘



을 선택한 다음 노트북을 선택합니다.

기본적으로 SQL 쿼리 셀은 선택한 노트북에 나타납니다.

3. SQL 쿼리 셀에서 다음 중 하나를 수행합니다.

- 쿼리를 입력합니다.
- 복사한 쿼리를 붙여넣습니다.

4. (선택 사항) 더하기 아이콘



을 선택한 다음 마크다운을 선택하여 표준 마크다운 구문을 사용하여 설명 또는 설명 텍스트를 제공할 수 있는 마크다운 셀을 추가합니다.

5. (선택 사항) 더하기 아이콘



을 선택한 다음 SQL을 선택하여 SQL 셀을 삽입합니다.

연필 아이콘



을 사용하여 노트북의 이름을 바꿀 수 있습니다.

노트북 메뉴 아이콘



에서 다음 작업을 수행할 수도 있습니다.



Share with my team(내 팀과 공유) - 태그로 정의된 대로 팀과 노트북을 공유합니다. 팀과 노트북을 공유하려면 기본 태그 `sqlworkbench-team`이 계정의 나머지 팀원과 동일한 값으로 설정되어 있는지 확인합니다. 예를 들어, 관리자는 회계 부서의 모든 사람에 대해 값을 `accounting-team`으로 설정할 수 있습니다. 예시는 [쿼리 편집기 v2를 사용하는 데 필요한 권한](#)을 확인하세요.



Export(내보내기) - 노트북을 `.ipynb` 확장자를 가진 로컬 파일로 내보냅니다.



쿼리 가져오기 - 로컬 파일에서 노트북의 셀로 쿼리를 가져옵니다. 확장자가 .sql 및 .txt인 파일을 가져올 수 있습니다.



Save version(버전 저장) - 노트북 버전을 만듭니다. 노트북 버전을 확인하려면 저장된 노트북으로 이동하여 Version history(버전 기록)를 엽니다.



Duplicate(복제) - 노트북 사본을 만들고 새 노트북 탭에서 엽니다.



Shortcuts(바로 가기) - 노트북을 작성할 때 사용할 수 있는 바로 가기를 표시합니다.

노트북으로 가져오기

전체 노트북 또는 개별 SQL 셀을 쿼리 에디터 v2 노트북으로 가져올 수 있습니다.

로컬 파일에서 전체 노트북을 내 노트북으로 가져오려면



가져오기를 선택한 다음, 노트북 가져오기를 선택합니다. 노트북이 들어 있는 ..ipynb 파일로 이동합니다. 노트북을 현재 열려 있는 노트북 폴더로 가져옵니다. 그런 다음 노트북 편집기를 사용하여 노트북을 열 수 있습니다.

로컬 파일에서 노트북의 SQL 셀로 쿼리를 가져오려면



가져오기를 선택한 다음, 쿼리 가져오기를 선택합니다. 쿼리 가져오기 창에서 화면의 지시에 따라 새 노트북 또는 기존 노트북으로 쿼리로 가져올 수 있는 파일과 폴더를 선택합니다. 파일의 확장자는 .sql 또는 .txt여야 합니다. 각 쿼리는 최대 10,000자까지 가능합니다. 기존 노트북에 추가할 때는 저장된 노트북 목록의 모든 노트북 중에서 원하는 노트북을 선택합니다. 가져온 쿼리는 노트북 끝에 SQL 셀로 추가됩니다. 새 노트북을 선택하는 경우, 노트북 이름을 선택하면 현재 열려 있는 저장된 노트북 폴더에 노트북이 만들어집니다.

Note

macOS에서 TextEdit 애플리케이션을 사용하여 .sql 파일을 만들 때 파일에 숨겨진 확장자가 추가되는 문제가 발생할 수 있습니다. 예를 들어 TextEdit에서 만든 Test.sql이라는 이름

의 파일은 Test.sql.rtf로 저장될 수 있습니다. 쿼리 에디터 v2는 확장자가 .rtf인 파일을 지원하지 않습니다. 그러나 TextEdit을 사용하여 .sql 파일을 만든 후 일반 텍스트 파일로 저장하면 파일에 숨겨진 .txt 확장자가 추가로 붙습니다. 예를 들어 Text.sql이라는 파일은 Text.sql.txt로 저장될 수 있습니다. .rtf 확장자와 달리, 쿼리 에디터 v2는 .txt 확장자를 가진 파일을 지원하므로, 노트북으로 쿼리를 가져올 때 Text.sql.txt가 지원됩니다.

노트북 데모를 보려면 다음 비디오를 시청하세요. [쿼리 에디터 v2의 Amazon Redshift SQL Notebooks](#)

AWS Glue Data Catalog 쿼리

쿼리 에디터 v2를 사용하여 특정 SQL 명령을 사용하고 이 섹션에 요약된 권한을 부여하여 AWS Glue Data Catalog에 카탈로그화된 데이터를 쿼리할 수 있습니다. 기본적으로 AWS Glue Data Catalog는 쿼리 편집기 v2 데이터베이스에 awsgluecatalog라는 이름으로 나열됩니다. 일부 Amazon Redshift AWS 리전에서는 AWS Glue Data Catalog 쿼리를 사용할 수 없습니다. SHOW 명령을 사용하여 이 기능을 사용할 수 있는지 확인하세요. AWS Glue에 대한 자세한 내용은 AWS Glue 개발자 안내서의 [AWS Glue란 무엇입니까?](#)를 참조하세요.

Note

AWS Glue Data Catalog 쿼리는 Amazon Redshift RA3 노드 유형 클러스터와 Amazon Redshift Serverless에서만 지원됩니다.

다음 SQL 명령을 사용하여 데이터 웨어하우스를 구성하고 카탈로그화된 AWS Glue 데이터베이스 객체를 볼 수 있습니다.

- SHOW - 현재 연결된 데이터 웨어하우스에 대해 awsgluecatalog가 마운트되었는지 여부를 표시합니다. 예를 들어 data_catalog_auto_mount 파라미터 값을 표시하려면 다음을 실행합니다.

```
SHOW data_catalog_auto_mount;
```

자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서에서 [SHOW](#)를 참조하세요.

- ALTER SYSTEM - data_catalog_auto_mount의 시스템 수준 구성을 변경합니다. 예를 들어 data_catalog_auto_mount 파라미터 값을 on으로 변경하려면 다음을 실행합니다.

```
ALTER SYSTEM SET data_catalog_auto_mount = on;
```

변경 사항은 프로비저닝된 클러스터가 재부팅되거나 서버리스 작업 그룹이 자동으로 일시 중지되었다가 다시 시작될 때 적용됩니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 가이드에서 [시스템 변경](#)을 참조하세요.

- SHOW SCHEMAS - 스키마 목록을 표시합니다. awsdatalog이라는 데이터베이스의 스키마는 AWS Glue Data Catalog에 카탈로그화된 AWS Glue를 나타냅니다. 예를 들어 이러한 스키마를 표시하려면 다음을 실행합니다.

```
SHOW SCHEMAS FROM DATABASE awsdatalog;
```

자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서에서 [SHOW SCHEMAS](#)를 참조하세요.

- SHOW TABLES - 스키마에 있는 테이블 목록을 표시합니다. 예를 들어 myglue 스키마에 있는 awsdatalog라는 이름의 AWS Glue Data Catalog 데이터베이스에 있는 테이블을 표시하려면 다음을 실행합니다.

```
SHOW TABLES FROM SCHEMA awsdatalog.myschema;
```

자세한 내용은 Amazon Redshift Database 개발자 안내서의 [SHOW TABLES](#)를 참조하세요.

- SHOW COLUMNS - 테이블의 열 목록을 표시합니다. 예를 들어 myglue 스키마 및 mytable 테이블에 있는 awsdatalog라는 이름의 AWS Glue Data Catalog 데이터베이스에 있는 테이블을 표시하려면 다음을 실행합니다.

```
SHOW COLUMNS FROM TABLE awsdatalog.myglue.mytable;
```

자세한 내용을 알아보려면 Amazon Redshift 데이터베이스 개발자 안내서의 [SHOW COLUMNS](#)를 참조하세요.

IAM 사용자 또는 역할에 AWS Glue Data Catalog을 쿼리할 수 있는 권한을 부여하려면

1. 트리 뷰 창에서 데이터베이스 사용자 이름 및 암호 인증 방법을 사용하여 프로비저닝된 클러스터나 서버리스 작업 그룹에 있는 최초 데이터베이스에 연결합니다. 예를 들어 클러스터 또는 작업 그룹을 만들 때 사용한 관리자 사용자 및 암호를 사용하여 dev 데이터베이스에 연결합니다.
2. 편집기 탭에서 다음 SQL 문을 실행하여 IAM 사용자에게 AWS Glue Data Catalog 액세스 권한을 부여합니다.

```
GRANT USAGE ON DATABASE awsdatalog to "IAM:myIAMUser"
```

여기서 ***IAM:myIAMUser***는 AWS Glue Data Catalog에 대한 사용 권한을 부여할 IAM 사용자입니다. IAM 역할에 대한 ***IAM:myIAMRole***에 사용 권한을 부여하는 방법도 있습니다.

- 트리 뷰 창에서 이전에 만든 클러스터 또는 작업 그룹에 대한 연결을 편집하거나 삭제합니다. 다음 방법 중 하나를 사용하여 클러스터 또는 작업 그룹에 연결합니다.
 - 클러스터에서 `awsdatacatalog` 데이터베이스에 연결하려면, IAM ID를 사용하는 임시 보안 인증 인증 방법을 사용해야 합니다. 이 인증 방법에 대한 자세한 내용은 [Amazon Redshift 데이터베이스에 연결](#) 단원을 참조하세요. 쿼리 편집기 v2 관리자가 연결 창에 이 인증 방법을 표시하도록 계정의 계정 설정을 구성해야 할 수 있습니다.
 - 미리 보기 작업 그룹에서 `awsdatacatalog` 데이터베이스에 연결하려면 페더레이션 사용자 인증 방법을 사용해야 합니다. 이 인증 방법에 대한 자세한 내용은 [Amazon Redshift 데이터베이스에 연결](#) 단원을 참조하세요.
- 부여된 권한이 있다면 IAM ID를 사용하여 AWS Glue Data Catalog를 상대로 SQL을 실행할 수 있습니다.

연결한 후에는 쿼리 편집기 v2를 사용하여 AWS Glue Data Catalog에 카탈로그화된 데이터를 쿼리할 수 있습니다. 쿼리 편집기 v2 트리 뷰 창에서 클러스터 또는 작업 그룹과 `awsdatacatalog` 데이터베이스를 선택합니다. 편집기나 노트북 창에서 올바른 클러스터 또는 작업 그룹이 선택되어 있는지 확인합니다. 선택한 데이터베이스는 최초 Amazon Redshift 데이터베이스(예: `dev`)여야 합니다. 쿼리 작성에 대한 자세한 내용은 [Amazon Redshift 쿼리 작성](#) 및 [Amazon Redshift의 노트북](#) 단원을 참조하세요. `awsdatacatalog`라는 이름이 지정된 데이터베이스는 계정의 외부 Data Catalog 데이터베이스를 참조하도록 예약되어 있습니다. `awsdatacatalog` 데이터베이스에 대한 쿼리는 읽기 전용일 수만 있습니다. `SELECT` 문에서 테이블을 참조하려면 세 부분으로 구성된 표기법을 사용해야 합니다. 여기서 첫 번째 부분은 데이터베이스 이름이고, 두 번째 부분은 AWS Glue 데이터베이스 이름이며, 세 번째 부분은 AWS Glue 테이블 이름입니다.

```
SELECT * FROM awsdatacatalog.<aws-glue-db-name>.<aws-glue-table-name>;
```

AWS Glue Data Catalog 데이터를 읽고 Amazon Redshift 테이블을 채우는 다양한 시나리오를 수행할 수 있습니다.

다음 예제 SQL은 AWS Glue에 정의된 두 테이블을 조인합니다.

```
SELECT pn.emp_id, alias, role, project_name
FROM "awsdatacatalog"."empl_db"."project_name_table" pn,
"awsdatacatalog"."empl_db"."project_alias_table" pa
WHERE pn.emp_id = pa.emp_id;
```

다음 예제 SQL은 Amazon Redshift 테이블을 만들고 두 AWS Glue 테이블의 조인에서 얻은 데이터로 테이블을 채웁니다.

```
CREATE TABLE dev.public.glue AS
SELECT pn.emp_id, alias, role, project_name
FROM "awsdatacatalog"."empl_db"."project_name_table" pn,
"awsdatacatalog"."empl_db"."project_alias_table" pa
WHERE pn.emp_id = pa.emp_id;
```

Amazon S3 테이블 쿼리(미리 보기)

Query Editor V2를 사용하여 AWS Glue Data Catalog에 탑재된 Amazon S3 테이블 카탈로그에 저장된 데이터를 쿼리할 수 있습니다. Amazon S3 테이블 카탈로그는 만드는 시점에 AWS Glue Data Catalog에 탑재되며, 동일한 계정에서 동일한 AWS 리전에 있는 모든 프로비저닝된 클러스터 및 서버리스 작업 그룹에 외부 데이터베이스로 자동으로 표시됩니다. Amazon Redshift를 사용하여 Amazon S3 테이블에 액세스하는 방법에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [Accessing Amazon S3 tables with Amazon Redshift](#)를 참조하세요.

데이터 레이크 쿼리

이 자습서의 태스크 세트에 따라 Amazon S3 데이터 레이크의 데이터를 쿼리할 수 있습니다. 먼저, [AWS Glue Data Catalog](#)에서 외부 데이터베이스를 참조하는 외부 스키마를 생성합니다. 그런 다음 Amazon S3 데이터 레이크의 데이터를 쿼리할 수 있습니다.

데모: 데이터 레이크 쿼리

데이터 레이크를 쿼리하는 방법의 데모를 보려면 다음 동영상을 시청하세요. [Amazon Redshift 쿼리 편집기 v2에서 데이터 레이크 쿼리](#)

사전 조건

쿼리 에디터 v2에서 데이터 레이크를 사용하기 전에 Amazon Redshift 환경에서 다음을 설정했는지 확인하세요.

- AWS Glue를 사용하여 Amazon S3 데이터를 크롤링하고 AWS Lake Formation에 대해 Data Catalog를 활성화하세요.
- AWS Glue가 활성화된 AWS Lake Formation용 Data Catalog로 Amazon Redshift에 대한 IAM 역할을 생성합니다. 이 절차에 대한 자세한 내용은 [AWS Lake Formation에 대해 활성화된 AWS Glue Data Catalog를 사용하여 Amazon Redshift용 IAM 역할 만들기](#)를 참조하세요. Redshift Spectrum 및

Lake Formation 사용에 대한 자세한 내용은 [AWS Lake Formation과 함께 Redshift Spectrum 사용](#)을 참조하세요.

- Lake Formation 데이터베이스에서 쿼리할 테이블에 SELECT 권한을 부여합니다. 이 절차에 대한 자세한 내용은 [Lake Formation 데이터베이스에서 쿼리할 테이블에 SELECT 권한 부여](#)를 참조하세요.

Lake Formation 콘솔(<https://console.aws.amazon.com/lakeformation/>)의 권한 섹션, 데이터 레이크 권한 페이지에서 IAM 역할, AWS Glue 데이터베이스 및 테이블에 적절한 권한이 있는지 확인할 수 있습니다.

- 연결된 사용자에게 Amazon Redshift 데이터베이스에서 스키마를 생성하고 데이터 레이크의 데이터에 액세스할 수 있는 권한이 있는지 확인하세요. 쿼리 에디터 v2에서 데이터베이스에 연결할 때는 보안 인증 정보가 포함된 인증 방법을 선택합니다. 보안 인증 정보는 데이터베이스 사용자 또는 IAM 사용자일 수 있습니다. 연결된 사용자에게는 적절한 권한 및 데이터베이스 권한(예:superuser)이 있어야 합니다. 클러스터 또는 작업 그룹을 생성한 Amazon Redshift admin 사용자에게는 superuser 권한이 있으며 스키마를 생성하고 Redshift 데이터베이스를 관리할 수 있습니다. 쿼리 에디터 v2에서 데이터베이스에 연결하는 방법에 대한 자세한 내용은 [Amazon Redshift 데이터베이스에 연결](#) 섹션을 참조하세요.

외부 스키마 생성

Amazon S3 데이터 레이크의 데이터를 쿼리하려면 먼저 외부 스키마를 생성합니다. 외부 스키마는 [AWS Glue Data Catalog](#)에서 외부 데이터베이스를 참조합니다.

1. 쿼리 에디터 v2의 편집기 보기에서



생

성을 선택한 다음 스키마를 선택합니다.

2. 스키마 이름(Schema name)을 입력합니다.
3. 스키마 유형으로 외부를 선택합니다.
4. Data Catalog 세부 정보에서 리전은 기본적으로 Redshift 데이터베이스가 있는 AWS 리전입니다.
5. 외부 스키마가 매핑되고 AWS Glue 테이블에 대한 참조가 포함된 AWS Glue 데이터베이스를 선택합니다.
6. Amazon S3에서 데이터를 쿼리하는 데 필요한 권한이 있는 Amazon Redshift용 IAM 역할을 선택합니다.
7. Data Catalog에 대한 권한이 있는 IAM 역할을 선택할 수도 있습니다(선택 사항).
8. 스키마 생성을 선택합니다.

스키마가 트리 보기 패널의 데이터베이스 아래에 나타납니다.

스키마를 만들 때 데이터베이스에 대한 권한 거부 오류가 표시되면 연결된 사용자에게 스키마를 만들 수 있는 데이터베이스 권한이 있는지 확인하세요.

Amazon S3 데이터 레이크에서 데이터 쿼리

이전 절차에서 생성한 스키마를 사용합니다.

1. 트리 보기 패널에서 스키마를 선택합니다.
2. 테이블 정의를 보려면 테이블을 선택합니다. 테이블 열 및 데이터 유형이 표시됩니다.
3. 테이블을 쿼리하려면 테이블을 선택하고 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)에서 테이블 선택을 선택합니다.
4. 편집기에서 쿼리를 실행합니다.

다음 예제 SQL은 쿼리 에디터 v2에서 `flightscsv`라는 이름이 지정된 AWS Glue 테이블의 모든 행을 쿼리하기 위해 생성되었습니다. 출력에 표시된 열과 행은 간소화를 위해 잘려 있습니다.

```
SELECT * FROM "dev"."mydatalake_schema"."flightscsv";
```

year	quarter	month	dom	day_of_week	fl_date	unique_carrier	airline_id
year	carrier	tail_num	fl_num				
2016	4	10	19	3	10/19/16	00	20304
00		N753SK	3086				
2016	4	10	19	3	10/19/16	00	20304
00		N753SK	3086				
2016	4	10	19	3	10/19/16	00	20304
00		N778SK	3087				
2016	4	10	19	3	10/19/16	00	20304
00		N778SK	3087				
...							

데이터 공유

다른 클러스터의 사용자가 데이터를 쿼리할 수 있도록 `datashare`를 생성할 수 있습니다. 공유하려는 데이터가 포함된 클러스터를 생산자 클러스터라고 합니다. 공유하려는 데이터베이스 객체를 대상으로 생산자 클러스터에 `datashare`를 생성합니다. 스키마, 테이블, 보기 및 SQL 사용자 정의 함수 (UDF)를 공유할 수 있습니다. 데이터를 공유하려는 클러스터를 소비자 클러스터라고 합니다. 소비자 클러스터

에서 `datashare`에서 데이터베이스를 생성합니다. 그런 다음 소비자 클러스터의 사용자가 데이터를 쿼리할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 가이드의 [Getting started with data sharing](#)를 참조하세요.

datashare 생성

클러스터에서 생산자 클러스터로 사용할 `datashare`를 생성합니다. `Datashare`에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 가이드의 [Amazon Redshift에서 데이터 공유 시 고려 사항](#) 섹션을 참조하세요.

1. 사용하려는 생산자 클러스터에서 데이터베이스를 선택합니다.
2. `Datashare`를 만듭니다. 예:

```
create datashare mysource;
```

3. `Datashare`에 대한 권한을 설정합니다. 예:

```
grant alter, share on datashare mysource to admin;
```

4. 공유하려는 데이터베이스 개체에 대한 사용 권한을 설정합니다. 예:

```
alter datashare mysource add schema public;
```

```
alter datashare mysource add table public.event;
```

5. `Datashare`에 액세스하기 위해 소비자 클러스터 네임스페이스에 대한 권한을 설정합니다. 예:

```
grant usage on datashare mysource to namespace '2b12345-1234-5678-9012-bb1234567890';
```

Datashares 보기

생산자 클러스터에서 생성한 `datashares`를 표시할 수 있습니다.

1. 생산자 클러스터를 선택합니다.
2. `Datashares`를 표시합니다. 예:

```
show datashares;
```

```
share_name share_owner source_database consumer_database share_type createdate
is_publicaccessible share_acl producer_account producer_namespace
test_datashare 100 db_producer NULL OUTBOUND 2/15/2022 FALSE admin
123456789012 p1234567-8765-4321-p10987654321
```

소비자 데이터베이스 생성

소비자 클러스터에서 datashare에서 데이터베이스를 생성합니다. 이 단계에서는 동일한 계정의 두 클러스터 간에 데이터를 공유하는 방법을 설명합니다. AWS 계정 간 데이터 공유에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 가이드의 [AWS 계정 간 데이터 공유](#)를 참조하세요.

SQL 명령 또는 쿼리 편집기 v2 트리 보기 패널을 사용하여 데이터베이스를 만들 수 있습니다.

SQL을 사용하려면

1. 계정 및 생산자 클러스터의 네임스페이스에서 datashare에서 데이터베이스를 생성합니다. 예:

```
create database share_db from datashare mysource of account '123456789012'
namespace 'p1234567-8765-4321-p10987654321';
```

2. 사용자가 데이터베이스와 스키마에 액세스할 수 있도록 권한을 설정합니다. 예:

```
grant usage on database share_db to usernames;
```

```
grant usage on schema public to usernames;
```

쿼리 편집기 v2 트리 보기 패널을 사용하려면

1.  생(Create)을 선택한 후 데이터베이스(Database)를 선택합니다.
2. 데이터베이스 이름(Database name)을 입력합니다.
3. (선택 사항) 사용자 및 그룹(Users and groups)을 선택하고 데이터베이스 사용자(Database user)를 선택합니다.
4. datashare를 사용하여 생성(Create using a datashare)을 선택합니다.
5. Datashare를 선택합니다.

생

6. 데이터베이스 생성을 선택합니다.

쿼리 편집기 v2 트리 보기 패널에 새



datash

데이터베이스가 표시됩니다.

7. 사용자가 데이터베이스와 스키마에 액세스할 수 있도록 권한을 설정합니다. 예:

```
grant usage on database share_db to usernames;
```

```
grant usage on schema public to usernames;
```

datashare 객체 쿼리

데이터베이스 클러스터에서 데이터베이스, 스키마 및 객체 이름의 세 부분으로 구성된 표기법으로 표현된 완전한 객체 이름을 사용하여 데이터 공유를 쿼리할 수 있습니다.

1. 쿼리 편집기 v2 트리 보기 패널에서 스키마를 선택합니다.
2. 테이블 정의를 보려면 테이블을 선택합니다.

테이블 열 및 데이터 유형이 표시됩니다.

3. 테이블을 쿼리하려면 테이블을 선택하고 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 사용하여 Select table(테이블 선택)을 선택합니다.
4. SELECT 명령을 사용하여 테이블 쿼리 예:

```
select top 10 * from test_db.public.event;
```

쿼리 에디터 v2를 사용하여 예약된 쿼리

Amazon Redshift Query Editor V2를 사용하면 SQL 쿼리를 자동화하여 예약에 따라 실행할 수 있습니다. 예약된 쿼리는 지정된 시간 또는 간격으로 자동 실행되는 SQL 스테이트먼트로, 이를 통해 반복되는 데이터 작업 및 분석 작업을 효율적으로 관리할 수 있습니다. 배치 처리를 간소화하거나, 정기 보고서를 작성하거나, Amazon Redshift 환경 내에서 데이터 파이프라인을 유지 관리하려는 경우 쿼리를 예약할 수 있습니다.

예약된 쿼리를 사용하면 추출, 전환, 적재(ETL) 워크플로를 자동화하고, 최신 인사이트로 대시보드를 새로고침하고, 다양한 데이터 관리 경로를 운영하는 것이 가능합니다. 다음 페이지에서는 Amazon

Redshift 워크로드를 최적화하기 위해 예약된 쿼리를 작성, 구성 및 관리하는 프로세스를 자세히 설명합니다.

쿼리 에디터 v2를 사용하여 쿼리 예약 생성

Amazon Redshift 쿼리 에디터 v2를 사용하여 SQL 문의 실행 일정을 생성할 수 있습니다. 비즈니스 요구 사항에 맞는 시간 간격으로 SQL 문의 실행 일정을 생성합니다. 예약된 쿼리를 실행할 시간이 되면 Amazon EventBridge에서 쿼리가 시작되고 Amazon Redshift Data API가 사용됩니다.

SQL 문 실행 일정을 생성하려면

1. 편집기



뷰에서



일정을 선택하여 SQL 문의 실행 일정을 생성할 수 있습니다.

2. 일정을 정의할 때 다음 정보를 제공합니다.

- 쿼리를 실행하는 데 필요한 권한을 맡는 IAM 역할. 이 IAM 역할은 클러스터 또는 작업 그룹에도 연결됩니다.
- 클러스터 또는 작업 그룹 액세스 권한을 부여하기 위한 AWS Secrets Manager 또는 임시 보안 인증 정보의 인증 값. 이러한 인증 방법은 데이터 API에서 지원됩니다. 자세한 내용은 [예약된 쿼리 인증](#) 섹션을 참조하세요.
- 데이터베이스가 있는 클러스터 또는 작업 그룹.
- 쿼리할 데이터가 있는 데이터베이스의 이름.
- 예약된 쿼리의 이름과 설명입니다. 쿼리 편집기 v2는 사용자가 제공한 예약된 쿼리 이름 앞에 "QS2-"를 접두사로 붙입니다. 쿼리 에디터 v1은 예약된 쿼리 이름 앞에 'QS-'를 붙입니다.
- 일정에 따라 실행할 SQL 문.
- 일정을 정의하는 일정 빈도 및 반복 옵션 또는 cron 형식 값. 자세한 내용을 알아보려면 Amazon CloudWatch Events 사용 설명서의 [Cron 표현식](#)을 참조하세요.
- 필요에 따라 표준 Amazon SNS 알림을 사용하여 예약된 쿼리를 모니터링할 수 있습니다. Amazon SNS 알림에 제공한 이메일 주소를 확인해야 할 수 있습니다. 이메일에서 Amazon SNS 알림을 받을 이메일 주소를 확인할 수 있는 링크를 확인합니다. 자세한 정보는 Amazon Simple Notification Service 개발자 안내서의 [이메일 알림](#)을 참조하세요. 쿼리가 실행 중이지만

SNS 주제에 게시된 메시지가 표시되지 않는 경우 Amazon EventBridge 사용 설명서의 [내 규칙이 실행되지만 내 Amazon SNS 주제에 어떤 메시지도 게시되지 않음](#)을 참조하세요.

3. 쿼리 예약을 선택하여 일정을 저장 및 활성화하고 예약된 쿼리 뷰의 쿼리 목록에 해당 일정을 추가합니다.

예약된 쿼리



뷰에는 클러스터 및 작업 그룹에 대한 모든 예약된 쿼리가 나열됩니다. 이 뷰에서는 쿼리 예약 세부 정보를 표시하고, 일정을 활성화 또는 비활성화하고, 일정을 편집하고, 예약된 쿼리를 삭제할 수 있습니다. 쿼리 세부 정보를 볼 때 일정과 함께 쿼리를 실행한 기록도 볼 수 있습니다.

Note

쿼리 예약 실행은 24시간 동안만 일정 기록 목록에서 확인할 수 있습니다. 일정에 따라 실행되는 쿼리는 쿼리 편집기 v2의 쿼리 기록 뷰에 나타나지 않습니다.

쿼리 예약 데모

쿼리 예약 데모를 보려면 다음 동영상을 시청하세요. [쿼리 예약 동영상 데모](#).

쿼리 예약을 위한 권한 설정

쿼리를 예약하려면 일정을 정의하는 AWS Identity and Access Management(IAM) 사용자 및 일정과 연결된 IAM 역할에 Amazon EventBridge와 Amazon Redshift Data API를 사용할 수 있는 IAM 권한을 구성해야 합니다. 예약된 쿼리에서 이메일을 받으려면 선택적으로 지정하는 Amazon SNS 알림도 구성해야 합니다.

다음은 AWS 관리형 정책을 사용하여 권한을 제공하는 작업에 대한 설명입니다. 하지만 환경에 따라 허용되는 권한의 범위를 좁힐 수 있습니다.

쿼리 에디터 v2에 로그인한 IAM 사용자의 경우 IAM 콘솔(<https://console.aws.amazon.com/iam/>)을 사용하여 IAM 사용자를 편집합니다.

- Amazon Redshift 및 쿼리 에디터 v2 작업을 실행할 수 있는 권한 외에도 IAM 사용자에게 해당 AmazonEventBridgeFullAccess 및 AmazonRedshiftDataFullAccess AWS 관리형 정책을 연결합니다.
- 또는 역할에 권한을 할당하고 사용자에게 역할을 할당합니다.

예약된 문을 정의할 때 지정하는 IAM 역할의 리소스 ARN에 대한 sts:AssumeRole 권한을 허용하는 정책을 연결합니다. 역할 수입에 대한 자세한 내용은 IAM 사용 설명서의 [사용자에게 역할 전환 권한 부여](#)를 참조하세요.

다음 예에서는 123456789012 계정에서 IAM 역할 myRedshiftRole을 수입하는 권한 정책을 보여줍니다. IAM 역할 myRedshiftRole은 예약된 쿼리가 실행되는 클러스터 또는 작업 그룹에 연결된 IAM 역할이기도 합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AssumeIAMRole",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": [
        "arn:aws:iam::123456789012:role/myRedshiftRole"
      ]
    }
  ]
}
```

쿼리를 예약하는 데 사용된 IAM 역할의 신뢰 정책을 업데이트하여 IAM 사용자가 쿼리를 맡을 수 있도록 합니다.

```
{
  "Sid": "AssumeRole",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::123456789012:user/myIAMUsername"
  },
  "Action": "sts:AssumeRole"
}
```

예약된 쿼리 실행을 허용하도록 지정하는 IAM 역할의 경우 IAM 콘솔(<https://console.aws.amazon.com/iam/>)을 사용하여 IAM 역할을 편집합니다.

- AmazonRedshiftDataFullAccess 및 AmazonEventBridgeFullAccess AWS 관리형 정책을 IAM 역할에 연결합니다. AmazonRedshiftDataFullAccess 관리형 정책은 RedshiftDataFullAccess 키로 태그가 지정된 Redshift Serverless 작업 그룹에 대해서만 redshift-serverless:GetCredentials 권한을 허용합니다.

예약된 쿼리 인증

쿼리를 예약할 때 SQL 실행 시 다음 인증 방법 중 하나를 사용합니다. 각 방법마다 쿼리 에디터 v2에서 서로 다른 입력 조합이 필요합니다. 이러한 인증 방법은 SQL 문을 실행하는 데 사용되는 데이터 API에서 지원됩니다.

쿼리를 실행하는 데 사용되는 데이터베이스 사용자 또는 역할에는 필요한 데이터베이스 권한이 있어야 합니다. 예를 들어 mytable 테이블에 IAMR:MyRedshiftQEv2Scheduler 권한을 부여하려면 다음 SQL 명령을 실행합니다.

```
GRANT all ON TABLE mytable TO "IAMR:MyRedshiftQEv2Scheduler";
```

클러스터 또는 작업 그룹의 데이터베이스 사용자 목록을 보려면 시스템 뷰 PG_USER_INFO를 쿼리하세요.

Note

쿼리를 예약하는 모든 Redshift Serverless 작업 그룹에는 RedshiftDataFullAccess 키로 태그가 지정됩니다. 자세한 내용은 [Amazon Redshift Data API에 대한 액세스 권한 부여](#) 섹션을 참조하세요.

작업 그룹에 태그를 지정하는 대신 redshift-serverless:GetCredentials를 허용하는 인라인 정책을 IAM 역할(일정과 함께 지정됨)에 추가할 수 있습니다. 예제:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UseTemporaryCredentialsForAllServerlessWorkgroups",
      "Effect": "Allow",
      "Action": "redshift-serverless:GetCredentials",
      "Resource": [
        "arn:aws:redshift-serverless:*:*:workgroup/*"
      ]
    }
  ]
}
```

```

    ]
  }
}

```

AWS Secrets Manager

이 방법으로 AWS Secrets Manager에 저장되어 있는 secret-arn에 대한 보안 암호 값을 제공합니다. 이 보안 암호에는 데이터베이스에 연결하기 위한 자격 증명이 포함되어 있습니다. 클러스터 또는 작업 그룹을 생성할 때 적절한 보안 인증을 사용하여 암호를 생성했을 수 있습니다. 보안 암호는 키 RedshiftDataFullAccess로 태그가 지정되어야 합니다. 태그 키가 아직 없는 경우 AWS Secrets Manager 콘솔을 사용하여 추가합니다. 보안 암호 생성에 대한 정보는 [데이터베이스 연결 자격 증명을 위한 보안 암호 생성](#) 섹션을 참조하세요.

최소 권한에 대한 자세한 내용은 AWS Secrets Manager User Guide의 [Creating and Managing Secrets with AWS Secrets Manager](#)를 참조하세요.

임시 자격 증명

이 방법을 사용하는 경우 클러스터의 데이터베이스에 연결할 때 데이터베이스 이름과 데이터베이스 사용자 값을 입력합니다. 작업 그룹의 데이터베이스에 연결할 때 데이터베이스 이름만 입력해야 합니다.

클러스터에 연결할 때, AmazonRedshiftDataFullAccess 정책은 redshift_data_api_user라는 데이터베이스 사용자에게 redshift:GetClusterCredentials에 대한 권한을 허용합니다. 다른 데이터베이스 사용자를 사용하여 SQL 문을 실행하려면 redshift:GetClusterCredentials를 허용하는 정책을 클러스터에 연결된 IAM 역할에 추가합니다. 다음 예제 정책에서는 데이터베이스 사용자를 awsuser 및 myuser를 허용합니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UseTemporaryCredentialsForAllDbUsers",
      "Effect": "Allow",
      "Action": "redshift:GetClusterCredentials",
      "Resource": [
        "arn:aws:redshift:*:dbuser:*/awsuser",

```

```

        "arn:aws:redshift:*:*:dbuser:*/myuser"
    ]
}
}

```

쿼리 예약 기록을 볼 수 있는 권한 설정

사용자가 쿼리 예약 기록을 볼 수 있도록 허용하려면 IAM 역할(예약과 함께 지정됨) 신뢰 관계를 편집하여 권한을 추가합니다.

다음은 IAM 사용자 *myIAMusername*이 쿼리 예약 기록을 볼 수 있도록 허용하는 IAM 역할 내 신뢰 정책의 예입니다. IAM 사용자에게 sts:AssumeRole 권한을 허용하는 대신 IAM 역할에 이 권한을 허용하도록 선택할 수 있습니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "redshift.amazonaws.com",
          "redshift-serverless.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Sid": "AssumeRole",
      "Effect": "Allow",

```

```

        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/myIAMUsername"
        },
        "Action": "sts:AssumeRole"
    }
]
}

```

예약된 쿼리 모니터링

이메일 알림을 보내도록 지정한 Amazon SNS 주제의 경우 쿼리 편집기 v2에서 SNS 알림 섹션으로 이동하여 모니터링을 켜고 SNS 주제 생성으로 주제를 생성하여 Amazon SNS 주제를 생성합니다. 쿼리 편집기 v2가 Amazon SNS 주제를 생성하고 Amazon EventBridge에 대한 액세스 정책에 서비스 주체를 추가합니다. 다음은 Amazon SNS 주제에 생성된 액세스 정책 예입니다. 이 예제에서는 AWS 리전 *us-west-2*, AWS 계정 *123456789012*, Amazon SNS 주제 *select-version-pdx-testunload*가 사용되었습니다.

JSON

```

{
  "Version": "2012-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "Allow_Publish_Events",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sns:Publish",
      "Resource": "arn:aws:sns:us-west-2:123456789012:select-version-pdx-testunload"
    }
  ]
}

```

예약된 쿼리가 실행되면 Amazon SNS에서 AWS 알림 이메일을 보냅니다. 다음 예는 Amazon SNS 알림 주제 *may25a-SNS*를 사용하여 AWS 계정 *123456789012* 내 AWS 리전 *eu-north-1*에서 실행된 예약된 쿼리 *QS2-may25a*에 대해 *myemail@example.com*으로 보낸 이메일을 보여줍니다.

```
{
  "version": "0",
  "id": "8e4323ec-5258-7138-181b-91290e30ff9b",
  "detail-type": "Scheduled Event",
  "source": "aws.events",
  "account": "123456789012",
  "time": "2023-05-25T15:22:00Z",
  "region": "eu-north-1",
  "resources": [
    "arn:aws:events:eu-north-1:123456789012:rule/QS2-may25a"
  ],
  "detail": {}
}
```

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:

<https://sns.eu-north-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:eu-north-1:123456789012:may25a-SNS:0c1a3d05-39c2-4507-bc3d-47250513d7b0&Endpoint=myemail@example.com>

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>

쿼리 예약 설정 문제 해결

쿼리 예약에 문제가 있는 경우 다음을 고려하세요.

쿼리가 실행되지 않음

일정에 사용된 IAM 역할에 임시 클러스터 보안 인증을 가져올 수 있는 권한이 있는지 확인합니다. 프로비저닝된 클러스터에 대한 권한은 `redshift:GetClusterCredentialsWithIAM`입니다. Redshift Serverless 작업 그룹에 대한 권한은 `redshift-serverless:GetCredentials`입니다.

예약된 기록이 표시되지 않음

AWS 콘솔에 로그인하는 데 사용된 IAM 사용자 또는 IAM 역할이 쿼리 예약에 사용된 IAM 역할의 신뢰 정책에 추가되지 않았습니까.

AWS Secrets Manager를 사용하여 예약된 쿼리를 연결할 때 암호에 `RedshiftDataFullAccess` 키 태그가 지정되었는지 확인합니다.

예약된 쿼리가 AWS Secrets Manager 연결을 사용하는 경우 쿼리를 예약하는 데 사용된 IAM 역할에는 역할에 연결된 관리형 정책 `SecretsManagerReadWrite`에 상응하는 정책이 있어야 합니다.

쿼리 기록 상태가 **Failed**입니다.

쿼리가 실패한 이유에 대한 자세한 내용은 `SYS_QUERY_HISTORY` 시스템 뷰를 참조하세요. 반복적인 문제는 쿼리를 실행하는 데 사용된 데이터베이스 사용자 또는 역할에 SQL을 실행하는 데 필요한 권한이 없을 수 있다는 것입니다. 자세한 내용은 [예약된 쿼리 인증](#) 섹션을 참조하세요.

다음 SQL은 SYS_QUERY_HISTORY 뷰를 쿼리하여 실패한 쿼리를 반환합니다.

```
SELECT user_id, query_id, transaction_id, session_id, database_name, query_type,
       status, error_message, query_text
FROM sys_query_history
WHERE status = 'failed';
```

실패한 특정 예약 쿼리에 대한 세부 정보를 찾으려면 [AWS CloudShell을 사용하여 예약된 쿼리 결과 보기](#)를 참조하세요.

AWS CloudShell을 사용하여 예약된 쿼리 결과 보기

AWS CloudShell을 사용하여 예약 쿼리에 대한 세부 정보를 찾을 수 있습니다. 다음 절차에 표시된 AWS CLI 명령을 실행하려면 적절한 권한이 있어야 합니다.

예약 쿼리 결과를 보려면 다음을 수행하세요.

1. AWS 콘솔에서 AWS CloudShell 명령 프롬프트를 엽니다. AWS CloudShell에 대한 자세한 내용은 AWS CloudShell 사용 설명서에서 [AWS CloudShell이란 무엇인가요](#)를 참조하세요.
2. 예약된 쿼리의 IAM 역할을 가정합니다. 역할을 가정하려면 쿼리 편집기 v2에서 예약된 쿼리와 연결된 IAM 역할을 찾아 AWS CloudShell의 AWS CLI 명령에서 사용합니다. 예를 들어 역할 scheduler의 경우, 예약된 쿼리에서 사용되는 역할을 가정하기 위해 AWS STS 명령을 입력합니다.

```
aws sts assume-role --role-arn "arn:aws:iam::123456789012:role/scheduler" --role-session-name "scheduler-test"
```

반환되는 보안 인 증은 다음과 유사합니다.

```
{
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
    "SessionToken": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY...",
    "Expiration": "2023-08-18T18:19:44+00:00"
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "AROA35B2NH6WBTP70NL4E:scheduler-test",
    "Arn": "arn:aws:sts::123456789012:assumed-role/scheduler/scheduler-test"
  }
}
```

```
}
```

3. IAM 역할을 가정하여 표시된 보안 인증을 사용하여 AWS CLI에서 환경 변수를 생성합니다. 이러한 토큰은 만료 시간 전에 사용해야 합니다. 예를 들어 AWS CloudShell에 다음을 입력합니다.

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
export AWS_SESSION_TOKEN=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY...
```

4. 실패한 쿼리의 오류를 보려면 AWS CLI 명령을 실행하여 문을 설명합니다. SQL 문 ID는 쿼리 편집기 v2에서 예약된 쿼리의 일정 기록 섹션에 표시된 ID에서 가져옵니다.

```
aws redshift-data describe-statement --id 130d2620-05d2-439c-b7cf-815d9767f513
```

이 예에서 예약된 SQL `select * from users limit 100`은 `users` 테이블이 존재하지 않는다는 SQL 오류를 발생시킵니다.

```
{
  "CreatedAt": "2023-08-18T17:39:15.563000+00:00",
  "Duration": -1,
  "Error": "ERROR: relation \"users\" does not exist",
  "HasResultSet": false,
  "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "QueryString": "select * from users limit 100\n-RequestID=a1b2c3d4-5678-90ab-cdef-EXAMPLE22222; TraceID=1-633c5642-4039308d03f3a0ba53dbdf6f",
  "RedshiftPid": 1073766651,
  "RedshiftQueryId": 0,
  "ResultRows": -1,
  "ResultSize": -1,
  "Status": "FAILED",
  "UpdatedAt": "2023-08-18T17:39:16.116000+00:00",
  "WorkgroupName": "default"
}
```

쿼리 결과 시각화

쿼리를 실행하고 결과가 표시되면 Chart(차트)를 설정하여 현재 결과 페이지의 그래픽 시각화를 표시할 수 있습니다. 다음 컨트롤을 사용하여 차트의 내용, 구조 및 모양을 정의할 수 있습니다.



트레이스

차트에서 관련 그래픽 표시 집합을 나타냅니다. 한 차트에서 여러 트레이스를 정의할 수 있습니다.

Type

트레이스 유형을 정의하여 데이터를 다음 중 하나로 나타낼 수 있습니다.

- 산점도 또는 거품형 차트를 위한 분산형 차트입니다.
- 세로 또는 가로 막대로 데이터 범주를 나타내는 막대형 차트입니다.
- 채워진 영역을 정의하는 영역 차트입니다.
- 막대를 사용하여 빈도 분포를 나타내는 히스토그램입니다.
- 각 슬라이스가 전체의 백분율을 나타내는 원형으로 데이터를 표현하기 위한 원형 차트입니다.
- 프로세스의 다양한 단계를 통해 데이터를 나타내는 깔때기형 또는 깔때기 영역 차트입니다.
- OHLC(open-high-low-close) 차트는 일반적으로 시간 간격을 나타내는 x축을 따라 열린 값, 높은 값, 낮은 값, 닫힌 값을 나타내는 재무 데이터에 자주 사용됩니다.
- 타임라인에서 범주에 대한 값 범위를 나타내는 원통형 차트입니다.
- 일련의 중간 값을 통해 초기 값의 증가 또는 감소를 나타내는 폭포 차트입니다. 값은 시간 간격이나 범주를 나타낼 수 있습니다.
- 시간에 따른 값의 변화를 나타내는 꺾은선형 차트입니다.

X축

X축을 따라 표시할 값이 포함된 테이블 열을 지정합니다. 설명 값이 포함된 열은 일반적으로 차원 데이터를 나타냅니다. 양적 값을 포함하는 열은 일반적으로 실제 데이터를 나타냅니다.

Y축

Y축을 따라 표시할 값이 포함된 테이블 열을 지정합니다. 설명 값이 포함된 열은 일반적으로 차원 데이터를 나타냅니다. 양적 값을 포함하는 열은 일반적으로 실제 데이터를 나타냅니다.

서브플롯

차트 데이터 표시를 추가로 정의할 수 있습니다.

변형

변환을 정의하여 트레이스 데이터를 필터링할 수 있습니다. 분할 변환을 사용하면 단일 소스 트레이스에서 여러 기록을 표시할 수 있습니다. 집계 변환을 사용하면 트레이스를 평균값 또는 최솟값으로 표시할 수 있습니다. 정렬 변환을 사용하면 트레이스를 정렬할 수 있습니다.

일반 모양

배경색, 여백 색, 디자인 팔레트의 색 눈금, 텍스트 스타일 및 크기, 제목 스타일 및 크기, 모드 표시 줄의 기본값을 설정할 수 있습니다. 끌기, 클릭 및 가리키기의 상호 작용을 정의할 수 있습니다. 메타 텍스트를 정의할 수 있습니다. 트레이스, 축, 범례 및 주석의 기본 모양을 정의할 수 있습니다.

차트를 생성하려면

1. 쿼리를 실행하고 결과를 가져옵니다.
2. 차트(Charts)를 설정합니다.
3. 기록(Trace)을 선택하고 데이터 시각화를 시작합니다.
4. 다음 중 하나에서 차트 스타일을 선택합니다.
 - 분산형
 - 막대형
 - 영역
 - 히스토그램(Histogram)
 - 파이형
 - 퍼널
 - 퍼널 영역
 - OHLC(Open-High-Low-Close)
 - 캔들스틱
 - 폭포
 - 행
5. 스타일(Style)을 선택하여 색상, 축, 범례 및 주석을 포함한 모양을 사용자 지정합니다. 텍스트, 도형 및 이미지를 추가할 수 있습니다.
6. 주석(Annotations)을 선택하여 텍스트, 모양 및 이미지를 추가합니다.
7. 차트 표시 내용을 업데이트하려면 새로 고침(Refresh)을 선택합니다. 전체 화면(Full screen)을 선택하여 차트 표시를 확장합니다.

예: 파이형 차트를 생성하여 쿼리 결과 시각화

다음 예에서는 샘플 데이터베이스의 Sales 테이블을 사용합니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 가이드의 [샘플 데이터베이스](#)를 참조하세요.

다음은 파이형 차트에 대한 데이터를 제공하기 위해 실행하는 쿼리입니다.

```
select top 5 eventname, count(salesid) totalorders, sum(pricepaid) totalsales
from sales, event
where sales.eventid=event.eventid group by eventname
order by 3;
```

총 매출을 기준으로 상위 이벤트에 대한 파이형 차트를 생성하려면

1. 쿼리를 실행합니다.
2. 쿼리 결과 영역에서 차트(Chart)를 설정합니다.
3. 추적(Trace)을 선택합니다.
4. 유형(Type)에서 파이(Pie)를 선택합니다.
5. 값(Values)에서 totalsales를 선택합니다.
6. 레이블(Labels)에서 eventname을 선택합니다.
7. 스타일(Style)을 선택하고 일반(General)을 선택합니다.
8. 색조(Colorscales)에서 카테고리(Categorical)를 선택한 다음 Pastel2를 선택합니다.



예: 매출과 판매를 비교하기 위한 혼합형 차트 생성

이 예제의 단계를 수행하여 매출 데이터의 막대형 차트와 판매 데이터의 꺾은선형 그래프를 결합하는 차트를 만듭니다. 다음 예제에서는 tickit 샘플 데이터베이스의 Sales 테이블을 사용합니다. 자세한 내용은 Amazon Redshift Database Developer Guide Amazon Redshift 데이터베이스 개발자 가이드의 [Sample database](#)(샘플 데이터베이스)를 참조하세요.

다음은 차트의 데이터를 제공하기 위해 실행하는 쿼리입니다.

```
select eventname, total_price, total_qty_sold
from (select eventid, total_price, total_qty_sold, ntile(1000) over(order by
total_price desc) as percentile
      from (select eventid, sum(pricepaid) total_price, sum(qtysold) total_qty_sold
            from tickit.sales
            group by eventid)) Q, tickit.event E
where Q.eventid = E.eventid
and percentile = 1
order by total_price desc;
```

매출과 판매를 비교하기 위한 혼합형 차트를 생성하려면

1. 쿼리를 실행합니다.
2. 쿼리 결과 영역에서 차트(Chart)를 설정합니다.
3. trace o(트레이스 o)의 Type(유형)에서 Bar(막대형)를 선택합니다.
4. X는 eventname을 선택합니다.
5. Y는 total_price를 선택합니다.

X축을 따라 이벤트 이름을 표시하는 막대형 차트가 표시됩니다.

6. Style(스타일)에서 Traces(트레이스)를 선택합니다.
7. Name(이름)에 Revenue를 입력합니다.
8. Style(스타일)에서 Axes(축)를 선택합니다.
9. Titles(제목)에서 Y를 선택하고 Revenue를 입력합니다.

왼쪽 Y 축에 Revenue 레이블이 표시됩니다.

10. Structure(구조)에서 Traces(트레이스)를 선택합니다.
11.  Trace(트레이스)를 선택합니다.

트레이스 1 옵션이 표시됩니다.

12. 유형(Type)에서 Line(꺾은선형)를 선택합니다.
13. X는 eventname을 선택합니다.
14. Y는 total_qty_sold를 선택합니다.

15. Axes To Use(사용할 축)의 Y Axis(Y 축)에 대해



선택합니다.

Y Axis(Y 축)에 Y2가 표시됩니다.

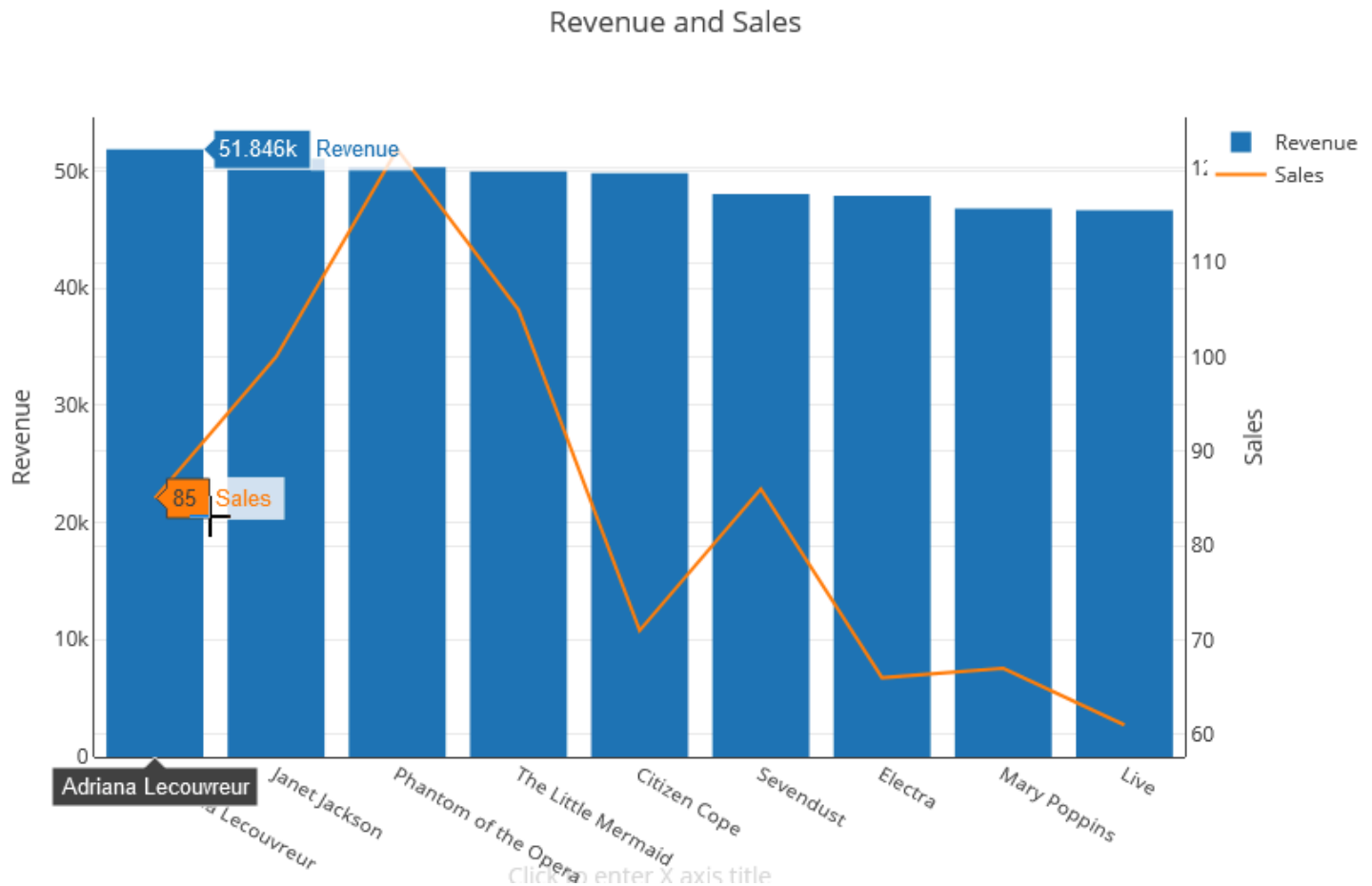
16. Style(스타일)에서 Axes(축)를 선택합니다.

17. Titles(제목)에서 Y2를 선택합니다.

18. Name(이름)에 Sales를 입력합니다.

19. Lines(선)에서 Y:Sales를 선택합니다.

20. AXis line(축 선)에서 Show(표시)를 선택하고 Position(위치)에서 Right(오른쪽)를 선택합니다.



데모: Amazon Redshift 쿼리 편집기 v2를 사용하여 시각화 작성

시각화를 작성하는 방법에 대한 데모를 보려면 다음 동영상을 시청하세요. [Amazon Redshift 쿼리 편집기 v2를 사용하여 시각화 작성](#)

팀으로 협업 및 공유

팀과 쿼리를 공유할 수 있습니다.

팀은 공동으로 작업하면서 쿼리 편집기 v2 리소스를 공유하는 사용자 집합으로 정의됩니다. 관리자는 IAM 역할에 태그를 추가하여 팀을 생성할 수 있습니다. 자세한 내용은 [쿼리 편집기 v2를 사용하는 데 필요한 권한](#) 섹션을 참조하세요.

쿼리 저장 및 검색

팀과 쿼리를 공유하려면 먼저 쿼리를 저장하세요. 저장된 쿼리를 보고 삭제할 수 있습니다.

쿼리를 저장하려면

1. 쿼리를 준비하고 저장(Save)을 선택합니다.
2. 쿼리 제목을 입력합니다.
3. 저장을 선택합니다.

저장된 쿼리를 찾아보려면

1. 탐색 창에서 쿼리(Queries)를 선택합니다.
2. 내 쿼리(My queries), 나에 의해 공유됨(Shared by me) 또는 내 팀에 공유됨(Shared to my team) 상태의 쿼리를 볼 수 있습니다. 이러한 쿼리는 개별 쿼리로 표시되거나 사용자가 생성한 폴더 내에 나타날 수 있습니다.

쿼리 공유

팀과 쿼리를 공유할 수 있습니다. 또한 저장된 쿼리의 기록을 보고 쿼리 버전을 관리할 수 있습니다.

팀과 쿼리를 공유하려면 기본 태그 sqlworkbench-team이 계정의 나머지 팀원과 동일한 값으로 설정되어 있는지 확인합니다. 예를 들어, 관리자는 회계 부서의 모든 사람에 대해 값을 accounting-team으로 설정할 수 있습니다. 문제 해결 예는 [쿼리 편집기 v2를 사용하는 데 필요한 권한](#)을(를) 참조하세요.

팀과 쿼리를 공유하려면

1. 탐색 창에서 쿼리(Queries)를 선택합니다.
2. 공유하려는 쿼리의 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 내 팀과 공유(Share with my team)를 선택합니다.

3. 쿼리를 공유할 팀을 하나 이상 선택한 다음 공유 옵션 저장(Save sharing options)을 선택합니다.

쿼리 버전 관리

SQL 쿼리를 저장할 때마다 쿼리 편집기 v2에서 이를 새 버전으로 저장합니다. 이전 쿼리 버전을 찾아 보거나, 쿼리 사본을 저장하거나, 쿼리를 복원할 수 있습니다.

쿼리 버전을 관리하려면

1. 탐색 창에서 쿼리(Queries)를 선택합니다.
2. 작업할 쿼리에 대한 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 엽니다.
3. 버전 기록(Version history)을 선택하여 쿼리 버전 목록을 엽니다.
4. 버전 기록(Version history) 페이지에서 다음 작업을 수행할 수 있습니다.
 - 선택 항목으로 되돌리기(Revert to selected) - 선택한 버전으로 되돌리고 이 버전으로 작업을 계속합니다.
 - 다른 이름으로 선택 항목 저장(Save selected as) - 편집기에서 새 쿼리를 생성합니다.

Amazon Redshift Query Editor V1을 사용하여 데이터베이스 쿼리

쿼리 편집기를 사용하면 Amazon Redshift 클러스터에서 호스팅하는 데이터베이스에 대한 쿼리를 손쉽게 실행할 수 있습니다. 클러스터를 생성한 후 바로 Amazon Redshift 콘솔에서 쿼리 편집기를 사용해 쿼리를 실행할 수 있습니다.

Note

이 원본 쿼리 편집기를 사용하면 Amazon Redshift Serverless 내의 데이터는 쿼리할 수 없습니다. 대신 Amazon Redshift 쿼리 편집기 v2를 사용하십시오.

2021년 2월에 업데이트된 쿼리 편집기가 배포되고 쿼리 편집기를 사용할 수 있는 권한 부여 권한이 변경되었습니다. 새 쿼리 편집기는 Amazon Redshift Data API를 사용하여 쿼리를 실행합니다. AWS 관리형 AWS Identity and Access Management(IAM) 정책인 AmazonRedshiftQueryEditor 정책이 필요한 권한을 포함하도록 업데이트되었습니다. 사용자 정의 IAM 정책이 있는 경우 해당 정책을 업데이트해야 합니다. AmazonRedshiftQueryEditor를 가이드로 사용합니다. AmazonRedshiftQueryEditor의 변경 사항으로 다음과 같은 것들이 있습니다.

- 쿼리 편집기 문 결과를 관리하기 위한 권한은 문 소유자 사용자를 필요로 합니다.
- Secrets Manager를 사용하여 데이터베이스에 연결할 수 있는 권한이 추가되었습니다.

자세한 내용은 [Amazon Redshift 콘솔 쿼리 편집기 사용에 필요한 권한](#) 섹션을 참조하세요.

새 쿼리 에디터에서 클러스터에 연결하면 2가지 인증 방법 중 하나를 사용할 수 있습니다.

쿼리 편집기를 사용해 다음을 수행할 수 있습니다.

- 단일 SQL 문 쿼리를 실행합니다.
- 100MB 크기의 결과 집합을 쉼표 구분 값(CSV) 파일로 다운로드합니다.
- 다시 사용할 수 있도록 쿼리를 저장합니다. 유럽(파리) 리전, 아시아 태평양(오사카) 리전, 아시아 태평양(홍콩) 리전 또는 중동(바레인) 리전에서는 쿼리를 저장할 수 없습니다.
- 사용자 정의 테이블에서 쿼리 런타임 세부 정보를 확인합니다.
- 나중에 실행되도록 쿼리를 예약합니다.
- 쿼리 편집기에서 생성한 쿼리 기록을 봅니다.
- Enhanced VPC Routing을 사용하여 클러스터에 대해 쿼리를 실행합니다.

쿼리 편집기 고려 사항

쿼리 편집기를 사용할 때 쿼리 작업에 대해 다음을 고려합니다.

- 최대 쿼리 기간은 24시간입니다.
- 최대 쿼리 결과 크기는 100MB입니다. 호출이 100MB를 초과하는 응답 데이터를 반환하면 호출이 종료됩니다.
- 쿼리 결과의 최대 보존 시간은 24시간입니다.
- 최대 쿼리 문 크기는 100KB입니다.
- 클러스터는 Amazon VPC 서비스 기반의 Virtual Private Cloud(VPC)에 있어야 합니다.
- 쿼리 편집기에서는 트랜잭션을 사용할 수 없습니다. 트랜잭션에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [BEGIN](#) 섹션을 참조하세요.
- 쿼리는 최대 3,000자까지 저장할 수 있습니다.

SQL 클라이언트 도구를 사용하여 Amazon Redshift 데이터 웨어하우스에 연결

JDBC(Java Database Connectivity), Python 및 ODBC(Open Database Connectivity) 연결을 통해 SQL 클라이언트 도구에서 Amazon Redshift 데이터 웨어하우스로 연결할 수 있습니다. Amazon Redshift는 SQL 클라이언트 도구 또는 라이브러리를 제공하거나 설치하지 않습니다. 이러한 도구 또는 라이브러리를 사용하여 데이터 웨어하우스의 데이터로 작업하려면 클라이언트 컴퓨터나 Amazon EC2 인스턴스에 해당 도구를 설치합니다. JDBC, Python 또는 ODBC 드라이버를 지원하는 SQL 클라이언트 도구라면 대부분 사용할 수 있습니다.

이 주제 끝에 소개된 섹션 목록을 사용하여 JDBC, Python 또는 ODBC 연결을 사용하도록 클라이언트 컴퓨터 또는 Amazon EC2 인스턴스를 구성하는 과정을 살펴보세요. 서버에 클라이언트 연결과 관련된 보안 옵션에 대해서도 알 수 있습니다. 또한 [Amazon Redshift RSQL](#) 같은 SQL 클라이언트 도구를 이용한 설정 및 연결 관련 정보를 찾아보세요. 아직 사용할 비즈니스 인텔리전스 도구가 없는 경우 이러한 도구를 사용해 볼 수 있습니다. 이 섹션에서 데이터 연결에 대해 알아볼 수도 있습니다. 마지막으로 데이터 웨어하우스에 연결하면서 문제가 발생하는 경우, 해결책을 확인할 수 있는 문제 해결 정보도 살펴볼 수 있습니다.

클라이언트 도구와의 연결을 위한 권장 사항

IP 주소를 사용하여 Redshift 클러스터에 연결하는 경우, 중단이 발생하거나 연결이 끊기고 새 가용 영역(AZ)에서 클러스터가 온라인 상태가 되면 추가 다운타임이 발생할 수 있습니다. 하지만 여전히 IP 주소를 사용하여 애플리케이션을 Redshift에 연결하려면 클러스터의 Virtual Private Cloud(VPC) 엔드포인트에 연결된 프라이빗 IP 주소를 사용하세요. 속성 탭 아래의 네트워크 및 보안에 있는 클러스터 세부 정보에서 정보를 찾을 수 있습니다.

Note

애플리케이션이 리더 노드의 IP 주소를 사용하여 Redshift 클러스터에 액세스하는 경우 권장되는 모범 사례는 클러스터 엔드포인트 URL을 사용하도록 변경하는 것입니다. 자세한 내용은 [Amazon Redshift에서 연결 구성](#)을 참조하세요.

주제

- [Amazon Redshift에서 연결 구성](#)
- [연결을 위한 보안 옵션 구성](#)
- [클라이언트 도구 및 코드에서 연결](#)

- [인증 프로파일을 사용하여 Amazon Redshift에 연결](#)
- [Amazon Redshift 연결 문제 해결](#)

Amazon Redshift에서 연결 구성

다음 섹션에서 SQL 클라이언트 도구에서 클러스터에 연결하도록 JDBC, Python 및 ODBC 연결을 구성하는 방법을 확인해보세요. 이 섹션에서는 JDBC, Python 및 ODBC 연결을 설정하는 방법에 대해 설명합니다. SSL(Secure Sockets Layer) 및 서버 인증서를 사용하여 클라이언트와 서버 간의 통신을 암호화하는 방법에 대해서도 설명합니다.

Amazon Redshift용 JDBC, Python 및 ODBC 드라이버

클러스터 데이터를 사용하려면 클라이언트 컴퓨터 또는 인스턴스에서 연결하기 위한 JDBC, Python 또는 ODBC 드라이버가 있어야 합니다. JDBC, Python 또는 ODBC 데이터 액세스 API 작업을 사용할 수 있는 애플리케이션을 코딩하거나, 혹은 JDBC, Python 또는 ODBC를 지원하는 SQL 클라이언트 도구를 사용하세요.

Amazon Redshift를 통해 JDBC, Python 및 ODBC 드라이버를 다운로드할 수 있습니다. 이 드라이버는 지원에서 지원됩니다. PostgreSQL 드라이버는 테스트되지 않았으며 Amazon Redshift 팀에서 지원하지 않습니다. Amazon Redshift 클러스터에 연결할 때 Amazon Redshift 전용 드라이버를 사용하세요. Amazon Redshift 드라이버는 다음과 같은 이점이 있습니다.

- IAM, SSO 및 페더레이션 인증 지원.
- 새로운 Amazon Redshift 데이터 유형 지원.
- 인증 프로파일 지원.
- Amazon Redshift 개선 사항과 함께 향상된 성능.

JDBC 및 ODBC 드라이버를 다운로드하여 클러스터 연결을 구성하는 방법에 대한 자세한 내용은 [Amazon Redshift용 JDBC 드라이버 버전 2.x 연결 구성](#), [Amazon Redshift Python 커넥터](#) 및 [Amazon Redshift용 ODBC 드라이버 버전 2.x 연결 구성](#) 섹션을 참조하세요.

IAM 역할에 대한 모범 사례를 비롯한 IAM ID 관리에 대한 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#) 섹션을 참조하세요.

클러스터 연결 문자열 찾기

SQL 클라이언트 도구를 사용하여 클러스터에 연결하려면 클러스터 연결 문자열이 있어야 합니다. Amazon Redshift 콘솔의 클러스터 세부 정보 페이지에서 클러스터 연결 문자열을 확인할 수 있습니다.

클러스터의 연결 문자열을 찾으려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 목록에서 클러스터 이름을 선택하여 세부 정보를 엽니다.
3. JDBC URL 및 ODBC URL 연결 문자열은 일반 정보(General information) 섹션에서 추가 세부 정보와 함께 사용할 수 있습니다. 각 문자열은 클러스터가 실행되는 AWS 리전을 기반으로 합니다. 해당 연결 문자열 옆에 있는 아이콘을 클릭하여 복사합니다.

클러스터 엔드포인트에 연결하려면 [DescribeClusters API 요청](#)의 클러스터 엔드포인트 URL을 사용할 수 있습니다. 다음은 클러스터 엔드포인트 예시입니다.

```
mycluster.cmeaswqeuae.us-east-2.redshift.amazonaws.com
```

클러스터에 대한 사용자 지정 도메인 이름을 설정한 경우 이를 사용하여 클러스터에 연결할 수도 있습니다. 사용자 지정 도메인 이름 생성에 대한 자세한 내용은 [사용자 지정 도메인 이름 설정](#)을 참조하세요.

Note

연결할 때 클러스터 노드의 IP 주소 또는 VPC 엔드포인트의 IP 주소를 사용하지 마세요. 불필요한 중단을 방지하기 위해 항상 Redshift 엔드포인트를 사용하세요. 예외적으로 엔드포인트 URL을 사용하지 않는 유일한 상황은 사용자 지정 도메인 이름을 사용하는 경우입니다. 자세한 내용은 [클라이언트 연결에 사용자 지정 도메인 이름 사용](#)을 참조하세요.

Amazon Redshift용 JDBC 드라이버 버전 2.x 연결 구성

JDBC 드라이버 버전 2.x 연결을 사용하여 여러 서드 파티 SQL 클라이언트 도구에서 Amazon Redshift 클러스터에 연결할 수 있습니다. Amazon Redshift JDBC 커넥터는 오픈 소스 솔루션을 제공합니다. 소스 코드를 탐색하고, 개선 사항을 요청하고, 문제를 보고하고, 기여를 제공할 수 있습니다.

JDBC 드라이버 변경 사항에 대한 최신 정보는 [변경 로그](#)를 참조하세요.

기본적으로 Amazon Redshift JDBC 드라이버는 연결이 시간 초과되는 것을 방지하기 위해 TCP KeepAlive를 사용하도록 구성됩니다. 연결 URL에서 관련 속성을 설정하여 드라이버가 KeepAlive 패

킷 전송을 시작하는 시기를 지정하거나 기능을 해제할 수 있습니다. 연결 URL의 구문에 대한 자세한 내용은 [연결 URL 구축](#) 섹션을 참조하세요.

속성	설명
TCPKeepAlive	TCP KeepAlive를 해제하려면 이 속성을 FALSE로 설정합니다.

주제

- [Amazon Redshift JDBC 드라이버, 버전 2.1 다운로드](#)
- [Amazon Redshift JDBC 드라이버, 버전 2.2 설치](#)
- [JDBC URL 가져오기](#)
- [연결 URL 구축](#)
- [JDBC와 Apache Maven의 연결 구성](#)
- [인증 및 SSL 구성](#)
- [로깅 구성](#)
- [데이터 형식 변환](#)
- [준비된 명령문 지원 사용](#)
- [JDBC 드라이버 버전 2.2과 1.x의 차이점](#)
- [JDBC 드라이버 버전 2.x용 초기화\(.ini\) 파일 생성](#)
- [JDBC 드라이버 버전 2.x 구성 옵션](#)
- [JDBC 드라이버 버전 2.x의 이전 버전](#)

Amazon Redshift JDBC 드라이버, 버전 2.1 다운로드

Note

Amazon Redshift JDBC 2.x 드라이버는 스레드 세이프가 설계되지 않았습니다. 두 개 이상의 스레드가 동시에 동일한 연결을 사용하려고 하면 교착 상태, 오류, 잘못된 결과 또는 기타 예상치 못한 동작이 발생할 수 있습니다.

다중 스레드 애플리케이션이 있는 경우 동시 액세스를 방지하기 위해 드라이버에 대한 액세스를 동기화하는 것이 좋습니다.

Amazon Redshift는 JDBC 4.2 API와 호환되는 도구용 드라이버를 제공합니다. 이 드라이버의 클래스 이름은 `com.amazon.redshift.Driver`입니다.

JDBC 드라이버를 설치하고 JDBC 드라이버 라이브러리를 참조하며 드라이버 클래스를 등록하는 방법에 대한 자세한 내용은 다음 주제를 참조하세요.

Amazon Redshift JDBC 드라이버 버전 2.x를 사용하는 각 컴퓨터에 대해 Java 런타임 환경(JRE) 8.0이 설치되어 있는지 확인합니다.

데이터베이스 인증을 위해 Amazon Redshift JDBC 드라이버를 사용하는 경우 Java 클래스 경로에 AWS SDK for Java 1.11.118 이상이 있는지 확인합니다. AWS SDK for Java가 설치되어 있지 않으면 JDBC 4.2 호환 드라이버와 AWS SDK용 드라이버 종속 라이브러리가 포함된 ZIP 파일을 다운로드합니다.

- [JDBC 4.2 호환 드라이버 버전 2.x 및 AWS SDK 드라이버 종속 라이브러리](#)

이 ZIP 파일에는 JDBC 4.2 호환 드라이버 버전 2.x와 AWS SDK for Java 1.x 드라이버 종속 라이브러리 파일이 포함되어 있습니다. JDBC 드라이버와 동일한 위치에 종속 jar 파일의 압축을 풉니다. JDBC 드라이버만 CLASSPATH에 있어야 합니다.

전체 AWS SDK for Java 1.x는 이 ZIP 파일에 포함되어 있지 않습니다. 그러나 AWS Identity and Access Management(IAM) 데이터베이스 인증에 필요한 AWS SDK for Java 1.x 드라이버 종속 라이브러리는 포함되어 있습니다.

IAM 데이터베이스 인증에 필요한 AWS SDK가 있는 이 Amazon Redshift JDBC 드라이버를 사용합니다.

전체 AWS SDK for Java 1.x를 설치하려면 AWS SDK for Java Developer Guide의 [AWS SDK for Java 1.x](#)를 참조하세요.

- [JDBC 4.2 호환 드라이버 버전 2.x\(AWS SDK 제외\)](#)

JDBC 드라이버 버전 2.x 소프트웨어 라이선스를 검토하고 로그 파일을 변경합니다.

- [JDBC 드라이버 버전 2.x 라이선스](#)
- [JDBC 드라이버 버전 2.x 변경 로그](#)

JDBC 드라이버 버전 1.2.27.1051 이상은 Amazon Redshift 저장 프로시저를 지원합니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Amazon Redshift에서 저장 프로시저 생성](#) 섹션을 참조하세요.

Amazon Redshift JDBC 드라이버, 버전 2.2 설치

Amazon Redshift JDBC 4.2 호환 드라이버 버전 2.x와 AWS SDK용 드라이버 종속 라이브러리를 설치하려면 ZIP 아카이브에서 원하는 디렉터리로 파일을 추출합니다.

Amazon Redshift JDBC 4.2 호환 드라이버 버전 2.x(AWS SDK 제외)를 설치하려면 원하는 디렉터리에 JAR 파일을 복사합니다.

Amazon Redshift JDBC 드라이버를 사용하여 Amazon Redshift 데이터 스토어에 액세스하려면 다음 설명에 따라 구성을 수행해야 합니다.

주제

- [JDBC 드라이버 라이브러리 참조](#)
- [드라이버 클래스 등록](#)

JDBC 드라이버 라이브러리 참조

데이터에 연결하는 데 사용하는 JDBC 애플리케이션이나 Java 코드는 드라이버 JAR 파일에 액세스해야 합니다. ZIP 아카이브에서 추출한 모든 JAR 파일을 애플리케이션 또는 코드에 지정합니다.

JDBC 애플리케이션에서 드라이버 사용

JDBC 애플리케이션은 일반적으로 드라이버 라이브러리 파일 목록 추가를 위한 구성 옵션 집합을 제공합니다. 제공된 옵션을 사용하여 ZIP 아카이브의 모든 JAR 파일을 애플리케이션의 드라이버 구성의 일부로 포함합니다. 자세한 내용은 JDBC 애플리케이션 설명서를 참조하세요.

Java 코드에서 드라이버 사용

클래스 경로에 모든 드라이버 라이브러리 파일을 포함해야 합니다. 클래스 경로는 Java Runtime Environment가 클래스와 기타 리소스 파일을 검색하는 경로입니다. 운영 체제의 클래스 경로를 설정하려면 Java SE 설명서에서 자세한 내용을 참조하세요.

- Windows: <https://docs.oracle.com/javase/7/docs/technotes/tools/windows/classpath.html>
- Linux 및 Solaris: <https://docs.oracle.com/javase/7/docs/technotes/tools/solaris/classpath.html>
- macOS: 기본 macOS 클래스 경로는 JDBC 드라이버가 설치된 디렉터리입니다.

드라이버 클래스 등록

애플리케이션에 적합한 클래스를 등록해야 합니다. 다음 클래스를 사용하여 Amazon Redshift JDBC 드라이버를 Amazon Redshift 데이터 스토어에 연결합니다.

- Driver 클래스는 `java.sql.Driver`를 확장합니다.
- DataSource 클래스는 `javax.sql.DataSource` 및 `javax.sql.ConnectionPoolDataSource`를 확장합니다.

드라이버는 JDBC 버전과 독립된 다음과 같은 정규화된 클래스 이름을 지원합니다.

- `com.amazon.redshift.jdbc.Driver`
- `com.amazon.redshift.jdbc.DataSource`

다음 예에서는 DriverManager 클래스를 사용하여 JDBC 4.2 연결을 설정하는 방법을 보여줍니다.

```
private static Connection connectViaDM() throws Exception
{
    Connection connection = null;
    connection = DriverManager.getConnection(CONNECTION_URL);
    return connection;
}
```

다음 예에서는 DataSource 클래스를 사용하여 연결을 설정하는 방법을 보여줍니다.

```
private static Connection connectViaDS() throws Exception
{
    Connection connection = null;
    11
    Amazon Redshift JDBC Driver Installation and Configuration Guide
    DataSource ds = new com.amazon.redshift.jdbc.DataSource
    ();
    ds.setURL(CONNECTION_URL);
    connection = ds.getConnection();
    return connection;
}
```

JDBC URL 가져오기

SQL 클라이언트 도구에서 Amazon Redshift 클러스터에 연결하려면 먼저 클러스터의 JDBC URL을 알아야 합니다. JDBC URL의 형식은 다음과 같습니다.

`jdbc:redshift://endpoint:port/database`

위에 나와 있는 형식의 필드는 다음과 같은 값을 가집니다.

Field	값
<code>jdbc</code>	연결 프로토콜입니다.
<code>redshift</code>	Amazon Redshift 드라이버를 사용하여 데이터베이스에 연결하도록 지정하는 하위 프로토콜입니다.
<code><i>endpoint</i></code>	Amazon Redshift 클러스터의 엔드포인트입니다.
<code><i>port</i></code>	클러스터를 시작할 때 지정한 포트 번호입니다. 방화벽이 있는 경우 이 포트를 사용할 수 있도록 열려 있는지 확인하세요.
<code><i>database</i></code>	클러스터에서 생성한 데이터베이스입니다.

다음은 JDBC URL의 예입니다. `jdbc:redshift://examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com:5439/dev`

URL 값에 다음 URI 예약 문자가 포함되어 있는 경우 해당 값은 URL 인코딩된 값이어야 합니다.

- ;
- +
- {
- }
- [
-]
- &
- =
- ?

• 빈 공간

예를 들어 PWD 값이 password:password인 경우 해당 값을 사용하는 연결 URL은 다음과 같습니다.

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/
dev;UID=amazon;PWD=password%3Apassword
```

JDBC 연결을 만드는 방법에 대한 자세한 내용은 [클러스터 연결 문자열 찾기](#) 섹션을 참조하세요.

클라이언트 컴퓨터가 데이터베이스에 연결하지 못하는 경우에는 가능한 문제를 해결할 수 있습니다. 자세한 내용은 [Amazon Redshift 연결 문제 해결](#) 섹션을 참조하세요.

연결 URL 구축

연결 URL을 사용하여 액세스 중인 데이터 스토어에 대한 연결 정보를 제공합니다. 다음은 Amazon Redshift JDBC 드라이버 버전 2.x의 연결 URL 형식입니다. 여기서 [Host]는 Amazon Redshift 서버의 엔드포인트이고 [Port]는 서버가 클라이언트 요청을 수신 대기하는 데 사용하는 TCP(Transmission Control Protocol) 포트의 번호입니다.

```
jdbc:redshift://[Host]:[Port]
```

다음은 몇 가지 옵션 설정을 지정하는 연결 URL의 형식입니다.

```
jdbc:redshift://[Host]:[Port]/[database];[Property1]=[Value];
[Property2]=[Value];
```

URL 값에 다음 URI 예약 문자가 포함되어 있는 경우 해당 값은 URL 인코딩된 값이어야 합니다.

- ;
- +
- {
- }
- [
-]
- &

- =
- ?
- 빈 공간

예를 들어 PWD 값이 password:password인 경우 해당 값을 사용하는 연결 URL은 다음과 같습니다.

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/
dev;UID=amazon;PWD=password%3Apassword
```

AWS에서 미국 서부(캘리포니아 북부) 리전에 있는 Amazon Redshift 클러스터의 포트 9000에 연결하려는 경우를 예로 들어 보겠습니다. 또한 dev라는 데이터베이스에 액세스하고 데이터베이스 사용자 이름과 암호를 사용하여 연결을 인증하려고 합니다. 이 경우 다음과 같은 연결 URL을 사용합니다.

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/
dev;UID=amazon;PWD=amazon
```

다음 문자를 사용하여 나머지 URL 문자열에서 구성 옵션을 구분할 수 있습니다.

- ;
- ?

예를 들어 다음과 같은 URL 문자열은 동일합니다.

```
jdbc:redshift://my_host:5439/dev;ssl=true;defaultRowFetchSize=100
```

```
jdbc:redshift://my_host:5439/dev?ssl=true;defaultRowFetchSize=100
```

다음 문자를 사용하여 URL 문자열에서 구성 옵션을 서로 구분할 수 있습니다.

- ;
- &

예를 들어 다음과 같은 URL 문자열은 동일합니다.

```
jdbc:redshift://my_host:5439/dev;ssl=true;defaultRowFetchSize=100
```

```
jdbc:redshift://my_host:5439/dev;ssl=true&defaultRowFetchSize=100
```

다음 URL 예에서는 로그 수준 6과 로그 경로를 지정합니다.

```
jdbc:redshift://redshift.amazonaws.com:5439/dev;DSILogLevel=6;LogPath=/home/user/logs;
```

연결 URL에서 속성을 복제하지 않습니다.

지정할 수 있는 구성 옵션의 전체 목록은 [JDBC 드라이버 버전 2.x 구성 옵션](#) 섹션을 참조하세요.

Note

연결할 때 클러스터 노드의 IP 주소 또는 VPC 엔드포인트의 IP 주소를 사용하지 마세요. 불필요한 중단을 방지하기 위해 항상 Redshift 엔드포인트를 사용하세요. 예외적으로 엔드포인트 URL을 사용하지 않는 유일한 상황은 사용자 지정 도메인 이름을 사용하는 경우입니다. 자세한 내용은 [클라이언트 연결에 사용자 지정 도메인 이름 사용](#)을 참조하세요.

JDBC와 Apache Maven의 연결 구성

Apache Maven은 소프트웨어 프로젝트 관리 및 이해 도구입니다. AWS SDK for Java는 Apache Maven 프로젝트를 지원합니다. 자세한 내용은 AWS SDK for Java Developer Guide의 [Using the SDK with Apache Maven](#) 섹션을 참조하세요.

Apache Maven을 사용하는 경우 Amazon Redshift JDBC 드라이버를 사용하여 Amazon Redshift 클러스터에 연결하는 프로젝트를 구성하고 구축할 수 있습니다. 이를 위해 JDBC 드라이버를 프로젝트의 pom.xml 파일에 종속성으로 추가하세요. Maven을 사용해 프로젝트를 빌드하고 JDBC 연결을 사용하려면 다음 섹션에서 설명하는 단계를 따르세요.

JDBC 드라이버를 Maven 종속성으로 구성하려면

1. pom.xml 파일의 리포지토리 섹션에 Amazon 리포지토리 또는 Maven Central 리포지토리를 추가합니다.

Note

다음 코드 예제의 URL은 브라우저에서 사용하면 오류를 반환합니다. 이 URL은 Maven 프로젝트의 컨텍스트에만 사용하세요.

보안 소켓 계층(SSL)을 사용하여 연결하려면 다음 리포지토리를 pom.xml 파일에 추가합니다.

```
<repositories>
  <repository>
    <id>redshift</id>
    <url>https://s3.amazonaws.com/redshift-maven-repository/release</url>
  </repository>
</repositories>
```

Maven Central 리포지토리의 경우 pom.xml 파일에 다음을 추가합니다.

```
<repositories>
  <repository>
    <id>redshift</id>
    <url>https://repo1.maven.org/maven2</url>
  </repository>
</repositories>
```

2. pom.xml 파일의 종속성 섹션에서 사용할 드라이버 버전을 선언합니다.

Amazon Redshift는 JDBC 4.2 API와 호환되는 도구용 드라이버를 제공합니다. 이러한 드라이버가 지원하는 기능에 대한 자세한 내용은 [Amazon Redshift JDBC 드라이버, 버전 2.1 다운로드](#) 섹션을 참조하세요.

다음 예에서 *driver-version*을 사용 중인 드라이버 버전(예: 2.1.0.1)으로 바꿉니다. JDBC 4.2 호환 드라이버의 경우 다음을 사용합니다.

```
<dependency>
  <groupId>com.amazon.redshift</groupId>
  <artifactId>redshift-jdbc42</artifactId>
  <version>driver-version</version>
</dependency>
```

이 드라이버의 클래스 이름은 com.amazon.redshift.Driver입니다.

IAM 데이터베이스 인증을 사용할 때는 Amazon Redshift Maven 드라이버에 다음과 같은 선택적 종속성이 필요합니다.

```
<dependency>
```

```

    <groupId>com.amazonaws</groupId>
    <artifactId>aws-java-sdk-core</artifactId>
    <version>1.12.23</version>
    <scope>runtime</scope>
    <optional>true</optional>
</dependency>
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>aws-java-sdk-redshift</artifactId>
    <version>1.12.23</version>
    <scope>runtime</scope>
    <optional>true</optional>
</dependency>
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>aws-java-sdk-sts</artifactId>
    <version>1.12.23</version>
    <scope>runtime</scope>
    <optional>true</optional>
</dependency>

```

Amazon Redshift JDBC 드라이버를 최신 버전으로 업그레이드하거나 변경하려면 먼저 종속 항목의 버전 섹션을 최신 버전의 드라이버로 수정합니다. 그런 다음 다음과 같이 Maven Clean Plugin으로 프로젝트를 정리하세요.

```
mvn clean
```

인증 및 SSL 구성

무단 액세스로부터 데이터를 보호하기 위해 Amazon Redshift 데이터 스토어는 사용자 자격 증명을 사용하여 모든 연결을 인증할 것을 요구합니다. 또한 일부 데이터 스토어는 단방향 인증 여부에 관계없이 보안 소켓 계층(SSL) 프로토콜을 통한 연결을 요구합니다.

Amazon Redshift JDBC 드라이버 버전 2.x는 이러한 인증 프로토콜을 완벽하게 지원합니다.

드라이버에서 지원하는 SSL 버전은 사용 중인 JVM 버전에 따라 다릅니다. 각 Java 버전에서 지원하는 SSL 버전에 대한 자세한 내용은 Java Platform Group Product Management Blog의 [Diagnosing TLS, SSL, and HTTPS](#)를 참조하세요.

연결에 사용되는 SSL 버전은 드라이버와 서버 모두에서 지원하는 가장 높은 버전으로 연결 시 결정됩니다.

연결하려는 Amazon Redshift 서버의 보안 요구 사항에 따라 연결을 인증하도록 Redshift JDBC 드라이버 버전 2.x를 구성합니다.

연결을 인증하려면 항상 Redshift 사용자 이름과 암호를 입력해야 합니다. 서버에서 SSL 활성화 및 필요 여부에 따라 SSL을 통해 연결하도록 드라이버를 구성해야 할 수도 있습니다. 또는 단방향 SSL 인증을 사용하여 클라이언트(드라이버 자체)가 서버의 ID를 확인하도록 할 수 있습니다.

연결 URL에서 드라이버에 구성 정보를 제공합니다. 연결 URL의 구문에 대한 자세한 내용은 [연결 URL 구축](#) 섹션을 참조하세요.

SSL은 전송 계층 보안과 보안 소켓 계층(TLS/SSL)을 나타냅니다. 이 드라이버는 TLS/SSL의 업계 표준 버전을 지원합니다.

IAM 인증 구성

IAM 인증을 사용하여 Amazon Redshift 서버에 연결하는 경우 다음 속성을 데이터 원본 연결 문자열의 일부로 설정합니다.

IAM 인증에 대한 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#) 섹션을 참조하세요.

IAM 인증을 사용하려면 다음 연결 문자열 형식 중 하나를 사용합니다.

연결 문자열	설명
<code>jdbc:redshift:iam:// [host]:[port]/[db]</code>	일반 연결 문자열입니다. 드라이버는 호스트에서 ClusterID와 리전을 유추합니다.
<code>jdbc:redshift:iam:// [cluster-id]: [region]/[db]</code>	드라이버는 주어진 ClusterID 및 리전에 따라 호스트 정보를 검색합니다.
<code>jdbc:redshift:iam:// [host]/[db]</code>	드라이버는 기본적으로 포트 5439이며 호스트에서 ClusterID와 리전을 유추합니다. 클러스터를 생성, 수정 또는 마이그레이션할 때 선택한 포트에 따라 선택한 포트에 대한 액세스를 허용합니다.

프로파일 지정

IAM 인증을 사용하는 경우 프로파일 이름 아래에 필수 또는 선택적 연결 속성을 추가로 지정할 수 있습니다. 이를 통해 연결 문자열에 특정 정보를 직접 넣는 것을 피할 수 있습니다. Profile 속성을 사용하여 연결 문자열에 프로파일 이름을 지정합니다.

AWS 자격 증명 파일에 프로파일을 추가할 수 있습니다. 이 파일의 기본 위치는 ~/.aws/credentials입니다.

환경 변수 AWS_CREDENTIAL_PROFILES_FILE에 경로를 설정하여 기본값을 변경할 수 있습니다.

프로파일에 대한 자세한 내용은 AWS SDK for Java의 [AWS 자격 증명 작업](#) 섹션을 참조하세요.

인스턴스 프로파일 자격 증명 사용

IAM 역할과 연결된 Amazon EC2 인스턴스에서 애플리케이션을 실행하는 경우 인스턴스 프로파일 자격 증명을 사용하여 연결할 수 있습니다.

이렇게 하려면 앞의 테이블에 나와 있는 IAM 연결 문자열 형식 중 하나를 사용하고 dbuser 연결에 사용하는 Amazon Redshift 사용자 이름으로 dbuser 연결 속성을 설정합니다.

인스턴스 프로파일에 대한 자세한 내용은 IAM User Guide의 [Access Management](#) 섹션을 참조하세요.

자격 증명 공급자 사용

드라이버에서 다음 서비스의 자격 증명 공급자 플러그인도 지원합니다.

- AWS IAM Identity Center
- Active Directory Federation Service(ADFS)
- JSON 웹 토큰(JWT) 서비스
- Microsoft Azure Active Directory(AD) 서비스 및 브라우저
- Okta 서비스
- PingFederate 서비스
- Okta, Ping, ADFS 등의 SAML 서비스용 브라우저 SAML

이러한 서비스 중 하나를 사용하는 경우 연결 URL에서 다음 속성을 지정해야 합니다.

- Plugin_Name – 자격 증명 공급자 플러그인 클래스의 정규화된 클래스 경로입니다.

- IdP_Host: – Amazon Redshift에 인증하는 데 사용하는 서비스의 호스트입니다.
- IdP_Port – 인증 서비스의 호스트가 수신 대기하는 포트입니다. Okta에는 필요하지 않습니다.
- User – idp_host 서버의 사용자 이름입니다.
- Password – idp_host 사용자 이름과 연결된 암호입니다.
- DbUser – 연결에 사용하는 Amazon Redshift 사용자 이름입니다.
- SSL_Insecure – IDP 서버 인증서를 확인해야 하는지 여부를 나타냅니다.
- Client_ID – Azure AD 포털의 사용자 이름과 연결된 클라이언트 ID입니다. Azure AD에만 사용됩니다.
- Client_Secret – Azure AD 포털의 클라이언트 ID와 연결된 클라이언트 암호입니다. Azure AD에만 사용됩니다.
- IdP_Tenant – Amazon Redshift 애플리케이션의 Azure AD 테넌트 ID입니다. Azure AD에만 사용됩니다.
- App_ID – Amazon Redshift 애플리케이션의 Okta 앱 ID입니다. Okta에만 사용됩니다.
- App_Name – Amazon Redshift 애플리케이션의 Okta 앱 이름(옵션)입니다. Okta에만 사용됩니다.
- Partner_SPID – 파트너 SPID(서비스 공급자 ID) 값(옵션)입니다. PingFederate에만 사용됩니다.
- Idc_Region - AWS IAM Identity Center 인스턴스가 위치한 AWS 리전입니다. AWS IAM Identity Center에만 사용됩니다.
- Issuer_Url - AWS IAM Identity Center 서버의 인스턴스 엔드포인트입니다. AWS IAM Identity Center에만 사용됩니다.

이러한 서비스 중 하나에 브라우저 플러그인을 사용하는 경우 연결 URL에 다음이 포함될 수도 있습니다.

- Login_URL – 브라우저 플러그인을 통해 SAML(Security Assertion Markup Language) 또는 Azure AD 서비스를 사용할 때 자격 증명 공급자의 웹 사이트에 있는 리소스의 URL입니다. 브라우저 플러그인을 사용하는 경우 이 파라미터는 필수입니다.
- Listen_Port - 브라우저 플러그인을 통해 SAML, Azure AD 또는 AWS IAM ID 센터 서비스를 사용할 때 드라이버가 ID 제공업체로부터 SAML 응답을 가져오는 데 사용하는 포트입니다.
- IdP_Response_Timeout - 브라우저 플러그인을 통해 SAML, Azure AD 또는 AWS IAM ID 센터 서비스를 사용할 때 드라이버가 ID 제공업체의 SAML 응답을 기다리는 시간(초)입니다.

추가 연결 문자열 속성에 대한 자세한 내용은 [JDBC 드라이버 버전 2.x 구성 옵션](#) 섹션을 참조하세요.

사용자 이름 및 암호만 사용

연결하려는 서버가 SSL을 사용하지 않는 경우 연결을 인증하는 데 Redshift 사용자 이름과 암호만 입력하면 됩니다.

Redshift 사용자 이름 및 암호만 사용하여 인증을 구성하려면 다음과 같이 합니다.

1. UID 속성을 Amazon Redshift 서버 액세스를 위한 Redshift 사용자 이름으로 설정합니다.
2. PWD 속성을 Redshift 사용자 이름에 해당하는 암호로 설정합니다.

자격 증명 확인 없이 SSL 사용

연결 중인 서버에서 SSL을 사용하지만 자격 증명 확인이 필요하지 않은 경우 비검증 SSL 팩토리를 사용하도록 드라이버를 구성할 수 있습니다.

자격 증명 확인 없이 SSL 연결을 구성하려면

1. UID 속성을 Amazon Redshift 서버 액세스를 위한 Redshift 사용자 이름으로 설정합니다.
2. PWD 속성을 Redshift 사용자 이름에 해당하는 암호로 설정합니다.
3. SSLFactory 속성 `com.amazon.redshift.ssl.NonValidatingFactory`로 설정합니다.

단방향 SSL 인증 사용

연결하려는 서버가 SSL을 사용하고 인증서가 있는 경우 단방향 인증을 사용하여 서버의 자격 증명을 확인하도록 드라이버를 구성할 수 있습니다.

단방향 인증에는 서버의 자격 증명 확인을 위한 서명된 신뢰할 수 있는 SSL 인증서가 필요합니다. 특정 인증서를 사용하거나 적절한 인증서가 들어 있는 TrustStore에 액세스하도록 드라이버를 구성할 수 있습니다. 인증서 또는 TrustStore를 지정하지 않으면 드라이버가 기본 Java TrustStore(일반적으로 `jssecacerts` 또는 `cacerts`)를 사용합니다.

단방향 SSL 인증을 구성하려면

1. UID 속성을 Amazon Redshift 서버 액세스를 위한 Redshift 사용자 이름으로 설정합니다.
2. PWD 속성을 Redshift 사용자 이름에 해당하는 암호로 설정합니다.
3. SSL 속성을 `true`로 설정합니다.
4. SSLRootCert 속성을 루트 CA 인증서의 위치로 설정합니다.

5. 기본 Java TrustStore 중 하나를 사용하지 않는 경우 다음 중 하나를 수행합니다.
 - 서버 인증서를 지정하려면 SSLRootCert 속성을 인증서의 전체 경로로 설정합니다.
 - TrustStore를 지정하려면 다음을 수행합니다.
 - a. keytool 프로그램으로 사용하려는 TrustStore에 서버 인증서를 추가합니다.
 - b. 드라이버를 사용하여 Java 애플리케이션을 시작할 때 사용할 TrustStore와 암호를 지정합니다. 예:

```
-Djavax.net.ssl.trustStore=[TrustStoreName]
-Djavax.net.ssl.trustStorePassword=[TrustStorePassword]
-Djavax.net.ssl.trustStoreType=[TrustStoreType]
```

6. 그중 하나의 경로를 선택하세요.
 - 인증서를 검증하려면 SSLMode 속성을 verify-ca로 설정합니다.
 - 인증서를 검증하고 인증서의 호스트 이름을 확인하려면 SSLMode 속성을 verify-full로 설정합니다.

로깅 구성

문제 진단에 도움이 되도록 드라이버에서 로그를 설정할 수 있습니다.

다음 방법을 사용하여 드라이버 정보를 로그할 수 있습니다.

- 로그된 정보를 .log 파일에 저장하려면 [로그 파일 사용](#) 섹션을 참조하세요.
- 로그된 정보를 DriverManager에 지정된 LogStream 또는 LogWriter로 보내려면 [LogStream 또는 LogWriter 사용](#) 섹션을 참조하세요.

연결 URL에서 드라이버에 구성 정보를 제공합니다. 연결 URL의 구문에 대한 자세한 내용은 [연결 URL 구축](#) 섹션을 참조하세요.

로그 파일 사용

문제를 포착할 수 있을 만큼만 로깅을 길게 설정합니다. 로깅은 성능을 떨어뜨릴 뿐만 아니라 대용량의 디스크 공간을 소비할 수 있기 때문입니다.

연결 URL에서 LogLevel 키를 설정하여 로깅을 설정하고 로그 파일에 포함되는 세부 정보의 양을 지정합니다. 다음 표에는 Amazon Redshift JDBC 드라이버 버전 2.x에서 제공하는 로깅 수준이 나열되어 있습니다. 아래로 갈수록 더 자세한 세부 정보가 로그됩니다.

LogLevel 값	설명
1	드라이버 중단으로 이어질 심각한 오류 이벤트를 로그합니다.
2	드라이버가 계속 실행될 수 있는 오류 이벤트를 로그합니다.
3	작업을 수행하지 않으면 발생할 수 있는 이벤트를 로그합니다. 이 수준의 로깅과 이 수준 이상의 로깅 수준은 사용자의 쿼리도 기록합니다.
4	드라이버의 진행 상황을 설명하는 일반 정보를 로그합니다.
5	드라이버 디버깅에 유용한 세부 정보를 로그합니다.
6	모든 드라이버 활동을 로그합니다.

로그 파일을 사용하는 로깅을 설정하려면

1. LogLevel 속성을 로그 파일에 포함하려는 정보 수준으로 설정합니다.
2. LogPath 속성을 로그 파일을 저장할 폴더의 전체 경로로 설정합니다.

예를 들어 다음 연결 URL은 로깅 수준 3을 사용하고 로그 파일을 C:\temp 폴더에 저장합니다. jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/Default;DSILogLevel=3;LogPath=C:\temp

3. 새 설정이 적용되도록 하려면 JDBC 애플리케이션을 다시 시작하고 서버에 다시 연결합니다.

Amazon Redshift JDBC 드라이버가 LogPath 속성에 지정된 위치에 다음 로그 파일을 생성합니다.

- 연결과 관련되지 않은 드라이버 활동을 로그하는 redshift_jdbc.log 파일입니다.
- 각 데이터베이스 연결에 대한 redshift_jdbc_connection_[Number].log 파일입니다. 여기서 [Number]는 각 로그 파일을 식별하는 번호입니다. 이 파일은 연결과 관련된 드라이버 활동을 로그합니다.

LogPath 값이 유효하지 않으면 드라이버가 로그된 정보를 표준 출력 스트림(System.out)으로 전송합니다.

LogStream 또는 LogWriter 사용

문제를 포착할 수 있을 만큼만 로깅을 길게 설정합니다. 로깅은 성능을 떨어뜨릴 뿐만 아니라 대용량의 디스크 공간을 소비할 수 있기 때문입니다.

연결 URL에 LogLevel 키를 설정하여 로깅을 설정하고 DriverManager에 지정된 LogStream 또는 LogWriter로 전송되는 세부 정보의 양을 지정합니다.

LogStream 또는 LogWriter를 사용하는 로깅을 설정하려면

1. 드라이버의 진행률을 설명하는 일반 정보를 로그하도록 드라이버를 구성하려면 LogLevel 속성을 1 또는 INFO로 설정합니다.
2. 새 설정이 적용되도록 하려면 JDBC 애플리케이션을 다시 시작하고 서버에 다시 연결합니다.

데이터 형식 변환

Amazon Redshift JDBC 드라이버 버전 2.x는 Amazon Redshift, SQL 및 Java 데이터 유형 간 변환으로 많은 일반 데이터 유형을 지원합니다.

다음 표에는 지원되는 데이터 형식 매핑이 나와 있습니다.

Amazon Redshift 형식	SQL 형식	Java 형식
BIGINT	SQL_BIGINT	Long
BOOLEAN	SQL_BIT	Boolean
CHAR	SQL_CHAR	문자열
DATE	SQL_TYPE_DATE	java.sql.Date
DECIMAL	SQL_NUMERIC	BigDecimal
DOUBLE PRECISION	SQL_DOUBLE	Double
GEOMETRY	SQL_LONGVARBINARY	byte[]
INTEGER	SQL_INTEGER	Integer

Amazon Redshift 형식	SQL 형식	Java 형식
OID	SQL_BIGINT	Long
SUPER	SQL_LONGVARCHAR	문자열
REAL	SQL_REAL	Float
SMALLINT	SQL_SMALLINT	Short
TEXT	SQL_VARCHAR	문자열
TIME	SQL_TYPE_TIME	java.sql.Time
TIMETZ	SQL_TYPE_TIME	java.sql.Time
TIMESTAMP	SQL_TYPE_TIMESTAMP	java.sql.Timestamp
TIMESTAMPTZ	SQL_TYPE_TIMESTAMP	java.sql.Timestamp
VARCHAR	SQL_VARCHAR	문자열

준비된 명령문 지원 사용

Amazon Redshift JDBC 드라이버는 준비된 문을 지원합니다. 준비된 문을 사용하여 동일한 연결 중 여러 번 실행해야 하는 파라미터화된 쿼리의 성능을 향상시킬 수 있습니다.

준비된 문은 서버 측에서 컴파일되지만 즉시 실행되지 않는 SQL 문입니다. 컴파일된 문은 객체 또는 연결을 닫을 때까지 PreparedStatement 객체로 서버에 저장됩니다. 해당 객체가 존재하는 동안 문을 다시 컴파일하지 않고도 여러 파라미터 값을 사용하여 준비된 문을 필요한 만큼 실행할 수 있습니다. 이로 인해 오버헤드가 줄어 쿼리 집합을 더 빠르게 실행할 수 있습니다.

준비된 문에 대한 자세한 내용은 [Oracle에서 제공하는 JDBC Basics 튜토리얼](#)의 "Using Prepared Statements" 섹션을 참조하세요.

쿼리가 여러 개 포함된 문을 준비할 수 있습니다. 예를 들어 다음의 준비된 문에는 2개의 INSERT 쿼리가 포함되어 있습니다.

```
PreparedStatement pstmt = conn.prepareStatement("INSERT INTO
MyTable VALUES (1, 'abc'); INSERT INTO CompanyTable VALUES
(1, 'abc');");
```

이러한 쿼리가 동일한 준비된 문 내에 지정된 다른 쿼리의 결과에 의존하지 않도록 주의합니다. 쿼리는 준비 단계에서 실행되지 않기 때문에 결과가 아직 반환되지 않았으며 동일한 준비된 문의 다른 쿼리에서 사용할 수 없습니다.

예를 들어 테이블을 생성한 다음 새로 생성된 테이블에 값을 삽입하는 다음의 준비된 문은 허용되지 않습니다.

```
PreparedStatement pstmt = conn.prepareStatement("CREATE
TABLE MyTable(col1 int, col2 varchar); INSERT INTO myTable
VALUES (1, 'abc');");
```

이 문을 준비하려고 하면 서버에서 대상 테이블(myTable)이 아직 존재하지 않는다는 오류를 반환합니다. CREATE 쿼리를 실행해야 INSERT 쿼리를 준비할 수 있습니다.

JDBC 드라이버 버전 2.2과 1.x의 차이점

이 섹션에서는 JDBC 드라이버 버전 2.2와 1.x에서 반환되는 정보의 차이점에 대해 설명합니다. JDBC 드라이버 버전 1.x는 공급이 중단되었습니다.

다음 표에는 JDBC 드라이버의 각 버전에 대해 `getDatabaseProductName()` 및 `getDatabaseProductVersion()` 함수에서 반환된 `DatabaseMetadata` 정보가 나열되어 있습니다. JDBC 드라이버 버전 2.2은 연결을 설정하는 동안 값을 가져옵니다. JDBC 드라이버 버전 1.x는 쿼리의 결과로 값을 가져옵니다.

JDBC 드라이버 버전	<code>getDatabaseProductName()</code> 결과	<code>getDatabaseProductVersion()</code> 결과
2.2	Redshift	8.0.2
1.x	PostgreSQL	08.00.0002

다음 표에는 JDBC 드라이버의 각 버전에 대해 `getTypeInfo` 함수에서 반환된 `DatabaseMetadata` 정보가 나열되어 있습니다.

JDBC 드라이버 버전	<code>getTypeInfo</code> 결과
2.2	Redshift 데이터 형식과 일치

JDBC 드라이버 버전	getTypeInfo 결과
1.x	PostgreSQL 데이터 형식과 일치

JDBC 드라이버 버전 2.x용 초기화(.ini) 파일 생성

Amazon Redshift JDBC 드라이버 버전 2.x용 초기화(.ini) 파일을 사용하여 시스템 수준 구성 파라미터를 지정할 수 있습니다. 예를 들어 애플리케이션마다 페더레이션 IdP 인증 파라미터가 다를 수 있습니다. .ini 파일은 SQL 클라이언트가 필요한 구성 파라미터를 가져올 수 있는 공통 위치를 제공합니다.

SQL 클라이언트에 대한 구성 옵션이 들어 있는 JDBC 드라이버 버전 2.x 초기화(.ini) 파일을 생성할 수 있습니다. 파일의 기본 이름은 `rsjdbc.ini`입니다. JDBC 드라이버 버전 2.x는 다음 위치(우선 순위에 따라 나열됨)에서 .ini 파일을 확인합니다.

- IniFile 연결 URL 또는 SQL 클라이언트의 연결 속성 대화 상자에 있는 파라미터. IniFile 파라미터에 파일 이름을 포함한 .ini 파일의 전체 경로가 포함되어 있는지 확인합니다. IniFile 파라미터에 대한 자세한 내용은 [INIFILE](#) 섹션을 참조하세요. IniFile 파라미터가 .ini 파일의 위치를 잘못 지정하면 오류가 표시됩니다.
- 파일 이름을 포함한 전체 경로가 포함된 `AMAZON_REDSHIFT_JDBC_INI_FILE`과 같은 환경 변수. `rsjdbc.ini`를 사용하거나 파일 이름을 지정합니다. `AMAZON_REDSHIFT_JDBC_INI_FILE` 환경 변수가 .ini 파일의 위치를 잘못 지정하면 오류가 표시됩니다.
- 드라이버 JAR 파일이 있는 디렉터리.
- 사용자 홈 디렉터리.
- 시스템의 임시 디렉터리.

.ini 파일을 [DRIVER]와 같은 섹션으로 구성할 수 있습니다. 각 섹션에는 다양한 연결 파라미터를 지정하는 키-값 페어가 있습니다. IniSection 파라미터를 사용하여 .ini 파일에 섹션을 지정할 수 있습니다. IniSection 파라미터에 대한 자세한 내용은 [IniSection](#) 섹션을 참조하세요.

다음은 [DRIVER], [DEV], [QA] 및 [PROD]에 대한 섹션이 있는 .ini 파일 형식의 예입니다. [DRIVER] 섹션은 모든 연결에 적용할 수 있습니다.

```
[DRIVER]
key1=val1
key2=val2
```

```
[DEV]
key1=val1
key2=val2

[QA]
key1=val1
key2=val2

[PROD]
key1=val1
key2=val2
```

JDBC 드라이버 버전 2.x는 우선 순위에 따라 나열된 다음 위치에서 구성 파라미터를 로드합니다.

- 애플리케이션 코드의 기본 구성 파라미터.
- .ini 파일의 [DRIVER] 섹션 속성(포함된 경우).
- 연결 URL 또는 SQL 클라이언트의 연결 속성 대화 상자에 IniSection 옵션이 제공되는 경우 사용자 정의 섹션 구성 파라미터.
- getConnection 호출에 지정된 연결 속성 객체의 속성.
- 연결 URL에 지정된 구성 파라미터.

JDBC 드라이버 버전 2.x 구성 옵션

아래에서 Amazon Redshift JDBC 드라이버 버전 2.2에 지정할 수 있는 옵션에 대한 설명을 찾을 수 있습니다. 구성 옵션은 대소문자를 구분하지 않습니다.

연결 URL을 사용하여 구성 속성을 설정할 수 있습니다. 자세한 내용은 [연결 URL 구축](#) 섹션을 참조하세요.

주제

- [AccessKeyID](#)
- [AllowDBUserOverride](#)
- [App_ID](#)
- [App_Name](#)
- [ApplicationName](#)
- [AuthProfile](#)

- [AutoCreate](#)
- [Client_ID](#)
- [Client_Secret](#)
- [ClusterID](#)
- [압축](#)
- [connectTimeout](#)
- [connectionTimezone](#)
- [databaseMetadataCurrentDbOnly](#)
- [DbUser](#)
- [DbGroups](#)
- [DBNAME](#)
- [defaultRowFetchSize](#)
- [DisableIsValidQuery](#)
- [enableFetchRingBuffer](#)
- [enableMultiSqlSupport](#)
- [fetchRingBufferSize](#)
- [ForceLowercase](#)
- [groupFederation](#)
- [HOST](#)
- [IAMDisableCache](#)
- [IAMDuration](#)
- [Idc_Client_Display_Name](#)
- [Idc_Region](#)
- [IdP_Host](#)
- [IdP_Partition](#)
- [IdP_Port](#)
- [IdP_Tenant](#)
- [IdP_Response_Timeout](#)

- [INIFILE](#)
- [IniSection](#)
- [isServerless](#)
- [Issuer_Url](#)
- [Listen_Port](#)
- [Login_URL](#)
- [loginTimeout](#)
- [loginToRp](#)
- [LogLevel](#)
- [LogPath](#)
- [OverrideSchemaPatternType](#)
- [Partner_SPID](#)
- [암호](#)
- [Plugin_Name](#)
- [PORT](#)
- [Preferred_Role](#)
- [프로필](#)
- [PWD](#)
- [queryGroup](#)
- [readOnly](#)
- [리전](#)
- [reWriteBatchedInserts](#)
- [reWriteBatchedInsertsSize](#)
- [roleArn](#)
- [roleSessionName](#)
- [범위](#)
- [SecretAccessKey](#)
- [SessionToken](#)

- [serverlessAcctId](#)
- [serverlessWorkGroup](#)
- [socketFactory](#)
- [socketTimeout](#)
- [SSL](#)
- [SSL_Insecure](#)
- [SSLCert](#)
- [SSLFactory](#)
- [SSLKey](#)
- [SSLMode](#)
- [SSLPassword](#)
- [SSLRootCert](#)
- [StsEndpointUrl](#)
- [tcpKeepAlive](#)
- [토큰](#)
- [token_type](#)
- [UID](#)
- [User](#)
- [webIdentityToken](#)

AccessKeyID

- 기본값 – 없음
- 데이터 형식 – 문자열

이 파라미터를 지정하여 사용자 또는 역할에 대한 IAM 액세스 키를 입력할 수 있습니다. 일반적으로 기존 문자열 또는 사용자 프로파일을 보고 키를 찾을 수 있습니다. 이 파라미터를 지정할 경우 SecretAccessKey 파라미터도 지정해야 합니다. JDBC URL에 전달된 경우 AccessKeyID는 URL 인코딩되어야 합니다.

이 파라미터는 선택 사항입니다.

AllowDBUserOverride

- 기본값 - 0
- 데이터 형식 - 문자열

이 옵션은 드라이버가 SAML 어설션의 DbUser 값을 사용하는지 아니면 연결 URL의 DbUser 연결 속성에 지정된 값을 사용하는지를 지정합니다.

이 파라미터는 선택 사항입니다.

1

드라이버가 SAML 어설션의 DbUser 값을 사용합니다.

SAML 어설션이 DBUser에 대한 값을 지정하지 않으면 드라이버가 DBUser 연결 속성에 지정된 값을 사용합니다. 연결 속성도 값을 지정하지 않으면 드라이버가 연결 프로파일에 지정된 값을 사용합니다.

0

드라이버가 DBUser 연결 속성에 지정된 DBUser 값을 사용합니다.

DBUser 연결 속성이 값을 지정하지 않으면 드라이버가 연결 프로파일에 지정된 값을 사용합니다. 연결 프로파일도 값을 지정하지 않으면 드라이버가 SAML 어설션의 값을 사용합니다.

App_ID

- 기본값 - 없음
- 데이터 형식 - 문자열

Amazon Redshift 애플리케이션과 연결된 Okta 제공 고유 ID입니다.

Okta 서비스를 통해 인증하는 경우 이 파라미터는 필수입니다.

App_Name

- 기본값 - 없음
- 데이터 형식 - 문자열

Amazon Redshift에 대한 연결을 인증하는 데 사용하는 Okta 애플리케이션의 이름입니다.

이 파라미터는 선택 사항입니다.

ApplicationName

- 기본값 – null
- 데이터 형식 – 문자열

감사 목적으로 Amazon Redshift에 전달할 애플리케이션의 이름입니다.

이 파라미터는 선택 사항입니다.

AuthProfile

- 기본값 – 없음
- 데이터 형식 – 문자열

Amazon Redshift에 연결하는 데 사용할 인증 프로파일의 이름입니다.

이 파라미터는 선택 사항입니다.

AutoCreate

- 기본값 – false
- 데이터 형식 – 부울

이 옵션은 지정된 사용자가 없을 때 드라이버가 새 사용자를 생성하도록 하는지 여부를 지정합니다.

이 파라미터는 선택 사항입니다.

true

DBUser 또는 고유 ID(UID)로 지정된 사용자가 없으면 해당 이름의 사용자가 새로 생성됩니다.

false

드라이버로 인해 새 사용자가 생성되지 않습니다. 지정된 사용자가 없으면 인증에 실패합니다.

Client_ID

- 기본값 – 없음

- 데이터 형식 – 문자열

Azure AD 서비스를 사용하여 연결을 인증할 때 사용할 클라이언트 ID입니다.

Azure AD 서비스를 통해 인증하는 경우 이 파라미터는 필수입니다.

Client_Secret

- 기본값 – 없음
- 데이터 형식 – 문자열

Azure AD 서비스를 사용하여 연결을 인증할 때 사용할 클라이언트 암호입니다.

Azure AD 서비스를 통해 인증하는 경우 이 파라미터는 필수입니다.

ClusterID

- 기본값 – 없음
- 데이터 형식 – 문자열

연결하려는 Amazon Redshift 클러스터의 이름입니다. 드라이버는 지정된 호스트에서 이 파라미터를 감지하려고 시도합니다. Network Load Balancer(NLB)를 사용하며 IAM을 통해 연결하는 경우, 드라이버가 이를 감지하지 못하므로 이 연결 옵션을 사용하여 설정할 수 있습니다.

이 파라미터는 선택 사항입니다.

압축

- 기본값 - 끄기
- 데이터 형식 – 문자열

Amazon Redshift 서버와 클라이언트 또는 드라이버 간의 유선 프로토콜 통신에 사용되는 압축 방법입니다.

이 파라미터는 선택 사항입니다.

다음 값을 지정할 수 있습니다.

- lz4

Amazon Redshift와의 유선 프로토콜 통신에 사용되는 압축 방법을 lz4로 설정합니다.

- 꺼짐

Amazon Redshift와의 유선 프로토콜 통신에 압축을 사용하지 않습니다.

connectTimeout

- 기본값 - 10
- 데이터 형식 - 정수

소켓 연결 작업에 사용할 제한 시간 값입니다. Amazon Redshift 연결을 설정하는 데 필요한 시간이 이 값을 초과하면 연결을 사용할 수 없는 것으로 간주됩니다. 제한 시간은 초 단위로 지정됩니다. 값이 0이면 제한 시간을 지정하지 않는 것입니다.

이 파라미터는 선택 사항입니다.

connectionTimezone

- 기본값 - LOCAL
- 데이터 형식 - 문자열

세션 수준 시간대입니다.

이 파라미터는 선택 사항입니다.

다음 값을 지정할 수 있습니다.

LOCAL

세션 수준 시간대를 LOCAL JVM 시간대로 구성합니다.

SERVER

Amazon Redshift 서버에서 사용자에게 대해 설정된 시간대로 세션 수준 시간대를 구성합니다. 다음 명령으로 사용자에게 대한 세션 수준 시간대를 구성할 수 있습니다.

```
ALTER USER  
[...]
```

```
SET TIMEZONE TO [...];
```

databaseMetadataCurrentDbOnly

- 기본값 – true
- 데이터 형식 – 부울

이 옵션은 메타데이터 API가 액세스 가능한 모든 데이터베이스에서 데이터를 검색할지 아니면 연결된 데이터베이스에서만 데이터를 검색할지를 지정합니다.

이 파라미터는 선택 사항입니다.

다음 값을 지정할 수 있습니다.

true

애플리케이션이 단일 데이터베이스에서 메타데이터를 검색합니다.

false

애플리케이션이 액세스 가능한 모든 데이터베이스에서 메타데이터를 검색합니다.

DbUser

- 기본값 – 없음
- 데이터 형식 – 문자열

Amazon Redshift 계정에 사용할 사용자 ID입니다. AutoCreate 속성을 사용하도록 설정한 경우 현재 존재하지 않는 ID를 사용할 수 있습니다.

이 파라미터는 선택 사항입니다.

DbGroups

- 기본값 – PUBLIC
- 데이터 형식 – 문자열

DBUser가 현재 세션에 대해 조인하는 기존 데이터베이스 그룹 이름의 쉼표로 구분된 목록입니다.

이 파라미터는 선택 사항입니다.

DBNAME

- 기본값 – null
- 데이터 형식 – 문자열

연결할 데이터베이스의 이름입니다. 이 옵션을 사용하여 JDBC 연결 URL에 데이터베이스 이름을 지정할 수 있습니다.

이 파라미터는 필수 사항입니다. 연결 URL 또는 클라이언트 애플리케이션의 연결 속성에 데이터베이스 이름을 지정해야 합니다.

defaultRowFetchSize

- 기본값 - 0
- 데이터 형식 – 정수

이 옵션은 getFetchSize의 기본값을 지정합니다.

이 파라미터는 선택 사항입니다.

다음 값을 지정할 수 있습니다.

0

단일 작업으로 모든 행을 가져옵니다.

양의 정수

ResultSet의 각 가져오기 반복에 대해 데이터베이스에서 가져올 행 수입니다.

DisableIsValidQuery

- 기본값 – False
- 데이터 형식 – 부울

이 옵션은 Connection.isValid () 메서드를 사용하여 데이터베이스 연결이 활성 상태인지 여부를 확인할 때 드라이버가 새 데이터베이스 쿼리를 제출할지 여부를 지정합니다.

이 파라미터는 선택 사항입니다.

true

드라이버가 `Connection.isValid()`를 사용하여 데이터베이스 연결이 활성 상태인지 여부를 확인할 때 쿼리를 제출하지 않습니다. 이로 인해 데이터베이스 서버가 예기치 않게 종료된 경우 드라이버가 데이터베이스 연결을 활성 상태로 잘못 식별할 수 있습니다.

false

드라이버가 `Connection.isValid()`를 사용하여 데이터베이스 연결이 활성 상태인지 여부를 확인할 때 쿼리를 제출합니다.

enableFetchRingBuffer

- 기본값 – true
- 데이터 형식 – 부울

이 옵션은 드라이버가 별도의 스레드에서 링 버퍼를 사용하여 행을 가져오도록 지정합니다. `fetchRingBufferSize` 파라미터는 링 버퍼 크기를 지정합니다.

링 버퍼는 JDBC에서 자동 메모리 관리를 구현하여 데이터 검색 작업 중에 메모리 부족(OOM) 오류를 방지합니다. 링 버퍼는 버퍼링된 데이터의 실제 크기를 실시간으로 모니터링하여 드라이버의 총 메모리 사용량이 정의된 한도 내에 있도록 합니다. 버퍼 용량에 도달하면 드라이버가 데이터 가져오기 작업을 일시 중지하여 수동 개입 없이 메모리 오버플로를 방지합니다. 이 내장 보호 기능은 사용자의 구성 없이도 OOM 오류를 자동으로 제거합니다.

트랜잭션이 세미콜론으로 구분된 여러 SQL 명령이 포함된 명령문을 감지하면 해당 트랜잭션의 페치 링 버퍼가 거짓으로 설정됩니다. `enableFetchRingBuffer`의 값은 변경되지 않습니다.

이 파라미터는 선택 사항입니다.

Note

링 버퍼가 비활성화되고 가져오기 크기가 제대로 구성되지 않은 경우 메모리 부족(OOM) 문제가 발생할 수 있습니다. 가져오기 크기 구성에 대한 자세한 내용은 [여기](#)를 참조하세요.

enableMultiSqlSupport

- 기본값 – true
- 데이터 형식 – 부울

이 옵션은 Statement에서 세미콜론으로 구분된 여러 SQL 명령을 처리할지 여부를 지정합니다.

이 파라미터는 선택 사항입니다.

다음 값을 지정할 수 있습니다.

true

드라이버가 Statement 객체에서 세미콜론으로 구분된 여러 SQL 명령을 처리합니다.

false

드라이버가 단일 Statement에서 여러 SQL 명령에 대한 오류를 반환합니다.

fetchRingBufferSize

- 기본값 – 1G
- 데이터 형식 – 문자열

이 옵션은 결과 집합을 가져오는 동안 사용되는 링 버퍼의 크기를 지정합니다. 크기를 바이트 단위로 지정할 수 있습니다(예: 1KB의 경우 1K, 5,000바이트의 경우 5000, 1MB의 경우 1M, 1GB의 경우 1G). 또한 힙 메모리의 비율을 지정할 수 있습니다. 드라이버는 제한에 도달하면 행 가져오기를 중지합니다. 애플리케이션이 행을 읽고 링 버퍼의 공간을 확보하면 가져오기가 다시 시작됩니다.

이 파라미터는 선택 사항입니다.

ForceLowercase

- 기본값 – false
- 데이터 형식 – 부울

이 옵션은 드라이버가 Single Sign-On 인증을 사용할 때 자격 증명 공급자에서 Amazon Redshift로 전송된 모든 데이터베이스 그룹(DbGroups)을 소문자로 표시할지 여부를 지정합니다.

이 파라미터는 선택 사항입니다.

true

드라이버가 ID 공급자에서 전송된 모든 데이터베이스 그룹을 소문자로 표시합니다.

false

드라이버가 데이터베이스 그룹을 변경하지 않습니다.

groupFederation

- 기본값 – false
- 데이터 형식 – 부울

이 옵션은 Amazon Redshift IDP 그룹을 사용할지 여부를 지정합니다. 이 옵션은 GetClusterCredentialsV2 API에서 지원됩니다.

이 파라미터는 선택 사항입니다.

true

Amazon Redshift IDP(자격 증명 공급자) 그룹을 사용합니다.

false

사용자 페더레이션을 위해 STS API 및 GetClusterCredentials를 사용하고 연결을 위해 DbGroups를 명시적으로 지정합니다.

HOST

- 기본값 – null
- 데이터 형식 – 문자열

연결할 Amazon Redshift 서버의 호스트 이름입니다. 이 옵션을 사용하여 JDBC 연결 URL에 호스트 이름을 지정할 수 있습니다.

이 파라미터는 필수 사항입니다. 연결 URL 또는 클라이언트 애플리케이션의 연결 속성에 호스트 이름을 지정해야 합니다.

IAMDisableCache

- 기본값 – false

- 데이터 형식 – 부울

이 옵션은 IAM 자격 증명이 캐시되는지 여부를 지정합니다.

이 파라미터는 선택 사항입니다.

true

IAM 자격 증명이 캐시되지 않습니다.

false

IAM 자격 증명이 캐시되지 않습니다. 예를 들어 API 게이트웨이에 대한 요청이 제한될 때 성능이 향상됩니다.

IAMDURATION

- 기본값 – 900
- 데이터 형식 – 정수

임시 IAM 자격 증명이 만료될 때까지 남은 시간(초)입니다.

- 최솟값 – 900
- 최댓값 – 3,600

이 파라미터는 선택 사항입니다.

Idc_Client_Display_Name

- 기본값 - Amazon Redshift JDBC 드라이버
- 데이터 형식 – 문자열

BrowserIdcAuthPlugin을 사용하는 클라이언트에 사용할 표시 이름입니다.

이 파라미터는 선택 사항입니다.

Idc_Region

- 기본값 – 없음

- 데이터 형식 – 문자열

IAM Identity Center 인스턴스가 위치한 AWS 리전입니다.

이 파라미터는 `plugin_name` 구성 옵션에서 `BrowserIdcAuthPlugin`을 사용하여 인증하는 경우에만 필요합니다.

`IdP_Host`

- 기본값 – 없음
- 데이터 형식 – 문자열

Amazon Redshift에 인증하는 데 사용하는 자격 증명 공급자(IdP) 호스트입니다. 연결 문자열이나 프로파일에 지정할 수 있습니다.

이 파라미터는 선택 사항입니다.

`IdP_Partition`

- 기본값 – 없음
- 데이터 형식 – 문자열

ID 제공업체(IdP)가 구성된 클라우드 파티션을 지정합니다. 드라이버가 연결하는 IdP 인증 엔드포인트를 결정합니다.

이 파라미터를 비워 두면 드라이버는 기본적으로 상용 파티션으로 설정됩니다. 가능한 값은 다음과 같습니다.

- `us-gov`: IdP가 Azure Government에 구성된 경우 이 값을 사용합니다. 예를 들어 Azure AD Government는 `login.microsoftonline.us` 엔드포인트를 사용합니다.
- `cn`: IdP가 중국 클라우드 파티션에 구성된 경우 이 값을 사용합니다. 예를 들어 Azure AD China는 `login.chinacloudapi.cn` 엔드포인트를 사용합니다.

이 파라미터는 선택 사항입니다.

`IdP_Port`

- 기본값 – 없음

- 데이터 형식 – 문자열

자격 증명 공급자(IdP)가 사용하는 포트입니다. 연결 문자열이나 프로파일에 포트를 지정할 수 있습니다. 기본 포트는 5439입니다. 클러스터를 생성, 수정 또는 마이그레이션할 때 선택한 포트에 따라 선택한 포트에 대한 액세스를 허용합니다.

이 파라미터는 선택 사항입니다.

IdP_Tenant

- 기본값 – 없음
- 데이터 형식 – 문자열

Amazon Redshift 애플리케이션의 Azure AD 테넌트 ID입니다.

Azure AD 서비스를 통해 인증하는 경우 이 파라미터는 필수입니다.

IdP_Response_Timeout

- 기본값 – 120
- 데이터 형식 – 정수

브라우저 플러그인을 통해 SAML 또는 Azure AD 서비스를 사용할 때 드라이버가 자격 증명 공급자의 SAML 응답을 기다리는 시간(초)입니다.

이 파라미터는 선택 사항입니다.

INIFILE

- 기본값 – 없음
- 데이터 형식 – 문자열

파일 이름을 포함한 .ini 파일의 전체 경로입니다. 예:

```
IniFile="C:\tools\rsjdbc.ini"
```

.ini 파일에 대한 자세한 내용은 [JDBC 드라이버 버전 2.x용 초기화\(.ini\) 파일 생성](#) 섹션을 참조하세요.

이 파라미터는 선택 사항입니다.

IniSection

- 기본값 – 없음
- 데이터 형식 – 문자열

구성 옵션을 포함하는 .ini 파일의 섹션 이름입니다. .ini 파일에 대한 자세한 내용은 [JDBC 드라이버 버전 2.x용 초기화\(.ini\) 파일 생성](#) 섹션을 참조하세요.

다음 예에서는 .ini 파일의 [Prod] 섹션을 지정합니다.

```
IniSection="Prod"
```

이 파라미터는 선택 사항입니다.

isServerless

- 기본값 – false
- 데이터 형식 – 부울

이 옵션은 Amazon Redshift 엔드포인트 호스트가 서버리스 인스턴스인지 여부를 지정합니다. 드라이버는 지정된 호스트에서 이 파라미터를 감지하려고 시도합니다. Network Load Balancer(NLB)를 사용하는 경우 드라이버가 이를 감지하지 못하므로, 여기에서 설정하면 됩니다.

이 파라미터는 선택 사항입니다.

true

Amazon Redshift 엔드포인트 호스트가 서버리스 인스턴스입니다.

false

Amazon Redshift 엔드포인트 호스트가 프로비저닝된 클러스터입니다.

Issuer_Url

- 기본값 – 없음
- 데이터 형식 – 문자열

AWS IAM Identity Center 서버의 인스턴스 엔드포인트를 가리킵니다.

이 파라미터는 `plugin_name` 구성 옵션에서 `BrowserIdcAuthPlugin`을 사용하여 인증하는 경우에만 필요합니다.

Listen_Port

- 기본값 – 7890
- 데이터 형식 – 정수

드라이버가 브라우저 플러그인을 통해 SAML, Azure AD 또는 AWS IAM Identity Center 서비스를 사용할 때 ID 제공업체 또는 권한 부여 코드로부터 SAML 응답을 받는 데 사용하는 포트입니다.

이 파라미터는 선택 사항입니다.

Login_URL

- 기본값 – 없음
- 데이터 형식 – 문자열

브라우저 플러그인을 통해 SAML 또는 Azure AD 서비스를 사용할 때 자격 증명 공급자의 웹 사이트에 있는 리소스의 URL입니다.

브라우저 플러그인을 통해 SAML 또는 Azure AD 서비스로 인증하는 경우 이 파라미터는 필수입니다.

loginTimeout

- 기본값 - 0
- 데이터 형식 – 정수

서버에 연결하고 인증하면서 시간 제한이 지날 때까지 대기하는 시간(초)입니다. 연결 구성이 이 제한 시간보다 오래 걸리면 연결이 중단됩니다.

이 속성이 0으로 설정되는 경우에는 연결 시간 제한이 없습니다.

이 파라미터는 선택 사항입니다.

loginToRp

- 기본값 – `urn:amazon:webservicess`
- 데이터 형식 – 문자열

AD FS 인증 유형에 사용할 신뢰 당사자 트러스트입니다.

이 파라미터는 선택 사항입니다.

LogLevel

- 기본값 - 0
- 데이터 형식 - 정수

이 속성을 사용하여 드라이버에서 로깅을 설정하거나 해제하고 로그 파일에 포함되는 세부 정보의 양을 지정합니다.

문제를 포착할 수 있을 만큼만 로깅을 사용합니다. 로깅은 성능을 떨어뜨릴 뿐만 아니라 대용량의 디스크 공간을 소비할 수 있기 때문입니다.

이 파라미터는 선택 사항입니다.

파라미터를 다음 값 중 하나로 설정합니다.

0

모든 로깅을 사용 중지합니다.

1

드라이버를 중단시키는 매우 심각한 오류 이벤트를 로그하는 FATAL 수준에서 로깅을 사용합니다.

2

드라이버가 계속 실행될 수 있는 오류 이벤트를 로그하는 ERROR 수준에서 로깅을 사용합니다.

3

작업을 수행하지 않으면 오류가 발생할 수 있는 이벤트를 로그하는 WARNING 수준에서 로깅을 사용합니다.

4

드라이버의 진행 상황을 설명하는 일반 정보를 로그하는 INFO 수준에서 로깅을 사용합니다.

5

드라이버 디버깅에 유용한 세부 정보를 로그하는 DEBUG 수준에서 로깅을 사용합니다.

6

모든 드라이버 활동을 로그하는 TRACE 수준에서 로깅을 사용합니다.

로깅을 사용하면 드라이버가 LogPath 속성에 지정된 위치에 다음 로그 파일을 생성합니다.

- **redshift_jdbc.log** – 연결과 관련되지 않은 드라이버 활동을 로그하는 파일입니다.
- **redshift_jdbc_connection_[Number].log** – 각 데이터베이스 연결에 대한 파일입니다. 여기서 [Number]는 각 로그 파일을 다른 로그 파일과 구별하는 숫자입니다. 이 파일은 연결과 관련된 드라이버 활동을 로그합니다.

LogPath 값이 유효하지 않으면 드라이버가 로그된 정보를 표준 출력 스트림인 System.out으로 전송합니다.

LogPath

- 기본값 – 현재 작업 디렉터리.
- 데이터 형식 – 문자열

DSILogLevel 속성을 사용할 때 드라이버가 로그 파일을 저장하는 폴더의 전체 경로입니다.

연결 URL이 모든 JDBC 애플리케이션과 호환되는지 확인하려면 다른 백슬래시를 입력하여 파일 경로에서 백슬래시(\)를 이스케이프 처리하는 것이 좋습니다.

이 파라미터는 선택 사항입니다.

OverrideSchemaPatternType

- 기본값 – null
- 데이터 형식 – 정수

이 옵션은 getTables 호출에 사용하는 쿼리 유형을 재정의할지 여부를 지정합니다.

0

스키마 없는 범용 쿼리

1

로컬 스키마 쿼리

2

외부 스키마 쿼리

이 파라미터는 선택 사항입니다.

Partner_SPID

- 기본값 – 없음
- 데이터 형식 – 문자열

PingFederate 서비스를 사용하여 연결을 인증할 때 사용할 파트너 SPID(서비스 공급자 ID) 값입니다.

이 파라미터는 선택 사항입니다.

암호

- 기본값 – 없음
- 데이터 형식 – 문자열

IDP를 통해 IAM 인증을 사용하여 연결할 때 IDP_Host 서버의 암호입니다. 표준 인증을 사용할 때 Amazon Redshift 데이터베이스 암호에 PWD 대신 사용할 수 있습니다.

이 파라미터는 선택 사항입니다.

Plugin_Name

- 기본값 – 없음
- 데이터 형식 – 문자열

특정 자격 증명 공급자 플러그인을 구현하기 위한 정규화된 클래스 이름입니다.

이 파라미터는 선택 사항입니다.

다음과 같은 공급자 옵션이 지원됩니다.

- **AdfsCredentialsProvider** – Active Directory Federation Service입니다.
- **AzureCredentialsProvider** – Microsoft Azure Active Directory(AD) Service입니다.
- **BasicJwtCredentialsProvider** – JSON Web Tokens (JWT) Service입니다.
- **BasicSamlCredentialsProvider** - 많은 SAML 서비스 공급자와 함께 사용할 수 있는 SAML(Security Assertion Markup Language) 자격 증명입니다.

- **BrowserAzureCredentialsProvider** – Browser Microsoft Azure Active Directory(AD) Service입니다.
- **BrowserAzureOAuth2CredentialsProvider** – 네이티브 인증용 Browser Microsoft Azure Active Directory(AD) Service입니다.
- **BrowserIdcAuthPlugin** - AWS IAM Identity Center를 사용하는 인증 플러그인입니다.
- **BrowserSamlCredentialsProvider** – Okta, Ping, ADFS 등의 SAML 서비스용 Browser SAML입니다.
- **IdpTokenAuthPlugin** – AWS IAM Identity Center 토큰 또는 AWS IAM Identity Center에 연결된 웹 ID 제공업체의 OpenID Connect(OIDC) JSON 기반 자격 증명 토큰(JWT)을 수락하는 권한 부여 플러그인입니다.
- **OktaCredentialsProvider** – Okta Service입니다.
- **PingCredentialsProvider** – PingFederate Service입니다.

PORT

- 기본값 – null
- 데이터 형식 – 정수

연결할 Amazon Redshift 서버의 포트입니다. 이 옵션을 사용하여 JDBC 연결 URL에 포트를 지정할 수 있습니다.

이 파라미터는 선택 사항입니다.

Preferred_Role

- 기본값 – 없음
- 데이터 형식 – 문자열

Amazon Redshift에 연결하는 동안 수임하려는 IAM 역할입니다.

이 파라미터는 선택 사항입니다.

프로필

- 기본값 – 없음
- 데이터 형식 – 문자열

IAM 인증에 사용할 프로파일의 이름입니다. 이 프로파일에는 연결 문자열에 지정되지 않은 추가 연결 속성이 포함되어 있습니다.

이 파라미터는 선택 사항입니다.

PWD

- 기본값 – 없음
- 데이터 형식 – 문자열

속성 UID를 사용하여 제공한 Amazon Redshift 사용자 이름에 해당하는 암호입니다.

이 파라미터는 선택 사항입니다.

queryGroup

- 기본값 – null
- 데이터 형식 – 문자열

이 옵션은 적절한 쿼리 그룹에 쿼리를 할당하여 런타임에 쿼리를 대기열에 할당합니다. 쿼리 그룹이 세션에 대해 설정됩니다. 연결에서 실행되는 모든 쿼리는 이 쿼리 그룹에 속합니다.

이 파라미터는 선택 사항입니다.

readOnly

- 기본값 – false
- 데이터 형식 – 부울

이 속성은 드라이버가 읽기 전용 모드인지 여부를 지정합니다.

이 파라미터는 선택 사항입니다.

true

연결이 읽기 전용 모드이며 데이터 스토어에 쓸 수 없습니다.

false

연결이 읽기 전용 모드가 아니며 데이터 스토어에 쓸 수 있습니다.

리전

- 기본값 – null
- 데이터 형식 – 문자열

이 옵션은 클러스터가 위치한 AWS 리전을 지정합니다. StsEndPoint 옵션을 지정하면 리전 옵션이 무시됩니다. Redshift GetClusterCredentials API 작업도 리전 옵션을 사용합니다.

이 파라미터는 선택 사항입니다.

rewriteBatchedInserts

- 기본값 – false
- 데이터 형식 – 부울

이 옵션을 사용하면 최적화를 통해 호환 가능한 INSERT 문을 다시 작성하고 배치로 결합할 수 있습니다.

이 파라미터는 선택 사항입니다.

rewriteBatchedInsertsSize

- 기본값 – 128
- 데이터 형식 – 정수

이 옵션을 사용하면 최적화를 통해 호환 가능한 INSERT 문을 다시 작성하고 배치로 결합할 수 있습니다. 이 값은 2의 거듭제곱으로 기하급수적으로 증가해야 합니다.

이 파라미터는 선택 사항입니다.

roleArn

- 기본값 – 없음
- 데이터 형식 – 문자열

역할의 Amazon 리소스 이름(ARN)입니다. Plugin_Name 옵션에 BasicJwtCredentialsProvider를 지정할 때 이 파라미터를 지정해야 합니다. 다음 형식으로 ARN을 지정합니다.

arn:partition:service:region:account-id:resource-id

Plugin_Name 옵션에 BasicJwtCredentialsProvider를 지정하는 경우 이 파라미터는 필수입니다.

roleSessionName

- 기본값 – jwt_redshift_session
- 데이터 형식 – 문자열

말은 역할 세션의 식별자입니다. 일반적으로 애플리케이션 사용자와 연결된 이름 또는 식별자를 전달합니다. 애플리케이션에서 사용하는 임시 보안 자격 증명은 해당 사용자와 연결됩니다. Plugin_Name 옵션에 BasicJwtCredentialsProvider를 지정할 때 이 파라미터를 지정할 수 있습니다.

이 파라미터는 선택 사항입니다.

범위

- 기본값 – 없음
- 데이터 형식 – 문자열

사용자가 동의할 수 있는 공백으로 구분된 범위 목록입니다. Microsoft Azure 애플리케이션이 해당 사용자가 호출하려는 API에 대한 동의를 얻을 수 있도록 이 파라미터를 지정합니다. Plugin_Name 옵션에 BrowserAzureOAuth2CredentialsProvider를 지정할 때 이 파라미터를 지정할 수 있습니다.

이 파라미터는 BrowserAzureOAuth2CredentialsProvider 플러그인에 반드시 필요합니다.

SecretAccessKey

- 기본값 – 없음
- 데이터 형식 – 문자열

사용자 또는 역할에 대한 IAM 액세스 키입니다. 이 파라미터를 지정하는 경우 AccessKeyId도 지정해야 합니다. JDBC URL에 전달된 SecretAccessKey는 URL 인코딩되어야 합니다.

이 파라미터는 선택 사항입니다.

SessionToken

- 기본값 – 없음
- 데이터 형식 – 문자열

인증에 사용하는 IAM 역할과 연결된 임시 IAM 세션 토큰입니다. JDBC URL에 전달된 경우 임시 IAM 세션 토큰은 URL 인코딩되어야 합니다.

이 파라미터는 선택 사항입니다.

`serverlessAcctId`

- 기본값 – null
- 데이터 형식 – 문자열

Amazon Redshift Serverless 계정 ID입니다. 드라이버는 지정된 호스트에서 이 파라미터를 감지하려고 시도합니다. Network Load Balancer(NLB)를 사용하는 경우 드라이버가 이를 감지하지 못하므로, 여기에서 설정하면 됩니다.

이 파라미터는 선택 사항입니다.

`serverlessWorkGroup`

- 기본값 – null
- 데이터 형식 – 문자열

Amazon Redshift Serverless 작업 그룹 이름입니다. 드라이버는 지정된 호스트에서 이 파라미터를 감지하려고 시도합니다. Network Load Balancer(NLB)를 사용하는 경우 드라이버가 이를 감지하지 못하므로, 여기에서 설정하면 됩니다.

이 파라미터는 선택 사항입니다.

`socketFactory`

- 기본값 – null
- 데이터 형식 – 문자열

이 옵션은 소켓 생성을 위한 소켓 팩토리를 지정합니다.

이 파라미터는 선택 사항입니다.

`socketTimeout`

- 기본값 – 0

- 데이터 형식 – 정수

소켓 읽기 작업에서 제한 시간이 지날 때까지 대기하는 시간(초)입니다. 작업이 이 임계값보다 오래 걸리면 연결이 닫힙니다. 이 속성이 0으로 설정되는 경우에는 연결 시간 제한이 없습니다.

이 파라미터는 선택 사항입니다.

SSL

- 기본값 – TRUE
- 데이터 형식 – 문자열

이 속성을 사용하여 연결에 SSL을 설정하거나 해제합니다.

이 파라미터는 선택 사항입니다.

다음 값을 지정할 수 있습니다.

TRUE

드라이버가 SSL을 통해 서버에 연결합니다.

FALSE

드라이버가 SSL을 사용하지 않고 서버에 연결합니다. IAM 인증에서는 이 옵션이 지원되지 않습니다.

또는 AuthMech 속성을 구성할 수 있습니다.

SSL_Insecure

- 기본값 – true
- 데이터 형식 – 문자열

이 속성은 IDP 호스트 서버 인증서를 확인해야 하는지 여부를 나타냅니다.

이 파라미터는 선택 사항입니다.

다음 값을 지정할 수 있습니다.

true

드라이버가 IDP 서버 인증서의 신뢰성을 확인하지 않습니다.

false

드라이버가 IDP 서버 인증서의 신뢰성을 확인합니다.

SSLCert

- 기본값 – 없음
- 데이터 형식 – 문자열

SSL 사용 시 Amazon Redshift 서버 인스턴스 확인을 위한 신뢰할 수 있는 추가 CA 인증서가 포함된 .pem 또는 .crt 파일의 전체 경로입니다.

SSLKey가 지정된 경우 이 파라미터는 필수입니다.

SSLFactory

- 기본값 – 없음
- 데이터 형식 – 문자열

서버 인증서를 사용하지 않고 TLS/SSL을 통해 서버에 연결할 때 사용할 SSL 팩토리입니다.

SSLKey

- 기본값 – 없음
- 데이터 형식 – 문자열

SSLCert에 지정된 인증서 확인을 위한 PKCS8 키 파일이 포함된.der 파일의 전체 경로입니다.

SSLCert가 지정된 경우 이 파라미터는 필수입니다.

SSLMode

- 기본값 – verify-ca
- 데이터 형식 – 문자열

이 속성을 사용하여 TLS/SSL 사용 시 드라이버가 인증서를 검증하는 방법을 지정합니다.

이 파라미터는 선택 사항입니다.

다음 값을 지정할 수 있습니다.

[verify-ca]

드라이버가 신뢰할 수 있는 인증 기관(CA)에서 인증서를 제공하는지 확인합니다.

[verify-full]

드라이버가 신뢰할 수 있는 CA에서 인증서를 제공하고 인증서의 호스트 이름이 연결 URL에 지정된 호스트 이름과 일치하는지 확인합니다.

SSLPassword

- 기본값 - 0
- 데이터 형식 - 문자열

SSLKey에 지정된 암호화된 키 파일의 암호입니다.

SSLKey가 지정되고 키 파일이 암호화된 경우 이 파라미터는 필수입니다.

SSLRootCert

- 기본값 - 없음
- 데이터 형식 - 문자열

SSL 사용 시 Amazon Redshift 서버 인스턴스 확인을 위한 루트 CA 인증서가 포함된 .pem 또는 .cert 파일의 전체 경로입니다.

StsEndpointUrl

- 기본값 - null
- 데이터 형식 - 문자열

AWS Security Token Service(AWS STS) 엔드포인트를 지정할 수 있습니다. 이 옵션을 지정하면 리전 옵션이 무시됩니다. 이 엔드포인트에는 보안 프로토콜(HTTPS)만 지정할 수 있습니다.

tcpKeepAlive

- 기본값 – TRUE
- 데이터 형식 – 문자열

이 속성을 사용하여 TCP KeepAlive를 설정하거나 해제합니다.

이 파라미터는 선택 사항입니다.

다음 값을 지정할 수 있습니다.

TRUE

드라이버가 TCP KeepAlive를 사용하여 연결 시간 초과를 방지합니다.

FALSE

드라이버가 TCP KeepAlive를 사용하지 않습니다.

토큰

- 기본값 – 없음
- 데이터 형식 – 문자열

AWS IAM Identity Center에서 제공한 액세스 토큰 또는 AWS IAM Identity Center와 연결된 웹 ID 제공업체가 제공한 OpenID Connect(OIDC) JSON 웹 토큰(JWT)입니다. 애플리케이션은 AWS IAM Identity Center 또는 AWS IAM Identity Center와 연결된 자격 증명 공급자를 통해 애플리케이션 사용자를 인증하여 이 토큰을 생성해야 합니다.

이 파라미터는 IdpTokenAuthPlugin과 함께 작동합니다.

token_type

- 기본값 – 없음
- 데이터 형식 – 문자열

IdpTokenAuthPlugin에서 사용되는 토큰 유형입니다.

다음 값을 지정할 수 있습니다.

ACCESS_TOKEN

AWS IAM Identity Center에서 제공한 액세스 토큰을 사용하는 경우 이 값을 입력합니다.

EXT_JWT

AWS IAM Identity Center와 통합된 웹 기반 ID 제공업체가 제공하는 OpenID Connect(OIDC) JSON 웹 토큰(JWT)을 사용하는 경우 이 값을 입력합니다.

이 파라미터는 IdpTokenAuthPlugin과 함께 작동합니다.

UID

- 기본값 – 없음
- 데이터 형식 – 문자열

데이터베이스에 액세스하는 데 사용하는 데이터베이스 사용자 이름입니다.

이 파라미터는 필수 사항입니다.

User

- 기본값 – 없음
- 데이터 형식 – 문자열

IDP를 통해 IAM 인증을 사용하여 연결 시 idp_host 서버의 사용자 이름입니다. 표준 인증을 사용하는 경우 Amazon Redshift 데이터베이스 사용자 이름에 사용할 수 있습니다.

이 파라미터는 선택 사항입니다.

webIdentityToken

- 기본값 – 없음
- 데이터 형식 – 문자열

자격 증명 공급자가 제공하는 OAuth 2.1 액세스 토큰 또는 OpenID Connect ID 토큰입니다. 애플리케이션에서 웹 자격 증명 공급자로 애플리케이션 사용자를 인증하여 이 토큰을 가져와야 합니다. Plugin_Name 옵션에 BasicJwtCredentialsProvider를 지정할 때 이 파라미터를 지정해야 합니다.

Plugin_Name 옵션에 BasicJwtCredentialsProvider를 지정하는 경우 이 파라미터는 필수입니다.

JDBC 드라이버 버전 2.x의 이전 버전

도구에 특정 버전의 드라이버가 필요한 경우에만 Amazon Redshift JDBC 드라이버 버전 2.x의 이전 버전을 다운로드합니다.

다음은 이전의 JDBC 4.2 호환 JDBC 드라이버 버전 2.x 드라이버입니다.

- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.4/redshift-jdbc42-2.2.4.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.3/redshift-jdbc42-2.2.3.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.2/redshift-jdbc42-2.2.2.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.1/redshift-jdbc42-2.2.1.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.0/redshift-jdbc42-2.2.0.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.34/redshift-jdbc42-2.1.0.34.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.33/redshift-jdbc42-2.1.0.33.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.32/redshift-jdbc42-2.1.0.32.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.30/redshift-jdbc42-2.1.0.30.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.29/redshift-jdbc42-2.1.0.29.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.28/redshift-jdbc42-2.1.0.28.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.26/redshift-jdbc42-2.1.0.26.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.25/redshift-jdbc42-2.1.0.25.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.24/redshift-jdbc42-2.1.0.24.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.23/redshift-jdbc42-2.1.0.23.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.22/redshift-jdbc42-2.1.0.22.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.21/redshift-jdbc42-2.1.0.21.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.20/redshift-jdbc42-2.1.0.20.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.19/redshift-jdbc42-2.1.0.19.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.18/redshift-jdbc42-2.1.0.18.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.17/redshift-jdbc42-2.1.0.17.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.16/redshift-jdbc42-2.1.0.16.zip>

- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.15/redshift-jdbc42-2.1.0.15.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.14/redshift-jdbc42-2.1.0.14.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.13/redshift-jdbc42-2.1.0.13.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.12/redshift-jdbc42-2.1.0.12.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.11/redshift-jdbc42-2.1.0.11.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.10/redshift-jdbc42-2.1.0.10.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.8/redshift-jdbc42-2.1.0.8.zip>
- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.7/redshift-jdbc42-2.1.0.7.zip>

Amazon Redshift Python 커넥터

Python용 Amazon Redshift 커넥터를 사용하면 [AWS Python용 SDK\(Boto3\)](#), pandas 및 Numerical Python(NumPy)과 작업을 통합할 수 있습니다. pandas에 대한 자세한 내용은 [pandas GitHub 리포지토리](#)를 참조하세요. NumPy에 대한 자세한 내용은 [NumPy GitHub 리포지토리](#)를 참조하세요.

Amazon Redshift Python 커넥터는 오픈 소스 솔루션을 제공합니다. 소스 코드를 탐색하고, 개선 사항을 요청하고, 문제를 보고하고, 기여를 제공할 수 있습니다.

Amazon Redshift Python 커넥터를 사용하려면 Python 버전 3.6 이상이 있어야 합니다. 자세한 내용은 [Amazon Redshift Python 드라이버 사용권 계약](#)을 참조하세요.

Amazon Redshift Python 커넥터는 다음을 제공합니다.

- AWS Identity and Access Management(IAM) 인증. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#) 섹션을 참조하세요.
- 페더레이션 API 액세스를 사용한 자격 증명 공급자 인증. 페더레이션 API 액세스는 다음과 같은 기업 자격 증명 공급자에 대해 지원됩니다.
 - Azure AD. 자세한 내용은 AWS 빅 데이터 블로그 게시물 [Federate Amazon Redshift access with Microsoft Azure AD single sign-on](#)을 참조하세요.
 - Active Directory Federation Services. 자세한 내용은 AWS 빅 데이터 블로그 게시물 [Federate access to your Amazon Redshift cluster with Active Directory Federation Services \(AD FS\): Part 1](#)을 참조하세요.
 - Okta. 자세한 내용은 AWS 빅 데이터 블로그 게시물 [Federate Amazon Redshift access with Okta as an identity provider](#)를 참조하세요.

- PingFederate. 자세한 내용은 [PingFederate 사이트](#)를 참조하세요.
- JumpCloud. 자세한 내용은 [JumpCloud 사이트](#)를 참조하세요.
- Amazon Redshift 데이터 유형.

Amazon Redshift Python 커넥터는 Python Database API 사양 2.0을 구현합니다. 자세한 내용은 Python 웹 사이트의 [PEP 249—Python Database API Specification v2.0](#)을 참조하세요.

주제

- [Amazon Redshift Python 커넥터 설치](#)
- [Amazon Redshift Python 커넥터의 구성 옵션](#)
- [Python 커넥터 가져오기](#)
- [NumPy와 Python 커넥터 통합](#)
- [pandas와 Python 커넥터 통합](#)
- [자격 증명 공급자 플러그인 사용](#)
- [Amazon Redshift Python 커넥터 사용 예](#)
- [Amazon Redshift Python 커넥터용 API 참조](#)

Amazon Redshift Python 커넥터 설치

다음 방법을 사용하여 Amazon Redshift Python 커넥터를 설치할 수 있습니다.

- Python 패키지 인덱스(PyPi)
- Conda
- GitHub 리포지토리 복제

PyPI에서 Python 커넥터 설치

Python 패키지 인덱스(PyPi)에서 Python 커넥터를 설치하려면 pip를 사용할 수 있습니다. 다음 명령으로 실행하세요.

```
>>> pip install redshift_connector
```

가상 환경 내에 커넥터를 설치할 수 있습니다. 다음 명령으로 실행하세요.


```
>>> pip install redshift_connector
```

커넥터를 사용하여 pandas와 NumPy를 설치할 수도 있습니다.

```
>>> pip install 'redshift_connector[full]'
```

pip에 대한 자세한 내용은 [pip 웹 사이트](#)를 참조하세요.

Conda에서 Python 커넥터 설치

Anaconda.org에서 Python 커넥터를 설치할 수 있습니다.

```
>>>conda install -c conda-forge redshift_connector
```

AWS에서 GitHub 리포지토리를 복제하여 Python 커넥터 설치

소스에서 Python 커넥터를 설치하려면 AWS에서 GitHub 리포지토리를 복제합니다. Python 및 virtualenv를 설치한 후 다음 명령을 실행하여 환경을 설정하고 필요한 종속성을 설치합니다.

```
$ git clone https://github.com/aws/amazon-redshift-python-driver.git
$ cd amazon-redshift-python-driver
$ virtualenv venv
$ . venv/bin/activate
$ python -m pip install -r requirements.txt
$ python -m pip install -e .
$ python -m pip install redshift_connector
```

Amazon Redshift Python 커넥터의 구성 옵션

아래에서 Amazon Redshift JDBC Python 커넥터에 지정할 수 있는 옵션에 대한 설명을 찾을 수 있습니다. 아래 옵션은 달리 지정하지 않는 한 사용 가능한 최신 커넥터 버전에 적용됩니다.

access_key_id

- 기본값 – 없음
- 데이터 유형 – 문자열

IAM 데이터베이스 인증용으로 구성한 IAM 사용자 또는 역할에 대한 액세스 키 ID입니다.

이 파라미터는 선택 사항입니다.

allow_db_user_override

- 기본값 – False
- 데이터 유형 – 부울

True

커넥터가 SAML(Security Assertion Markup Language) 어설션의 DbUser 값을 사용하도록 지정합니다.

False

DbUser 연결 파라미터의 값이 사용되도록 지정합니다.

이 파라미터는 선택 사항입니다.

app_name

- 기본값 – 없음
- 데이터 유형 – 문자열

인증에 사용되는 자격 증명 공급자(IdP) 애플리케이션의 이름입니다.

이 파라미터는 선택 사항입니다.

application_name

- 기본값 – 없음
- 데이터 유형 – 문자열

감사 목적으로 Amazon Redshift에 전달할 클라이언트 애플리케이션의 이름입니다. 입력하는 애플리케이션 이름은 [SYS_CONNECTION_LOG](#) 테이블의 'application_name' 열에 나타납니다. 이렇게 하면 문제를 디버깅할 때 연결 소스를 추적하고 문제를 해결할 수 있습니다.

이 파라미터는 선택 사항입니다.

auth_profile

- 기본값 – 없음

- 데이터 유형 – 문자열

연결 속성이 JSON인 Amazon Redshift 인증 프로파일의 이름입니다. 연결 파라미터 이름 지정에 대한 자세한 내용은 RedshiftProperty 클래스를 참조하세요. 최종 사용자가 제공하고 해당 사항이 있는 경우 RedshiftProperty 클래스에는 IAM 인증 프로세스 중에 생성된 연결 파라미터가 저장됩니다 (예: 임시 IAM 자격 증명). 자세한 내용은 [RedshiftProperty 클래스](#)를 참조하세요.

이 파라미터는 선택 사항입니다.

auto_create

- 기본값 – False
- 데이터 유형 – 부울

사용자가 존재하지 않는 경우 사용자를 생성할지 여부를 나타내는 값입니다.

이 파라미터는 선택 사항입니다.

client_id

- 기본값 – 없음
- 데이터 유형 – 문자열

Azure IdP의 클라이언트 ID입니다.

이 파라미터는 선택 사항입니다.

client_secret

- 기본값 – 없음
- 데이터 유형 – 문자열

Azure IdP의 클라이언트 보안 암호입니다.

이 파라미터는 선택 사항입니다.

cluster_identifier

- 기본값 – 없음

- 데이터 유형 – 문자열

Amazon Redshift 클러스터의 클러스터 식별자입니다.

이 파라미터는 선택 사항입니다.

credentials_provider

- 기본값 – 없음
- 데이터 유형 – 문자열

Amazon Redshift로 인증하는 데 사용되는 IdP입니다. 유효한 값은 다음과 같습니다.

- AdfsCredentialsProvider
- AzureCredentialsProvider
- BrowserAzureCredentialsProvider
- BrowserAzureOAuth2CredentialsProvider
- BrowserIdcAuthPlugin - AWS IAM Identity Center를 사용하는 인증 플러그인입니다.
- BrowserSamlCredentialsProvider
- IdpTokenAuthPlugin – AWS IAM Identity Center 토큰 또는 AWS IAM Identity Center에 연결된 웹 ID 제공업체의 OpenID Connect(OIDC) JSON 기반 자격 증명 토큰(JWT)을 수락하는 권한 부여 플러그인입니다.
- PingCredentialsProvider
- OktaCredentialsProvider

이 파라미터는 선택 사항입니다.

데이터베이스

- 기본값 – 없음
- 데이터 유형 – 문자열

연결할 데이터베이스의 이름입니다.

이 파라미터는 필수 사항입니다.

database_metadata_current_db_only

- 기본값 – True
- 데이터 유형 – 부울

애플리케이션이 다중 데이터베이스 datashare 카탈로그를 지원하는지 여부를 나타내는 값입니다. 기본값 True는 애플리케이션이 이전 버전과의 호환성을 위해 다중 데이터베이스 datashare 카탈로그를 지원하지 않음을 나타냅니다.

이 파라미터는 선택 사항입니다.

db_groups

- 기본값 – 없음
- 데이터 유형 – 문자열

DbUser가 표시한 사용자가 현재 세션에 대해 조인하는 기존 데이터베이스 그룹 이름의 쉼표로 구분된 목록입니다.

이 파라미터는 선택 사항입니다.

db_user

- 기본값 – 없음
- 데이터 유형 – 문자열

Amazon Redshift에 사용할 사용자 ID입니다.

이 파라미터는 선택 사항입니다.

endpoint_url

- 기본값 – 없음
- 데이터 유형 – 문자열

Amazon Redshift 엔드포인트 URL입니다. 이 옵션은 AWS 내부 전용입니다.

이 파라미터는 선택 사항입니다.

group_federation

- 기본값 – False
- 데이터 유형 – 부울

이 옵션은 Amazon Redshift IDP 그룹을 사용할지 여부를 지정합니다.

이 파라미터는 선택 사항입니다.

true

Amazon Redshift IDP(자격 증명 공급자) 그룹을 사용합니다.

false

사용자 페더레이션을 위해 STS API 및 GetClusterCredentials를 사용하고 연결을 위해 db_groups를 지정합니다.

host

- 기본값 – 없음
- 데이터 유형 – 문자열

Amazon Redshift 클러스터의 호스트 이름입니다.

이 파라미터는 선택 사항입니다.

iam

- 기본값 – False
- 데이터 유형 – 부울

IAM 인증이 사용됩니다.

이 파라미터는 필수 사항입니다.

iam_disable_cache

- 기본값 – False

- 데이터 유형 – 부울

이 옵션은 IAM 자격 증명이 캐시되는지 여부를 지정합니다. 기본적으로 IAM 자격 증명이 캐시되지 않습니다. API 게이트웨이에 대한 요청이 제한될 때 성능이 향상됩니다.

이 파라미터는 선택 사항입니다.

`idc_client_display_name`

- 기본값 - Amazon Redshift Python 커넥터
- 데이터 형식 – 문자열

`BrowserIdcAuthPlugin`을 사용하는 클라이언트에 사용할 표시 이름입니다.

이 파라미터는 선택 사항입니다.

`idc_region`

- 기본값 – 없음
- 데이터 형식 – 문자열

AWS IAM Identity Center 인스턴스가 위치한 AWS 리전입니다.

이 파라미터는 `credentials_provider` 구성 옵션에서 `BrowserIdcAuthPlugin`을 사용하여 인증하는 경우에만 필요합니다.

`idp_partition`

- 기본값 – 없음
- 데이터 형식 – 문자열

ID 제공업체(IdP)가 구성된 클라우드 파티션을 지정합니다. 드라이버가 연결하는 IdP 인증 엔드포인트를 결정합니다.

이 파라미터를 비워 두면 드라이버는 기본적으로 상용 파티션으로 설정됩니다. 가능한 값은 다음과 같습니다.

- `us-gov`: IdP가 Azure Government에 구성된 경우 이 값을 사용합니다. 예를 들어 Azure AD Government는 `login.microsoftonline.us` 엔드포인트를 사용합니다.

- `cn`: IdP가 중국 클라우드 파티션에 구성된 경우 이 값을 사용합니다. 예를 들어 Azure AD China는 `login.chinacloudapi.cn` 엔드포인트를 사용합니다.

이 파라미터는 선택 사항입니다.

`idpPort`

- 기본값 – 7890
- 데이터 유형 – 정수

IdP가 SAML 어설션을 전송하는 수신 포트입니다.

이 파라미터는 필수 사항입니다.

`idp_response_timeout`

- 기본값 – 120
- 데이터 유형 – 정수

IdP에서 SAML 어설션 검색에 대한 시간 제한입니다.

이 파라미터는 필수 사항입니다.

`idp_tenant`

- 기본값 – 없음
- 데이터 유형 – 문자열

IdP 테넌트입니다.

이 파라미터는 선택 사항입니다.

`issuer_url`

- 기본값 – 없음
- 데이터 형식 – 문자열

AWS IAM Identity Center 서버의 인스턴스 엔드포인트를 가리킵니다.

이 파라미터는 `credentials_provider` 구성 옵션에서 `BrowserIdcAuthPlugin`을 사용하여 인증하는 경우에만 필요합니다.

`listen_port`

- 기본값 – 7890
- 데이터 유형 – 정수

드라이버가 브라우저 플러그인을 통해 SAML, Azure AD 또는 AWS IAM Identity Center 서비스를 사용할 때 ID 제공업체 또는 권한 부여 코드로부터 SAML 응답을 받는 데 사용하는 포트입니다.

이 파라미터는 선택 사항입니다.

`login_url`

- 기본값 – 없음
- 데이터 유형 – 문자열

IdP의 Single Sign-On URL입니다.

이 파라미터는 선택 사항입니다.

`max_prepared_statements`

- 기본값 - 1000
- 데이터 유형 – 정수

연결당 캐시되는 준비된 명령문의 최대 개수입니다. 이 파라미터를 0으로 설정하면 캐싱 메커니즘이 비활성화됩니다. 이 파라미터에 음수를 입력하면 기본값으로 설정됩니다.

이 파라미터는 선택 사항입니다.

`numeric_to_float`

- 기본값 – False
- 데이터 유형 – 부울

이 옵션은 커넥터가 숫자 데이터 형식 값을 `decimal.Decimal`에서 `float`로 변환하는지 여부를 지정합니다. 기본적으로 커넥터는 숫자 데이터 형식 값을 `decimal.Decimal`로 수신하며 변환하지 않습니다.

결과가 반올림될 수 있으므로 정밀도가 필요한 사용 사례에는 `numeric_to_float`를 사용하지 않는 것이 좋습니다.

`decimal.Decimal` 및 `decimal.Decimal`과 `float`의 장단점에 대한 자세한 내용은 Python 웹 사이트에서 [decimal - Decimal fixed point and floating point arithmetic](#)(decimal - 10진수 고정 소수점 및 부동 소수점 연산)을 참조하세요.

이 파라미터는 선택 사항입니다.

`partner_sp_id`

- 기본값 – 없음
- 데이터 유형 – 문자열

Ping 인증에 사용되는 파트너 SP ID입니다.

이 파라미터는 선택 사항입니다.

암호

- 기본값 – 없음
- 데이터 유형 – 문자열

인증에 사용할 암호입니다.

이 파라미터는 선택 사항입니다.

포트

- 기본값 - 5439
- 데이터 유형 – 정수

Amazon Redshift 클러스터의 포트 번호입니다.

이 파라미터는 필수 사항입니다.

`preferred_role`

- 기본값 – 없음

- 데이터 유형 – 문자열

현재 연결에 선호되는 IAM 역할입니다.

이 파라미터는 선택 사항입니다.

principal_arn

- 기본값 – 없음
- 데이터 유형 – 문자열

정책을 생성할 사용자 또는 IAM 역할의 Amazon 리소스 이름(ARN)입니다. 역할에 정책을 연결한 후 사용자에게 해당 역할을 할당하여 액세스할 수 있도록 하는 것이 좋습니다.

이 파라미터는 선택 사항입니다.

profile

- 기본값 – 없음
- 데이터 유형 – 문자열

AWS 자격 증명이 포함된 AWS 자격 증명 파일의 프로파일 이름입니다.

이 파라미터는 선택 사항입니다.

provider_name

- 기본값 – 없음
- 데이터 유형 – 문자열

Redshift Native Authentication Provider의 이름입니다.

이 파라미터는 선택 사항입니다.

region

- 기본값 – 없음
- 데이터 유형 – 문자열

클러스터가 있는 AWS 리전입니다.

이 파라미터는 선택 사항입니다.

role_arn

- 기본값 – 없음
- 데이터 유형 – 문자열

발신자가 수임하는 역할의 Amazon 리소스 이름(ARN)입니다. 이 파라미터는 JwcCredentialsProvider로 표시된 공급자가 사용합니다.

JwcCredentialsProvider 공급자의 경우 이 파라미터는 필수입니다. 그렇지 않으면 이 파라미터는 선택 항목입니다.

role_session_name

- 기본값 – jwt_redshift_session
- 데이터 유형 – 문자열

말은 역할 세션의 식별자입니다. 일반적으로 애플리케이션 사용자와 연결된 이름 또는 식별자를 전달합니다. 애플리케이션에서 사용하는 임시 보안 자격 증명은 해당 사용자와 연결됩니다. 이 파라미터는 JwcCredentialsProvider로 표시된 공급자가 사용합니다.

이 파라미터는 선택 사항입니다.

범위

- 기본값 – 없음
- 데이터 유형 – 문자열

사용자가 동의할 수 있는 공백으로 구분된 범위 목록입니다. 애플리케이션이 해당 사용자가 호출하려는 API에 대한 동의를 얻을 수 있도록 이 파라미터를 지정합니다. credentials_provider 옵션에 BrowserAzureOAuth2CredentialsProvider를 지정할 때 이 파라미터를 지정할 수 있습니다.

이 파라미터는 BrowserAzureOAuth2CredentialsProvider 플러그인에 반드시 필요합니다.

secret_access_key_id

- 기본값 – 없음

- 데이터 유형 – 문자열

IAM 데이터베이스 인증용으로 구성된 IAM 사용자 또는 역할에 대한 보안 액세스 키 ID입니다.

이 파라미터는 선택 사항입니다.

session_token

- 기본값 – 없음
- 데이터 유형 – 문자열

IAM 데이터베이스 인증용으로 구성된 IAM 사용자 또는 역할에 대한 액세스 키 ID입니다. 임시 AWS 자격 증명이 사용 중인 경우 이 파라미터가 필요합니다.

이 파라미터는 선택 사항입니다.

serverless_acct_id

- 기본값 – 없음
- 데이터 유형 – 문자열

Amazon Redshift Serverless 계정 ID입니다.

이 파라미터는 선택 사항입니다.

serverless_work_group

- 기본값 – 없음
- 데이터 유형 – 문자열

Amazon Redshift Serverless 작업 그룹 이름입니다.

이 파라미터는 선택 사항입니다.

ssl

- 기본값 – True
- 데이터 유형 – 부울

보안 소켓 계층(SSL)을 사용합니다.

이 파라미터는 필수 사항입니다.

ssl_insecure

- 기본값 – False
- 데이터 유형 – 부울

IdP 호스트의 서버 SSL 인증서 검증을 비활성화할지 여부를 지정하는 값입니다. 이 파라미터를 True로 설정하면 IdP 호스트의 서버 SSL 인증서 검증이 비활성화됩니다. 프로덕션 환경에서는 기본값 False를 유지하는 것이 좋습니다.

이 파라미터는 선택 사항입니다.

sslmode

- 기본값 – verify-ca
- 데이터 유형 – 문자열

Amazon Redshift에 대한 연결 보안입니다. 다음 중 하나를 지정할 수 있습니다.

- [verify-ca]
- [verify-full]

이 파라미터는 필수 사항입니다.

tcp_keepalive

- 기본값 – True
- 데이터 유형 – 부울

TCP keepalive를 사용하여 연결이 시간 초과되지 않도록 할지 여부입니다. 다음 값을 지정할 수 있습니다.

- True: 드라이버가 TCP keepalive를 사용하여 연결이 시간 초과되지 않도록 합니다.
- False: 드라이버가 TCP keepalive를 사용하지 않습니다.

이 파라미터는 선택 사항입니다.

`tcp_keepalive_count`

- 기본값 – 없음
- 데이터 유형 – 정수

연결이 비활성 상태로 간주되기 전에 전송할 응답이 없는 프로브의 수입입니다. 예를 들어, 값을 3으로 설정하면 드라이버가 응답이 없는 keepalive 패킷을 3회 전송한 후 연결이 더 이상 활성 상태가 아니라고 판단합니다.

이 파라미터가 지정되지 않으면 Amazon Redshift는 시스템의 기본값을 사용합니다.

이 파라미터는 선택 사항입니다.

`tcp_keepalive_interval`

- 기본값 – 없음
- 데이터 유형 – 정수

드라이버가 이전 킵얼라이브 프로브에 대한 확인 응답을 받지 못한 경우, 후속 keepalive 프로브 간 간격(초)입니다. 이 파라미터를 지정할 경우 양의 정수여야 합니다.

이 파라미터가 지정되지 않으면 Amazon Redshift는 시스템의 기본값을 사용합니다.

이 파라미터는 선택 사항입니다.

`tcp_keepalive_idle`

- 기본값 – 없음
- 데이터 유형 – 정수

드라이버가 첫 번째 keepalive 프로브를 전송하기까지의 비활성 상태 지속 시간(초)입니다. 예를 들어 값을 120으로 설정하면 드라이버는 첫 번째 keepalive 패킷을 전송하기 전 2분 동안 비활성 상태를 대기합니다. 이 파라미터를 지정할 경우 양의 정수여야 합니다.

이 파라미터가 지정되지 않으면 Amazon Redshift는 시스템의 기본값을 사용합니다.

이 파라미터는 선택 사항입니다.

제한 시간

- 기본값 – 없음
- 데이터 유형 – 정수

서버 연결 시간이 초과되기까지의 시간(초)입니다.

이 파라미터는 선택 사항입니다.

토큰

- 기본값 – 없음
- 데이터 형식 – 문자열

AWS IAM Identity Center에서 제공한 액세스 토큰 또는 AWS IAM Identity Center와 연결된 웹 ID 제공업체가 제공한 OpenID Connect(OIDC) JSON 웹 토큰(JWT)입니다. 애플리케이션은 AWS IAM Identity Center 또는 AWS IAM Identity Center와 연결된 자격 증명 공급자를 통해 애플리케이션 사용자를 인증하여 이 토큰을 생성해야 합니다.

이 파라미터는 IdpTokenAuthPlugin과 함께 작동합니다.

token_type

- 기본값 – 없음
- 데이터 형식 – 문자열

IdpTokenAuthPlugin에서 사용되는 토큰 유형입니다.

다음 값을 지정할 수 있습니다.

ACCESS_TOKEN

AWS IAM Identity Center에서 제공한 액세스 토큰을 사용하는 경우 이 값을 입력합니다.

EXT_JWT

AWS IAM Identity Center와 통합된 웹 기반 ID 제공업체가 제공하는 OpenID Connect(OIDC) JSON 웹 토큰(JWT)을 사용하는 경우 이 값을 입력합니다.

이 파라미터는 IdpTokenAuthPlugin과 함께 작동합니다.

user

- 기본값 – 없음
- 데이터 유형 – 문자열

권한 부여에 사용할 사용자 이름입니다.

이 파라미터는 선택 사항입니다.

web_identity_token

- 기본값 – 없음
- 데이터 유형 – 문자열

자격 증명 공급자가 제공하는 OAuth 2.0 액세스 토큰 또는 OpenID Connect ID 토큰입니다. 웹 자격 증명 공급자를 통해 애플리케이션을 사용하는 사용자를 인증하여 애플리케이션이 이 토큰을 가져오도록 합니다. JwtCredentialsProvider로 표시된 공급자는 이 파라미터를 사용합니다.

JwtCredentialsProvider 공급자의 경우 이 파라미터는 필수입니다. 그렇지 않으면 이 파라미터는 선택 항목입니다.

Python 커넥터 가져오기

Python 커넥터를 가져오려면 다음 명령을 실행합니다.

```
>>> import redshift_connector
```

AWS 자격 증명을 사용하여 Amazon Redshift 클러스터에 연결하려면 다음 명령을 실행합니다.

```
conn = redshift_connector.connect(  
    host='examplecluster.abc123xyz789.us-west-1.redshift.amazonaws.com',  
    port=5439,  
    database='dev',  
    user='awsuser',  
    password='my_password'  
)
```

NumPy와 Python 커넥터 통합

다음은 Python 커넥터를 NumPy와 통합하는 예입니다.

```
>>> import numpy
#Connect to the cluster
>>> import redshift_connector
>>> conn = redshift_connector.connect(
    host='examplecluster.abc123xyz789.us-west-1.redshift.amazonaws.com',
    port=5439,
    database='dev',
    user='awsuser',
    password='my_password'
)

# Create a Cursor object
>>> cursor = conn.cursor()

# Query and receive result set
cursor.execute("select * from book")

result: numpy.ndarray = cursor.fetch_numpy_array()
print(result)
```

다음은 결과입니다.

```
[['One Hundred Years of Solitude' 'Gabriel García Márquez']
['A Brief History of Time' 'Stephen Hawking']]
```

pandas와 Python 커넥터 통합

다음은 Python 커넥터를 pandas와 통합하는 예입니다.

```
>>> import pandas

#Connect to the cluster
>>> import redshift_connector
>>> conn = redshift_connector.connect(
    host='examplecluster.abc123xyz789.us-west-1.redshift.amazonaws.com',
    port=5439,
    database='dev',
    user='awsuser',
    password='my_password'
)

# Create a Cursor object
```

```
>>> cursor = conn.cursor()

# Query and receive result set
cursor.execute("select * from book")
result: pandas.DataFrame = cursor.fetch_dataframe()
print(result)
```

자격 증명 공급자 플러그인 사용

자격 증명 공급자 플러그인을 사용하는 방법에 대한 일반적인 내용은 [IAM 자격 증명 제공 옵션](#) 섹션을 참조하세요. IAM 역할에 대한 모범 사례를 비롯한 IAM ID 관리에 대한 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#) 섹션을 참조하세요.

ADFS 자격 증명 공급자 플러그인을 사용한 인증

다음은 ADFS(Active Directory Federation Service) 자격 증명 공급자 플러그인을 사용하여 Amazon Redshift 데이터베이스에 연결하는 사용자를 인증하는 예입니다.

```
>>> con = redshift_connector.connect(
    iam=True,
    database='dev',
    host='my-testing-cluster.abc.us-east-2.redshift.amazonaws.com',
    cluster_identifier='my-testing-cluster',
    credentials_provider='AdfsCredentialsProvider',
    user='brooke@myadfshostname.com',
    password='Hunter2',
    idp_host='myadfshostname.com'
)
```

Azure 자격 증명 공급자 플러그인을 사용한 인증

다음은 Azure 자격 증명 공급자 플러그인을 사용한 인증의 예입니다. 다음과 같이 Azure Enterprise 애플리케이션에 대한 `client_id` 및 `client_secret` 값을 생성할 수 있습니다.

```
>>> con = redshift_connector.connect(
    iam=True,
    database='dev',
    host='my-testing-cluster.abc.us-east-2.redshift.amazonaws.com',
    cluster_identifier='my-testing-cluster',
    credentials_provider='AzureCredentialsProvider',
    user='brooke@myazure.org',
    password='Hunter2',
    idp_tenant='my_idp_tenant',
```

```

client_id='my_client_id',
client_secret='my_client_secret',
preferred_role='arn:aws:iam:123:role/DataScientist'
)

```

AWS IAM Identity Center 자격 증명 공급자 플러그인을 사용한 인증

다음은 AWS IAM Identity Center 자격 증명 공급자 플러그인을 사용한 인증의 예입니다.

```

with redshift_connector.connect(
    credentials_provider='BrowserIdcAuthPlugin',
    host='my-testing-cluster.abc.us-east-2.redshift.amazonaws.com',
    database='dev',
    idc_region='us-east-1',
    issuer_url='https://identitycenter.amazonaws.com/ssoins-790723ebe09c86f9',
    idp_response_timeout=60,
    listen_port=8100,
    idc_client_display_name='Test Display Name',
    # port value of 5439 is specified by default
)

```

Azure 브라우저 자격 증명 공급자 플러그인을 사용한 인증

다음은 Azure 브라우저 자격 증명 공급자 플러그인을 사용하여 Amazon Redshift 데이터베이스에 연결하는 사용자를 인증하는 예입니다.

멀티 팩터 인증은 사용자가 로그인 보안 인증 정보를 제공하는 브라우저에서 발생합니다.

```

>>>con = redshift_connector.connect(
    iam=True,
    database='dev',
    host='my-testing-cluster.abc.us-east-2.redshift.amazonaws.com',
    cluster_identifier='my-testing-cluster',
    credentials_provider='BrowserAzureCredentialsProvider',
    idp_tenant='my_idp_tenant',
    client_id='my_client_id',
)

```

Okta 자격 증명 공급자 플러그인을 사용한 인증

다음은 Okta 자격 증명 공급자 플러그인을 사용한 인증의 예입니다. Okta 애플리케이션을 통해 `idp_host`, `app_id` 및 `app_name`에 대한 값을 얻을 수 있습니다.

```
>>> con = redshift_connector.connect(
    iam=True,
    database='dev',
    host='my-testing-cluster.abc.us-east-2.redshift.amazonaws.com',
    cluster_identifier='my-testing-cluster',
    credentials_provider='OktaCredentialsProvider',
    user='brooke@myazure.org',
    password='hunter2',
    idp_host='my_idp_host',
    app_id='my_first_appetizer',
    app_name='dinner_party'
)
```

일반 SAML 브라우저 자격 증명 공급자 플러그인과 함께 JumpCloud를 사용한 인증

다음은 일반 SAML 브라우저 자격 증명 공급자 플러그인과 함께 JumpCloud를 사용한 인증의 예입니다.

암호 파라미터가 필요합니다. 그러나 브라우저에서 멀티 팩터 인증이 이루어지므로 이 파라미터를 입력할 필요는 없습니다.

```
>>> con = redshift_connector.connect(
    iam=True,
    database='dev',
    host='my-testing-cluster.abc.us-east-2.redshift.amazonaws.com',
    cluster_identifier='my-testing-cluster',
    credentials_provider='BrowserSamlCredentialsProvider',
    user='brooke@myjumpcloud.org',
    password='',
    login_url='https://sso.jumpcloud.com/saml2/plustwo_melody'
)
```

Amazon Redshift Python 커넥터 사용 예

다음은 Amazon Redshift Python 커넥터 사용 방법의 예입니다. 실행하려면 먼저 Python 커넥터를 설치해야 합니다. Amazon Redshift Python 커넥터 설치에 대한 자세한 내용은 [Amazon Redshift Python 커넥터 설치](#) 섹션을 참조하세요. Python 커넥터와 함께 사용할 수 있는 구성 옵션에 대한 자세한 내용은 [Amazon Redshift Python 커넥터의 구성 옵션](#) 단원을 참조하세요.

주제

- [AWS 보안 인증 정보를 사용하여 Amazon Redshift 클러스터에 연결하고 쿼리 수행](#)

- [autocommit 사용](#)
- [커서 파라미터 스타일 구성](#)
- [COPY를 사용하여 Amazon S3 버킷에서 데이터 복사 및 UNLOAD를 사용하여 Amazon S3 버킷에 데이터 쓰기](#)

AWS 보안 인증 정보를 사용하여 Amazon Redshift 클러스터에 연결하고 쿼리 수행

다음 예시는 AWS 보안 인증 정보를 사용하여 Amazon Redshift 클러스터에 연결한 후 테이블을 쿼리하고 쿼리 결과를 가져오는 방법을 안내합니다.

```
#Connect to the cluster
>>> import redshift_connector
>>> conn = redshift_connector.connect(
    host='examplecluster.abc123xyz789.us-west-1.redshift.amazonaws.com',
    database='dev',
    port=5439,
    user='awsuser',
    password='my_password'
)

# Create a Cursor object
>>> cursor = conn.cursor()

# Query a table using the Cursor
>>> cursor.execute("select * from book")

#Retrieve the query result set
>>> result: tuple = cursor.fetchall()
>>> print(result)
>> (['One Hundred Years of Solitude', 'Gabriel García Márquez'], ['A Brief History of Time', 'Stephen Hawking'])
```

autocommit 사용

autocommit 속성은 Python 데이터베이스 API 사양에 따라 기본적으로 해제되어 있습니다. 트랜잭션이 진행되지 않도록 롤백 명령을 수행한 후 다음 명령을 사용하여 연결의 autocommit 속성을 설정할 수 있습니다.

```
#Connect to the cluster
```

```
>>> import redshift_connector
>>> conn = redshift_connector.connect(...)

# Run a rollback command
>>> conn.rollback()

# Turn on autocommit
>>> conn.autocommit = True
>>> conn.run("VACUUM")

# Turn off autocommit
>>> conn.autocommit = False
```

커서 파라미터 스타일 구성

커서의 파라미터 스타일은 `cursor.paramstyle`을 통해 수정할 수 있습니다. 사용되는 기본 파라미터 스타일은 `format`입니다. 이 파라미터의 유효한 값은 `qmark`, `numeric`, `named`, `format`, `pyformat`입니다.

다음은 다양한 파라미터 스타일을 사용하여 샘플 SQL 문에 파라미터를 전달하는 예제입니다.

```
# qmark
redshift_connector.paramstyle = 'qmark'
sql = 'insert into foo(bar, jar) VALUES(?, ?)'
cursor.execute(sql, (1, "hello world"))

# numeric
redshift_connector.paramstyle = 'numeric'
sql = 'insert into foo(bar, jar) VALUES(:1, :2)'
cursor.execute(sql, (1, "hello world"))

# named
redshift_connector.paramstyle = 'named'
sql = 'insert into foo(bar, jar) VALUES(:p1, :p2)'
cursor.execute(sql, {"p1":1, "p2":"hello world"})

# format
redshift_connector.paramstyle = 'format'
sql = 'insert into foo(bar, jar) VALUES(%s, %s)'
cursor.execute(sql, (1, "hello world"))

# pyformat
redshift_connector.paramstyle = 'pyformat'
```

```
sql = 'insert into foo(bar, jar) VALUES(%(bar)s, %(jar)s)'
cursor.execute(sql, {"bar": 1, "jar": "hello world"})
```

COPY를 사용하여 Amazon S3 버킷에서 데이터 복사 및 UNLOAD를 사용하여 Amazon S3 버킷에 데이터 쓰기

다음 예에서는 Amazon S3 버킷에서 테이블로 데이터를 복사한 다음 해당 테이블에서 버킷으로 언로드하는 방법을 보여줍니다.

다음 데이터가 포함된 category_csv.txt 텍스트 파일이 S3 버킷에 업로드됩니다.

```
12,Shows,Musicals,Musical theatre
13,Shows,Plays,"All ""non-musical"" theatre"
14,Shows,Opera,"All opera, light, and ""rock"" opera"
15,Concerts,Classical,"All symphony, concerto, and choir concerts"
```

다음은 Amazon Redshift 데이터베이스에 먼저 연결하는 Python 코드의 예입니다. 그런 다음 category라는 테이블을 생성하고 S3 버킷의 CSV 데이터를 테이블로 복사합니다.

```
#Connect to the cluster and create a Cursor
>>> import redshift_connector
>>> with redshift_connector.connect(...) as conn:
>>> with conn.cursor() as cursor:

#Create an empty table
>>> cursor.execute("create table category (catid int, cargroup varchar, catname
    varchar, catdesc varchar)")

#Use COPY to copy the contents of the S3 bucket into the empty table
>>> cursor.execute("copy category from 's3://testing/category_csv.txt' iam_role
    'arn:aws:iam::123:role/RedshiftCopyUnload' csv;")

#Retrieve the contents of the table
>>> cursor.execute("select * from category")
>>> print(cursor.fetchall())

#Use UNLOAD to copy the contents of the table into the S3 bucket
>>> cursor.execute("unload ('select * from category') to 's3://testing/
    unloaded_category_csv.txt' iam_role 'arn:aws:iam::123:role/RedshiftCopyUnload' csv;")

#Retrieve the contents of the bucket
>>> print(cursor.fetchall())
```



```
>> ([12, 'Shows', 'Musicals', 'Musical theatre'], [13, 'Shows', 'Plays', 'All "non-musical" theatre'], [14, 'Shows', 'Opera', 'All opera, light, and "rock" opera'], [15, 'Concerts', 'Classical', 'All symphony, concerto, and choir concerts'])
```

autocommit을 true로 설정하지 않은 경우 execute() 문을 실행한 후 conn.commit()으로 커밋합니다.

데이터는 다음 콘텐츠가 들어 있는 S3 버킷의 unloaded_category_csv.text0000_part00 파일로 업로드됩니다.

```
12,Shows,Musicals,Musical theatre
13,Shows,Plays,"All ""non-musical"" theatre"
14,Shows,Opera,"All opera, light, and ""rock"" opera"
15,Concerts,Classical,"All symphony, concerto, and choir concerts"
```

Amazon Redshift Python 커넥터용 API 참조

다음에서 Amazon Redshift Python 커넥터 API 작업에 대한 설명을 찾을 수 있습니다.

redshift_connector

다음에서 redshift_connector API 작업에 대한 설명을 찾을 수 있습니다.

connect(user, database, password[, port, ...])

Amazon Redshift 클러스터에 대한 연결을 설정합니다. 이 기능은 사용자 입력을 검증하고 선택적으로 자격 증명 공급자 플러그인을 사용하여 인증한 다음 연결 객체를 구성합니다.

apilevel

지원되는 DBAPI 레벨은 현재 "2.0"입니다.

```
paramstyle, str(object='') -> str str(bytes_or_buffer[, encoding[, errors]])
-> str
```

전역적으로 사용할 데이터베이스 API 파라미터 스타일입니다.

연결

다음에서 Amazon Redshift Python 커넥터의 연결 API 작업에 대한 설명을 찾을 수 있습니다.

__init__(user, password, database[, host, ...])

원시 연결 객체를 초기화합니다.

cursor

이 연결에 바인딩된 커서 객체를 생성합니다.

commit

현재 데이터베이스 트랜잭션을 커밋합니다.

rollback

현재 데이터베이스 트랜잭션을 롤백합니다.

close

데이터베이스 연결을 닫습니다.

execute(cursor, operation, vals)

지정된 SQL 명령을 실행합니다. `redshift_connector.paramstyle` 값에 따라 파라미터를 시퀀스 또는 매핑으로 제공할 수 있습니다.

run(sql[, stream])

지정된 SQL 명령을 실행합니다. COPY 명령과 함께 사용할 스트림을 제공할 수도 있습니다.

xid(format_id, global_transaction_id, ...)

트랜잭션 ID를 생성합니다. postgres에서는 `global_transaction_id` 파라미터만 사용됩니다. `format_id` 및 `branch_qualifier`는 postgres에서 사용되지 않습니다. `global_transaction_id`는 튜플(`format_id`, `global_transaction_id`, `branch_qualifier`)을 반환하는 postgres에서 지원하는 모든 문자열 식별자일 수 있습니다.

tpc_begin(xid)

형식 ID, 글로벌 트랜잭션 ID 및 분기 한정자로 구성된 트랜잭션 ID `xid`로 TPC 트랜잭션을 시작합니다.

tpc_prepare

`.tpc_begin`으로 시작된 트랜잭션의 첫 번째 단계를 수행합니다.

tpc_commit([xid])

인수 없이 호출 시 `.tpc_commit`은 이전에 `.tpc_prepare()`로 준비된 TPC 트랜잭션을 커밋합니다.

tpc_rollback([xid])

인수 없이 호출 시 `.tpc_rollback`은 TPC 트랜잭션을 롤백합니다.

tpc_recover

.tpc_commit(xid) 또는 .tpc_rollback(xid)과 함께 사용하기에 적합한 보류 중인 트랜잭션 ID 목록을 반환합니다.

커서

다음에서 커서 API 작업에 대한 설명을 찾을 수 있습니다.

`__init__(connection[, paramstyle])`

원시 커서 객체를 초기화합니다.

`insert_data_bulk(filename, table_name, parameter_indices, column_names, delimiter, batch_size)`

대량 INSERT 문을 실행합니다.

`execute(operation[, args, stream, ...])`

데이터베이스 작업을 실행합니다.

`executemany(operation, param_sets)`

데이터베이스 작업을 준비한 다음 제공된 모든 파라미터 시퀀스 또는 매핑에 대해 실행합니다.

`fetchone`

쿼리 결과 집합의 다음 행을 가져옵니다.

`fetchmany([num])`

쿼리 결과 집합의 다음 행 집합을 가져옵니다.

`fetchall`

쿼리 결과의 나머지 행을 모두 가져옵니다.

`close`

지금 커서를 닫습니다.

`__iter__`

커서 객체를 반복하여 쿼리에서 행을 검색할 수 있습니다.

`fetch_dataframe([num])`

마지막 쿼리 결과의 데이터 프레임을 반환합니다.

```
write_dataframe(df, table)
```

동일한 구조 데이터 프레임을 Amazon Redshift 데이터베이스에 씁니다.

```
fetch_numpy_array([num])
```

마지막 쿼리 결과의 NumPy 배열을 반환합니다.

```
get_catalogs
```

Amazon Redshift는 단일 연결에서 여러 카탈로그를 지원하지 않습니다. Amazon Redshift는 현재 카탈로그만 반환합니다.

```
get_tables([catalog, schema_pattern, ...])
```

시스템 내에서 사용자 정의된 고유한 공용 테이블을 반환합니다.

```
get_columns([catalog, schema_pattern, ...])
```

Amazon Redshift 데이터베이스의 특정 테이블에 있는 모든 열 목록을 반환합니다.

AdfsCredentialsProvider 플러그인

다음은 Amazon Redshift Python 커넥터에 대한 AdfsCredentialsProvider 플러그인 API 작업의 구문입니다.

```
redshift_connector.plugin.AdfsCredentialsProvider()
```

AzureCredentialsProvider 플러그인

다음은 Amazon Redshift Python 커넥터에 대한 AzureCredentialsProvider 플러그인 API 작업의 구문입니다.

```
redshift_connector.plugin.AzureCredentialsProvider()
```

BrowserAzureCredentialsProvider 플러그인

다음은 Amazon Redshift Python 커넥터에 대한 BrowserAzureCredentialsProvider 플러그인 API 작업의 구문입니다.

```
redshift_connector.plugin.BrowserAzureCredentialsProvider()
```

BrowserSamlCredentialsProvider 플러그인

다음은 Amazon Redshift Python 커넥터에 대한 BrowserSamlCredentialsProvider 플러그인 API 작업의 구문입니다.

```
redshift_connector.plugin.BrowserSamlCredentialsProvider()
```

OktaCredentialsProvider 플러그인

다음은 Amazon Redshift Python 커넥터에 대한 OktaCredentialsProvider 플러그인 API 작업의 구문입니다.

```
redshift_connector.plugin.OktaCredentialsProvider()
```

PingCredentialsProvider 플러그인

다음은 Amazon Redshift Python 커넥터에 대한 PingCredentialsProvider 플러그인 API 작업의 구문입니다.

```
redshift_connector.plugin.PingCredentialsProvider()
```

SamlCredentialsProvider 플러그인

다음은 Amazon Redshift Python 커넥터에 대한 SamlCredentialsProvider 플러그인 API 작업의 구문입니다.

```
redshift_connector.plugin.SamlCredentialsProvider()
```

Apache Spark용 Amazon Redshift 통합

[Apache Spark](#)는 기계 학습, 스트림 처리 또는 그래픽 분석을 수행하는 데 도움이 되는 분산 처리 프레임워크 및 프로그래밍 모델입니다. Apache Hadoop과 마찬가지로, Spark는 빅 데이터 워크로드에 일반적으로 사용되는 오픈 소스 분산형 처리 시스템입니다. Spark에는 최적화된 비순환 방향 그래프 (DAG) 실행 엔진이 있으며 메모리 안에 데이터를 능동적으로 캐시합니다. 이렇게 하면 특히 특정 알고리즘 및 대화형 쿼리의 성능을 향상시킬 수 있습니다.

이 통합은 Amazon Redshift와 Amazon Redshift Serverless에서 데이터를 읽고 쓰는 Apache Spark 애플리케이션을 빌드할 수 있는 Spark 커넥터를 제공합니다. 이러한 애플리케이션은 애플리케이션 성능이나 데이터의 트랜잭션 일관성을 손상시키지 않습니다. 이 통합은 [Amazon EMR](#) 및 [AWS Glue](#)에 자동으로 포함되므로 데이터 수집 및 변환 파이프라인의 일부로 Amazon Redshift에 데이터를 액세스하고 로드하는 Apache Spark 작업을 즉시 실행할 수 있습니다.

현재 이 통합과 함께 Spark 버전 3.3.x, 3.4.x, 3.5.x 및 4.0.0을 사용할 수 있습니다.

이 통합은 다음을 제공합니다.

- AWS Identity and Access Management(IAM) 인증. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.
- 조건자 및 쿼리 푸시다운을 통한 성능 향상.
- Amazon Redshift 데이터 형식.
- Amazon Redshift와 Amazon Redshift Serverless에 대한 연결.

Spark 커넥터 사용 시 고려 사항 및 제한 사항

- tempdir URI는 Amazon S3 위치를 가리킵니다. 이 임시 디렉터리는 자동으로 정리되지 않으며 추가 비용이 발생할 수 있습니다. Amazon Simple Storage Service 사용 설명서의 [Amazon S3 수명 주기 정책](#)을 사용하여 Amazon S3 버킷에 대한 보존 규칙을 정의하는 것이 좋습니다.
- 기본적으로 Amazon S3와 Redshift 간의 복사는 S3 버킷과 Redshift 클러스터가 서로 다른 AWS 리전에 있는 경우 작동하지 않습니다. 별도의 AWS 리전을 사용하려면 tempdir_region 파라미터를 tempdir에 사용되는 S3 버킷의 리전으로 설정합니다.
- tempformat 파라미터를 사용하여 Parquet 데이터를 쓰는 경우 교차 리전은 S3와 Redshift 사이에 씁니다.
- [Amazon S3 서버 측 암호화](#)를 사용하여 사용된 Amazon S3 버킷을 암호화하는 것이 좋습니다.
- [Amazon S3 버킷에 대한 퍼블릭 액세스를 차단](#)하는 것이 좋습니다.
- Amazon Redshift 클러스터는 공개적으로 액세스할 수 없도록 하는 것이 좋습니다.
- [Amazon Redshift 감사 로깅](#)을 켜는 것이 좋습니다.
- [Amazon Redshift 저장된 데이터 암호화](#)를 켜는 것이 좋습니다.
- Amazon EMR의 Spark에서 Amazon Redshift로의 JDBC 연결에 대해 SSL을 켜는 것이 좋습니다.
- Amazon Redshift 인증 파라미터에 대해 aws_iam_role 파라미터를 사용하여 IAM 역할을 전달하는 것이 좋습니다.

Spark 커넥터를 사용한 인증

다음 다이어그램은 Amazon S3, Amazon Redshift, Spark 드라이버 및 Spark 실행기 간의 인증을 설명합니다.

Redshift와 Spark 간의 인증

Amazon Redshift 제공 JDBC 드라이버 버전 2.x 드라이버를 사용해 로그인 자격 증명을 지정하여 Spark 커넥터를 Amazon Redshift에 연결할 수 있습니다. IAM을 사용하려면 [IAM 인증을 사용하도록 JDBC URL을 구성](#)하세요. Amazon EMR 또는 AWS Glue에서 Redshift 클러스터에 연결하려면 IAM 역할에 임시 IAM 자격 증명을 검색하는 데 필요한 권한이 있는지 확인하세요. 다음 목록은 IAM 역할이 자격 증명을 검색하고 Amazon S3 작업을 실행하는 데 필요한 모든 권한을 설명합니다.

- [Redshift:GetClusterCredentials](#)(프로비저닝된 Redshift 클러스터용)
- [Redshift:DescribeClusters](#)(프로비저닝된 Redshift 클러스터용)
- [Redshift:GetWorkgroup](#)(Amazon Redshift Serverless 작업 그룹)
- [Redshift:GetCredentials](#)(Amazon Redshift Serverless 작업 그룹)
- [s3:ListBucket](#)
- [s3:GetBucket](#)
- [s3:GetObject](#)
- [s3:PutObject](#)
- [s3:GetBucketLifecycleConfiguration](#)

GetClusterCredentials에 대한 자세한 내용은 [GetClusterCredentials에 대한 IAM 정책](#)을 참조하세요.

또한 COPY 및 UNLOAD 작업 중에 Amazon Redshift가 IAM 역할을 맡을 수 있는지 확인해야 합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      }
    }
  ]
}
```

```

    },
    "Action": "sts:AssumeRole"
  }
]
}

```

최신 JDBC 드라이버를 사용하는 경우 드라이버는 Amazon Redshift 자체 서명 인증서에서 ACM 인증서로의 전환을 자동으로 관리합니다. 하지만 [JDBC URL에 SSL 옵션을 지정](#)해야 합니다.

다음은 Amazon Redshift에 연결하기 위해 JDBC 드라이버 URL 및 `aws_iam_role`을 지정하는 방법의 예입니다.

```

df.write \
  .format("io.github.spark_redshift_community.spark.redshift ") \
  .option("url", "jdbc:redshift:iam://<the-rest-of-the-connection-string>") \
  .option("dbtable", "<your-table-name>") \
  .option("tempdir", "s3a://<your-bucket>/<your-directory-path>") \
  .option("aws_iam_role", "<your-aws-role-arn>") \
  .mode("error") \
  .save()

```

Amazon S3와 Spark 간의 인증

IAM 역할을 사용하여 Spark와 Amazon S3 간에 인증하는 경우 다음 방법 중 하나를 사용하세요.

- Java용 AWS SDK는 `DefaultAWSCredentialsProviderChain` 클래스에 의해 구현된 기본 자격 증명 공급자 체인을 사용하여 자동으로 AWS 자격 증명을 찾으려고 시도합니다. 자세한 내용은 [기본 자격 증명 공급자 체인 사용](#)을 참조하세요.
- [Hadoop 구성 속성](#)을 통해 AWS 키를 지정할 수 있습니다. 예를 들어 `tempdir` 구성이 `s3n://` 파일 시스템을 가리키는 경우 Hadoop XML 구성 파일에서 `fs.s3n.awsAccessKeyId` 및 `fs.s3n.awsSecretAccessKey` 속성을 설정하거나 `sc.hadoopConfiguration.set()`을 호출하여 Spark의 전역 Hadoop 구성을 변경합니다.

예를 들어 `s3n` 파일 시스템을 사용하는 경우 다음을 추가합니다.

```

sc.hadoopConfiguration.set("fs.s3n.awsAccessKeyId", "YOUR_KEY_ID")
sc.hadoopConfiguration.set("fs.s3n.awsSecretAccessKey", "YOUR_SECRET_ACCESS_KEY")

```

`s3a` 파일 시스템을 사용하는 경우 다음을 추가합니다.


```
sc.hadoopConfiguration.set("fs.s3a.access.key", "YOUR_KEY_ID")
sc.hadoopConfiguration.set("fs.s3a.secret.key", "YOUR_SECRET_ACCESS_KEY")
```

Python을 사용하는 경우 다음 작업을 사용합니다.

```
sc._jsc.hadoopConfiguration().set("fs.s3n.awsAccessKeyId", "YOUR_KEY_ID")
sc._jsc.hadoopConfiguration().set("fs.s3n.awsSecretAccessKey",
    "YOUR_SECRET_ACCESS_KEY")
```

- tempdir URL에서 인증 키를 인코딩합니다. 예를 들어 URI s3n://ACCESSKEY:SECRETKEY@bucket/path/to/temp/dir은 키 쌍(ACCESSKEY, SECRETKEY)을 인코딩합니다.

Redshift와 Amazon S3 간의 인증

쿼리에서 COPY 및 UNLOAD 명령을 사용하는 경우 사용자 대신 쿼리를 실행할 수 있도록 Amazon Redshift에 대한 Amazon S3 액세스 권한도 부여해야 합니다. 이렇게 하려면 먼저 [다른 AWS 서비스에 액세스할 수 있도록 Amazon Redshift에 권한을 부여한 다음 IAM 역할을 사용하여 COPY 및 UNLOAD 작업에 권한을 부여합니다.](#)

가장 좋은 방법은 권한 정책을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에 할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.

와의 통합AWS Secrets Manager

AWS Secrets Manager에 저장된 시크릿에서 Redshift 사용자 이름 및 암호 보안 인증 정보를 검색할 수 있습니다. Redshift 보안 인증을 자동으로 제공하려면 secret.id 파라미터를 사용하세요. Redshift 보안 인증 시크릿을 만드는 방법에 대한 자세한 내용은 [AWS Secrets Manager 데이터베이스 보안 암호 생성](#)을 참조하세요.

GroupID	ArtifactID	지원되는 리비전	설명
com.amazonaws.secretsmanager	aws-secretsmanager-jdbc	1.0.12	Java용 AWS Secrets Manager SQL 연결 라이브러리를 사용하면 Java 개발자가 AWS Secrets Manager에 저장된 암호

GroupID	ArtifactID	지원되는 리비전	설명
			호를 사용하여 SQL 데이터베이스에 쉽게 연결할 수 있습니다.

Note

인정 조항: 이 문서에는 [Apache 2.0 라이선스](#)에 따라 라이선스가 부여된 [Apache Software Foundation](#)에서 개발한 샘플 코드 및 언어가 포함되어 있습니다.

푸시다운을 통한 성능 향상

Spark 커넥터는 조건자 및 쿼리 푸시다운을 자동으로 적용하여 성능을 최적화합니다. 이 지원은 쿼리에서 지원되는 함수를 사용하는 경우 Spark 커넥터가 함수를 SQL 쿼리로 변환하고 Amazon Redshift에서 쿼리를 실행함을 의미합니다. 이 최적화로 인해 검색되는 데이터가 적어지므로 Apache Spark는 더 적은 데이터를 처리하고 더 나은 성능을 가질 수 있습니다. 기본적으로 푸시다운은 자동으로 활성화됩니다. 비활성화하려면 `autopushdown`을 `false`로 설정하세요.

```
import sqlContext.implicits._val
sample= sqlContext.read
    .format("io.github.spark_redshift_community.spark.redshift")
    .option("url",jdbcURL )
    .option("tempdir", tempS3Dir)
    .option("dbtable", "event")
    .option("autopushdown", "false")
    .load()
```

푸시다운에서는 다음 함수가 지원됩니다. 이 목록에 없는 함수를 사용하는 경우 Spark 커넥터는 Amazon Redshift 대신 Spark에서 함수를 수행하므로 성능이 최적화되지 않습니다. Spark의 전체 함수 목록은 [내장 함수](#)를 참조하세요.

- 집계 함수
 - avg
 - count
 - 최대
 - min

- sum
- stddev_samp
- stddev_pop
- var_samp
- var_pop
- 부울 연산자
 - in
 - isnull
 - isnotnull
 - contains
 - endswith
 - startswith
- 논리 연산자
 - 및
 - 또는
 - not(또는 !)
- 수학 함수
 - +
 - -
 - *
 - /
 - -(단항)
 - abs
 - acos
 - asin
 - atan
 - ceil
 - cos
 - exp
 - floor

- greatest
- least
- log10
- pi
- pow
- round
- sin
- sqrt
- tan
- 기타 함수
 - cast
 - coalesce
 - decimal
 - 다음과 같은 경우
 - in
- 관계형 연산자
 - !=
 - =
 - >
 - >=
 - <
 - <=
- 문자열 함수
 - ASCII
 - lpad
 - rpad
 - translate
 - upper
 - lower
- length

- trim
- ltrim
- rtrim
- like
- substring
- concat
- 시간 및 날짜 함수
 - add_months
 - 날짜
 - date_add
 - date_sub
 - date_trunc
 - timestamp
 - trunc
- 수학적 연산
 - CheckOverflow
 - 정밀성 증진
- 관계형 연산
 - Aliases(예: AS)
 - 사례: 언제
 - Distinct
 - InSet
 - Joins and cross joins
 - 한도
 - Unions, union all
 - ScalarSubquery
 - Sorts(오름차순 및 내림차순)
 - UnscaledValue

기타 구성 옵션

이 페이지에서 Amazon Redshift Spark 커넥터에 지정할 수 있는 옵션에 관한 설명을 확인합니다.

문자열 열의 최대 크기

Redshift는 VARCHAR(256)로 저장되는 테이블 생성 시 문자열 열을 텍스트 열로 생성합니다. 더 큰 크기를 지원하는 열을 원하면 maxlength를 사용하여 문자열 열의 최대 길이를 지정할 수 있습니다. 다음은 maxlength를 지정하는 방법의 예입니다.

```
columnLengthMap.foreach { case (colName, length) =>
  val metadata = new MetadataBuilder().putLong("maxlength", length).build()
  df = df.withColumn(colName, df(colName).as(colName, metadata))
}
```

열 유형

열 유형을 설정하려면 redshift_type 필드를 사용합니다.

```
columnTypeMap.foreach { case (colName, colType) =>
  val metadata = new MetadataBuilder().putString("redshift_type", colType).build()
  df = df.withColumn(colName, df(colName).as(colName, metadata))
}
```

열에 대한 압축 인코딩

열에서 특정 압축 인코딩을 사용하려면 인코딩 필드를 사용하세요. 지원되는 압축 인코딩의 전체 목록은 [압축 인코딩](#)을 참조하세요.

열에 대한 설명

설명을 설정하려면 description 필드를 사용합니다.

Redshift와 Amazon S3 간의 인증

기본적으로 결과는 Parquet 형식으로 Amazon S3에 업로드됩니다. 파이프를 구분된 텍스트 파일로 결과를 업로드하려면 다음 옵션을 지정합니다.

```
.option("unload_s3_format", "TEXT")
```

푸시다운 문

파라미터	필수	기본값	설명
spark.datasource.redshift.community.autopushdown.lazyMode	아니요	True	커넥터가 푸시다운 문 Redshift를 느리게 실행할지 여부를 지정합니다. true인 경우 Spark 커넥터는 쿼리를 실행하기 전에 모든 관련 모델과 정보를 검색하므로 일반적으로 성능이 향상됩니다. false인 경우 Spark 커넥터는 기본 Spark 드라이버 스레드에서 즉시 푸시다운 문을 실행하고 표현식 간에 직렬화됩니다.

커넥터 파라미터

Spark SQL의 파라미터 맵 또는 OPTIONS는 다음 설정을 지원합니다.

파라미터	필수	기본값	설명
dbtable	예, query를 지정하지 않는 한 필요	해당 사항 없음	Redshift에서 생성하거나 읽을 테이블입니다. 이 파라미터는 Redshift에 데이터를 다시 저장할 때 필요합니다.

파라미터	필수	기본값	설명
쿼리	예, dbtable을 지정하지 않는 한 필요	해당 사항 없음	Redshift에서 읽을 쿼리입니다.
user	아니요	해당 사항 없음	Redshift 사용자 이름입니다. password 파라미터와 함께 사용해야 합니다. user 및 password가 URL의 파라미터가 아닌 경우에만 유효합니다. 둘 다 사용하면 오류가 발생합니다.
password	아니요	해당 사항 없음	Redshift 암호입니다. user 파라미터와 함께 사용해야 합니다. user 및 password가 URL의 파라미터가 아닌 경우에만 유효합니다. 둘 다 사용하면 오류가 발생합니다.

파라미터	필수	기본값	설명
url	아니요	해당 사항 없음	JDBC URL입니다. 형식은 jdbc:subprotocol://host:port/database?user=username&password=password입니다. subprotocol은 로드한 JDBC 드라이버에 따라 postgresql 또는 Redshift가 될 수 있습니다. 하나의 Redshift 호환 드라이버가 클래스 경로에 있어야 하며 이 URL과 일치해야 합니다. host와 port는 Redshift 마스터 노드를 가리켜야 하므로 드라이버 애플리케이션에서 액세스할 수 있도록 보안 그룹 및/또는 VPC를 구성해야 합니다. database는 Redshift 데이터베이스 이름입니다. user 및 password는 데이터베이스에 액세스하기 위한 보안 인증 정보로 JDBC의 경우 이 URL에 포함되

파라미터	필수	기본값	설명
			어야 하며 데이터베이스 사용자는 테이블에 액세스하는 데 필요한 권한이 있어야 합니다.
aws_iam_role	IAM 역할을 사용하여 Redshift COPY/UNLOAD 작업을 승인하는 경우에만 필요	해당 사항 없음	Redshift 클러스터에 연결된 IAM 역할의 전체 지정된 ARN입니다.
forward_spark_s3_credentials	아니요	False	이 라이브러리가 Spark가 Amazon S3에 연결하는 데 사용하는 자격 증명을 자동으로 검색할지와 해당 자격 증명을 JDBC 드라이버를 통해 Redshift로 전달할지를 나타냅니다. 이러한 자격 증명은 JDBC 쿼리의 일부로 전송됩니다. 따라서 이 옵션을 사용할 때는 JDBC 연결을 통한 SSL 암호화를 사용하도록 설정하는 것이 좋습니다.
temporary_aws_access_key_id	아니요	해당 사항 없음	AWS 액세스 키입니다. S3 버킷에 대한 쓰기 권한이 있어야 합니다.

파라미터	필수	기본값	설명
temporary_aws_secret_access_key	아니요	해당 사항 없음	액세스 키에 해당하는 AWS 비밀 액세스 키입니다.
temporary_aws_session_token	아니요	해당 사항 없음	제공된 액세스 키에 해당하는 AWS 세션 토큰입니다.
tempdir	아니요	해당 사항 없음	Amazon S3에서 쓰기 가능한 위치입니다. 읽을 때 데이터를 언로드하고 쓸 때 Avro 데이터를 Redshift로 로드하는 데 사용됩니다. 일반 ETL 파이프라인의 일부로 Spark 용 Redshift 데이터 소스를 사용하는 경우 버킷에 수명 주기 정책 을 설정하고 이를 이 데이터의 임시 위치로 사용하는 것이 유용할 수 있습니다.

파라미터	필수	기본값	설명
jdbcdriver	아니요	JDBC URL의 하위 프로토콜에 의해 결정됨	사용할 JDBC 드라이버의 클래스 이름입니다. 이 클래스는 클래스 경로에 있어야 합니다. 대부분의 경우 적절한 드라이버 클래스 이름이 JDBC URL의 하위 프로토콜에 의해 자동으로 결정되어야 하므로 이 옵션을 지정할 필요가 없습니다.
diststyle	아니요	짝수	테이블을 생성할 때 사용할 Redshift 분산 스타일 입니다. 유효한 옵션은 EVEN, KEY 또는 ALL입니다. KEY를 사용할 때 distkey 옵션으로 분산 키도 설정해야 합니다.
distkey	아니요, DISTSTYLE _KEY를 사용하지 않는 한 필요하지 않음	해당 사항 없음	테이블을 생성할 때 분산 키로 사용할 테이블의 열 이름입니다.
sortkeyspec	아니요	해당 사항 없음	Redshift 정렬 키 전체 정의입니다.

파라미터	필수	기본값	설명
include_column_list	아니요	False	이 라이브러리가 스키마에서 열을 자동으로 추출하고 열 매핑 옵션 에 따라 COPY 명령에 추가해야 하는지 여부를 나타냅니다.
description	아니요	해당 사항 없음	테이블에 대한 설명입니다. 설명은 SQL COMMENT 명령으로 설정되며 대부분의 쿼리 도구에 나타납니다. 개별 열에 대한 설명을 설정하려면 description 메타데이터를 참조하세요.

파라미터	필수	기본값	설명
preactions	아니요	해당 사항 없음	COPY 명령을 로드하기 전에 실행할 SQL 명령의 세미콜론으로 구분된 목록입니다. 새 데이터를 로드하기 전에 DELETE 명령 또는 유사한 명령을 실행하는 것이 유용할 수 있습니다. 명령에 %s가 포함되어 있으면 런타임 전에 테이블 이름의 형식이 지정됩니다(스테이징 테이블을 사용하는 경우). 이 명령이 실패하면 예외로 처리됩니다. 스테이징 테이블을 사용하는 경우 사전 조치가 실패하면 변경 사항이 되돌려지고 백업 테이블이 복원됩니다.

파라미터	필수	기본값	설명
extracopyoptions	아니요	해당 사항 없음	데이터를 로드할 때 Redshift COPY 명령에 추가할 추가 옵션 목록(예: TRUNCATECOLUMNS 또는 MAXERROR n)입니다. 사용 가능한 파라미터의 전체 목록은 선택적 파라미터를 참조하세요. 이러한 옵션은 COPY 명령 끝에 추가되므로 명령 끝에 의미가 있는 옵션만 사용할 수 있습니다. 여기서는 가능한 대부분의 사용 사례를 다루어야 합니다.

파라미터	필수	기본값	설명
sse_kms_key	아니요	해당 사항 없음	AWS 기본 암호화가 아닌 Redshift UNLOAD 작업 중에 S3에서 서버 측 암호화에 사용할 AWS KMS 키 ID입니다. Redshift IAM 역할은 쓰기를 위해 KMS 키에 액세스할 수 있어야 하며 Spark IAM 역할은 읽기 작업을 위해 키에 액세스할 수 있어야 합니다. 암호화된 데이터를 읽으려면 Spark의 IAM 역할에 적절한 액세스 권한이 있는 한 변경할 필요가 없습니다 (AWS에서 처리).
tempformat	아니요	AVRO	Redshift에 쓸 때 Amazon S3에 임시 파일을 저장하는 형식입니다. 유효한 값은 AVRO, CSV, CSV GZIP(압축 CSV) 및 PARQUET입니다.
csvnullstring(실험용)	아니요	Null	CSV 임시 형식을 사용할 때 null에 쓸 문자열 값입니다. 실제 데이터에 나타나지 않는 값이어야 합니다.

파라미터	필수	기본값	설명
autopushdown	아니요	True	SQL 작업에 대한 Spark 논리 계획을 캡처하고 분석하여 조건자 및 쿼리 푸시다운을 적용할지를 나타냅니다. 작업은 SQL 쿼리로 변환된 다음 Redshift에서 실행되어 성능이 향상됩니다.
autopushdown.s3_result_cache	아니요	False	동일한 쿼리가 동일한 Spark 세션에서 다시 실행될 필요가 없도록 쿼리 SQL을 캐시하여 데이터 Amazon S3 경로 매핑을 메모리에서 언로드합니다. autopushdown이 켜져 있을 때만 지원됩니다. 캐시된 결과에 오래된 정보가 포함될 수 있으므로 읽기 및 쓰기 작업을 혼합할 때는 이 파라미터를 사용하지 않는 것이 좋습니다.

파라미터	필수	기본값	설명
unload_s3_format	아니요	Parquet	쿼리 결과를 언로드하는 형식입니다. 유효한 옵션은 쿼리 결과를 파이프로 구분된 텍스트 형식으로 언로드하도록 지정하는 Parquet 및 Text입니다.
extraunloadoptions	아니요	해당 사항 없음	Redshift UNLOAD 명령에 추가할 추가 옵션입니다. 일부 옵션은 커넥터 내에 설정된 다른 옵션과 충돌할 수 있으므로 모든 옵션의 작동이 보장되지 않습니다.
copydelay	아니요	30000	Redshift COPY 작업에 대한 재시도 사이의 지연 시간(밀리초)입니다.
copyretrycount	아니요	2	Redshift COPY 작업을 재시도할 횟수입니다.

파라미터	필수	기본값	설명
tempdir_region	아니요	해당 사항 없음	tempdir이 있는 AWS 리전입니다. 이 옵션을 설정하면 커넥터의 읽기 및 쓰기 작업 중 복사 및 언로드 작업의 일부로 이 값을 자동으로 제공할 뿐만 아니라 tempdir과의 상호 작용에 대한 커넥터 성능이 향상됩니다. 이 설정은 다음과 같은 상황에서 권장됩니다. 1) 커넥터가 AWS 외부에서 실행되는 경우 (자동 리전 검색이 실패하여 커넥터 성능에 부정적인 영향을 미침) 2) 이 설정을 사용하면 extracopy options 및 extraunload options 파라미터를 사용하여 자동으로 리전을 제공할 필요가 없으므로, tempdir이 Redshift 클러스터와 다른 리전에 있는 경우 이 파라미터를 사용하

파라미터	필수	기본값	설명
			더라도 PARQUET을 tempformat 으로 사용하는 경우 tempdir이 Redshift 클러스터와 다른 리전에 있을 수 없습니다. 3) 커넥터가 tempdir과 다른 리전에서 실행되는 경우 (tempdir 커넥터의 액세스 성능이 향상됨)
secret.id	아니요	해당 사항 없음	AWS Secrets Manager에 저장된 암호의 이름 또는 ARN입니다. 이 파라미터를 사용하여 사용자, 암호 및 DbUser 보안 인증 정보가 JDBC URL에 전달되지 않은 경우 또는 다른 옵션으로 전달된 경우에만 자동으로 Redshift 보안 인증을 제공할 수 있습니다.

파라미터	필수	기본값	설명
secret.region	아니요	해당 사항 없음	secret.id 값을 검색할 기본 AWS 리전 (예: 미국 동부(버지니아 북부))입니다. 이 리전을 지정하지 않으면 커넥터는 기본 보안 인증 제공업체 체인을 사용하여 secret.id 의 리전을 확인하려고 시도합니다. 외부에서 커넥터를 사용하는 경우와 같이 일부 경우에는 커넥터가 리전을 찾지 못할 수 있습니다. 다음과 같은 상황에서 이 설정을 사용하는 것이 좋습니다. 1) 커넥터가 AWS 외부에서 실행 중인 경우 자동 리전 검색이 실패하여 Redshift를 통한 인증이 불가능합니다. 커넥터가 secret.id 가 아닌 다른 리전에서 실행 중인 경우, 커넥터의 비밀 액세스 성능이 향상됩니다.

파라미터	필수	기본값	설명
secret.vpcEndpointUrl	아니요	해당 사항 없음	기본 보안 인증 제공 업체 체인 을 재정의 할 때 AWS Secrets Manager용 PrivateLink DNS 엔드포인트 URL입니다.
secret.vpcEndpointRegion	아니요	해당 사항 없음	기본 보안 인증 제공 업체 체인 을 재정의 할 때 AWS Secrets Manager용 PrivateLink DNS 엔드포인트 리전입니다.
jdbc.*	아니요	해당 사항 없음	기본 JDBC 드라이버에 전달할 추가 파라미터로, 와일드카드는 JDBC 파라미터의 이름입니다(예: jdbc.ssl). JDBC 드라이버에 전달되기 전에 접두사 jdbc가 제거됩니다. Redshift JDBC 드라이버에 사용할 수 있는 모든 옵션을 보려면 JDBC 드라이버 버전 2.x 구성 옵션 을 참조하세요.

파라미터	필수	기본값	설명
레이블	아니요	" "	커넥터로 쿼리를 실행할 때 쿼리 그룹 집합에 포함할 식별자입니다. 100자 이하여야 하며 모든 문자는 유효한 unicodeIdentifierParts 여야 합니다. 식별자가 100자를 초과하는 경우 초과 문자는 제거됩니다. 커넥터로 쿼리를 실행할 때 쿼리 그룹은 다음과 같은 JSON 형식의 문자열로 설정됩니다. <pre>{ "spark-redshift-connector": { "svc": "5.1.0-amzn-1-spark_3.3", "op": "Read", "l1": "" } }</pre> . 이 옵션은 l1 키의 값을 대체합니다.

Note

인정 조항: 이 문서에는 [Apache 2.0 라이선스](#)에 따라 라이선스가 부여된 [Apache Software Foundation](#)에서 개발한 샘플 코드 및 언어가 포함되어 있습니다.

지원되는 데이터 유형

Amazon Redshift의 다음 데이터 형식은 Spark 커넥터에서 지원됩니다. Amazon Redshift에서 지원되는 데이터 형식의 전체 목록은 [데이터 유형](#)을 참조하세요. 아래 테이블에 없는 데이터 형식은 Spark 커넥터에서 지원되지 않습니다.

데이터 유형	에일리어스
SMALLINT	INT2
INTEGER	INT, INT4
BIGINT	INT8
DECIMAL	NUMERIC
REAL	FLOAT4
DOUBLE PRECISION	FLOAT8, FLOAT
BOOLEAN	BOOL
CHAR	CHARACTER, NCHAR, BPCHAR
VARCHAR	CHARACTER VARYING, NVARCHAR, TEXT
DATE	
TIMESTAMP	표준 시간대가 없는 타임스탬프
TIMESTAMPTZ	표준 시간대가 있는 타임스탬프
SUPER	
TIME	표준 시간대가 없는 시간
TIMETZ	표준 시간대가 있는 시간
VARBYTE	VARBINARY, BINARY VARYING

복잡한 데이터 형식

spark 커넥터를 사용하여 ArrayType, MapType, StructType과 같은 Spark 복합 데이터 형식을 Redshift SUPER 데이터 형식 열에 읽고 쓸 수 있습니다. 읽기 작업 중에 스키마를 제공하는 경우 열의 데이터는 중첩된 형식을 포함하여 Spark에서 해당 복합 형식으로 변환됩니다. 또한 autopushdown을 활성화하면 중첩된 속성, 맵 값 및 배열 인덱스의 투영이 Redshift로 푸시다운되므로 데이터의 일부만 액세스할 때 중첩된 전체 데이터 구조를 더 이상 언로드할 필요가 없습니다.

커넥터에서 데이터 프레임을 작성할 때 MapType(StringType 사용), StructType 또는 ArrayType 형식의 모든 열은 Redshift SUPER 데이터 형식 열에 기록됩니다. 이러한 중첩된 데이터 구조를 작성할 때 tempformat 파라미터는 CSV, CSV GZIP 또는 PARQUET 형식이어야 합니다. AVRO를 사용하면 예외가 발생합니다. 키 형식이 StringType이 아닌 MapType 데이터 구조를 작성하는 경우에도 예외가 발생합니다.

StructType

다음 예제에서는 구조체를 포함하는 SUPER 데이터 형식으로 테이블을 만드는 방법을 보여 줍니다.

```
create table contains_super (a super);
```

그런 다음 커넥터를 사용하여 다음 예제와 같은 스키마를 사용하여 테이블의 SUPER 열 a에서 StringType 필드 hello를 쿼리할 수 있습니다.

```
import org.apache.spark.sql.types._

val sc = // existing SparkContext
val sqlContext = new SQLContext(sc)

val schema = StructType(StructField("a", StructType(StructField("hello",
  StringType) :: Nil)) :: Nil)

val helloDF = sqlContext.read
  .format("io.github.spark_redshift_community.spark.redshift")
  .option("url", jdbcURL )
  .option("tempdir", tempS3Dir)
  .option("dbtable", "contains_super")
  .schema(schema)
  .load().selectExpr("a.hello")
```

다음 예제는 열 a에 구조체를 작성하는 방법을 보여줍니다.

```
import org.apache.spark.sql.types._
import org.apache.spark.sql._

val sc = // existing SparkContext
val sqlContext = new SQLContext(sc)

val schema = StructType(StructField("a", StructType(StructField("hello",
  StringType) :: Nil)) :: Nil)
val data = sc.parallelize(Seq(Row(Row("world"))))
val mydf = sqlContext.createDataFrame(data, schema)

mydf.write.format("io.github.spark_redshift_community.spark.redshift").
  option("url", jdbcUrl).
  option("dbtable", tableName).
  option("tempdir", tempS3Dir).
  option("tempformat", "CSV").
  mode(SaveMode.Append).save
```

MapType

MapType을 사용하여 데이터를 표현하려는 경우 스키마에서 MapType 데이터 구조를 사용하여 맵의 키에 해당하는 값을 검색할 수 있습니다. MapType 데이터 구조의 모든 키는 문자열 형식이어야 하며 모든 값은 int와 같은 동일한 형식이어야 한다는 점에 유의합니다.

다음 예는 열 hello에서 hello 키의 값을 가져오는 방법을 보여 줍니다.

```
import org.apache.spark.sql.types._

val sc = // existing SparkContext
val sqlContext = new SQLContext(sc)

val schema = StructType(StructField("a", MapType(StringType, IntegerType)) :: Nil)

val helloDF = sqlContext.read
  .format("io.github.spark_redshift_community.spark.redshift")
  .option("url", jdbcURL )
  .option("tempdir", tempS3Dir)
  .option("dbtable", "contains_super")
  .schema(schema)
  .load().selectExpr("a['hello']")
```

ArrayType

열에 구조체 대신 배열이 포함된 경우 커넥터를 사용하여 배열의 첫 번째 요소를 쿼리할 수 있습니다.

```
import org.apache.spark.sql.types._

val sc = // existing SparkContext
val sqlContext = new SQLContext(sc)

val schema = StructType(StructField("a", ArrayType(IntegerType)):: Nil)

val helloDF = sqlContext.read
  .format("io.github.spark_redshift_community.spark.redshift")
  .option("url", jdbcURL )
  .option("tempdir", tempS3Dir)
  .option("dbtable", "contains_super")
  .schema(schema)
  .load().selectExpr("a[0]")
```

제한 사항

복잡한 데이터 형식을 Spark 커넥터와 함께 사용하면 다음과 같은 제한 사항이 있습니다.

- 중첩된 모든 구조체 필드 이름과 맵 키는 소문자여야 합니다. 대문자가 포함된 복잡한 필드 이름을 쿼리하는 경우 스키마를 생략하고 `from_json spark` 함수를 사용하여 반환된 문자열을 로컬로 변환하는 해결 방법을 시도해 볼 수 있습니다.
- 읽기 또는 쓰기 작업에 사용되는 모든 맵 필드에는 `StringType` 키만 있어야 합니다.
- 복잡한 형식을 Redshift에 쓰기 위한 임시 형식 값은 CSV, CSV GZIP 및 PARQUET 만 지원됩니다. AVRO 를 사용하려고 하면 예외가 발생합니다.

Amazon Redshift용 ODBC 드라이버 버전 2.x 연결 구성

ODBC 연결을 사용하여 여러 서드 파티 SQL 클라이언트 도구 및 애플리케이션에서 Amazon Redshift 클러스터에 연결할 수 있습니다. 클라이언트 도구가 JDBC를 지원하는 경우에는 JDBC의 구성 용이성 때문에 ODBC가 아닌 JDBC 연결 유형을 사용하는 것이 좋을 수도 있습니다. 하지만 클라이언트 도구가 JDBC를 지원하지 않는 경우, 이 섹션의 단계에 따라 클라이언트 컴퓨터 또는 Amazon EC2 인스턴스에서 ODBC 연결을 설정할 수 있습니다.

Amazon Redshift는 Linux, Windows, Mac 운영 체제용의 64비트 ODBC 드라이버를 제공합니다. 32비트 ODBC 드라이버는 단종되었습니다. 긴급 보안 패치를 제외하고 32비트 ODBC 드라이버로의 추가 업데이트는 릴리스되지 않습니다.

ODBC 드라이버 변경 사항에 대한 최신 정보는 [변경 로그](#)를 참조하세요.

주제

- [ODBC URL 가져오기](#)
- [Microsoft Windows에서 Amazon Redshift ODBC 드라이버 사용](#)
- [Linux에서 Amazon Redshift ODBC 드라이버 사용](#)
- [Apple macOS에서 Amazon Redshift ODBC 드라이버 사용](#)
- [인증 방법](#)
- [데이터 유형 변환](#)
- [ODBC 드라이버 옵션](#)
- [이전 ODBC 드라이버 버전](#)

ODBC URL 가져오기

Amazon Redshift는 Amazon Redshift 콘솔에 클러스터의 ODBC URL을 표시합니다. 이 URL에는 클라이언트 컴퓨터와 데이터베이스 간의 연결을 설정하는 데 필요한 정보가 포함되어 있습니다.

ODBC URL의 형식은 다음과 같습니다.

```
Driver={driver}; Server=endpoint_host; Database=database_name; UID=user_name;  
PWD=password; Port=port_number
```

위에 나와 있는 형식의 필드는 다음과 같은 값을 가집니다.

Field	값
<i>Driver</i>	사용할 64비트 ODBC 드라이버의 이름은 Amazon Redshift ODBC 드라이버 (x64)입니다.
<i>Server</i>	Amazon Redshift 클러스터의 엔드포인트 호스트입니다.

Field	값
<i>Database</i>	클러스터에서 생성한 데이터베이스입니다.
<i>UID</i>	데이터베이스 연결 권한을 갖는 데이터베이스 사용자 계정의 사용자 이름입니다. 이 값은 클러스터 수준 권한이 아니라 데이터베이스 수준 권한이지만, 클러스터를 시작할 때 설정한 Redshift 관리자 계정을 사용할 수 있습니다.
<i>PWD</i>	데이터베이스에 연결하기 위한 데이터베이스 사용자 계정 암호입니다.
<i>Port</i>	클러스터를 시작할 때 지정한 포트 번호입니다. 방화벽이 있는 경우에는 이 포트가 사용할 수 있게 열려있어야 합니다.

다음은 ODBC URL의 예입니다.

```
Driver={Amazon Redshift ODBC Driver (x64)}; Server=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com; Database=dev; UID=adminuser;
PWD=insert_your_admin_user_password_here; Port=5439
```

ODBC URL을 찾을 수 있는 위치에 대한 자세한 내용은 [클러스터 연결 문자열 찾기](#)를 참조하세요.

Microsoft Windows에서 Amazon Redshift ODBC 드라이버 사용

Amazon Redshift 데이터 웨어하우스에 액세스하는 클라이언트 컴퓨터에 Amazon Redshift ODBC 드라이버를 설치해야 합니다. 드라이버를 설치하는 컴퓨터마다 다음과 같은 시스템 요구 사항이 적용됩니다.

- 시스템에 대한 관리자 권한
- 시스템이 다음 시스템 요구 사항 충족
 - 다음 운영 체제 중 하나:
 - Windows 10 또는 8.1
 - Windows Server 2019, 2016 또는 2012
 - 사용 가능한 디스크 공간 100MB
 - 설치된 64비트 Windows용 Visual Studio 2015용 Visual C++ 재배포 가능 패키지 Microsoft 웹사이트의 [Visual Studio 2022용 Visual C++ 재배포 가능 패키지 다운로드](#)에서 설치 패키지를 다운로드할 수 있습니다.

Amazon Redshift ODBC 드라이버 다운로드 및 설치

Windows 운영 체제용 Amazon Redshift ODBC 드라이버를 다운로드하여 설치하려면 다음 절차를 따르세요. Amazon Redshift와 함께 사용하도록 인증되었고 특정 드라이버가 필요한 서드 파티 애플리케이션을 실행하고 있다면 다른 드라이버만 사용합니다.

ODBC 드라이버를 다운로드하고 설치하려면:

1. [64비트 ODBC 드라이버 버전 2.1.15.0](#)을 다운로드합니다.

이 드라이버의 이름은 Amazon Redshift ODBC 드라이버(x64)입니다.

2. [Amazon Redshift ODBC 드라이버 버전 2.x 라이선스](#)를 검토하세요.
3. .msi 파일을 두 번 클릭한 후 마법사의 단계에 따라 드라이버를 설치합니다.

ODBC 연결을 위한 시스템 DSN 항목 생성

ODBC 드라이버를 다운로드하여 설치한 후에는 데이터 원본 이름(DSN) 항목을 클라이언트 컴퓨터 또는 Amazon EC2 인스턴스에 추가합니다. SQL 클라이언트 도구는 이 데이터 원본을 사용하여 Amazon Redshift 데이터베이스에 연결할 수 있습니다.

사용자 DSN 대신 시스템 DSN을 생성하는 것이 좋습니다. 일부 애플리케이션은 다른 데이터베이스 사용자 계정을 사용하여 데이터를 로드하며, 다른 데이터베이스 사용자 계정으로 생성된 사용자 DSN을 감지하지 못할 수 있습니다.

Note

AWS Identity and Access Management(IAM) 자격 증명 또는 자격 증명 공급자(IdP) 자격 증명을 사용한 인증의 경우 추가 단계가 필요합니다. 자세한 내용은 [IAM 자격 증명을 사용하도록 JDBC 또는 ODBC 연결 구성](#)을 참조하세요.

ODBC 연결을 위한 시스템 DSN 항목을 생성하려면:

1. 시작 메뉴에서 'ODBC 데이터 소스'를 입력합니다. ODBC Data Sources(ODBC 데이터 소스)를 선택합니다.

Amazon Redshift에 연결하는 데 사용하는 클라이언트 애플리케이션과 동일한 비트를 가진 ODBC 데이터 원본 관리자를 선택해야 합니다.

2. ODBC Data Source Administrator(ODBC 데이터 소스 관리자)에서 Driver(드라이버) 탭을 선택하고 Amazon Redshift ODBC Driver (x64) 드라이버 폴더를 찾습니다.
3. 해당 컴퓨터의 모든 사용자가 이용할 수 있도록 드라이버를 구성하려면 시스템 DSN(System DSN) 탭을 선택하고, 본인의 데이터베이스 사용자 계정에서만 이용할 수 있도록 드라이버를 구성하려면 사용자 DSN(User DSN) 탭을 선택합니다.
4. 추가를 선택합니다. 새 데이터 원본 생성(Create New Data Source) 창이 열립니다.
5. Amazon Redshift ODBC 드라이버(x64)를 선택하고 Finish(마침)를 선택합니다. Amazon Redshift ODBC 드라이버 DSN 설정(Amazon Redshift ODBC Driver DSN Setup) 창이 열립니다.
6. Connection Settings(연결 설정) 섹션에 다음 정보를 입력합니다.

- 데이터 소스 이름

데이터 소스 이름을 입력합니다. 예를 들어 Amazon Redshift 시작 안내서에 따른 경우 `exampleclusterdsn`을 입력하면 이 DSN과 연결할 클러스터를 기억하기 쉽습니다.

- Server

Amazon Redshift 클러스터의 엔드포인트 호스트를 지정합니다. 이 정보는 Amazon Redshift 콘솔의 클러스터 세부 정보 페이지에서 찾을 수 있습니다. 자세한 내용은 [Amazon Redshift에서 연결 구성](#)을 참조하세요.

- Port

데이터베이스가 사용하는 포트 번호를 입력합니다. 클러스터를 생성, 수정 또는 마이그레이션할 때 선택한 포트에 따라 선택한 포트에 대한 액세스를 허용합니다.

- 데이터베이스

Amazon Redshift 데이터베이스의 이름을 입력합니다. 데이터베이스 이름을 지정하지 않고 클러스터를 시작한 경우 `dev`를 입력합니다. 그렇지 않으면 시작 프로세스 중에 선택한 이름을 사용합니다. Amazon Redshift 시작 안내서에 따른 경우 `dev`를 입력합니다.

7. Authentication(인증) 섹션에서 표준 또는 IAM 인증을 구성하는 구성 옵션을 지정합니다.

8. SSL Options(SSL 옵션)을 선택하고 다음의 값을 지정합니다.

- 인증 모드

SSL(Secure Sockets Layer)의 처리 모드를 선택합니다. 테스트 환경에서는 `prefer`를 사용할 수 있습니다. 그러나 프로덕션 환경 및 안전한 데이터 교환이 필요한 경우 `verify-ca` 또는 `verify-full`을 사용합니다.

- 최소 TLS

선택 사항으로, 드라이버가 데이터 스토어에서 연결을 암호화하는 데 사용하도록 허용하는 TLS/SSL의 최소 버전을 선택합니다. 예를 들어 TLS 1.2를 지정하면 연결을 암호화하는 데 TLS 1.1을 사용할 수 없습니다. 기본 버전은 TLS 1.2입니다.

9. Proxy(프록시) 탭에서 프록시 연결 설정을 지정합니다.

10.Cursor(커서) 탭에서 쿼리 결과를 SQL 클라이언트 도구 또는 애플리케이션에 반환하는 방법에 대한 옵션을 지정합니다.

11고급 옵션에서 logLevel, logPath, compression 및 기타 옵션에 대한 값을 지정합니다.

12테스트를 선택합니다. 클라이언트 컴퓨터가 Amazon Redshift 데이터베이스에 연결할 수 있으면 Connection successful(연결 성공) 메시지가 표시됩니다. 클라이언트 컴퓨터가 데이터베이스에 연결하지 못하는 경우에는 로그 파일을 생성하고 AWS Support 팀에 문의하여 문제를 해결할 수 있습니다. 로그 생성에 대한 자세한 내용은 (LINK)를 참조하세요.

13.확인을 선택합니다.

Linux에서 Amazon Redshift ODBC 드라이버 사용

Amazon Redshift 데이터 웨어하우스에 액세스하는 클라이언트 컴퓨터에 Amazon Redshift ODBC 드라이버를 설치해야 합니다. 드라이버를 설치하는 컴퓨터마다 다음과 같은 시스템 요구 사항이 적용됩니다.

- 시스템에 대한 루트 액세스 권한
- 다음 배포판 중 하나:
 - Red Hat® Enterprise Linux®(RHEL) 8 이상
 - CentOS 8 이상
- 사용 가능한 디스크 공간 150MB
- unixODBC 2.2.14 이상
- glibc 2.26 이상

Amazon Redshift ODBC 드라이버 다운로드 및 설치

Linux용 Amazon Redshift ODBC 드라이버 버전 2.x를 다운로드하고 설치하려면:

1. 다음 드라이버를 다운로드합니다.

- [x86 64비트 RPM 드라이버 버전 2.1.15.0](#)
- [ARM 64비트 RPM 드라이버 버전 2.1.15.0](#)

 Note

32비트 ODBC 드라이버는 단종되었습니다. 긴급 보안 패치를 제외하고 추가 업데이트는 릴리스되지 않습니다.

- 패키지를 다운로드한 위치로 이동한 다음, 다음 명령 중 하나를 실행합니다. Linux 배포판에 따라 올바른 명령을 사용하세요.

RHEL 및 CentOS 운영 체제에서 다음 명령을 실행합니다.

```
yum --nogpgcheck localinstall RPMFileName
```

RPMFileName을 RPM 패키지 파일 이름으로 바꿉니다. 예를 들어 다음은 64비트 드라이버를 설치하는 명령입니다.

```
yum --nogpgcheck localinstall AmazonRedshiftODBC-64-bit-2.x.xx.xxxx.x86_64.rpm
```

ODBC 드라이버 관리자를 사용하여 ODBC 드라이버 구성

Linux에서 ODBC 드라이버 관리자를 사용하여 ODBC 연결 설정을 구성합니다. ODBC 드라이버 관리자는 구성 파일을 사용하여 ODBC 데이터 원본과 드라이버를 정의 및 구성합니다. 또한 사용하는 운영 체제에 따라 ODBC 드라이버 관리자도 달라집니다.

unixODBC 드라이버 관리자를 사용하여 ODBC 드라이버 구성

Amazon Redshift ODBC 드라이버를 구성하려면 다음 파일이 필요합니다.

- amazon.redshiftodbc.ini
- odbc.ini
- odbcinst.ini

기본 위치에 드라이버를 설치한 경우에는 amazon.redshiftodbc.ini 구성 파일이 /opt/amazon/redshiftodbcx64에 있습니다.

또한 `/opt/amazon/redshiftodbcx64`에 샘플 `odbc.ini` 및 `odbcinst.ini` 파일이 있습니다. 이러한 파일을 Amazon Redshift ODBC 드라이버 및 데이터 원본 이름(DSN)을 구성하는 예제로 사용할 수 있습니다.

Amazon Redshift ODBC 드라이버 설치 디렉터리를 구성 파일용으로 사용하는 것은 바람직하지 않습니다. 설치된 디렉터리의 샘플 파일은 오직 예를 제공하기 위한 용도일 뿐입니다. 나중에 Amazon Redshift ODBC 드라이버를 다시 설치하거나 새 버전으로 업그레이드하면 설치 디렉터리를 덮어씁니다. 설치 디렉터리의 파일에 대한 변경 사항은 손실됩니다.

이를 방지하려면 `amazon.redshiftodbc.ini` 파일을 설치 디렉터리가 아닌 다른 디렉터리에 복사합니다. 이 파일을 사용자의 홈 디렉터리로 복사하는 경우에는 파일 이름 앞에 마침표(.)를 추가하여 숨김 파일로 지정하세요.

`odbc.ini` 및 `odbcinst.ini` 파일의 경우 사용자의 홈 디렉터리에 있는 구성 파일을 사용하거나 다른 디렉터리에 새 버전을 생성합니다. 기본적으로 Linux 운영 체제는 사용자의 홈 디렉터리(`/home/$USER` 또는 `~/.`)에 `odbc.ini` 파일과 `odbcinst.ini` 파일이 있어야 합니다. 이러한 기본 파일은 숨김 파일이기 때문에 파일 이름 앞에 마침표(.)가 있습니다. 이러한 파일은 `-a` 플래그를 사용하여 디렉터리 내용을 나열하는 경우에만 표시됩니다.

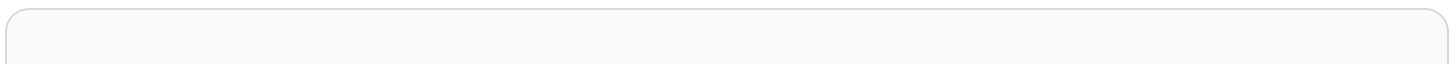
`odbc.ini` 및 `odbcinst.ini` 파일에 대해 어떤 옵션을 선택하든지 파일을 수정하여 드라이버 및 DSN 구성 정보를 추가합니다. 새로운 파일을 생성할 때도 마찬가지로 환경 변수를 설정하여 구성 파일의 위치를 지정해야 합니다.

기본적으로 ODBC 드라이버 관리자는 홈 디렉터리에 있는 숨겨진 버전의 `odbc.ini` 및 `odbcinst.ini` 구성 파일(`.odbc.ini` 및 `.odbcinst.ini`)을 사용하도록 구성됩니다. 또한 드라이버 설치 디렉터리에 있는 `amazon.redshiftodbc.ini` 파일을 사용하도록 구성됩니다. 이러한 구성 파일을 다른 위치에 저장하는 경우 드라이버 관리자가 파일을 찾을 수 있도록 다음에 설명된 환경 변수를 설정합니다.

unixODBC 사용하는 경우 다음과 같이 합니다.

- ODBCINI를 `odbc.ini` 파일의 전체 경로 및 파일 이름으로 설정합니다.
- ODBCSYSINI를 `odbcinst.ini` 파일이 포함된 디렉토리의 전체 경로로 설정합니다.
- AMAZONREDSHIFTODBCINI를 `amazon.redshiftodbc.ini` 파일의 전체 경로 및 파일 이름으로 설정합니다.

다음은 위의 값을 설정하는 예입니다.



```
export ODBCINI=/usr/local/odbc/odbc.ini
export ODBCSYSINI=/usr/local/odbc
export AMAZONREDSHIFTODBCINI=/etc/amazon.redshiftodbc.ini
```

Linux에서 데이터 소스 이름(DSN)을 사용하여 연결 구성

데이터 소스 이름(DSN)을 사용하여 데이터 스토어에 연결할 경우 데이터 소스 이름(DSN)을 정의하도록 `odbc.ini` 파일을 구성합니다. `odbc.ini` 파일의 속성을 설정하여 데이터 저장소에 대한 연결 정보를 지정하는 DSN을 생성합니다.

Linux 운영 체제에서는 다음 형식을 사용합니다.

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file
Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

다음 예에서는 Linux 운영 체제에서 64비트 ODBC 드라이버를 사용하는 `odbc.ini`에 대한 구성을 보여줍니다.

```
[ODBC Data Sources]
Amazon_Redshift_x64=Amazon Redshift ODBC Driver (x64)

[Amazon_Redshift_x64]
Driver=/opt/amazon/redshiftodbcx64/librsodbc64.so
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932Database=dev
locale=en-US
```

Linux에서 DSN 없이 연결 구성

DSN이 없는 연결을 통해 데이터 저장소에 연결하려면 `odbcinst.ini` 파일에 드라이버를 정의합니다. 그런 다음 애플리케이션에 DSN이 없는 연결 문자열을 제공합니다.

Linux 운영 체제에서는 다음 형식을 사용합니다.

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/driver_file

...
```

다음 예에서는 Linux 운영 체제에서 64비트 ODBC 드라이버를 사용하는 `odbcinst.ini`에 대한 구성을 보여줍니다.

```
[ODBC Drivers]
Amazon Redshift ODBC Driver (x64)=Installed

[Amazon Redshift ODBC Driver (x64)]
Description=Amazon Redshift ODBC Driver (64-bit)
Driver=/opt/amazon/redshiftodbcx64/librsodbc64.so
```

Apple macOS에서 Amazon Redshift ODBC 드라이버 사용

Amazon Redshift 데이터 웨어하우스에 액세스하는 클라이언트 컴퓨터에 Amazon Redshift ODBC 드라이버를 설치해야 합니다. 드라이버를 설치하는 컴퓨터마다 다음과 같은 시스템 요구 사항이 적용됩니다.

- 시스템에 대한 루트 액세스 권한
- Apple macOS 시스템 요구 사항:
 - 64비트 버전의 Apple macOS 버전 11.7 이상(예: Apple macOS Big Sur, Monterey, Ventura 이상)이 필요합니다. Redshift ODBC 드라이버는 64비트 클라이언트 애플리케이션만 지원합니다.
 - 사용 가능한 디스크 공간 150MB
 - 드라이버는 iODBC 3.52.9 이상 또는 unixODBC 2.3.7 이상으로 빌드된 애플리케이션을 지원합니다.

Amazon Redshift ODBC 드라이버 다운로드 및 설치

Apple macOS용 Amazon Redshift ODBC 드라이버를 다운로드하여 설치하려면 다음 절차를 따르세요. Amazon Redshift와 함께 사용하도록 인증되었고 특정 드라이버가 필요한 서드 파티 애플리케이션을 실행하고 있다면 다른 드라이버만 사용합니다.

ODBC 드라이버를 다운로드하고 설치하려면:

1. [64비트 ODBC 드라이버 버전 2.1.15.0](#)을 다운로드합니다.

이 드라이버는 x86_64 아키텍처와 arm64 아키텍처 모두에서 지원됩니다. 이 드라이버의 이름은 Amazon Redshift ODBC 드라이버(x64)입니다.

2. [Amazon Redshift ODBC 드라이버 버전 2.x 라이선스](#)를 검토하세요.
3. .pkg 파일을 두 번 클릭한 후 마법사의 단계에 따라 드라이버를 설치합니다. 또는 다음 명령을 실행합니다.

```
sudo installer -pkg PKGFileName -target /
```

PKGFileName을 pkg 패키지 파일 이름으로 바꿉니다. 예를 들어 다음은 64비트 드라이버를 설치하는 명령입니다.

```
sudo installer -pkg ./AmazonRedshiftODBC-64-bit.X.X.XX.X.universal.pkg -target /
```

ODBC 드라이버 관리자를 사용하여 ODBC 드라이버 구성

Mac에서 ODBC 드라이버 관리자를 사용하여 ODBC 연결 설정을 구성합니다. ODBC 드라이버 관리자는 구성 파일을 사용하여 ODBC 데이터 원본과 드라이버를 정의 및 구성합니다. 또한 사용하는 운영 체제에 따라 ODBC 드라이버 관리자도 달라집니다.

iODBC 또는 unixODBC 드라이버 관리자를 사용하여 ODBC 드라이버 구성

Amazon Redshift ODBC 드라이버를 구성하려면 다음 파일이 필요합니다.

- amazon.redshiftodbc.ini
- odbc.ini
- odbcinst.ini

기본 위치에 드라이버를 설치한 경우에는 `amazon.redshiftodbc.ini` 구성 파일이 `/opt/amazon/redshiftodbcx64`에 있습니다.

또한 `/opt/amazon/redshiftodbcx64`에 샘플 `odbc.ini` 및 `odbcinst.ini` 파일이 있습니다. 이러한 파일을 Amazon Redshift ODBC 드라이버 및 데이터 원본 이름(DSN)을 구성하는 예제로 사용할 수 있습니다. 설치된 디렉터리의 샘플 파일은 오직 예를 제공하기 위한 용도일 뿐입니다.

Amazon Redshift ODBC 드라이버 설치 디렉터리를 구성 파일용으로 사용하는 것은 바람직하지 않습니다. 나중에 Amazon Redshift ODBC 드라이버를 다시 설치하거나 새 버전으로 업그레이드하면 설치 디렉터리를 덮어씁니다. 설치 디렉터리의 파일에 대한 변경 사항은 손실됩니다.

이를 방지하려면 `odbc.ini`, `odbcinst.ini` 및 `amazon.redshiftodbc.ini` 파일을 설치 디렉터리가 아닌 다른 디렉터리에 복사합니다. 이러한 파일을 사용자의 홈 디렉터리로 복사하는 경우에는 파일 이름 앞에 마침표(.)를 추가하여 숨김 파일로 지정하세요.

파일을 수정하여 DSN 구성 정보를 추가합니다. 새로운 파일을 생성할 때도 마찬가지로 환경 변수를 설정하여 구성 파일의 위치를 지정해야 합니다.

다음은 이러한 환경 변수 설정의 예입니다.

```
export ODBCINI=/Library/ODBC/odbc.ini
export ODBCSYSINI=/Library/ODBC
export ODBCINSTINI=${ODBCSYSINI}/odbcinst.ini
```

명령줄 애플리케이션의 경우: 셸 시작 파일(예: `~/.bash_profile` 또는 `~/.zshrc`)에 내보내기 명령을 추가합니다.

지원되는 드라이버 관리자 버전은 [여기](#)를 참조하세요.

Apple macOS에서 데이터 소스 이름(DSN)을 사용하여 연결 구성

데이터 소스 이름(DSN)을 사용하여 데이터 스토어에 연결할 경우 데이터 소스 이름(DSN)을 정의하도록 `odbc.ini` 파일을 구성합니다. `odbc.ini` 파일의 속성을 설정하여 Redshift 데이터 웨어하우스에 대한 연결 정보를 지정하는 DSN을 생성합니다.

Apple macOS에서는 다음 형식을 사용합니다.

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
```

```
Driver=path/driver_file
Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

다음 예에서는 Apple macOS에서 64비트 ODBC 드라이버를 사용하는 `odbc.ini`에 대한 구성을 보여줍니다.

```
[ODBC Data Sources]
Amazon_Redshift_x64=Amazon Redshift ODBC Driver (x64)

[Amazon_Redshift_x64]
Driver=/opt/amazon/redshiftodbcx64/librsodbc64.dylib
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

Apple macOS에서 DSN 없이 연결 구성

DSN이 없는 연결을 통해 Redshift 데이터 웨어하우스에 연결하려면 `odbcinst.ini` 파일에 드라이버를 정의합니다. 그런 다음 애플리케이션에 DSN이 없는 연결 문자열을 제공합니다.

Apple macOS에서는 다음 형식을 사용합니다.

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/driver_file
...
```

다음 예에서는 Apple macOS에서 64비트 ODBC 드라이버를 사용하는 `odbcinst.ini`에 대한 구성을 보여줍니다.

```
[ODBC Drivers]
Amazon Redshift ODBC Driver (x64)=Installed
```

```
[Amazon Redshift ODBC Driver (x64)]
Description=Amazon Redshift ODBC Driver (64-bit)
Driver=/opt/amazon/redshiftodbcx64/librsodbc64.dylib
```

인증 방법

무단 액세스로부터 데이터를 보호하기 위해 Amazon Redshift 데이터 스토어는 사용자 자격 증명을 사용하여 모든 연결을 인증할 것을 요구합니다.

다음 표는 Amazon Redshift ODBC 드라이버 버전 2.x에 연결하는 데 사용할 수 있는 각 인증 방법별로 필수 및 선택적 연결 옵션을 보여줍니다.

인증 방법	필수	선택
표준	- Host - Port - 데이터베이스 - UID - 암호	
IAM 프로필	- Host - Port - 데이터베이스 - IAM - 프로필	- ClusterID - 리전(Region) - AutoCreate - EndpointURL - StsEndpointURL - InstanceProfile  Note 별도로 설정되지 않는 경우 호스트(Host)에서 ClusterID와 리전(Region)을 설정해야 합니다.
IAM 자격 증명	Host	ClusterID

인증 방법	필수	선택
	• Port • 데이터베이스 • IAM • AccessKeyID • SecretAccessKey	• 리전(Region) • AutoCreate • EndpointURL • StsEndpointURL • SessionToken • UID

 **Note**

별도로 설정되지 않는 경우 호스트(Host)에서 ClusterID와 리전(Region)을 설정해야 합니다.

인증 방법	필수	선택
AD FS	• Host • Port • 데이터베이스 • IAM • plugin_name • UID • 암호 • IdP_Host • IdP_Port	• ClusterID • 리전(Region) • AutoCreate • EndpointUrl • StsEndpointUrl • Preferred_Role • loginToRp • SSL_Insecure  Note 별도로 설정되지 않는 경우 호스트(Host)에서 ClusterID와 리전(Region)을 설정해야 합니다.

인증 방법	필수	선택
Azure AD	- Host - Port - 데이터베이스 - IAM - plugin_name - UID - 암호 - IdP_Tenant - Client_ID - Client_Secret	- ClusterID - 리전(Region) - AutoCreate - EndpointUrl - StsEndpointUrl - Preferred_Role - dbgroups_filter  Note 별도로 설정되지 않는 경우 호스트(Host)에서 ClusterID와 리전(Region)을 설정해야 합니다.
JWT	- Host - Port - 데이터베이스 - IAM - plugin_name - web_identity_token	provider_name

인증 방법	필수	선택
Okta	- Host - Port - 데이터베이스 - IAM - plugin_name - UID - 암호 - IdP_Host - App_Name - App_ID	- ClusterID - 리전(Region) - AutoCreate - EndpointUrl - StsEndpointUrl - Preferred_Role

 **Note**

별도로 설정되지 않는 경우 호스트(Host)에서 ClusterID와 리전(Region)을 설정해야 합니다.

인증 방법	필수	선택
Ping Federate	- Host - Port - 데이터베이스 - IAM - plugin_name - UID - 암호 - IdP_Host - IdP_Port	- ClusterID - 리전(Region) - AutoCreate - EndpointUrl - StsEndpointUrl - Preferred_Role - SSL_Insecure - partner_spid  Note 별도로 설정되지 않는 경우 호스트(Host)에서 ClusterID와 리전(Region)을 설정해야 합니다.

인증 방법	필수	선택
브라우저 Azure AD	• Host • Port • 데이터베이스 • IAM • plugin_name • IdP_Tenant • Client_ID • UID	• ClusterID • 리전(Region) • AutoCreate • EndpointUrl • StsEndpointUrl • Preferred_Role • dbgroups_filter • IdP_Response_Timeout • listen_port

 **Note**

별도로 설정되지 않는 경우 호스트(Host)에서 ClusterID와 리전(Region)을 설정해야 합니다.

인증 방법	필수	선택
브라우저 SAML	- Host - Port - 데이터베이스 - IAM - plugin_name - login_url - UID	- ClusterID - 리전(Region) - AutoCreate - EndpointUrl - StsEndpointUrl - Preferred_Role - dbgroups_filter - IdP_Response_Timeout - listen_port  Note 별도로 설정되지 않는 경우 호스트(Host)에서 ClusterID와 리전(Region)을 설정해야 합니다.
인증 프로필	- Host - Port - 데이터베이스 - AccessKeyID - SecretAccessKey	

인증 방법	필수	선택
브라우저 Azure AD OAUTH2	- Host - Port - 데이터베이스 - IAM - plugin_name - IdP_Tenant - Client_ID - UID	- ClusterID - 리전(Region) - EndpointUrl - IdP_Response_Timeout - listen_port - 범위 - provider_name  Note 별도로 설정되지 않는 경우 호스트(Host)에서 ClusterID와 리전(Region)을 설정해야 합니다.
AWS IAM Identity Center	- Host - 데이터베이스 - plugin_name - idc_region - issuer_url	- idc_client_display_name - idp_response_timeout - listen_port

외부 보안 인증 서비스 사용

Amazon Redshift ODBC 드라이버의 Windows 버전은 AD FS, Azure AD 및 Okta에 대한 기본 제공 지원 외에, 다른 보안 인증 서비스에 대한 지원도 제공합니다. 이 드라이버는 선택한 SAML 기반 보안 인증 공급자 플러그인을 사용하여 연결을 인증할 수 있습니다.

Windows에서 외부 보안 인증 서비스를 구성하려면

1. 필요에 따라 보안 인증 공급자 플러그인 및 기타 인증 파라미터를 지정하는 IAM 프로필을 생성합니다. 이 프로필은 ASCII로 인코딩해야 하며 다음 키-값 페어를 포함해야 합니다. 여기서 PluginPath는 플러그인 애플리케이션의 전체 경로입니다.

```
plugin_name = PluginPath
```

예:

```
plugin_name = C:\Users\kjson\myapp\CredServiceApp.exe
```

프로필을 생성하는 방법에 대한 자세한 내용은 Amazon Redshift 클러스터 관리 가이드에서 [구성 프로필 사용](#)을 참조하세요.

2. 이 프로필을 사용하도록 드라이버를 구성합니다. 이 드라이버는 프로필에 지정된 인증 설정을 감지하고 사용합니다.

데이터 유형 변환

Amazon Redshift ODBC 드라이버 버전 2.x는 Amazon Redshift 및 SQL 데이터 유형 간 변환으로 많은 일반 데이터 유형을 지원합니다.

다음 표에는 지원되는 데이터 형식 매핑이 나와 있습니다.

Amazon Redshift 형식	SQL 형식
BIGINT	SQL_BIGINT
BOOLEAN	SQL_BIT
CHAR	SQL_CHAR
DATE	SQL_TYPE_DATE
DECIMAL	SQL_NUMERIC
DOUBLE PRECISION	SQL_DOUBLE
GEOGRAPHY	SQL_LONGVARBINARY
GEOMETRY	SQL_LONGVARBINARY

Amazon Redshift 형식	SQL 형식
INTEGER	SQL_INTEGER
REAL	SQL_REAL
SMALLINT	SQL_SMALLINT
SUPER	SQL_LONGVARCHAR
TEXT	SQL_LONGVARCHAR
TIME	SQL_TYPE_TIME
TIMETZ	SQL_TYPE_TIME
TIMESTAMP	SQL_TYPE_TIMESTAMP
TIMESTAMPTZ	SQL_TYPE_TIMESTAMP
VARBYTE	SQL_LONGVARBINARY
VARCHAR	SQL_VARCHAR

ODBC 드라이버 옵션

드라이버 구성 옵션을 사용하여 Amazon Redshift ODBC 드라이버의 동작을 제어할 수 있습니다. 드라이버 옵션은 대소문자를 구분하지 않습니다.

Microsoft Windows에서는 일반적으로 데이터 원본 이름(DSN)을 구성할 때 드라이버 옵션을 설정합니다. 프로그래밍 방식으로 연결할 때 또는 HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI*your_DSN*에서 레지스트리 키를 추가하거나 변경하여 연결 문자열에 드라이버 옵션을 설정할 수도 있습니다.

Linux에서는 `odbc.ini` 및 `amazon.redshiftodbc.ini` 파일에서 드라이버 구성 옵션을 설정합니다. `amazon.redshiftodbc.ini` 파일에서 설정하는 구성 옵션이 모든 연결에 적용됩니다. 반대로 `odbc.ini` 파일에서 설정하는 구성 옵션은 연결에 따라 다릅니다. `odbc.ini`에서 설정하는 구성 옵션이 `amazon.redshiftodbc.ini`에서 설정하는 구성 옵션보다 우선합니다.

다음은 Amazon Redshift ODBC 드라이버 버전 2.x 드라이버에 지정할 수 있는 옵션에 대한 설명입니다.

AccessKeyID

- 기본값 – 없음
- 데이터 형식 – 문자열

사용자 또는 역할에 대한 IAM 액세스 키입니다. 이 파라미터를 설정할 경우 SecretAccessKey도 지정해야 합니다

이 파라미터는 선택 사항입니다.

app_id

- 기본값 – 없음
- 데이터 형식 – 문자열

Amazon Redshift 애플리케이션과 연결된 Okta 제공 고유 ID입니다.

이 파라미터는 선택 사항입니다.

ApplicationName

- 기본값 – 없음
- 데이터 유형 – 문자열

감사 목적으로 Amazon Redshift에 전달할 클라이언트 애플리케이션의 이름입니다. 입력하는 애플리케이션 이름은 [SYS_CONNECTION_LOG](#) 테이블의 'application_name' 열에 나타납니다. 이렇게 하면 문제를 디버깅할 때 연결 소스를 추적하고 문제를 해결할 수 있습니다.

이 파라미터는 선택 사항입니다.

app_name

- 기본값 – 없음
- 데이터 형식 – 문자열

Amazon Redshift에 대한 연결을 인증하는 데 사용하는 Okta 애플리케이션의 이름입니다.

이 파라미터는 선택 사항입니다.

AuthProfile

- 기본값 – 없음
- 데이터 형식 – 문자열

연결 설정을 관리하는 데 사용되는 인증 프로필입니다. 이 파라미터를 설정할 경우 AccessKeyId와 SecretAccessKey도 설정해야 합니다

이 파라미터는 선택 사항입니다.

AuthType

- 기본값 - 표준(Standard)
- 데이터 형식 – 문자열

이 옵션은 Amazon Redshift ODBC 드라이버 DSN 설정 대화 상자를 사용하여 DSN을 구성할 때 드라이버가 사용하는 인증 모드를 지정합니다.

- 표준(Standard): Amazon Redshift 사용자 이름과 암호를 사용하는 표준 인증입니다.
- AWS Profile(프로필): 프로필을 사용하는 IAM 인증입니다.
- AWS IAM Credentials(IAM 보안 인증 정보): IAM 보안 인증 정보를 사용하는 IAM 인증입니다.
- 자격 증명 공급자(Identity Provider): AD FS: Active Directory Federation Services(AD FS)를 사용하는 IAM 인증입니다.
- ID 제공업체: 인증 플러그인: AWS IAM Identity Center 토큰 또는 AWS IAM Identity Center에 연결된 웹 ID 제공업체의 OpenID Connect(OIDC) JSON 기반 자격 증명 토큰(JWT)을 수락하는 권한 부여 플러그인입니다.
- 자격 증명 공급자(Identity Provider): Azure AD: Azure AD 포털을 사용한 IAM 인증입니다.
- 자격 증명 공급자(Identity Provider): JWT: JSON 웹 토큰(JWT)을 사용한 IAM 인증입니다.
- 자격 증명 공급자(Identity Provider): Okta: Okta를 사용한 IAM 인증입니다.
- 자격 증명 공급자(Identity Provider): PingFederate: PingFederate를 사용한 IAM 인증입니다.

이 옵션은 Windows 드라이버에서 Amazon Redshift ODBC 드라이버 DSN 설정 대화 상자를 사용하여 DSN을 구성하는 경우에만 사용할 수 있습니다. 연결 문자열 또는 Windows 이외의 시스템을 사용하여 연결을 구성하면, 드라이버가 지정된 자격 증명에 따라 표준, AWS 프로필 또는 AWS IAM 보안 인증

중 어떤 인증 방법을 사용할지 자동으로 결정합니다. 자격 증명 공급자를 사용하려면 `plugin_name` 속성을 설정해야 합니다.

이 파라미터는 필수 사항입니다.

AutoCreate

- 기본값 - 0
- 데이터 형식 - 부울

지정된 사용자가 존재하지 않을 경우 드라이버가 새 사용자를 생성할지 여부를 지정하는 부울 값입니다.

- 1 | TRUE: UID로 지정된 사용자가 없는 경우 드라이버가 새 사용자를 생성합니다.
- 0 | FALSE: 드라이버가 새 사용자를 생성하지 않습니다. 지정된 사용자가 없으면 인증에 실패합니다.

이 파라미터는 선택 사항입니다.

CaFile

- 기본값 - 없음
- 데이터 형식 - 문자열

일부 IAM 인증 방식에 사용되는 CA 인증서 파일의 파일 경로입니다.

이 파라미터는 Linux에서만 사용 가능합니다.

이 파라미터는 선택 사항입니다.

client_id

- 기본값 - 없음
- 데이터 형식 - 문자열

Azure AD에서 Amazon Redshift 애플리케이션과 연결된 클라이언트 ID입니다.

Azure AD 서비스를 통해 인증하는 경우 이 파라미터는 필수입니다.

client_secret

- 기본값 – 없음
- 데이터 형식 – 문자열

Azure AD에서 Amazon Redshift 애플리케이션과 연결된 비밀 키입니다.

Azure AD 서비스를 통해 인증하는 경우 이 파라미터는 필수입니다.

ClusterId

- 기본값 – 없음
- 데이터 형식 – 문자열

연결하려는 Amazon Redshift 클러스터의 이름입니다. IAM 인증에 사용됩니다. 이 클러스터 ID는 Server 파라미터에 지정되어 있지 않습니다.

이 파라미터는 선택 사항입니다.

압축

- 기본값 - 끄기
- 데이터 형식 – 문자열

Amazon Redshift 서버와 클라이언트 또는 드라이버 간의 유선 프로토콜 통신에 사용되는 압축 방법입니다.

다음 값을 지정할 수 있습니다.

- lz4: Amazon Redshift와의 유선 프로토콜 통신에 사용되는 압축 방법을 lz4로 설정합니다.
- zstd: Amazon Redshift와의 유선 프로토콜 통신에 사용되는 압축 방법을 zstd로 설정합니다.
- off: Amazon Redshift와의 유선 프로토콜 통신에 압축을 사용하지 않습니다.

이 파라미터는 선택 사항입니다.

데이터베이스

- 기본값 – 없음

- 데이터 형식 – 문자열

액세스할 Amazon Redshift 데이터베이스의 이름입니다.

이 파라미터는 필수 사항입니다.

DatabaseMetadataCurrentDbOnly

- 기본값 – 1
- 데이터 형식 – 부울

드라이버가 여러 데이터베이스 및 클러스터의 메타데이터를 반환하는지 여부를 지정하는 부울 값입니다.

- 1 | TRUE: 드라이버가 현재 데이터베이스의 메타데이터만 반환합니다.
- 0 | FALSE: 드라이버가 여러 Amazon Redshift 데이터베이스 및 클러스터의 메타데이터를 반환합니다.

이 파라미터는 선택 사항입니다.

dbgroups_filter

- 기본값 – 없음
- 데이터 형식 – 문자열

Azure, 브라우저 Azure 및 브라우저 SAML 인증 유형을 사용할 경우 Amazon Redshift에 대한 SAML 응답에서 수신된 DbGroups를 필터링하기 위해 지정할 수 있는 정규식입니다.

이 파라미터는 선택 사항입니다.

드라이버

- 기본값 - Amazon Redshift ODBC 드라이버(x64)
- 데이터 형식 – 문자열

드라이버의 이름입니다. 지원되는 유일한 값은 Amazon Redshift ODBC 드라이버(x64)(Amazon Redshift ODBC Driver (x64))입니다.

DSN를 설정하지 않을 경우 이 파라미터가 필요합니다.

DSN

- 기본값 – 없음
- 데이터 형식 – 문자열

드라이버 데이터 소스의 이름입니다. 애플리케이션이 SQLDriverConnect API에 DSN을 지정합니다.

드라이버(Driver)를 설정하지 않을 경우 이 파라미터가 필요합니다.

EndpointUrl

- 기본값 – 없음
- 데이터 형식 – 문자열

IAM 인증을 위해 Amazon Redshift Coral Service와 통신하는 데 사용되는 재정의의 엔드포인트입니다.

이 파라미터는 선택 사항입니다.

ForceLowercase

- 기본값 - 0
- 데이터 형식 – 부울

드라이버가 Single Sing-On 인증을 사용할 때 자격 증명 공급자에서 Amazon Redshift로 전송된 모든 DbGroups를 소문자로 표시할지 여부를 지정하는 부울 값입니다.

- 1 | TRUE: 드라이버가 ID 공급자에서 전송된 모든 DbGroups를 소문자로 표시합니다.
- 0 | FALSE: 드라이버가 DbGroups를 변경하지 않습니다.

이 파라미터는 선택 사항입니다.

group_federation

- 기본값 - 0
- 데이터 형식 – 부울

프로비저닝된 클러스터에서 임시 클러스터 보안 인증 정보를 얻는 데 `getClusterCredentialsWithIAM` API를 사용할지 여부를 지정하는 부울입니다. 이 옵션을 사용하면 IAM 사용자가 프로비저닝된 클러스터의 Redshift 데이터베이스 역할과 통합할 수 있습니다. 이 옵션은 Redshift Serverless 네임스페이스에 적용되지 않습니다.

- 1 | TRUE: 드라이버가 `getClusterCredentialsWithIAM` API를 사용하여 프로비저닝된 클러스터에서 임시 클러스터 보안 인증 정보를 얻습니다.
- 0 | FALSE: 드라이버가 기본 `getClusterCredentials` API를 사용하여 프로비저닝된 클러스터에서 임시 클러스터 보안 인증 정보를 얻습니다.

이 파라미터는 선택 사항입니다.

`https_proxy_host`

- 기본값 – 없음
- 데이터 형식 – 문자열

IAM 인증 프로세스를 통과하도록 하려는 프록시 서버의 호스트 이름 또는 IP 주소입니다.

이 파라미터는 선택 사항입니다.

`https_proxy_password`

- 기본값 – 없음
- 데이터 형식 – 문자열

프록시 서버에 액세스하는 데 사용하는 암호입니다. IAM 인증에 사용됩니다.

이 파라미터는 선택 사항입니다.

`https_proxy_port`

- 기본값 – 없음
- 데이터 형식 – 정수

프록시 서버가 클라이언트 연결을 수신 대기하는 데 사용하는 포트 번호입니다. IAM 인증에 사용됩니다.

이 파라미터는 선택 사항입니다.

`https_proxy_username`

- 기본값 – 없음
- 데이터 형식 – 문자열

프록시 서버에 액세스하는 데 사용하는 사용자 이름입니다. IAM 인증에 사용됩니다.

이 파라미터는 선택 사항입니다.

`IAM`

- 기본값 - 0
- 데이터 형식 – 부울

드라이버가 IAM 인증 방법을 사용하여 연결을 인증하는지 여부를 지정하는 부울 값입니다.

- 1 | TRUE: 드라이버가 IAM 인증 방법(액세스 키와 비밀 키의 페어, 프로필 또는 보안 인증 서비스 사용) 중 하나를 사용합니다.
- 0 | FALSE: 드라이버가 표준 인증(데이터베이스 사용자 이름 및 암호 사용)을 사용합니다.

이 파라미터는 선택 사항입니다.

`idc_client_display_name`

- 기본값 - Amazon Redshift ODBC 드라이버
- 데이터 형식 – 문자열

BrowserIdcAuthPlugin을 사용하는 클라이언트에 사용할 표시 이름입니다.

이 파라미터는 선택 사항입니다.

`idc_region`

- 기본값 – 없음
- 데이터 형식 – 문자열

AWS IAM Identity Center 인스턴스가 위치한 AWS 리전입니다.

이 파라미터는 plugin_name 구성 옵션에서 BrowserIdcAuthPlugin을 사용하여 인증하는 경우에만 필요합니다.

idp_host

- 기본값 – 없음
- 데이터 형식 – 문자열

Amazon Redshift에 인증하는 데 사용하는 자격 증명 공급자(IdP) 호스트입니다.

이 파라미터는 선택 사항입니다.

idp_port

- 기본값 – 없음
- 데이터 형식 – 정수

Amazon Redshift에 인증하는 데 사용하는 자격 증명 공급자(IdP)용 포트입니다. 클러스터를 생성, 수정 또는 마이그레이션할 때 선택한 포트에 따라 선택한 포트에 대한 액세스를 허용합니다.

이 파라미터는 선택 사항입니다.

idp_response_timeout

- 기본값 – 120
- 데이터 형식 – 정수

브라우저 플러그인을 통해 SAML 또는 Azure AD 서비스를 사용할 때 드라이버가 자격 증명 공급자의 SAML 응답을 기다리는 시간(초)입니다.

이 파라미터는 선택 사항입니다.

idp_tenant

- 기본값 – 없음
- 데이터 형식 – 문자열

Amazon Redshift 애플리케이션과 연결된 Azure AD 테넌트 ID입니다.

Azure AD 서비스를 통해 인증하는 경우 이 파라미터는 필수입니다.

idp_partition

- 기본값 – 없음
- 데이터 형식 – 문자열

ID 제공업체(IdP)가 구성된 클라우드 파티션을 지정합니다. 드라이버가 연결하는 IdP 인증 엔드포인트를 결정합니다.

이 파라미터를 비워 두면 드라이버는 기본적으로 상용 파티션으로 설정됩니다. 가능한 값은 다음과 같습니다.

- us-gov: IdP가 Azure Government에 구성된 경우 이 값을 사용합니다. 예를 들어 Azure AD Government는 login.microsoftonline.us 엔드포인트를 사용합니다.
- cn: IdP가 중국 클라우드 파티션에 구성된 경우 이 값을 사용합니다. 예를 들어 Azure AD China는 login.chinacloudapi.cn 엔드포인트를 사용합니다.

이 파라미터는 선택 사항입니다.

idp_use_https_proxy

- 기본값 - 0
- 데이터 형식 – 부울

드라이버가 프록시 서버를 통해 아이덴티티 제공업체(IdP)에 대한 인증 프로세스를 통과하는지 여부를 지정하는 부울 값입니다.

- 1 | TRUE: 드라이버가 프록시 서버를 통해 IdP 인증 프로세스를 통과합니다.
- 0 | FALSE: 드라이버가 프록시 서버를 통해 IdP 인증 프로세스를 통과하지 않습니다.

이 파라미터는 선택 사항입니다.

InstanceProfile

- 기본값 - 0

- 데이터 형식 – 부울

인증에 프로필을 사용하도록 구성된 경우 드라이버가 Amazon EC2 인스턴스 프로파일을 사용하는지 여부를 지정하는 부울 값입니다.

- 1 | TRUE: 드라이버가 Amazon EC2 인스턴스 프로파일을 사용합니다.
- 0 | FALSE. 드라이버가 프로필 이름 옵션(Profile(프로필))에 지정된 함께 묶은 역할 프로필을 대신 사용합니다.

이 파라미터는 선택 사항입니다.

issuer_url

- 기본값 – 없음
- 데이터 형식 – 문자열

AWS IAM Identity Center 서버의 인스턴스 엔드포인트를 가리킵니다.

이 파라미터는 plugin_name 구성 옵션에서 BrowserIdcAuthPlugin을 사용하여 인증하는 경우에만 필요합니다.

KeepAlive

- 기본값 – 1
- 데이터 형식 – 부울

드라이버가 TCP KeepAlive를 사용하여 연결 시간 초과를 방지하는지 여부를 지정하는 부울 값입니다.

- 1 | TRUE: 드라이버가 TCP KeepAlive를 사용하여 연결 시간 초과를 방지합니다.
- 0 | FALSE. 드라이버가 TCP KeepAlive를 사용하지 않습니다.

이 파라미터는 선택 사항입니다.

KeepAliveCount

- 기본값 - 0
- 데이터 형식 – 정수

연결이 끊긴 것으로 간주할 때까지 손실될 수 있는 TCP keepalive 패킷의 수입니다. 이 파라미터를 0으로 설정하면 드라이버가 이 설정에 시스템 기본값을 사용합니다.

이 파라미터는 선택 사항입니다.

KeepAliveInterval

- 기본값 - 0
- 데이터 형식 - 정수

TCP keepalive가 재전송되는 시간 간격(초)입니다. 이 파라미터를 0으로 설정하면 드라이버가 이 설정에 시스템 기본값을 사용합니다.

이 파라미터는 선택 사항입니다.

KeepAliveTime

- 기본값 - 0
- 데이터 형식 - 정수

드라이버가 TCP keepalive 패킷을 전송할 때까지 아무런 작업 없이 대기하는 시간(초)입니다. 이 파라미터를 0으로 설정하면 드라이버가 이 설정에 시스템 기본값을 사용합니다.

이 파라미터는 선택 사항입니다.

listen_port

- 기본값 - 7890
- 데이터 형식 - 정수

드라이버가 브라우저 플러그인을 통해 SAML, Azure AD 또는 AWS IAM Identity Center 서비스를 사용할 때 ID 제공업체 또는 권한 부여 코드로부터 SAML 응답을 받는 데 사용하는 포트입니다.

이 파라미터는 선택 사항입니다.

login_url

- 기본값 - 없음
- 데이터 형식 - 문자열

일반 브라우저 SAML 플러그인을 사용할 때 자격 증명 공급자의 웹 사이트에 있는 리소스의 URL입니다.

브라우저 플러그인을 통해 SAML 또는 Azure AD 서비스로 인증하는 경우 이 파라미터는 필수입니다.

loginToRp

- 기본값 - urn:amazon:webservices
- 데이터 형식 - 문자열

AD FS 인증 유형에 사용할 신뢰 당사자 트러스트입니다.

이 문자열은 선택 사항입니다.

LogLevel

- 기본값 - 0
- 데이터 형식 - 정수

이 속성을 사용하여 드라이버에서 로깅을 활성화하거나 비활성화하고 로그 파일에 포함되는 세부 정보의 양을 지정합니다. 로깅은 성능을 저하시킬 뿐만 아니라 대용량의 디스크 공간을 소모할 수 있기 때문에, 문제를 포착하기에 충분한 시간 동안만 로깅을 활성화하는 것이 좋습니다.

이 속성을 다음 값 중 하나로 설정합니다.

- 0: 끄. 모든 로깅을 사용 중지합니다.
- 1: 오류. 드라이버가 계속 실행되지만 오류가 발생할 수 있는 오류 이벤트를 로깅합니다.
- 2: API_CALL. 함수 인수 값을 사용하여 ODBC API 함수 호출을 로깅합니다.
- 3: 정보. 드라이버의 진행 상황을 설명하는 일반 정보를 로깅합니다.
- 4: MSG_PROTOCOL. 드라이버의 메시지 프로토콜에 대한 자세한 정보를 로깅합니다.
- 5: DEBUG. 모든 드라이버 작업을 로깅합니다.
- 6: DEBUG_APPEND. 모든 드라이버 작업에 대한 로그를 계속 추가합니다.

로깅을 사용하면 LogPath 속성에 지정한 위치에 드라이버가 다음 로그 파일을 생성합니다.

- 연결의 핸드셰이크 중에 발생하는 드라이버 작업을 로깅하는 redshift_odbc.log.1 파일입니다.
- 데이터베이스에 연결한 후의 모든 드라이버 작업에 대한 redshift_odbc.log 파일입니다.

이 파라미터는 선택 사항입니다.

LogPath

- 기본값 - OS별 TEMP 디렉터리
- 데이터 형식 - 문자열

LogLevel이 0보다 높은 경우 드라이버가 로그 파일을 저장하는 폴더의 전체 경로입니다.

이 파라미터는 선택 사항입니다.

Min_TLS

- 기본값 - 1.2
- 데이터 형식 - 문자열

드라이버가 데이터 스토어에서 연결을 암호화하는 데 사용하도록 허용하는 TLS/SSL의 최소 버전입니다. 예를 들어 TLS 1.2를 지정하면 연결을 암호화하는 데 TLS 1.1을 사용할 수 없습니다.

Min_TLS는 다음 값으로 지정할 수 있습니다.

- 1.0: 연결에 TLS 1.0 이상을 사용해야 합니다.
- 1.1: 연결에 TLS 1.1 이상을 사용해야 합니다.
- 1.2: 연결에 TLS 1.2 이상을 사용해야 합니다.

이 파라미터는 선택 사항입니다.

partner_spid

- 기본값 - 없음
- 데이터 형식 - 문자열

PingFederate 서비스를 사용하여 연결을 인증할 때 사용할 파트너 SPID(서비스 공급자 ID) 값입니다.

이 파라미터는 선택 사항입니다.

Password | PWS

- 기본값 - 없음

- 데이터 형식 – 문자열

사용자 필드(UserID|User(사용자)|LogonID)에서 제공한 데이터베이스 사용자 이름에 해당하는 암호입니다.

이 파라미터는 선택 사항입니다.

plugin_name

- 기본값 – 없음
- 데이터 형식 – 문자열

인증에 사용할 보안 인증 공급자 플러그인 이름입니다.

다음과 같은 값이 지원됩니다.

- ADFS: Active Directory Federation Services를 사용하여 인증합니다.
- AzureAD: Microsoft Azure Active Directory(AD) Service를 사용하여 인증합니다..
- BrowserAzureAD: Microsoft Azure Active Directory(AD) Service용 브라우저 플러그인을 사용하여 인증합니다.
- BrowserIdcAuthPlugin : AWS IAM Identity Center를 사용하는 인증 플러그인입니다.
- BrowserSAML: Okta 또는 Ping과 같은 SAML 서비스용 브라우저 플러그인을 사용하여 인증합니다.
- IdpTokenAuthPlugin: AWS IAM Identity Center 토큰 또는 AWS IAM Identity Center에 연결된 웹 ID 제공업체의 OpenID Connect(OIDC) JSON 기반 자격 증명 토큰(JWT)을 수락하는 권한 부여 플러그인입니다.
- JWT: JSON 웹 토큰(JWT)을 사용하여 인증합니다.
- Ping: PingFederate 서비스를 사용하여 인증합니다.
- Okta: Okta 서비스를 사용하여 인증합니다.

이 파라미터는 선택 사항입니다.

Port | PortNumber

- 기본값 - 5439
- 데이터 형식 – 정수

Amazon Redshift 서버가 클라이언트 연결을 수신 대기하는 데 사용하는 TCP 포트의 번호입니다.

이 파라미터는 선택 사항입니다.

preferred_role

- 기본값 – 없음
- 데이터 형식 – 문자열

Amazon Redshift에 연결하는 동안 수입하려는 역할입니다. IAM 인증에 사용됩니다.

이 파라미터는 선택 사항입니다.

프로필

- 기본값 – 없음
- 데이터 형식 – 문자열

Amazon Redshift에 연결하기 위한 인증에 사용되는 사용자 AWS 프로필의 이름입니다.

- 인스턴스 프로파일 사용(Use Instance Profile) 파라미터(InstanceProfile 속성)가 1 | TRUE로 설정된 경우 해당 설정이 우선하며 드라이버가 Amazon EC2 인스턴스 프로파일을 대신 사용합니다.
- 프로필이 포함된 보안 인증 파일의 기본 위치는 ~/.aws/Credentials입니다.
AWS_SHARED_CREDENTIALS_FILE 환경 변수를 사용하여 다른 보안 인증 파일을 가리킬 수 있습니다.

이 파라미터는 선택 사항입니다.

provider_name

- 기본값 – 없음
- 데이터 형식 – 문자열

사용자가 CREATE IDENTITY PROVIDER 쿼리를 사용하여 생성한 인증 공급자입니다. 네이티브 Amazon Redshift 인증에 사용됩니다.

이 파라미터는 선택 사항입니다.

ProxyHost

- 기본값 – 없음
- 데이터 형식 – 문자열

연결하는 데 사용할 프록시 서버의 호스트 이름 또는 IP 주소입니다.

이 파라미터는 선택 사항입니다.

ProxyPort

- 기본값 – 없음
- 데이터 형식 – 정수

프록시 서버가 클라이언트 연결을 수신 대기하는 데 사용하는 포트 번호입니다.

이 파라미터는 선택 사항입니다.

ProxyPwd

- 기본 ValPrevious ODBC 드라이버 버전 - 없음
- 데이터 형식 – 문자열

프록시 서버에 액세스하는 데 사용하는 암호입니다.

이 파라미터는 선택 사항입니다.

ProxyUid

- 기본값 – 없음
- 데이터 형식 – 문자열

프록시 서버에 액세스하는 데 사용하는 사용자 이름입니다.

이 파라미터는 선택 사항입니다.

ReadOnly

- 기본값 - 0
- 데이터 형식 – 부울

드라이버가 읽기 전용 모드인지 여부를 지정하는 부울 값입니다.

- 1 | TRUE: 연결이 읽기 전용 모드이며 데이터 스토어에 쓸 수 없습니다.
- 0 | FALSE: 연결이 읽기 전용 모드가 아니며 데이터 스토어에 쓸 수 있습니다.

이 파라미터는 선택 사항입니다.

region

- 기본값 – 없음
- 데이터 형식 – 문자열

클러스터가 있는 AWS 리전입니다.

이 파라미터는 선택 사항입니다.

SecretAccessKey

- 기본값 – 없음
- 데이터 형식 – 문자열

사용자 또는 역할에 대한 IAM 비밀 키입니다. 이 파라미터를 설정할 경우 AccessKeyId도 설정해야 합니다

이 파라미터는 선택 사항입니다.

SessionToken

- 기본값 – 없음
- 데이터 형식 – 문자열

인증에 사용하는 IAM 역할과 연결된 임시 IAM 세션 토큰입니다.

이 파라미터는 선택 사항입니다.

Server | HostName | Host

- 기본값 – 없음

- 데이터 형식 – 문자열

연결할 엔드포인트 서버입니다.

이 파라미터는 필수 사항입니다.

ssl_insecure

- 기본값 - 0
- 데이터 형식 – 부울

드라이버가 IdP 서버 인증서의 신뢰성을 확인하는지 여부를 지정하는 부울 값입니다.

- 1 | TRUE: 드라이버가 IdP 서버 인증서의 신뢰성을 확인하지 않습니다.
- 0 | FALSE: 드라이버가 IdP 서버 인증서의 신뢰성을 확인합니다.

이 파라미터는 선택 사항입니다.

SSLMode

- 기본값 – verify-ca
- 데이터 형식 – 문자열

Amazon Redshift 연결 시 사용할 SSL 인증서 검증 모드입니다. 다음과 같은 값이 가능합니다.

- verify-full: SSL, 신뢰할 수 있는 인증 기관 및 인증서와 일치하는 서버 이름을 사용해서만 연결합니다.
- verify-ca: SSL 및 신뢰할 수 있는 인증 기관을 사용해서만 연결합니다.
- require: SSL을 사용해서만 연결합니다.
- prefer: 가능한 경우 SSL을 사용하여 연결합니다. 그렇지 않으면 SSL을 사용하지 않고 연결합니다.
- allow: 기본적으로 SSL을 사용하지 않고 연결합니다. 서버에서 SSL 연결을 요구하는 경우 SSL을 사용합니다.
- disable: SSL을 사용하지 않고 연결합니다.

이 파라미터는 선택 사항입니다.

StsConnectionTimeout

- 기본값 - 0
- 데이터 형식 - 정수

IAM 연결의 최대 대기 시간(초)입니다. 0으로 설정되거나 지정되지 않은 경우, 드라이버는 각 AWS STS 호출에 대해 60초 동안 대기합니다.

이 파라미터는 선택 사항입니다.

StsEndpointUrl

- 기본값 - 없음
- 데이터 형식 - 문자열

이 옵션은 AWS Security Token Service(AWS STS)와 통신하는 데 사용되는 재정의의 엔드포인트를 지정합니다..

이 파라미터는 선택 사항입니다.

토큰

- 기본값 - 없음
- 데이터 형식 - 문자열

AWS IAM Identity Center에서 제공한 액세스 토큰 또는 AWS IAM Identity Center와 연결된 웹 ID 제공업체가 제공한 OpenID Connect(OIDC) JSON 웹 토큰(JWT)입니다. 애플리케이션은 AWS IAM Identity Center 또는 AWS IAM Identity Center와 연결된 자격 증명 공급자를 통해 애플리케이션 사용자를 인증하여 이 토큰을 생성해야 합니다.

이 파라미터는 IdpTokenAuthPlugin과 함께 작동합니다.

token_type

- 기본값 - 없음
- 데이터 형식 - 문자열

IdpTokenAuthPlugin에서 사용되는 토큰 유형입니다.

다음 값을 지정할 수 있습니다.

ACCESS_TOKEN

AWS IAM Identity Center에서 제공한 액세스 토큰을 사용하는 경우 이 값을 입력합니다.

EXT_JWT

AWS IAM Identity Center와 통합된 웹 기반 ID 제공업체가 제공하는 OpenID Connect(OIDC) JSON 웹 토큰(JWT)을 사용하는 경우 이 값을 입력합니다.

이 파라미터는 IdpTokenAuthPlugin과 함께 작동합니다.

UID | User | LogonID

- 기본값 – 없음
- 데이터 형식 – 문자열

Amazon Redshift 서버에 액세스하는 데 사용하는 사용자 이름입니다.

데이터베이스 인증을 사용하는 경우 이 파라미터가 필요합니다.

UseUnicode

- 기본값 - 0
- 데이터 형식 – 부울

드라이버가 Redshift 데이터를 유니코드 또는 일반 SQL 유형으로 반환할지 여부를 지정하는 부울입니다.

- 1 | TRUE: 드라이버가 문자 데이터 유형에 대해 와이드 SQL 유형을 반환합니다.
 - SQL_CHAR 대신 SQL_WCHAR가 반환됩니다.
 - SQL_VARCHAR 대신 SQL_WVARCHAR가 반환됩니다.
 - SQL_LONGVARCHAR 대신 SQL_WLONGVARCHAR가 반환됩니다.
- 0 | FALSE: 드라이버가 문자 데이터 유형에 대해 일반 SQL 유형을 반환합니다.
 - SQL_WCHAR 대신 SQL_CHAR가 반환됩니다.
 - SQL_WVARCHAR 대신 SQL_VARCHAR가 반환됩니다.
 - SQL_WLONGVARCHAR 대신 SQL_LONGVARCHAR가 반환됩니다.

이 파라미터는 선택 사항입니다. 드라이버 버전 2.1.15 이상에서 사용할 수 있습니다.

web_identity_token

- 기본값 – 없음
- 데이터 형식 – 문자열

자격 증명 공급자가 제공하는 OAUTH 토큰입니다. JWT 플러그인에서 사용됩니다.

plugin_name 파라미터를 BasicJwtCredentialsProvider로 설정한 경우 이 파라미터가 필요합니다.

이전 ODBC 드라이버 버전

도구에 특정 버전의 드라이버가 필요한 경우에만 Amazon Redshift ODBC 드라이버 버전 2.x의 이전 버전을 다운로드합니다.

Microsoft Windows용 이전 ODBC 드라이버 버전 사용

Microsoft Windows용 Amazon Redshift ODBC 드라이버 버전 2.x의 이전 버전은 다음과 같습니다.

- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.14.0/AmazonRedshiftODBC64-2.1.14.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.13.0/AmazonRedshiftODBC64-2.1.13.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.12.0/AmazonRedshiftODBC64-2.1.12.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.11.0/AmazonRedshiftODBC64-2.1.11.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.10.0/AmazonRedshiftODBC64-2.1.10.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.9.0/AmazonRedshiftODBC64-2.1.9.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.9.0/AmazonRedshiftODBC64-2.1.9.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.8.0/AmazonRedshiftODBC64-2.1.8.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.7.0/AmazonRedshiftODBC64-2.1.7.0.msi>

- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.6.0/AmazonRedshiftODBC64-2.1.6.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.4.0/AmazonRedshiftODBC64-2.1.4.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.3.0/AmazonRedshiftODBC64-2.1.3.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.2.0/AmazonRedshiftODBC64-2.1.2.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.1.0/AmazonRedshiftODBC64-2.1.1.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.0.0/AmazonRedshiftODBC64-2.1.0.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.1.0/AmazonRedshiftODBC64-2.0.1.0.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.11/AmazonRedshiftODBC64-2.0.0.11.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.9/AmazonRedshiftODBC64-2.0.0.9.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.8/AmazonRedshiftODBC64-2.0.0.8.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.7/AmazonRedshiftODBC64-2.0.0.7.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.6/AmazonRedshiftODBC64-2.0.0.6.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.5/AmazonRedshiftODBC64-2.0.0.5.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.3/AmazonRedshiftODBC64-2.0.0.3.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.1/AmazonRedshiftODBC64-2.0.0.1.msi>

Linux용 이전 ODBC 드라이버 버전 사용

Linux용 Amazon Redshift ODBC 드라이버 버전 2.x의 이전 버전은 다음과 같습니다.

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.14.0/AmazonRedshiftODBC-64-bit-2.1.14.0.x86_64.rpm
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.14.0/AmazonRedshiftODBC-64-bit-2.1.14.0.aarch64.rpm>
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.13.0/AmazonRedshiftODBC-64-bit-2.1.13.0.x86_64.rpm
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.13.0/AmazonRedshiftODBC-64-bit-2.1.13.0.aarch64.rpm>
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.12.0/AmazonRedshiftODBC-64-bit-2.1.12.0.x86_64.rpm
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.12.0/AmazonRedshiftODBC-64-bit-2.1.12.0.aarch64.rpm>
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.11.0/AmazonRedshiftODBC-64-bit-2.1.11.0.x86_64.rpm
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.11.0/AmazonRedshiftODBC-64-bit-2.1.11.0.aarch64.rpm>
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.10.0/AmazonRedshiftODBC-64-bit-2.1.10.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.9.0/AmazonRedshiftODBC-64-bit-2.1.9.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.8.0/AmazonRedshiftODBC-64-bit-2.1.8.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.7.0/AmazonRedshiftODBC-64-bit-2.1.7.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.6.0/AmazonRedshiftODBC-64-bit-2.1.6.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.4.0/AmazonRedshiftODBC-64-bit-2.1.4.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.3.0/AmazonRedshiftODBC-64-bit-2.1.3.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.2.0/AmazonRedshiftODBC-64-bit-2.1.2.0.x86_64.rpm

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.1.0/AmazonRedshiftODBC-64-bit-2.1.1.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.0.0/AmazonRedshiftODBC-64-bit-2.1.0.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.1.0/AmazonRedshiftODBC-64-bit-2.0.1.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.11/AmazonRedshiftODBC-64-bit-2.0.0.11.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.9/AmazonRedshiftODBC-64-bit-2.0.0.9.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.8/AmazonRedshiftODBC-64-bit-2.0.0.8.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.7/AmazonRedshiftODBC-64-bit-2.0.0.7.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.6/AmazonRedshiftODBC-64-bit-2.0.0.6.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.5/AmazonRedshiftODBC-64-bit-2.0.0.5.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.3/AmazonRedshiftODBC-64-bit-2.0.0.3.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.1/AmazonRedshiftODBC-64-bit-2.0.0.1.x86_64.rpm

Apple macOS용 이전 ODBC 드라이버 버전 사용

Apple macOS용 Amazon Redshift ODBC 드라이버 버전 2.x의 이전 버전은 다음과 같습니다.

- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.14.0/AmazonRedshiftODBC-64-bit.2.1.14.0.universal.pkg>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.13.0/AmazonRedshiftODBC-64-bit.2.1.13.0.universal.pkg>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.12.0/AmazonRedshiftODBC-64-bit.2.1.12.0.universal.pkg>

ODBC 드라이버 버전 1.x 연결 구성

ODBC 연결을 사용하여 여러 서드 파티 SQL 클라이언트 도구 및 애플리케이션에서 Amazon Redshift 클러스터에 연결할 수 있습니다. 이를 위해 클라이언트 컴퓨터 또는 Amazon EC2 인스턴스에서 연결을 설정해야 합니다. 클라이언트 도구가 JDBC를 지원하는 경우에는 JDBC의 구성 용이성 때문에 ODBC가 아닌 JDBC 연결 유형을 사용하는 것이 좋을 수도 있습니다. 하지만 클라이언트 도구가 JDBC를 지원하지 않으면 이번 섹션의 단계에 따라 ODBC 연결을 구성해야 합니다.

Amazon Redshift는 Linux, Windows 및 macOS X 운영 체제용 64비트 ODBC 드라이버를 제공합니다. 32비트 ODBC 드라이버는 공급이 중단되었습니다. 긴급 보안 패치를 제외하고 추가 업데이트는 릴리스되지 않습니다.

ODBC 드라이버 기능과 사전 조건에 대한 최신 정보는 [Amazon Redshift ODBC 드라이버 릴리스 정보](#)를 참조하세요.

Amazon Redshift ODBC 드라이버의 설치 및 구성 정보는 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)를 참조하세요.

주제

- [ODBC URL 가져오기](#)
- [Microsoft Windows에서 Amazon Redshift ODBC 드라이버 사용](#)
- [Linux에서 Amazon Redshift ODBC 드라이버 사용](#)
- [macOS X에서 Amazon Redshift ODBC 드라이버 사용](#)
- [ODBC 드라이버 옵션](#)
- [이전 ODBC 드라이버 버전](#)

ODBC URL 가져오기

Amazon Redshift는 Amazon Redshift 콘솔에 클러스터의 ODBC URL을 표시합니다. 이 URL에는 클라이언트 컴퓨터와 데이터베이스 간의 연결을 설정하는 정보가 포함되어 있습니다.

ODBC URL의 형식은 다음과 같습니다.

Driver={*driver*};Server=*endpoint*;Database=*database_name*;UID=*user_name*;PWD=*password*

위에 나와 있는 형식의 필드는 다음과 같은 값을 가집니다.

Field	값
Driver	사용할 64비트 ODBC 드라이버의 이름은 Amazon Redshift(x64)입니다. 32비트 ODBC 드라이버의 이름은 Amazon Redshift(x86)입니다.
Server	Amazon Redshift 클러스터의 엔드포인트입니다.
Database	클러스터에서 생성한 데이터베이스입니다.
UID	데이터베이스 연결 권한을 갖는 사용자 계정의 사용자 이름입니다. 클러스터를 시작할 때 설정한 관리자 계정을 사용할 수 있지만 이 값은 Amazon Redshift 권한이 아니라 데이터베이스 사용 권한입니다.
PWD	데이터베이스에 연결하기 위한 사용자 계정 암호입니다.
Port	클러스터를 시작할 때 지정한 포트 번호입니다. 방화벽이 있는 경우에는 이 포트가 사용할 수 있게 열려있어야 합니다.

앞에 있는 테이블의 필드에는 다음과 같은 특수 문자가 포함될 수 있습니다.

```
[ ] { } ( ) , ; ? * = ! @
```

이러한 특수 문자를 사용하는 경우 값을 중괄호로 묶어야 합니다. 예를 들어, 연결 문자열의 암호 값 `Your;password123`은 `PWD={Your;password123};`으로 표시됩니다.

`Field=value` 쌍은 세미콜론으로 구분되므로 `}` 및 `;` 조합은 그 사이의 공백 개수에 상관없이 `Field={value};` 쌍의 끝으로 간주됩니다. 필드 값에서 `}` 시퀀스를 사용하지 않는 것이 좋습니다. 예를 들어 암호 값을 `PWD={This is a passwor} ;d};`로 설정하면 암호는 `This is a passwor} ;`이 되고 URL에 오류가 발생합니다.

다음은 ODBC URL의 예입니다.

```
Driver={Amazon Redshift (x64)};
      Server=examplecluster.abc123xyz789.us-
west-2.redshift.amazonaws.com;
      Database=dev;
      UID=adminuser;
      PWD=insert_your_admin_user_password_here;
```

Port=5439

ODBC 연결을 만드는 방법에 대한 자세한 내용은 [클러스터 연결 문자열 찾기](#) 섹션을 참조하세요.

Microsoft Windows에서 Amazon Redshift ODBC 드라이버 사용

Amazon Redshift 데이터 웨어하우스에 액세스하는 클라이언트 컴퓨터에 Amazon Redshift ODBC 드라이버를 설치합니다. 드라이버를 설치하는 컴퓨터마다 최소 시스템 요구 사항을 충족해야 합니다. 최소 시스템 요구 사항에 대한 자세한 내용은 [Amazon Redshift ODBC 커넥터 설치 및 구성 안내서](#)를 참조하세요.

주제

- [Amazon Redshift ODBC 드라이버 다운로드 및 설치](#)
- [ODBC 연결을 위한 시스템 DSN 항목 생성](#)

Amazon Redshift ODBC 드라이버 다운로드 및 설치

Windows 운영 체제용 Amazon Redshift ODBC 드라이버를 다운로드하려면 다음 절차를 따르세요. Amazon Redshift와 함께 사용하도록 인증되었고 특정 드라이버가 필요한 서드 파티 애플리케이션을 실행하고 있다면 이것과 다른 드라이버만 사용합니다.

ODBC 드라이버를 설치하려면

1. SQL 클라이언트 도구 또는 애플리케이션의 시스템 아키텍처에 따라 다음 중 하나를 다운로드합니다.

- [64비트 ODBC 드라이버 버전 1.6.3](#)

이 드라이버의 이름은 Amazon Redshift(x64)입니다.

- [32비트 ODBC 드라이버 버전 1.4.52](#)

이 드라이버의 이름은 Amazon Redshift(x86)입니다. 32비트 ODBC 드라이버는 공급이 중단되었습니다. 긴급 보안 패치를 제외하고 추가 업데이트는 릴리스되지 않습니다.

Note

SQL 클라이언트 도구 또는 애플리케이션의 시스템 아키텍처와 일치하는 MSI 패키지를 다운로드하세요. 예를 들어 SQL 클라이언트 도구가 64비트라면 드라이버도 64비트를 설치해야 합니다.

그런 다음 [Amazon Redshift ODBC 및 JDBC 드라이버 사용권 계약](#)을 다운로드하여 검토합니다.

2. .msi 파일을 두 번 클릭한 후 마법사의 단계에 따라 드라이버를 설치합니다.

ODBC 연결을 위한 시스템 DSN 항목 생성

ODBC 드라이버를 다운로드하여 설치한 후에는 데이터 원본 이름(DSN) 항목을 클라이언트 컴퓨터 또는 Amazon EC2 인스턴스에 추가합니다. SQL 클라이언트 도구는 이 데이터 원본을 사용하여 Amazon Redshift 데이터베이스에 연결합니다.

사용자 DSN 대신 시스템 DSN을 생성하는 것이 좋습니다. 일부 애플리케이션은 다른 사용자 계정을 사용하여 데이터를 로드합니다. 이러한 애플리케이션은 다른 사용자 계정으로 생성된 사용자 DSN을 감지하지 못할 수 있습니다.

Note

AWS Identity and Access Management(IAM) 자격 증명 또는 자격 증명 공급자(IdP) 자격 증명을 사용한 인증의 경우 추가 단계가 필요합니다. 자세한 내용은 [5단계: IAM 자격 증명을 사용하여 JDBC 또는 ODBC 연결 구성](#) 섹션을 참조하세요.

시스템 DSN 항목을 생성하는 방법에 대한 자세한 내용은 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)를 참조하세요.

Windows에서 ODBC 연결을 위한 시스템 DSN 항목을 생성하려면

1. 시작 메뉴에서 ODBC 데이터 원본을 엽니다.

Amazon Redshift에 연결하는 데 사용하는 클라이언트 애플리케이션과 동일한 비트를 가진 ODBC 데이터 원본 관리자를 선택해야 합니다.

2. ODBC 데이터 원본 관리자(ODBC Data Source Administrator)에서 드라이버(Driver) 탭을 선택하고 드라이버 폴더를 찾습니다.
 - Amazon Redshift ODBC 드라이버(64비트)
 - Amazon Redshift ODBC 드라이버(32비트)
3. 해당 컴퓨터의 모든 사용자가 이용할 수 있도록 드라이버를 구성하려면 시스템 DSN(System DSN) 탭을 선택하고, 본인의 사용자 계정에서만 이용할 수 있도록 드라이버를 구성하려면 사용자 DSN(User DSN) 탭을 선택합니다.
4. 추가를 선택합니다. 새 데이터 원본 생성(Create New Data Source) 창이 열립니다.
5. Amazon Redshift ODBC 드라이버를 선택하고 완료(Finish)를 선택합니다. Amazon Redshift ODBC 드라이버 DSN 설정(Amazon Redshift ODBC Driver DSN Setup) 창이 열립니다.
6. 연결 설정(Connection Settings)에서 다음 정보를 입력하세요.

데이터 소스 이름

데이터 소스 이름을 입력합니다. 나중에 클러스터 연결을 생성할 때 데이터 원본을 식별할 이름이라면 무엇이든 사용할 수 있습니다. 예를 들어 Amazon Redshift 시작 안내서에 따른 경우 `exampleclusterdsn`을 입력하면 이 DSN과 연결할 클러스터를 기억하기 쉽습니다.

Server

Amazon Redshift 클러스터의 엔드포인트를 지정합니다. 이 정보는 Amazon Redshift 콘솔의 클러스터 세부 정보 페이지에서 찾을 수 있습니다. 자세한 내용은 [Amazon Redshift에서 연결 구성](#) 섹션을 참조하세요.

Port

데이터베이스가 사용하는 포트 번호를 입력합니다. 클러스터를 시작할 때 사용하도록 구성된 포트를 사용해야 합니다.

데이터베이스

Amazon Redshift 데이터베이스의 이름을 입력합니다. 데이터베이스 이름을 지정하지 않고 클러스터를 시작한 경우 `dev`를 입력합니다. 그렇지 않으면 시작 프로세스 중에 선택한 이름을 사용합니다. Amazon Redshift 시작 안내서에 따른 경우 `dev`를 입력합니다.

7. 인증(Authentication)에서 표준 또는 IAM 인증을 구성하는 구성 옵션을 지정합니다. 인증 옵션에 대한 자세한 내용은 Amazon Redshift ODBC Connector Installation and Configuration Guide의 "Configuring Authentication on Windows"를 참조하세요.

8. SSL 설정(SSL Settings)에서 다음 값을 지정합니다.

SSL 인증

SSL(Secure Sockets Layer)의 처리 모드를 선택합니다. 테스트 환경에서는 `prefer`를 사용할 수 있습니다. 그러나 프로덕션 환경 및 안전한 데이터 교환이 필요한 경우 `verify-ca` 또는 `verify-full`을 사용합니다. Windows에서 SSL을 사용하는 방법에 대한 자세한 내용은 Amazon Redshift ODBC Connector Installation and Configuration Guide의 "Configuring SSL Verification on Windows" 섹션을 참조하세요.

9. 추가 옵션(Additional Options)에서 쿼리 결과를 SQL 클라이언트 도구 또는 애플리케이션에 반환하는 방법에 대한 옵션을 지정합니다. 자세한 내용은 Amazon Redshift ODBC Connector Installation and Configuration Guide의 "Configuring Additional Options on Windows" 섹션을 참조하세요.
10. 로깅 옵션(Logging Options)에서 로깅 옵션에 대한 값을 지정합니다. 자세한 내용은 Amazon Redshift ODBC Connector Installation and Configuration Guide의 "Configuring Logging Options on Windows" 섹션을 참조하세요.

그 다음에 확인(OK)를 선택합니다.

11. 데이터 형식 옵션(Data Type Options)에서 데이터 형식에 대한 값을 지정합니다. 자세한 내용은 Amazon Redshift ODBC Connector Installation and Configuration Guide의 "Configuring Data Type Options on Windows" 섹션을 참조하세요.

그 다음에 확인(OK)를 선택합니다.

12. 테스트를 선택합니다. 클라이언트 컴퓨터가 Amazon Redshift 데이터베이스에 연결할 수 있으면 연결 성공(Connection successful) 메시지가 표시됩니다.

클라이언트 컴퓨터가 데이터베이스에 연결하지 못하는 경우에는 가능한 문제를 해결할 수 있습니다. 자세한 내용은 [Amazon Redshift 연결 문제 해결](#) 섹션을 참조하세요.

13. Windows에서 TCP keepalive를 구성하여 연결 시간 초과가 발생하지 않도록 합니다. Windows에서 TCP KeepAlive를 구성하는 방법에 대한 자세한 내용은 Amazon Redshift ODBC Connector Installation and Configuration Guide를 참조하세요.
14. 문제 해결에 도움이 되도록 로깅을 구성합니다. Windows에서 로깅을 구성하는 방법에 대한 자세한 내용은 Amazon Redshift ODBC Connector Installation and Configuration Guide를 참조하세요.

Linux에서 Amazon Redshift ODBC 드라이버 사용

Amazon Redshift 데이터 웨어하우스에 액세스하는 클라이언트 컴퓨터에 Amazon Redshift ODBC 드라이버를 설치합니다. 드라이버를 설치하는 컴퓨터마다 최소 시스템 요구 사항을 충족해야 합니다. 최소 시스템 요구 사항에 대한 자세한 내용은 [Amazon Redshift ODBC 커넥터 설치 및 구성 안내서](#)를 참조하세요.

주제

- [Amazon Redshift ODBC 드라이버 다운로드 및 설치](#)
- [ODBC 드라이버 관리자를 사용하여 드라이버 구성](#)

Amazon Redshift ODBC 드라이버 다운로드 및 설치

이번 섹션에서 설명하는 단계에 따라 지원되는 Linux 배포에서 Amazon Redshift ODBC 드라이버를 다운로드하고 설치합니다. 설치 프로세스는 드라이버 파일을 다음 디렉터리에 설치합니다.

- /opt/amazon/redshiftodbc/lib/64(64비트 드라이버의 경우)
- /opt/amazon/redshiftodbc/ErrorMessage
- /opt/amazon/redshiftodbc/Setup
- /opt/amazon/redshiftodbc/lib/32(32비트 드라이버의 경우)

Amazon Redshift ODBC 드라이버를 설치하려면

1. SQL 클라이언트 도구 또는 애플리케이션의 시스템 아키텍처에 따라 다음 중 하나를 다운로드합니다.
 - [64비트 RPM 드라이버 버전 1.6.3](#)
 - [64비트 Debian 드라이버 버전 1.6.3](#)
 - [32비트 RPM 드라이버 버전 1.4.52](#)

이러한 드라이버의 이름은 각각 Amazon Redshift ODBC 드라이버입니다. 32비트 ODBC 드라이버는 공급이 중단되었습니다. 긴급 보안 패치를 제외하고 추가 업데이트는 릴리스되지 않습니다.

 Note

SQL 클라이언트 도구 또는 애플리케이션의 시스템 아키텍처와 일치하는 패키지를 다운로드하세요. 예를 들어 클라이언트 도구가 64비트라면 드라이버도 64비트를 설치해야 합니다.

그런 다음 [Amazon Redshift ODBC 및 JDBC 드라이버 사용권 계약](#)을 다운로드하여 검토합니다.

2. 패키지를 다운로드한 위치로 이동한 다음, 다음 명령 중 하나를 실행합니다. Linux 배포판에 따라 올바른 명령을 사용하세요.

- RHEL 및 CentOS 운영 체제에서 다음 명령을 실행합니다.

```
yum -nogpgcheck localinstall RPMFileName
```

*RPMFileName*을 RPM 패키지 파일 이름으로 바꿉니다. 예를 들어, 다음은 64비트 드라이버를 설치하는 명령입니다.

```
yum -nogpgcheck localinstall AmazonRedshiftODBC-64-bit-1.x.xx.xxxx-x.x86_64.rpm
```

- SLES에서 다음 명령을 실행합니다.

```
zypper install RPMFileName
```

*RPMFileName*을 RPM 패키지 파일 이름으로 바꿉니다. 예를 들어, 다음은 64비트 드라이버를 설치하는 명령입니다.

```
zypper install AmazonRedshiftODBC-1.x.x.xxxx-x.x86_64.rpm
```

- Debian에서 다음 명령을 실행합니다.

```
sudo apt install ./DEBFileName.deb
```

*DEBFileName.deb*를 Debian 패키지 파일 이름으로 바꿉니다. 예를 들어, 다음은 64비트 드라이버를 설치하는 명령입니다.

```
sudo apt install ./AmazonRedshiftODBC-1.x.x.xxxx-x.x86_64.deb
```

⚠ Important

드라이버 설치를 마친 후에는 시스템에서 사용할 수 있도록 구성해야 합니다. 드라이버 구성에 대한 자세한 내용은 [ODBC 드라이버 관리자를 사용하여 드라이버 구성](#) 섹션을 참조하세요.

ODBC 드라이버 관리자를 사용하여 드라이버 구성

Linux 운영 체제에서 ODBC 드라이버 관리자를 사용하여 ODBC 연결 설정을 구성합니다. ODBC 드라이버 관리자는 구성 파일을 사용하여 ODBC 데이터 원본과 드라이버를 정의 및 구성합니다. 또한 사용하는 운영 체제에 따라 ODBC 드라이버 관리자도 달라집니다. Linux의 경우 unixODBC 드라이버 관리자입니다.

Amazon Redshift ODBC 드라이버 구성에 대해 지원되는 ODBC 드라이버 관리자에 관한 자세한 내용은 Linux 운영 체제의 경우 [Linux에서 Amazon Redshift ODBC 드라이버 사용](#) 섹션을 참조하시기 바랍니다. 또한 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)에서 "Specifying ODBC Driver Managers on Non-Windows Machines"(Windows 이외의 컴퓨터에서 ODBC 드라이버 관리자 지정)를 참조하세요. 링크를 사용합니다.

Amazon Redshift ODBC 드라이버 구성에 필요한 파일은 `amazon.redshiftodbc.ini`, `odbc.ini` 및 `odbcinst.ini`입니다.

드라이버를 기본 위치에 설치한 경우에는 `amazon.redshiftodbc.ini` 구성 파일이 다음 디렉터리 중 한 곳에 만들어집니다.

- `/opt/amazon/redshiftodbc/lib/64`(Linux 운영 체제의 64비트 드라이버의 경우)
- `/opt/amazon/redshiftodbc/lib/32`(Linux 운영 체제의 32비트 드라이버의 경우)

또한 Linux의 `/opt/amazon/redshiftodbc/Setup`에는 `odbc.ini` 샘플과 `odbcinst.ini` 파일이 있습니다. 이러한 파일을 Amazon Redshift ODBC 드라이버 및 데이터 원본 이름(DSN)을 구성하는 예제로 사용할 수 있습니다.

Amazon Redshift ODBC 드라이버 설치 디렉터리를 구성 파일용으로 사용하는 것은 바람직하지 않습니다. Setup 디렉터리의 샘플 파일은 오직 예를 제공하기 위한 용도일 뿐입니다. 나중에 Amazon Redshift ODBC 드라이버를 다시 설치하거나 새 버전으로 업그레이드하면 설치 디렉터리를 덮어씁니다. 그러면 이러한 파일에 대한 변경 사항이 손실됩니다.

이를 방지하려면 `amazon.redshiftdbc.ini` 파일을 설치 디렉터리가 아닌 다른 디렉터리에 복사합니다. 이 파일을 사용자의 홈 디렉터리로 복사하는 경우에는 파일 이름 앞에 마침표(.)를 추가하여 숨김 파일로 지정하세요.

`odbc.ini` 및 `odbcinst.ini` 파일의 경우 사용자의 홈 디렉터리에 있는 구성 파일을 사용하거나 다른 디렉터리에 새 버전을 생성합니다. 기본적으로 Linux 운영 체제는 사용자의 홈 디렉터리(`/home/$USER` 또는 `~/`)에 `odbc.ini` 파일과 `odbcinst.ini` 파일이 있어야 합니다. 이러한 기본 파일은 숨김 파일이기 때문에 파일 이름 앞에 마침표(.)가 있습니다. 이러한 파일은 `-a` 플래그를 사용하여 디렉터리 내용을 나열하는 경우에만 표시됩니다.

`odbc.ini` 및 `odbcinst.ini` 파일에 대해 어떤 옵션을 선택하든지 파일을 수정하여 드라이버 및 DSN 구성 정보를 추가합니다. 새로운 파일을 생성할 때도 마찬가지로 환경 변수를 설정하여 구성 파일의 위치를 지정해야 합니다.

기본적으로 ODBC 드라이버 관리자는 홈 디렉터리에 있는 숨겨진 버전의 `odbc.ini` 및 `odbcinst.ini` 구성 파일(`.odbc.ini` 및 `.odbcinst.ini`)을 사용하도록 구성됩니다. 또한 드라이버 설치 디렉터리의 `amazon.redshiftdbc.ini` 하위 폴더에 있는 `/lib` 파일을 사용하도록 구성됩니다. 이러한 구성 파일을 다른 위치에 저장하는 경우 드라이버 관리자가 파일을 찾을 수 있도록 다음에 설명된 환경 변수를 설정합니다. 자세한 내용은 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)에서 "Specifying the Locations of the Driver Configuration Files"(드라이버 구성 파일 위치 지정)를 참조하세요. 링크를 사용합니다.

Linux 운영 체제에서 데이터 소스 이름 작성

DSN(데이터 원본 이름)을 사용하여 데이터 저장소에 연결할 경우 DSN을 정의하도록 `odbc.ini` 파일을 구성합니다. `odbc.ini` 파일의 속성을 설정하여 데이터 저장소에 대한 연결 정보를 지정하는 DSN을 생성합니다.

`odbc.ini` 파일 구성 방법에 대한 자세한 내용은 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)의 "Creating a Data Source Name on a Non-Windows Machine"(Windows 이외의 컴퓨터에서 데이터 소스 이름 생성)을 참조하세요.

Linux 운영 체제에서 사용하는 형식은 다음과 같습니다.

```
[ODBC Data Sources]
driver_name=dsn_name
```

```
[dsn_name]
Driver=path/driver_file

Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

다음 예에서는 Linux 운영 체제에서 64비트 ODBC 드라이버를 사용하는 odbc.ini에 대한 구성을 보여줍니다.

```
[ODBC Data Sources]
Amazon_Redshift_x64=Amazon Redshift (x64)

[Amazon Redshift (x64)]
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

다음 예에서는 Linux 운영 체제에서 32비트 ODBC 드라이버를 사용하는 odbc.ini에 대한 구성을 보여줍니다.

```
[ODBC Data Sources]
Amazon_Redshift_x32=Amazon Redshift (x86)

[Amazon Redshift (x86)]
Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

Linux 운영 체제에서 DSN 없이 연결 구성

DSN이 없는 연결을 통해 데이터 저장소에 연결하려면 odbcinst.ini 파일에 드라이버를 정의합니다. 그런 다음 애플리케이션에 DSN이 없는 연결 문자열을 제공합니다.

이 경우 odbcinst.ini 파일을 구성하는 방법에 대한 자세한 내용은 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)

에서 "Configuring a DSN-less Connection on a Non-Windows Machine"(Windows 이외의 컴퓨터에서 DSN이 없는 연결 구성)을 참조하세요. 링크를 사용합니다.

Linux 운영 체제에서 사용하는 형식은 다음과 같습니다.

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/driver_file
...
```

다음 예에서는 64비트 드라이버가 Linux 운영 체제의 기본 디렉터리에 설치되어 있는 경우의 odbcinst.ini 구성을 보여줍니다.

```
[ODBC Drivers]
Amazon Redshift (x64)=Installed

[Amazon Redshift (x64)]
Description=Amazon Redshift ODBC Driver (64-bit)
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
```

다음 예에서는 32비트 드라이버가 Linux 운영 체제의 기본 디렉터리에 설치되어 있는 경우의 odbcinst.ini 구성을 보여줍니다.

```
[ODBC Drivers]
Amazon Redshift (x86)=Installed

[Amazon Redshift (x86)]
Description=Amazon Redshift ODBC Driver (32-bit)
Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so
```

환경 변수 구성

올바른 ODBC 드라이버 관리자를 사용하여 올바른 드라이버를 로드합니다. 이를 위해 라이브러리 경로 환경 변수를 설정합니다. 자세한 내용은 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)의 "Specifying ODBC

Driver Managers on Non-Windows Machines"(Windows 이외의 컴퓨터에서 ODBC 드라이버 관리자 지정)를 참조하세요. 링크를 사용합니다.

기본적으로 ODBC 드라이버 관리자는 홈 디렉터리에 있는 숨겨진 버전의 `odbc.ini` 및 `odbcinst.ini` 구성 파일(`odbc.ini` 및 `odbcinst.ini`)을 사용하도록 구성됩니다. 또한 드라이버 설치 디렉터리의 `amazon.redshiftodbc.ini` 하위 폴더에 있는 `/lib` 파일을 사용하도록 구성됩니다. 이러한 구성 파일을 다른 위치에 저장하는 경우 드라이버 관리자가 파일을 찾을 수 있도록 환경 변수를 지정합니다. 자세한 내용은 Amazon Redshift ODBC Connector Installation and Configuration Guide의 "Specifying the Locations of the Driver Configuration Files" 섹션을 참조하세요.

연결 기능 구성

ODBC 설정에 대해 다음 연결 기능을 구성할 수 있습니다.

- 자격 증명을 제공하고 Amazon Redshift 데이터베이스 연결을 인증하도록 ODBC 드라이버를 구성합니다.
- 보안 소켓 계층(SSL)이 사용 설정된 Amazon Redshift 서버에 연결하는 경우 SSL에서 사용할 수 있는 소켓에 연결하도록 ODBC 드라이버를 구성합니다.
- 프록시 서버를 통해 Amazon Redshift에 연결하도록 ODBC 드라이버를 구성합니다.
- 쿼리가 너무 많은 메모리를 소비하지 않게 쿼리 처리 모드를 사용하도록 ODBC 드라이버를 구성합니다.
- 프록시 서버를 통해 IAM 인증 프로세스를 통과하도록 ODBC 드라이버를 구성합니다.
- TCP keepalive를 사용하여 연결 시간 초과가 발생하지 않도록 ODBC 드라이버를 구성합니다.

이러한 연결 기능에 대한 자세한 내용은 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)를 참조하세요.

macOS X에서 Amazon Redshift ODBC 드라이버 사용

Amazon Redshift 데이터 웨어하우스에 액세스하는 클라이언트 컴퓨터에 드라이버를 설치합니다. 드라이버를 설치하는 컴퓨터마다 최소 시스템 요구 사항을 충족해야 합니다. 최소 시스템 요구 사항에 대한 자세한 내용은 [Amazon Redshift ODBC 커넥터 설치 및 구성 안내서](#)를 참조하세요.

주제

- [Amazon Redshift ODBC 드라이버 다운로드 및 설치](#)
- [ODBC 드라이버 관리자를 사용하여 드라이버 구성](#)

Amazon Redshift ODBC 드라이버 다운로드 및 설치

이 섹션의 단계에 따라 지원되는 macOS X 버전에 Amazon Redshift ODBC 드라이버를 다운로드하고 설치합니다. 설치 프로세스는 드라이버 파일을 다음 디렉터리에 설치합니다.

- /opt/amazon/redshift/lib/universal
- /opt/amazon/redshift/ErrorMessage
- /opt/amazon/redshift/Setup

macOS X에 Amazon Redshift ODBC 드라이버를 설치하려면

1. macOS X에 Amazon Redshift ODBC 드라이버를 설치하려면 [macOS 드라이버 버전 1.6.3](#)을 다운로드합니다.
그런 다음 [Amazon Redshift ODBC 및 JDBC 드라이버 사용권 계약](#)을 다운로드하여 검토합니다.
2. AmazonRedshiftODBC.pkg를 두 번 클릭하여 설치 프로그램을 실행합니다.
3. 설치 프로그램의 단계에 따라 드라이버 설치 프로세스를 마칩니다. 설치를 수행하려면 라이선스 계약 조건에 동의해야 합니다.

Important

드라이버 설치를 마친 후에는 시스템에서 사용할 수 있도록 구성해야 합니다. 드라이버 구성에 대한 자세한 내용은 [ODBC 드라이버 관리자를 사용하여 드라이버 구성](#) 섹션을 참조하세요.

ODBC 드라이버 관리자를 사용하여 드라이버 구성

macOS X 운영 체제에서 ODBC 드라이버 관리자를 사용하여 ODBC 연결 설정을 구성합니다. ODBC 드라이버 관리자는 구성 파일을 사용하여 ODBC 데이터 원본과 드라이버를 정의 및 구성합니다. 또한 사용하는 운영 체제에 따라 ODBC 드라이버 관리자도 달라집니다. macOS X 운영 체제의 경우에는 iODBC 드라이버 관리자입니다.

Amazon Redshift ODBC 드라이버 구성에 대해 지원되는 ODBC 드라이버 관리자에 관한 자세한 내용은 macOS X 운영 체제의 경우 [macOS X에서 Amazon Redshift ODBC 드라이버 사용](#) 섹션을 참조하시기 바랍니다. 또한 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)에서 "Specifying ODBC Driver Managers on Non- Windows Machines"(Windows 이외의 컴퓨터에서 ODBC 드라이버 관리자 지정)를 참조하세요. 링크를 사용합니다.

Amazon Redshift ODBC 드라이버 구성에 필요한 파일은 `amazon.redshiftodbc.ini`, `odbc.ini` 및 `odbcinst.ini`입니다.

기본 위치에 드라이버를 설치한 경우에는 `amazon.redshiftodbc.ini` 구성 파일이 `/opt/amazon/redshift/lib`에 있습니다.

또한 macOS X의 `/opt/amazon/redshift/Setup`에는 `odbc.ini` 샘플과 `odbcinst.ini` 파일이 있습니다. 이러한 파일을 Amazon Redshift ODBC 드라이버 및 데이터 원본 이름(DSN)을 구성하는 예제로 사용할 수 있습니다.

Amazon Redshift ODBC 드라이버 설치 디렉터리를 구성 파일용으로 사용하는 것은 바람직하지 않습니다. Setup 디렉터리의 샘플 파일은 오직 예를 제공하기 위한 용도일 뿐입니다. 나중에 Amazon Redshift ODBC 드라이버를 다시 설치하거나 새 버전으로 업그레이드하면 설치 디렉터리를 덮어씁니다. 그러면 이러한 파일에 대한 변경 사항이 손실됩니다.

이를 방지하려면 `amazon.redshiftodbc.ini` 파일을 설치 디렉터리가 아닌 다른 디렉터리에 복사합니다. 이 파일을 사용자의 홈 디렉터리로 복사하는 경우에는 파일 이름 앞에 마침표(.)를 추가하여 숨김 파일로 지정하세요.

`odbc.ini` 및 `odbcinst.ini` 파일의 경우 사용자의 홈 디렉터리에 있는 구성 파일을 사용하거나 다른 디렉터리에 새 버전을 생성합니다. 기본적으로 macOS X 운영 체제는 사용자의 홈 디렉터리(`/home/$USER` 또는 `~/`)에 `odbc.ini` 파일과 `odbcinst.ini` 파일이 있어야 합니다. 이러한 기본 파일은 숨김 파일이기 때문에 파일 이름 앞에 마침표(.)가 있습니다. 이러한 파일은 `-a` 플래그를 사용하여 디렉터리 내용을 나열하는 경우에만 표시됩니다.

`odbc.ini` 및 `odbcinst.ini` 파일에 대해 어떤 옵션을 선택하든지 파일을 수정하여 드라이버 및 DSN 구성 정보를 추가합니다. 새로운 파일을 생성할 때도 마찬가지로 환경 변수를 설정하여 구성 파일의 위치를 지정해야 합니다.

기본적으로 ODBC 드라이버 관리자는 홈 디렉터리에 있는 숨겨진 버전의 `odbc.ini` 및 `odbcinst.ini` 구성 파일(`odbc.ini` 및 `.odbcinst.ini`)을 사용하도록 구성됩니다. 또한 드라이버 설치 디렉터리의 `amazon.redshiftodbc.ini` 하위 폴더에 있는 `/lib` 파일을 사용하도록 구성됩니다. 이러한 구성 파일을 다른 위치에 저장하는 경우 드라이버 관리자가 파일을 찾을 수 있도록 다음에 설명된 환경 변수를 설정합니다. 자세한 내용은 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)에서 "Specifying the Locations of the Driver Configuration Files"(드라이버 구성 파일 위치 지정)를 참조하세요. 링크를 사용합니다.

macOS X 운영 체제에서 데이터 소스 이름 작성

DSN(데이터 원본 이름)을 사용하여 데이터 저장소에 연결할 경우 DSN을 정의하도록 `odbc.ini` 파일을 구성합니다. `odbc.ini` 파일의 속성을 설정하여 데이터 저장소에 대한 연결 정보를 지정하는 DSN을 생성합니다.

`odbc.ini` 파일 구성 방법에 대한 자세한 내용은 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)의 "Creating a Data Source Name on a Non-Windows Machine"(Windows 이외의 컴퓨터에서 데이터 소스 이름 생성)을 참조하세요.

macOS X 운영 체제에서 사용하는 형식은 다음과 같습니다.

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/lib/amazonredshiftodbc.dylib

Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

다음 예에서는 macOS X 운영 체제에서 `odbc.ini` 구성을 보여줍니다.

```
[ODBC Data Sources]
Amazon_Redshift_dylib=Amazon Redshift DSN for macOS X

[Amazon Redshift DSN for macOS X]
Driver=/opt/amazon/redshift/lib/amazonredshiftodbc.dylib
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

macOS X 운영 체제에서 DSN 없이 연결 구성

DSN이 없는 연결을 통해 데이터 저장소에 연결하려면 `odbcinst.ini` 파일에 드라이버를 정의합니다. 그런 다음 애플리케이션에 DSN이 없는 연결 문자열을 제공합니다.

이 경우 `odbcinst.ini` 파일을 구성하는 방법에 대한 자세한 내용은 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)에서 "Configuring a DSN-less Connection on a Non-Windows Machine"(Windows 이외의 컴퓨터에서 DSN이 없는 연결 구성)을 참조하세요. 링크를 사용합니다.

macOS X 운영 체제에서 사용하는 형식은 다음과 같습니다.

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/lib/amazonredshiftodbc.dylib
...
```

다음 예에서는 드라이버가 macOS X 운영 체제의 기본 디렉터리에 설치되어 있는 경우 `odbcinst.ini` 구성을 보여줍니다.

```
[ODBC Drivers]
Amazon RedshiftODBC DSN=Installed

[Amazon RedshiftODBC DSN]
Description=Amazon Redshift ODBC Driver for macOS X
Driver=/opt/amazon/redshift/lib/amazonredshiftodbc.dylib
```

환경 변수 구성

올바른 ODBC 드라이버 관리자를 사용하여 올바른 드라이버를 로드합니다. 이를 위해 라이브러리 경로 환경 변수를 설정합니다. 자세한 내용은 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)의 "Specifying ODBC Driver Managers on Non-Windows Machines"(Windows 이외의 컴퓨터에서 ODBC 드라이버 관리자 지정)를 참조하세요. 링크를 사용합니다.

기본적으로 ODBC 드라이버 관리자는 홈 디렉터리에 있는 숨겨진 버전의 `odbc.ini` 및 `odbcinst.ini` 구성 파일(`odbc.ini` 및 `odbcinst.ini`)을 사용하도록 구성됩니다. 또한 드라이버 설치 디렉터리의 `amazon.redshiftodbc.ini` 하위 폴더에 있는 `/lib` 파일을 사용하도록 구성됩니다. 이러한 구성 파일을 다른 위치에 저장하는 경우 드라이버 관리자가 파일을 찾을 수 있도록 환경 변수를 지정합니다. 자세한 내용은 Amazon Redshift ODBC Connector Installation and Configuration Guide의 "Specifying the Locations of the Driver Configuration Files" 섹션을 참조하세요.

연결 기능 구성

ODBC 설정에 대해 다음 연결 기능을 구성할 수 있습니다.

- 자격 증명을 제공하고 Amazon Redshift 데이터베이스 연결을 인증하도록 ODBC 드라이버를 구성합니다.
- 보안 소켓 계층(SSL)이 사용 설정된 Amazon Redshift 서버에 연결하는 경우 SSL에서 사용할 수 있는 소켓에 연결하도록 ODBC 드라이버를 구성합니다.
- 프록시 서버를 통해 Amazon Redshift에 연결하도록 ODBC 드라이버를 구성합니다.
- 쿼리가 너무 많은 메모리를 소비하지 않게 쿼리 처리 모드를 사용하도록 ODBC 드라이버를 구성합니다.
- 프록시 서버를 통해 IAM 인증 프로세스를 통과하도록 ODBC 드라이버를 구성합니다.
- TCP keepalive를 사용하여 연결 시간 초과가 발생하지 않도록 ODBC 드라이버를 구성합니다.

이러한 연결 기능에 대한 자세한 내용은 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)를 참조하세요.

ODBC 드라이버 옵션

구성 옵션을 사용하여 Amazon Redshift ODBC 드라이버의 동작을 제어할 수 있습니다.

Microsoft Windows에서는 일반적으로 데이터 원본 이름(DSN)을 구성할 때 드라이버 옵션을 설정합니다. 프로그래밍 방식으로 연결할 때 또는 HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\your_DSN에서 레지스트리 키를 추가하거나 변경하여 연결 문자열에 드라이버 옵션을 설정할 수도 있습니다. DSN 구성에 대한 자세한 내용은 [Microsoft Windows에서 Amazon Redshift ODBC 드라이버 사용](#) 섹션을 참조하세요.

macOS X에서는 [ODBC 드라이버 관리자를 사용하여 드라이버 구성](#) 섹션에서 설명한 바와 같이 드라이버 구성 옵션을 odbc.ini 및 amazon.redshiftodbc.ini 파일에서 설정합니다. amazon.redshiftodbc.ini 파일에서 설정하는 구성 옵션이 모든 연결에 적용됩니다. 반대로 odbc.ini 파일에서 설정하는 구성 옵션은 연결에 따라 다릅니다. odbc.ini에서 설정하는 구성 옵션이 amazon.redshiftodbc.ini에서 설정하는 구성 옵션보다 우선합니다.

ODBC 드라이버 구성 옵션을 설정하는 방법에 대한 자세한 내용은 [Amazon Redshift ODBC Connector Installation and Configuration Guide](#)(Amazon Redshift ODBC 커넥터 설치 및 구성 가이드)를 참조하세요.

이전 ODBC 드라이버 버전

도구에 특정 버전의 드라이버가 필요한 경우에만 이전 버전의 Amazon Redshift ODBC 드라이버를 다운로드합니다.

Windows용 이전 ODBC 드라이버 버전

다음은 64비트 드라이버입니다.

- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.3.1006/AmazonRedshiftODBC64-1.6.3.1006.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.1.1000/AmazonRedshiftODBC64-1.6.1.1000.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.20.1024/AmazonRedshiftODBC64-1.5.20.1024.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.16.1019/AmazonRedshiftODBC64-1.5.16.1019.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.9.1011/AmazonRedshiftODBC64-1.5.9.1011.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.7.1007/AmazonRedshiftODBC64-1.5.7.1007.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.65.1000/AmazonRedshiftODBC64-1.4.65.1000.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.62.1000/AmazonRedshiftODBC64-1.4.62.1000.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.59.1000/AmazonRedshiftODBC64-1.4.59.1000.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.56.1000/AmazonRedshiftODBC64-1.4.56.1000.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.53.1000/AmazonRedshiftODBC64-1.4.53.1000.msi>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.52.1000/AmazonRedshiftODBC64-1.4.52.1000.msi>

32비트 드라이버는 단종되었으며 이전 버전은 지원되지 않습니다.

Linux용 이전 ODBC 드라이버 버전

다음은 64비트 드라이버 버전입니다.

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.3.1006/AmazonRedshiftODBC-64-bit-1.6.3.1006-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.1.1000/AmazonRedshiftODBC-64-bit-1.6.1.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.20.1024/AmazonRedshiftODBC-64-bit-1.5.20.1024-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.16.1019/AmazonRedshiftODBC-64-bit-1.5.16.1019-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.9.1011/AmazonRedshiftODBC-64-bit-1.5.9.1011-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.7.1007/AmazonRedshiftODBC-64-bit-1.5.7.1007-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.65.1000/AmazonRedshiftODBC-64-bit-1.4.65.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.62.1000/AmazonRedshiftODBC-64-bit-1.4.62.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.59.1000/AmazonRedshiftODBC-64-bit-1.4.59.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.59.1000/AmazonRedshiftODBC-64-bit-1.4.59.1000-1.x86_64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.56.1000/AmazonRedshiftODBC-64-bit-1.4.56.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.56.1000/AmazonRedshiftODBC-64-bit-1.4.56.1000-1.x86_64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.52.1000/AmazonRedshiftODBC-64-bit-1.4.52.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.52.1000/AmazonRedshiftODBC-64-bit-1.4.52.1000-1.x86_64.deb

32비트 드라이버는 단종되었으며 이전 버전은 지원되지 않습니다.

macOS X용 이전 ODBC 드라이버 버전

다음은 macOS X용 Amazon Redshift ODBC 드라이버 버전입니다.

- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.3.1006/AmazonRedshiftODBC-64-bit.1.6.3.1006.universal.pkg>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.1.1000/AmazonRedshiftODBC-64-bit.1.6.1.1000.universal.pkg>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.20.1024/AmazonRedshiftODBC-1.5.20.1024.arm64.dmg>
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.20.1024/AmazonRedshiftODBC-1.5.20.1024.x86_64.dmg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.16.1019/AmazonRedshiftODBC-1.5.16.1019.x86_64.dmg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.9.1011/AmazonRedshiftODBC-1.5.9.1011.x86_64.dmg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.7.1007/AmazonRedshiftODBC-1.5.7.1007.x86_64.dmg
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.65.1000/AmazonRedshiftODBC-1.4.65.1000.dmg>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.62.1000/AmazonRedshiftODBC-1.4.62.1000.dmg>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.59.1000/AmazonRedshiftODBC-1.4.59.1000.dmg>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.56.1000/AmazonRedshiftODBC-1.4.56.1000.dmg>
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.52.1000/AmazonRedshiftODBC-1.4.52.1000.dmg>

연결을 위한 보안 옵션 구성

Amazon Redshift는 데이터와 서버 인증서 암호화를 위한 보안 소켓 계층(SSL) 연결을 지원하여 클라이언트에서 연결할 서버 인증서를 검증합니다.

SSL

SSL 연결을 지원하기 위해 Amazon Redshift는 [AWS Certificate Manager\(ACM\)](#)가 발급한 SSL 인증서를 생성해 각 클러스터에 설치합니다. ACM 인증서는 대부분의 운영 체제, 웹 브라우저 및 클라이언트에서 공개적으로 신뢰합니다. SQL 클라이언트 또는 애플리케이션이 `sslmode` 연결 옵션이 `require`, `verify-ca` 또는 `verify-full`로 설정된 상태에서 SSL을 사용하여 Amazon Redshift에 연결하는 경우 인증서 번들을 다운로드해야 합니다. 클라이언트에 인증서가 필요한 경우 Amazon Redshift는 다음과 같이 번들 인증서를 제공합니다.

- <https://s3.amazonaws.com/redshift-downloads/amazon-trust-ca-bundle.crt>에서 번들을 다운로드합니다.
- 예상 MD5 체크섬 번호는 418dea9b6d5d5de7a8f1ac42e164cdcf입니다.
- sha256 체크섬 번호는 36dba8e4b8041cd14b9d60158893963301bcb92e1c456847784de2acb5bd550입니다.

<https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt>에 있던 이전 인증서 번들을 사용하지 마세요.

- 중국 AWS 리전에서는 <https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/amazon-trust-ca-bundle.crt>에서 번들을 다운로드합니다.
- 예상 MD5 체크섬 번호는 418dea9b6d5d5de7a8f1ac42e164cdcf입니다.
- sha256 체크섬 번호는 36dba8e4b8041cd14b9d60158893963301bcb92e1c456847784de2acb5bd550입니다.

<https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ca-bundle.crt> 및 <https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ssl-ca-cert.pem>에 있던 이전 인증서 번들을 사용하지 마세요.

Important

Amazon Redshift는 SSL 인증서 관리 방법을 바꿨습니다. 계속해서 SSL을 사용하여 클러스터에 연결하려면 현재 신뢰하는 루트 CA 인증서를 업데이트해야 할 수 있습니다. 자세한 내용은 [SSL 연결을 위해 ACM 인증서로 이전](#) 섹션을 참조하세요.

기본적으로 클러스터 데이터베이스는 SSL의 사용 여부와 상관없이 연결을 허용합니다. 클러스터가 SSL 연결을 요구하도록 구성하려면 클러스터와 연결된 파라미터 그룹에서 `require_ssl` 파라미터를 `true`로 설정해야 합니다.

Amazon Redshift는 연방 정보 처리 표준(FIPS) 140-2를 준수하는 SSL 모드를 지원합니다. FIPS 준수 SSL 모드는 기본적으로 비활성화되어 있습니다.

Important

시스템에서 FIPS를 준수해야 하는 경우에만 FIPS 준수 SSL 모드를 활성화합니다.

FIPS 준수 SSL 모드를 활성화하려면 Amazon Redshift 클러스터 또는 Redshift Serverless 작업 그룹에 연결된 파라미터 그룹에서 `use_fips_ssl` 파라미터와 `require_ssl` 파라미터를 `true`로 설정합니다. 클러스터의 파라미터 그룹 수정에 대한 자세한 내용은 [Amazon Redshift 파라미터 그룹](#) 섹션을 참조하세요. 작업 그룹의 파라미터 그룹 수정에 대한 자세한 내용은 [Amazon Redshift Serverless에 대한 FIPS 준수 SSL 연결 구성](#) 섹션을 참조하세요.

Amazon Redshift는 ECDHE(Elliptic Curve Diffie—Hellman Ephemeral) 키 계약 프로토콜을 지원합니다. ECDHE 프로토콜에서는 클라이언트와 서버가 비보안 채널을 통해 공유 비밀을 구성하는 데 사용하는 타원 곡선 공개-비공개 키 페어를 갖습니다. ECDHE를 사용하기 위해 Amazon Redshift에서 아무 것도 구성할 필요가 없습니다. ECDHE를 통해 클라이언트와 서버 간 통신을 암호화하는 SQL 클라이언트 도구에서 연결하는 경우 Amazon Redshift가 제공된 암호 목록을 사용해 적절히 연결합니다. 자세한 내용은 Wikipedia의 [Elliptic curve diffie—hellman](#) 및 OpenSSL 웹 사이트의 [Ciphers](#) 섹션을 참조하세요.

ODBC의 SSL 및 신뢰하는 CA 인증서

최신 Amazon Redshift ODBC 드라이버(버전 1.3.7.1000 이상)을 사용하여 연결하는 경우 이 섹션을 건너뛸 수 있습니다. 최신 드라이버를 다운로드하려면 [Amazon Redshift용 ODBC 드라이버 버전 2.x 연결 구성](#) 섹션을 참조하세요.

계속해서 SSL을 사용하여 클러스터에 연결하려면 현재 신뢰하는 루트 CA 인증서를 업데이트해야 할 수 있습니다. 자세한 내용은 [SSL](#) 섹션을 참조하세요.

다운로드한 인증서가 예상 MD5 체크섬 번호와 일치하는지 확인할 수 있습니다. 이렇게 하려면 Linux 운영 체제에서 MD5sum 프로그램을 사용하거나 Windows 및 macOS X 운영 체제에서 다른 도구를 사용할 수 있습니다.

ODBC DSN에 포함되어 있는 `sslmode` 설정은 클라이언트 연결이나 서버 인증서 확인 시 암호화 처리 방법을 결정합니다. Amazon Redshift는 클라이언트 연결에서 다음 `sslmode` 값을 지원합니다.

- `disable`

SSL이 비활성화되고 연결이 암호화되지 않았습니다.

- allow

서버에서 요구할 경우 SSL이 사용됩니다.

- prefer

서버에서 지원할 경우 SSL이 사용됩니다. Amazon Redshift는 SSL을 지원하므로 `sslmode`를 `prefer`로 설정할 때 SSL을 사용합니다.

- require

SSL이 필요합니다.

- verify-ca

SSL을 사용하고, 서버 인증서의 검증을 받아야 합니다.

- verify-full

SSL을 사용해야 합니다. 또한 서버 인증서의 검증을 받아야 하며, 서버 호스트 이름이 인증서의 호스트 이름 속성과 일치해야 합니다.

클라이언트와 서버 간의 연결에서 SSL 사용 여부 및 서버 인증서 확인 여부를 확인할 수 있습니다. 이렇게 하려면 클라이언트의 ODBC DSN에 대한 `sslmode` 설정과 서버의 Amazon Redshift 클러스터에 대한 `require_SSL` 설정을 검토해야 합니다. 다음 표는 클라이언트와 서버 설정을 다양하게 조합한 암호화 결과를 설명한 것입니다.

sslmode(클라이언트)	require_SSL(서버)	결과
disable	false	연결이 암호화되지 않았습니다.
disable	true	서버가 SSL을 요구하지만 클라이언트는 SSL이 비활성화되어 있어서 연결이 이루어지지 않습니다.
allow	true	연결이 암호화됩니다.
allow	false	연결이 암호화되지 않았습니다.

sslmode(클라이언트)	require_SSL(서버)	결과
prefer 또는 require	true	연결이 암호화됩니다.
prefer 또는 require	false	연결이 암호화됩니다.
verify-ca	true	연결이 암호화되고, 서버 인증서가 검증됩니다.
verify-ca	false	연결이 암호화되고, 서버 인증서가 검증됩니다.
verify-full	true	연결이 암호화되고, 서버 인증서 및 호스트 이름이 검증됩니다.
verify-full	false	연결이 암호화되고, 서버 인증서 및 호스트 이름이 검증됩니다.

서버 인증서를 사용하여 Microsoft Windows 기반 ODBC와 연결하기

SSL 및 서버 인증서를 사용하여 클러스터에 연결하려면 먼저 인증서를 클라이언트 컴퓨터 또는 Amazon EC2 인스턴스에 다운로드합니다. 그런 다음 ODBC DSN을 구성합니다.

1. Amazon Redshift 인증 기관 번들을 클라이언트 컴퓨터의 드라이버 설치 디렉터리에 있는 lib 폴더로 다운로드하고 root.crt 파일로 저장합니다. 다운로드 정보는 [SSL](#) 섹션을 참조하세요.
2. ODBC Data Source Administrator(ODBC 데이터 원본 관리자)를 열고 ODBC 연결에 필요한 시스템 DSN 항목을 추가 또는 편집합니다. DNS 별칭을 사용하지 않는 경우 SSL 모드(SSL Mode)로 verify-full을 선택합니다. DNS 별칭을 사용하는 경우에는 verify-ca를 선택합니다. 그런 다음 저장(Save)을 선택합니다.

ODBC DSN 구성에 대한 자세한 내용은 [Amazon Redshift용 ODBC 드라이버 버전 2.x 연결 구성](#) 섹션을 참조하세요.

Java의 SSL 및 서버 인증서

SSL은 클라이언트와 클러스터 사이에 전송되는 데이터를 암호화하여 하나의 보안 계층으로서 역할을 합니다. 서버 인증서를 사용하여 해당 클러스터가 Amazon Redshift 클러스터라는 사실을 검증하면 보안 계층이 추가됩니다. 프로비저닝하는 모든 클러스터에 자동으로 설치되는 서버 인증서를 검사하기 때문에 이러한 검증이 가능합니다. JDBC에서 서버 인증서를 사용하는 방법에 대한 자세한 내용은 PostgreSQL 설명서에서 [클라이언트 구성](#)을 참조하세요.

Java에서 신뢰하는 CA 인증서를 사용해 연결

Important

Amazon Redshift는 SSL 인증서 관리 방법을 바꿨습니다. 계속해서 SSL을 사용하여 클러스터에 연결하려면 현재 신뢰하는 루트 CA 인증서를 업데이트해야 할 수 있습니다. 자세한 내용은 [SSL](#) 섹션을 참조하세요.

신뢰하는 CA 인증서를 사용해 연결하려면

redshift-keytool.jar 파일을 사용하여 Amazon Redshift 인증 기관 번들에 들어 있는 CA 인증서를 Java TrustStore 또는 프라이빗 TrustStore로 가져올 수 있습니다.

1. Java 명령줄 -Djavax.net.ssl.trustStore 옵션을 사용하는 경우 가능하면 명령줄에서 이 옵션을 제거합니다.
2. [redshift-keytool.jar](#)를 다운로드합니다.
3. 다음 중 하나를 수행하세요.

- Amazon Redshift 인증 기관 번들을 Java TrustStore로 가져오려면 다음 명령을 실행합니다.

```
java -jar redshift-keytool.jar -s
```

- Amazon Redshift 인증 기관 번들을 프라이빗 TrustStore로 가져오려면 다음 명령을 실행합니다.

```
java -jar redshift-keytool.jar -k <your_private_trust_store> -  
p <keystore_password>
```

SSL 연결을 위해 ACM 인증서로 이전

Amazon Redshift는 클러스터의 SSL 인증서를 [AWS Certificate Manager\(ACM\)](#)가 발급한 인증서로 바꿉니다. ACM은 대부분의 최신 시스템에서 신뢰하는 퍼블릭 인증 기관(CA)입니다. 계속해서 SSL을 사용하여 클러스터에 연결하려면 현재 신뢰하는 루트 CA 인증서를 업데이트해야 할 수 있습니다.

이러한 변경은 다음 사항이 모두 적용되는 경우에만 영향을 미칩니다.

- SQL 클라이언트 또는 애플리케이션이 sslMode 연결 옵션이 require, verify-ca 또는 verify-full 구성 옵션으로 설정된 상태에서 SSL을 사용해 Amazon Redshift 클러스터에 연결합니다.
- Amazon Redshift ODBC 또는 JDBC 드라이버를 사용하지 않거나 ODBC 버전 1.3.7.1000 또는 JDBC 버전 1.2.8.1005 이전의 Amazon Redshift 드라이버를 사용하고 있습니다.

이 변경 사항이 상용 Amazon Redshift 리전에서 영향을 미치는 경우 현재 신뢰하는 루트 CA 인증서를 2017년 10월 23일 이전에 업데이트해야 합니다. Amazon Redshift는 지금~2017년 10월 23일 사이에 ACM 인증서를 사용하도록 클러스터를 이전할 예정입니다. 이 변경은 클러스터의 성능 또는 가용성에 거의 또는 아예 영향을 미치지 않을 것입니다.

이 변경 사항이 AWS GovCloud (US)(미국) 리전에 영향을 미치는 경우 서비스 중단을 방지하기 위해 현재 신뢰하는 루트 CA 인증서를 2020년 4월 1일 이전에 업데이트해야 합니다. 이 날짜부터 SSL 암호화 연결을 사용하여 Amazon Redshift 클러스터에 연결하는 클라이언트에는 신뢰할 수 있는 인증 기관(CA)이 추가로 필요합니다. 클라이언트는 신뢰할 수 있는 인증 기관을 사용하여 Amazon Redshift 클러스터에 연결할 때 클러스터의 자격 증명을 확인합니다. 새로운 신뢰할 수 있는 CA가 포함된 업데이트된 인증서 번들을 사용하도록 SQL 클라이언트 및 애플리케이션을 업데이트하려면 작업이 필요합니다.

Important

2021년 1월 5일 중국 리전에서 Amazon Redshift는 클러스터의 SSL 인증서를 AWS Certificate Manager(ACM)가 발급한 인증서로 바꿉니다. 이 변경 사항이 중국(베이징) 리전 또는 중국(닝샤) 리전에 영향을 미치는 경우 서비스 중단을 방지하기 위해 현재 신뢰하는 루트 CA 인증서를 2021년 1월 5일 전에 업데이트해야 합니다. 이 날짜부터 SSL 암호화 연결을 사용하여 Amazon Redshift 클러스터에 연결하는 클라이언트에는 신뢰할 수 있는 인증 기관(CA)이 추가로 필요합니다. 클라이언트는 신뢰할 수 있는 인증 기관을 사용하여 Amazon Redshift 클러스터에 연결할 때 클러스터의 자격 증명을 확인합니다. 새로운 신뢰할 수 있는 CA가 포함된 업데이트된 인

증서 번들을 사용하도록 SQL 클라이언트 및 애플리케이션을 업데이트하려면 작업이 필요합니다.

- [최신 Amazon Redshift ODBC 또는 JDBC 드라이버 사용](#)
- [이전 Amazon Redshift ODBC 또는 JDBC 드라이버 사용](#)
- [다른 SSL 연결 유형 사용](#)

최신 Amazon Redshift ODBC 또는 JDBC 드라이버 사용

최신 Amazon Redshift ODBC 또는 JDBC 드라이버를 사용하는 것이 가장 기본적인 방법입니다. ODBC 버전 1.3.7.1000 및 JDBC 버전 1.2.8.1005부터 시작하는 Amazon Redshift 드라이버는 Amazon Redshift 자체 서명 인증서에서 ACM 인증서로의 이전을 자동으로 관리합니다. 최신 드라이버를 다운로드하려면 [Amazon Redshift용 JDBC 드라이버 버전 2.x 연결 구성](#) 섹션을 참조하세요.

최신 Amazon Redshift JDBC 드라이버를 사용하는 경우 JVM 옵션에서 -Djavax.net.ssl.trustStore를 사용하지 않는 것이 가장 좋습니다. -Djavax.net.ssl.trustStore를 사용해야 하는 경우에는 Redshift 인증 기관 번들을 이 번들이 가리키는 truststore로 가져옵니다. 다운로드 정보는 [SSL](#) 섹션을 참조하세요. 자세한 내용은 [Amazon Redshift 인증 기관 번들을 TrustStore로 가져오기](#) 섹션을 참조하세요.

이전 Amazon Redshift ODBC 또는 JDBC 드라이버 사용

- SSLCertPath를 사용하여 ODBC DSN을 구성한 경우 지정된 경로에 있는 인증서 파일을 덮어 씁니다.
- SSLCertPath가 설정되어 있지 않은 경우 드라이버 DLL 위치에 있는 root.crt라는 인증서 파일을 덮어 씁니다.

1.2.8.1005보다 이전 버전의 Amazon Redshift JDBC 드라이버를 사용해야 하는 경우 다음 중 하나를 수행하세요.

- JDBC 연결 문자열이 sslCert 옵션을 사용하는 경우 sslCert 옵션을 제거합니다. 그런 다음 Redshift 인증 기관 번들을 Java TrustStore로 가져옵니다. 다운로드 정보는 [SSL](#) 섹션을 참조하세요. 자세한 내용은 [Amazon Redshift 인증 기관 번들을 TrustStore로 가져오기](#) 섹션을 참조하세요.
- Java 명령줄 -Djavax.net.ssl.trustStore 옵션을 사용하는 경우 가능하면 명령줄에서 이 옵션을 제거합니다. 그런 다음 Redshift 인증 기관 번들을 Java TrustStore로 가져옵니다. 다운로드 정

보는 [SSL](#) 섹션을 참조하세요. 자세한 내용은 [Amazon Redshift 인증 기관 번들을 TrustStore로 가져오기](#) 섹션을 참조하세요.

Amazon Redshift 인증 기관 번들을 TrustStore로 가져오기

redshift-keytool.jar를 사용하여 Amazon Redshift 인증 기관 번들에 들어 있는 CA 인증서를 Java TrustStore 또는 프라이빗 TrustStore로 가져올 수 있습니다.

Amazon Redshift 인증 기관 번들을 TrustStore로 가져오려면

1. [redshift-keytool.jar](#)를 다운로드합니다.
2. 다음 중 하나를 수행하세요.
 - Amazon Redshift 인증 기관 번들을 Java TrustStore로 가져오려면 다음 명령을 실행합니다.

```
java -jar redshift-keytool.jar -s
```

- Amazon Redshift 인증 기관 번들을 프라이빗 TrustStore로 가져오려면 다음 명령을 실행합니다.

```
java -jar redshift-keytool.jar -k <your_private_trust_store> -  
p <keystore_password>
```

다른 SSL 연결 유형 사용

다음 중 하나를 사용하여 연결하는 경우 이 섹션의 단계를 따르세요.

- 오픈 소스 ODBC 드라이버
- 오픈 소스 JDBC 드라이버
- [Amazon Redshift RSQL](#) 명령줄 인터페이스
- libpq 기반으로 하는 모든 언어 바인딩(예: psycopg2(Python) 및 ruby-pg (Ruby))

다른 SSL 연결 유형을 사용하여 ACM 인증서를 사용하려면:

1. Amazon Redshift 인증 기관 번들을 다운로드합니다. 다운로드 정보는 [SSL](#) 섹션을 참조하세요.
2. 번들의 인증서를 root.crt 파일에 넣습니다.
 - Linux 및 macOS X 운영 체제에서 이 파일은 ~/.postgresql/root.crt입니다.

- Microsoft Windows에서는 %APPDATA%\postgresql\root.crt입니다.

클라이언트 도구 및 코드에서 연결

Amazon Redshift는 Amazon Redshift 쿼리 에디터 v2를 제공하여 클러스터와 작업 그룹에 연결할 수 있습니다. 자세한 내용은 [Query Editor V2를 사용하여 데이터베이스 쿼리](#) 섹션을 참조하세요.

이 섹션에서는 서드 파티 도구를 연결할 수 있는 몇 가지 옵션을 제공합니다. 또한 프로그래밍 방식으로 클러스터에 연결하는 방법에 대해서도 설명합니다.

주제

- [Amazon Redshift RSQL을 사용하여 연결](#)
- [Amazon Redshift RSQL을 사용하여 클러스터에 연결](#)
- [Amazon Redshift RSQL 메타 명령](#)
- [Amazon Redshift RSQL 변수](#)
- [Amazon Redshift RSQL 오류 코드](#)
- [Amazon Redshift RSQL 환경 변수](#)

Amazon Redshift RSQL을 사용하여 연결

Amazon Redshift RSQL은 Amazon Redshift 클러스터 및 데이터베이스와의 상호 작용을 위한 명령줄 클라이언트입니다. Amazon Redshift 클러스터에 연결하고, 데이터베이스 객체를 설명하고, 데이터를 쿼리하고, 다양한 출력 형식으로 쿼리 결과를 볼 수 있습니다.

Amazon Redshift RSQL은 Amazon Redshift에 특정한 추가 기능 집합과 함께 PostgreSQL psql 명령줄 도구의 기능을 지원합니다. 여기에는 다음이 포함됩니다.

- AD FS, PingIdentity, Okta, Azure ADm 또는 기타 SAML/JWT 기반 ID 제공업체를 통한 Single Sign-On을 사용할 수 있습니다. 멀티 팩터 인증(MFA)에 브라우저 기반 SAML 자격 증명 공급자를 사용할 수도 있습니다.
- 테이블 배포 키, 테이블 정렬 키, 후기 바인딩 보기(LBV) 및 구체화된 보기와 같은 Amazon Redshift 객체의 속성을 설명할 수 있습니다. AWS Glue 카탈로그 또는 Apache Hive Metastore, Amazon RDS for PostgreSQL, Amazon Aurora PostgreSQL 호환 버전, RDS for MySQL(평가판) 및 Amazon Aurora MySQL 호환 버전(평가판)의 외부 데이터베이스, Amazon Redshift 데이터 공유를 사용하여 공유되는 테이블에 있는 외부 테이블의 속성을 설명할 수도 있습니다.

- IF(\ELSEIF, \ELSE, \ENDIF), \GOTO, \LABEL 등의 향상된 제어 흐름 명령을 사용할 수도 있습니다.

입력 파라미터로 전달된 스크립트를 실행하는 Amazon Redshift RSQL 배치 모드를 사용하면 SQL 및 복잡한 비즈니스 로직을 모두 포함하는 스크립트를 실행할 수 있습니다. 기존 자체 관리형 온프레미스 데이터 웨어하우스가 있는 경우 Amazon Redshift RSQL을 사용하여 Teradata BTEQ 스크립트와 같은 기존 추출, 변환, 로드 및 자동화 스크립트를 대체할 수 있습니다. RSQL을 사용하면 절차 언어로 스크립트를 수동으로 다시 구현하지 않아도 됩니다.

Amazon Redshift RSQL은 Linux, Windows 및 macOS X 운영 체제에서 사용할 수 있습니다.

Amazon Redshift RSQL에 대한 문제를 보고하려면 <redshift-rsql-support@amazon.com>으로 이메일을 보내주세요.

주제

- [Amazon Redshift RSQL 시작하기](#)
- [Amazon Redshift RSQL 변경 로그](#)

Amazon Redshift RSQL 시작하기

Linux, macOS 또는 Microsoft Windows 운영 체제가 설치된 컴퓨터에 Amazon Redshift RSQL을 설치합니다.

RSQL 다운로드

- Linux 64비트 RPM: [RSQL 버전 1.1.2](#)
 - Linux 아티팩트 서명 키: [키](#)
 - Linux 아티팩트 서명된 해시: [해시](#)
- macOS 64비트 PKG: [RSQL 버전 1.1.2](#)
- Windows 64비트 MSI: [RSQL 버전 1.1.2](#)

이전 버전의 변경 로그 및 다운로드는 [Amazon Redshift RSQL 변경 로그](#)에서 확인하세요.

Linux용 RSQL 설치

아래 단계에 따라 Linux용 RSQL을 설치하세요.

1. 다음 명령을 사용하여 드라이버 관리자를 설치합니다.

```
sudo yum install unixODBC
```

2. ODBC 드라이버를 설치합니다. [Amazon Redshift ODBC 드라이버 다운로드 및 설치](#).
3. ini 파일을 홈 디렉터리에 복사합니다.

```
cp /opt/amazon/redshiftodbcx64/odbc.ini ~/.odbc.ini
```

4. 파일 위치를 가리키도록 환경 변수를 설정합니다.

```
export ODBCINI=~/.odbc.ini
export ODBCSYSINI=/opt/amazon/redshiftodbcx64/
export AMAZONREDSHIFTODBCINI=/opt/amazon/redshiftodbcx64/amazon.redshiftodbc.ini
```

5. 이제 다음 명령을 실행하여 RSQL을 설치할 수 있습니다.

```
sudo rpm -i AmazonRedshiftRsql-<version>.rhel.x86_64.rpm
```

Mac용 RSQL 설치

아래 단계에 따라 macOS X용 RSQL을 설치합니다.

1. 다음 명령을 사용하여 드라이버 관리자를 설치합니다.

```
brew install unixodbc --build-from-source
```

2. ODBC 드라이버를 설치합니다. [Amazon Redshift ODBC 드라이버 다운로드 및 설치](#).
3. ini 파일을 홈 디렉터리에 복사합니다.

```
cp /opt/amazon/redshift/Setup/odbc.ini ~/.odbc.ini
```

4. 파일 위치를 가리키도록 환경 변수를 설정합니다.

```
export ODBCINI=~/.odbc.ini
export ODBCSYSINI=/opt/amazon/redshift/Setup
export AMAZONREDSHIFTODBCINI=/opt/amazon/redshift/lib/amazon.redshiftodbc.ini
```

5. /usr/local/lib에 없으면 DYLD_LIBRARY_PATH를 libodbc.dylib의 위치로 설정합니다.

```
export DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:/usr/local/lib
```

6. pkg 파일을 두 번 클릭하여 설치 프로그램을 실행합니다.
7. 설치 관리자의 단계에 따라 설치를 완료합니다. 라이선스 계약 조건에 동의합니다.

Windows용 RSQL 설치

아래 단계에 따라 Windows용 RSQL을 설치합니다.

1. ODBC 드라이버를 설치합니다. [Amazon Redshift ODBC 드라이버 다운로드 및 설치](#).
2. RSQL 다운로드 파일을 두 번 클릭하여 설치 프로그램을 실행한 다음 화면 지시에 따라 설치를 완료합니다.

Amazon Redshift RSQL 변경 로그

1.1.2(2025년 12월 11일)

버그 수정

- \goto 및 \label 명령에서 실패를 일으킨 버그를 수정했습니다.
- 변수를 따옴표로 묶을 때 RSQL이 변수 값을 인쇄하지 못하는 버그를 수정했습니다.
- UseDeclareFetch가 활성화된 상태에서 쿼리 결과가 ODBC DSN 가져오기 파라미터 크기를 초과할 때 RSQL 충돌이 발생하는 버그를 수정했습니다.
- 페이지가 켜져 있더라도 여러 페이지의 결과가 한 번에 반환되는 문제를 수정했습니다.
- 트랜잭션 블록 내에서 쿼리가 실패할 때 RSQL 충돌을 트리거하는 버그를 수정했습니다.

1.1.1(2025-11-20)

버그 수정

- -c 플래그를 사용할 때 RSQL이 쿼리를 잘못 구문 분석하는 문제를 수정했습니다. 이 수정 사항은 모든 플랫폼에 적용됩니다.
- Mac에서 사용자가 RSQL에서 \s 명령을 사용하지 못하게 하는 버그를 수정했습니다.

1.1.0(2025-11-11)

버그 수정

- RSQL에서 예기치 않은 충돌을 일으키는 메모리 누수 문제를 해결했습니다.

- RSQL에서 OpenSSL 종속성을 제거했습니다.
- 동일한 환경에서 libpq/psql 설치와의 연결 충돌을 수정했습니다.
- Amazon Linux 2023, Windows 및 macOS의 플랫폼 호환성을 개선했습니다.
- 현재 표시 크기를 초과할 때 출력이 잘리는 문제를 수정했습니다.

1.0.8(2023-06-19)

버그 수정

- SHOW 명령으로 출력이 잘리는 문제를 수정했습니다.
- \de에 외부 Kinesis 스트림 및 Kafka 주제 설명을 위한 지원이 추가되었습니다.

1.0.7(2023-03-22)

버그 수정

- RSQL이 구체화된 뷰를 설명할 수 없는 문제가 수정되었습니다.
- Amazon Redshift Serverless를 사용할 때 stl_connection_log의 권한 거부 오류가 수정되었습니다.
- RSQL이 \GOTO 레이블을 잘못 처리할 수 있는 문제가 해결되었습니다.
- 자동 모드에서 SSL 메시지가 출력되는 문제가 해결되었습니다.
- 저장된 프로시저를 설명할 때 임의의 문자가 표시되는 문제가 해결되었습니다.
- 중복된 오류/정보 메시지가 출력되는 문제가 해결되었습니다.

New

- RSQL이 이제 ODBC 드라이버에서 직접 SSL 정보를 가져옵니다.

1.0.6(2023-02-21)

버그 수정

- Redshift 패치 1.0.46086(P173)에서 \d가 잘못된 정수 입력 구문: 'xid'라는 오류가 발생하는 문제를 수정했습니다.

New

- 지원되는 아키텍처를 반영하도록 설치 파일의 이름을 변경했습니다.

1.0.5(2022-06-27)

버그 수정

- SQL 오류 메시지를 표준 오류(stderr)로 보냅니다.
- ON_ERROR_STOP을 사용할 때 발생하는 종료 코드 문제를 해결했습니다. 이제 오류가 발생하면 스크립트가 종료되고 올바른 종료 코드를 반환합니다.
- Maxerror가 이제 대/소문자를 구분하지 않습니다.

New

- ODBC 2.x 드라이브에 대한 지원이 추가되었습니다.

1.0.4(2022-03-19)

- RSPASSWORD 환경 변수에 대한 지원을 추가합니다. Amazon Redshift Redshift에 연결할 암호를 설정합니다. 예를 들어 `export RSPASSWORD=TestPassw0rd`입니다.

1.0.3(2021-12-08)

버그 수정

- Windows OS에서 \c 또는 \logon을 사용하여 데이터베이스 간에 전환할 때 대화 상자 팝업이 수정되었습니다.
- ssl 정보를 확인할 때 발생하는 충돌이 수정되었습니다.

Amazon Redshift RSQL 이전 버전

링크 중 하나를 선택하여 운영 체제에 따라 필요한 Amazon Redshift RSQL 버전을 다운로드합니다.

Linux 64비트 RPM

- [RSQL 버전 1.1.1](#)

- [RSQL 버전 1.1.0](#)
- [RSQL 버전 1.0.8](#)
- [RSQL 버전 1.0.7](#)
- [RSQL 버전 1.0.6](#)
- [RSQL 버전 1.0.5](#)
- [RSQL 버전 1.0.4](#)
- [RSQL 버전 1.0.3](#)
- [RSQL 버전 1.0.1](#)

macOS 64비트 DMG/PKG

- [RSQL 버전 1.1.1](#)
- [RSQL 버전 1.1.0](#)
- [RSQL 버전 1.0.8](#)
- [RSQL 버전 1.0.7](#)
- [RSQL 버전 1.0.6](#)
- [RSQL 버전 1.0.5](#)
- [RSQL 버전 1.0.4](#)
- [RSQL 버전 1.0.3](#)
- [RSQL 버전 1.0.1](#)

Windows 64비트 MSI

- [RSQL 버전 1.1.1](#)
- [RSQL 버전 1.1.0](#)
- [RSQL 버전 1.0.8](#)
- [RSQL 버전 1.0.7](#)
- [RSQL 버전 1.0.6](#)
- [RSQL 버전 1.0.5](#)
- [RSQL 버전 1.0.4](#)
- [RSQL 버전 1.0.3](#)
- [RSQL 버전 1.0.1](#)

Amazon Redshift RSQL을 사용하여 클러스터에 연결

Amazon Redshift를 사용하면 RSQL로 클러스터에 연결하고 상호 작용할 수 있습니다. 이는 데이터를 쿼리하고, 데이터베이스 객체를 생성하고, Amazon Redshift 클러스터를 관리하는 안전한 방법을 제공하는 명령줄 도구입니다. 다음 섹션에서는 데이터 소스 이름(DSN)이 있거나 없는 RSQL을 사용하여 클러스터에 대한 연결을 설정하는 단계를 안내합니다.

DSN을 사용하지 않고 연결

1. Amazon Redshift 콘솔에서 연결할 클러스터를 선택하고 엔드포인트, 데이터베이스 및 포트를 기록해 둡니다.
2. 명령 프롬프트에서 명령줄 파라미터를 사용하여 연결 정보를 지정합니다.

```
rsql -h <endpoint> -U <username> -d <databasename> -p <port>
```

여기에는 다음이 적용됩니다.

- **<endpoint>**은(는) 이전 단계에서 기록한 엔드포인트(Endpoint)입니다.
- **<username>**은 클러스터 연결 권한이 있는 사용자의 이름입니다.
- **<databasename>**은(는) 이전 단계에서 입력한 데이터베이스 이름(Database Name)입니다.
- **<port>**는 이전 단계에서 기록한 포트(Port)입니다. **<port>**는 선택적 파라미터입니다.

예를 들면 다음과 같습니다.

```
rsql -h testcluster.example.amazonaws.com -U user1 -d dev -p 5439
```

3. 암호 프롬프트에서 **<username>** 사용자의 암호를 입력합니다.

성공적인 연결 응답은 다음과 같습니다.

```
% rsql -h testcluster.example.com -d dev -U user1 -p 5349
Password for user user1:
DSN-less Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.
```



```
(testcluster) user1@dev=#
```

연결 명령은 Linux, Mac OS 및 Windows에서 동일한 파라미터를 사용합니다.

DSN으로 연결

DSN을 통해 연결 속성 구성을 간소화하여 RSQL을 Amazon Redshift에 연결할 수 있습니다. 이 주제에는 ODBC 드라이버 설치 설명과 DSN 속성 설명이 포함되어 있습니다.

암호와 함께 DSN 연결 사용

다음은 암호를 사용하는 DSN 연결 구성의 예를 보여줍니다. 기본 <path to driver>은 Mac OSX의 경우 /opt/amazon/redshift/lib/libamazonredshiftodbc.dylib이고 Linux의 경우 /opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so입니다.

```
[testuser]
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
SSLMode=verify-ca
Min_TLS=1.2
boolsaschar=0
Host=<server endpoint>
Port=<database port>
Database=<dbname>
UID=<username>
PWD=<password>
sslmode=prefer
```

다음 출력은 성공적인 연결의 결과입니다.

```
% rsql -D testuser
DSN Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.

(testcluster) user1@dev=#
```

통합 인증 DSN 사용

Single Sign-On 인증을 위해 DSN을 구성할 수 있습니다. 다음은 Okta Single Sign-On을 사용하는 DSN 연결 구성의 예를 보여줍니다.

```
[testokta]
Driver=<path to driver>
SSLMode=verify-ca
Min_TLS=1.2
boolsaschar=0
Host=<server endpoint>
clusterid=<cluster id>
region=<region name>
Database=<dbname>
locale=en-US
iam=1
plugin_name=<plugin name>
uid=<okta username>
pwd=<okta password>
idp_host=<idp endpoint>
app_id=<app id>
app_name=<app name>
preferred_role=<role arn>
```

성공적인 연결의 샘플 출력입니다.

```
% rsql -D testokta
DSN Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.

(testcluster) user1@dev=#
```

다음은 Azure Single Sign-On을 사용하는 DSN 연결 구성의 예를 보여줍니다.

```
[testazure]
Driver=<path to driver>
SSLMode=verify-ca
```

```

Min_TLS=1.2
boolsaschar=0
Host=<server endpoint>
Port=<cluster port>
clusterid=<cluster id>
region=<region name>
Database=<dbname>
locale=en-us
iam=1
plugin_name=<plugin name>
uid=<azure username>
pwd=<azure password>
idp_tenant=<Azure idp tenant uuid>
client_id=<Azure idp client uuid>
client_secret=<Azure idp client secret>

```

IAM 프로파일과 함께 DSN 연결 사용

구성된 IAM 프로파일을 사용하여 Amazon Redshift에 연결할 수 있습니다. IAM 프로파일에 GetClusterCredentials를 호출할 수 있는 권한이 있어야 합니다. 다음 예에서는 사용할 DSN 속성을 보여줍니다. ClusterID 및 Region 파라미터는 Host가 examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com과 같이 Amazon에서 제공하는 엔드포인트가 아닌 경우에만 필요합니다.

```

[testiam]
Driver=Default
Host=testcluster.example.com
Database=dev
DbUser=testuser
ClusterID=rsqltestcluster
Region=us-east-1
IAM=1
Profile=default

```

Profile 키 값은 AWS CLI 자격 증명에서 선택하는 명명된 프로파일입니다. 이 예에서는 default라는 프로파일에 대한 자격 증명을 보여줍니다.

```

$ cat .aws/credentials
[default]
aws_access_key_id = ASIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

```

다음은 연결 응답을 보여줍니다.

```
$ rsql -D testiam
DSN Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.

(testcluster) testuser@dev=>
```

인스턴스 프로파일과 함께 DSN 연결 사용

Amazon EC2 인스턴스 프로파일을 사용하여 Amazon Redshift에 연결할 수 있습니다. 인스턴스 프로파일에 GetClusterCredentials를 호출할 수 있는 권한이 있어야 합니다. 사용할 DSN 속성은 아래 예를 참조하세요. ClusterID 및 Region 파라미터는 Host가 examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com과 같이 Amazon에서 제공하는 엔드포인트가 아닌 경우에만 필요합니다.

```
[testinstanceprofile]
Driver=Default
Host=testcluster.example.com
Database=dev
DbUser=testuser
ClusterID=rsqltestcluster
Region=us-east-1
IAM=1
Instanceprofile=1
```

다음은 연결 응답을 보여줍니다.

```
$ rsql -D testinstanceprofile
DSN Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.
```

```
(testcluster) testuser@dev=>
```

기본 자격 증명 공급자 체인에서 DSN 연결 사용

기본 자격 증명 공급자 체인을 사용하여 연결하기 위해 IAM 속성만 지정하면 Amazon Redshift RSQL 이 Java용 AWS SDK에서 [AWS 자격 증명으로 작업하기](#)에 나와 있는 순서대로 자격 증명 획득을 시도 합니다. 체인에서 제공자 하나 이상이 GetClusterCredentials 권한이 있어야 합니다. 이것은 예를 들어 ECS 컨테이너에서 연결할 때 유용합니다.

```
[iamcredentials]
Driver=Default
Host=testcluster.example.com
Database=dev
DbUser=testuser
ClusterID=rsqltestcluster
Region=us-east-1
IAM=1
```

Amazon Redshift RSQL 메타 명령

Amazon Redshift RSQL 메타 명령은 데이터베이스 또는 특정 데이터베이스 객체에 대한 정보 레코드를 반환합니다. 결과에는 다양한 열과 메타데이터가 포함될 수 있습니다. 다른 명령은 특정 작업을 수행합니다. 이러한 명령 앞에는 백슬래시를 붙입니다.

`\d[S+]`

로컬 사용자 생성 테이블, 일반 보기, 후기 바인딩 보기 및 구체화된 보기를 나열합니다. `\dS` 는 `\d`와 같은 테이블과 보기도 나열하지만 반환된 레코드에는 시스템 객체가 포함됩니다. `+`는 나열된 모든 객체에 대한 추가 메타데이터 열 `description`을 생성합니다. 다음은 명령의 결과로 반환된 샘플 레코드를 보여줍니다.

```
List of relations
 schema |   name   | type | owner
-----+-----+-----+-----
 public | category | table | awsuser
 public | date     | table | awsuser
 public | event    | table | awsuser
 public | listing  | table | awsuser
 public | sales    | table | awsuser
```

```
public | users      | table | awsuser
public | venue        | table | awsuser
(7 rows)
```

\d[S+] NAME

테이블, 보기 또는 인덱스에 대해 설명합니다. 열 이름 및 유형을 포함합니다. 또한 diststyle, 백업 구성, 생성 날짜(2018년 10월 이후 생성된 테이블) 및 제약 조건을 제공합니다. 예를 들어, \dS+ sample은 객체 속성을 반환합니다. S+를 추가하면 반환된 레코드에 추가 열이 포함됩니다.

```
Table "public.sample"
Column |          Type          | Collation | Nullable | Default Value |
Encoding | DistKey | SortKey
-----+-----+-----+-----+-----+
+-----+-----+-----+
col1   | smallint                |           | NO       |               |
none   | t                      | 1
col2   | character(100)          | case_sensitive | YES     |               |
none   | f                      | 2
col3   | character varying(100)  | case_sensitive | YES     |               |
text32k | f                      | 3
col4   | timestamp without time zone |           | YES     |               |
runlength | f                      | 0
col5   | super                   |           | YES     |               |
zstd   | f                      | 0
col6   | bigint                  |           | YES     |               |
az64   | f                      | 0
```

Diststyle: KEY

Backup: YES

Created: 2021-07-20 19:47:27.997045

Unique Constraints:

"sample_pkey" PRIMARY KEY (col1)

"sample_col2_key" UNIQUE (col2)

Foreign-key constraints:

"sample_col2_fkey" FOREIGN KEY (col2) REFERENCES lineitem(l_orderkey)

테이블의 배포 스타일 또는 Diststyle은 KEY, AUTO, EVEN 또는 ALL일 수 있습니다.

백업(Backup)은 스냅샷이 생성될 때 테이블이 백업되는지 여부를 나타냅니다. 유효한 값은 YES 또는 NO입니다.

생성됨(Created)은 테이블이 생성될 때의 타임스탬프입니다. 2018년 11월 이전에 생성된 Amazon Redshift 테이블에는 생성 날짜를 사용할 수 없습니다. 이 날짜 이전에 생성된 테이블은 n/a(해당 사항 없음)으로 표시됩니다.

고유 제약 조건(Unique Constraints)에는 테이블에 대한 고유 및 기본 키 제약 조건이 나열됩니다.

외래 키 제약 조건(Foreign-key constraints)에는 테이블에 대한 외래 키 제약 조건이 나열됩니다.

\dC[+] [PATTERN]

캐스트를 나열합니다. 소스 유형, 대상 유형 및 캐스트가 암시적인지 여부를 포함합니다.

다음은 \dC+ 결과의 하위 집합을 보여줍니다.

List of casts

source type		target type	function	
implicit?	description			
-----+-----+-----				
+-----+	+-----+			
"char"	character	bpchar	in	
assignment				
"char"	character varying	text	in	
assignment				
"char"	integer	int4	no	
"char"	text	text	yes	
"path"	point	point	no	
"path"	polygon	polygon	in	
assignment				
abstime	date	date	in	
assignment				
abstime	integer	(binary coercible)	no	
abstime	time without time zone	time	in	
assignment				
abstime	timestamp with time zone	timestamptz	yes	
abstime	timestamp without time zone	timestamp	yes	
bigint	bit	bit	no	

bigint		boolean		bool		yes
bigint		character		bpchar		in
assignment		character varying		text		in
assignment		double precision		float8		yes
bigint		integer		int4		in
assignment		numeric		numeric		yes
bigint		oid		oid		yes
bigint		real		float4		yes
bigint		regclass		oid		yes
bigint		regoper		oid		yes
bigint		regoperator		oid		yes
bigint		regproc		oid		yes
bigint		regprocedure		oid		yes
bigint		regtype		oid		yes
bigint		smallint		int2		in
assignment		super		int8_partiql		in
assignment						

\dd[S] [PATTERN]

다른 곳에 표시되지 않는 객체 설명을 보여줍니다.

\de

외부 테이블을 나열합니다. 여기에는 AWS Glue Data Catalog의 테이블, Hive Metastore 및 Amazon RDS/Aurora MySQL, Amazon RDS/Aurora PostgreSQL 및 Amazon Redshift 데이터 공유 테이블의 페더레이션 테이블이 포함됩니다.

\de NAME

외부 테이블을 설명합니다.

다음 예에서는 AWS Glue 외부 테이블을 보여줍니다.

```
# \de spectrum.lineitem

                                Glue External table "spectrum.lineitem"
  Column      | External Type | Redshift Type | Position | Partition Key | Nullable
-----+-----+-----+-----+-----+-----
l_orderkey    | bigint        | bigint        | 1        | 0              |
l_partkey     | bigint        | bigint        | 2        | 0              |
l_suppkey     | int           | int           | 3        | 0              |
l_linenumber  | int           | int           | 4        | 0              |
l_quantity    | decimal(12,2) | decimal(12,2) | 5        | 0              |
l_extendedprice | decimal(12,2) | decimal(12,2) | 6        | 0              |
l_discount    | decimal(12,2) | decimal(12,2) | 7        | 0              |
l_tax         | decimal(12,2) | decimal(12,2) | 8        | 0              |
l_returnflag  | char(1)       | char(1)       | 9        | 0              |
l_linestatus  | char(1)       | char(1)       | 10       | 0              |
l_shipdate    | date          | date          | 11       | 0              |
l_commitdate  | date          | date          | 12       | 0              |
l_receiptdate | date          | date          | 13       | 0              |
l_shipinstruct | char(25)      | char(25)      | 14       | 0              |
l_shipmode    | char(10)      | char(10)      | 15       | 0              |
l_comment     | varchar(44)   | varchar(44)   | 16       | 0              |

Location: s3://redshiftbucket/kfhose2019/12/31
Input_format: org.apache.hadoop.mapred.TextInputFormat
Output_format: org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat
Serialization_lib: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
Serde_parameters: {"field.delim": "|", "serialization.format": "|"}
Parameters:
{"EXTERNAL": "TRUE", "numRows": "178196721475", "transient_lastDdlTime": "1577771873"}
```

Hive 메타스토어 테이블입니다.

```
# \de emr.lineitem

                                Hive Metastore External Table "emr.lineitem"
  Column      | External Type | Redshift Type | Position | Partition Key | Nullable
-----+-----+-----+-----+-----+-----
l_orderkey    | bigint        | bigint        | 1        | 0              |
```

l_partkey	bigint	bigint	2	0	
l_suppkey	int	int	3	0	
l_linenumber	int	int	4	0	
l_quantity	decimal(12,2)	decimal(12,2)	5	0	
l_extendedprice	decimal(12,2)	decimal(12,2)	6	0	
l_discount	decimal(12,2)	decimal(12,2)	7	0	
l_tax	decimal(12,2)	decimal(12,2)	8	0	
l_returnflag	char(1)	char(1)	9	0	
l_linestatus	char(1)	char(1)	10	0	
l_commitdate	date	date	11	0	
l_receiptdate	date	date	12	0	
l_shipinstruct	char(25)	char(25)	13	0	
l_shipmode	char(10)	char(10)	14	0	
l_comment	varchar(44)	varchar(44)	15	0	
l_shipdate	date	date	16	1	

Location: s3://redshiftbucket/cetas

Input_format: org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat

Output_format: org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat

Serialization_lib: org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe

Serde_parameters: {"serialization.format":"1"}

Parameters: {"EXTERNAL":"TRUE", "numRows":"4307207",
"transient_lastDdlTime":"1626990007"}

PostgreSQL 외부 테이블입니다.

```
# \de pgrsql.alltypes
```

Postgres Federated Table "pgrsql.alltypes"

Column	External Type	Redshift Type	Position
Partition Key	Nullable		
-----+-----			-----
+-----+-----			
col1	bigint	bigint	1 0
col2	bigint	bigint	2 0
col5	boolean	boolean	3 0
col6	box	varchar(65535)	4 0
col7	bytea	varchar(65535)	5 0

col8	character(10)	character(10)	6	0
col9	character varying(10)	character varying(10)	7	0
col10	cidr	varchar(65535)	8	0
col11	circle	varchar(65535)	9	0
col12	date	date	10	0
col13	double precision	double precision	11	0
col14	inet	varchar(65535)	12	0
col15	integer	integer	13	0
col16	interval	varchar(65535)	14	0
col17	json	varchar(65535)	15	0
col18	jsonb	varchar(65535)	16	0
col19	line	varchar(65535)	17	0
col20	lseg	varchar(65535)	18	0
col21	macaddr	varchar(65535)	19	0
col22	macaddr8	varchar(65535)	20	0
col23	money	varchar(65535)	21	0
col24	numeric	numeric(38,20)	22	0
col25	path	varchar(65535)	23	0
col26	pg_lsn	varchar(65535)	24	0
col28	point	varchar(65535)	25	0
col29	polygon	varchar(65535)	26	0
col30	real	real	27	0

col31	smallint	smallint	28	0
col32	smallint	smallint	29	0
col33	integer	integer	30	0
col34	text	varchar(65535)	31	0
col35	time without time zone	varchar(65535)	32	0
col36	time with time zone	varchar(65535)	33	0
col37	timestamp without time zone	timestamp without time zone	34	0
col38	timestamp with time zone	timestamp with time zone	35	0
col39	tsquery	varchar(65535)	36	0
col40	tsvector	varchar(65535)	37	0
col41	txid_snapshot	varchar(65535)	38	0
col42	uuid	varchar(65535)	39	0
col43	xml	varchar(65535)	40	0

`\df[anptw][S+] [PATTERN]`

다양한 유형의 함수를 나열합니다. 예를 들어 `\df` 명령은 함수 목록을 반환합니다. 결과에는 이름, 반환된 데이터 유형, 액세스 권한 및 추가 메타데이터와 같은 속성이 포함됩니다. 함수 유형에는 트리거, 저장 프로시저, 윈도우 함수 및 기타 유형이 포함될 수 있습니다. 명령에 `S+`를 추가하면(예: `\dfantS+`) owner, security, access privileges 등의 추가 메타데이터 열이 포함됩니다.

`\dL[S+] [PATTERN]`

데이터베이스와 연결된 절차 언어에 대한 데이터를 나열합니다. 정보에는 `plpgsql`과 같은 이름과 신뢰할 수 있는지 여부, 액세스 권한 및 설명이 포함된 추가 메타데이터가 포함됩니다. 예를 들어 샘플 호출은 언어 및 해당 속성을 나열하는 `\dLS+`입니다. 명령에 `S+`를 추가하면 `call handler`, `access privileges` 등의 추가 메타데이터 열이 포함됩니다.

샘플 결과:

List of languages

name	trusted	internal language	call handler	description
validator			access privileges	
-----+-----+-----+-----				
+-----+-----+-----+-----				
+-----				
c	f	t	-	
fmgr_c_validator(oid)				
Dynamically-loaded C functions				
exfunc	f	f	exfunc_call_handler()	-
			rdsdb=U/rdsdb	
internal	f	t	-	
fmgr_internal_validator(oid)				
Built-in functions				
mlfunc	f	f	mlfunc_call_handler()	-
			rdsdb=U/rdsdb	
plpgsql	t	f	plpgsql_call_handler()	
plpgsql_validator(oid)				
plpythonu	f	f	plpython_call_handler()	
plpython_compiler(cstring,cstring,cstring,cstring,cstring)			rdsdb=U/rdsdb	
sql	t	t	-	
fmgr_sql_validator(oid)			=U/rdsdb	SQL-
language functions				

\dm[S+] [PATTERN]

구체화된 보기를 나열합니다. 예를 들어 \dmS+는 구체화된 보기와 해당 속성을 나열합니다. 명령에 S+를 추가하면 추가 메타데이터 열이 포함됩니다.

\dn[S+] [PATTERN]

스키마를 나열합니다. 명령에 S+를 추가하면(예: \dnS+) description, access privileges등의 추가 메타데이터 열이 포함됩니다.

\dp [PATTERN]

테이블, 보기 및 시퀀스 액세스 권한을 나열합니다.

`\dt[S+] [PATTERN]`

테이블을 나열합니다. 명령에 S+를 추가하면(예: `\dtS+`) 이 경우 `description` 등의 추가 메타데이터 열이 포함됩니다.

`\du`

데이터베이스의 사용자를 나열합니다. 해당 이름과 슈퍼 사용자 및 속성 등의 역할을 포함합니다.

`\dv[S+] [PATTERN]`

보기를 나열합니다. 스키마, 유형 및 소유자 데이터를 포함합니다. 명령에 S+를 추가하면(예: `\dvS+`) 추가 메타데이터 열이 포함됩니다.

`\H`

HTML 출력을 설정합니다. 이는 형식이 지정된 결과를 빠르게 반환하는 데 유용합니다. 예를 들어 `select * from sales;` `\H`는 `sales` 테이블의 결과를 HTML로 반환합니다. 테이블형 결과로 다시 전환하려면 `\q` 또는 `quiet`을 사용합니다.

`\i`

파일에서 명령을 실행합니다. 예를 들어, 작업 디렉터리에 `rsql_steps.sql`이 있다고 가정하면 `\i rsql_steps.sql`은 파일에서 명령을 실행합니다.

`\l[+] [PATTERN]`

데이터베이스를 나열합니다. 소유자, 인코딩 및 추가 정보가 포함됩니다.

`\q`

`quit` 또는 `\q` 명령은 데이터베이스 세션을 로그오프하고 RSQL을 종료합니다.

`\sv[+] VIEWNAME`

보기의 정의를 표시합니다.

`\timing`

예를 들어 쿼리의 실행 시간을 표시합니다.

`\z [PATTERN]`

`\dp`와 동일한 출력입니다.

\?

도움말 정보를 표시합니다. 이 선택적 파라미터는 설명할 항목을 지정합니다.

\EXIT

모든 데이터베이스 세션을 로그오프하고 Amazon Redshift RSQL을 종료합니다. 또한 선택적 종료 코드를 지정할 수 있습니다. 예를 들어, \EXIT 15는 Amazon Redshift RSQL 터미널을 종료하고 종료 코드 15를 반환합니다.

다음 예는 연결 및 RSQL 종료의 출력을 보여줍니다.

```
% rsq1 -D testuser
DSN Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.34.1000
Rsql Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.

(testcluster) user1@dev=# \exit 15

% echo $?
15
```

\EXPORT

RSQL이 후속 SQL SELECT 문에서 반환하는 데이터베이스 정보를 저장하는 데 사용하는 내보내기 파일의 이름을 지정합니다.

export_01.sql

```
\export report file='E:\\accounts.out'
\rset rformat off
\rset width 1500
\rset heading "General Title"
\rset titledashes on
select * from td_dwh.accounts;
\export reset
```

콘솔 출력

```
Rformat is off.
Target width is 1500.
Heading is set to: General Title
Titledashes is on.
(exported 40 rows)
```

\LOGON

데이터베이스에 연결합니다. 위치 구문이나 연결 문자열을 사용하여 연결 파라미터를 지정할 수 있습니다.

명령 구문은 `\logon {[DBNAME]- USERNAME|- HOST|- PORT|- [PASSWORD]] | conninfo}`입니다.

DBNAME은 연결할 데이터베이스의 이름입니다. USERNAME은 연결할 사용자 이름입니다. 기본 HOST는 localhost입니다. 기본 PORT는 5439입니다.

\LOGON 명령에 지정된 호스트 이름이 추가 \LOGON 명령에 대한 기본 호스트 이름이 됩니다. 기본 호스트 이름을 변경하려면 추가 \LOGON 명령에 새 HOST를 지정합니다.

user1에 대한 \LOGON 명령의 샘플 출력은 다음과 같습니다.

```
(testcluster) user1@redshiftdb=# \logon dev
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
You are now connected to database "dev" as user "user1".
(testcluster) user1@dev=#
```

user2에 대한 샘플 출력

```
(testcluster) user1@dev=# \logon dev user2 testcluster2.example.com
Password for user user2:
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
You are now connected to database "dev" as user "user2" on host
"testcluster2.example.com" at port "5439".
(testcluster2) user2@dev=#
```


\REMARK

\echo 명령의 확장입니다. \REMARK는 지정된 문자열을 출력 스트림에 출력합니다. \REMARK 는 출력을 분리하는 기능을 추가하여 \echo를 확장합니다.

다음 샘플은 명령의 출력을 보여줍니다.

```
(testcluster) user1@dev=# \remark 'hello//world'
hello
world
```

\RSET

\rset 명령은 명령 파라미터와 변수를 설정합니다. \rset에는 대화형 모드와 배치 모드가 모두 있습니다. -x와 같은 bash 옵션이나 --<arg>와 같은 인수의 옵션을 지원하지 않습니다.

다음과 같은 변수를 설정합니다.

- ERRORLEVEL
- HEADING 및 RTITLE
- RFORMAT
- MAXERROR
- TITLEDASHES
- WIDTH

다음 예에서는 헤딩을 지정합니다.

```
\rset heading "Winter Sales Report"
```

\rset 사용 방법의 추가 예제를 보려면 [Amazon Redshift RSQL 변수](#) 주제에서 여러 예를 확인할 수 있습니다.

\RUN

지정된 파일에 포함된 Amazon Redshift RSQL 스크립트를 실행합니다. \RUN은 파일의 헤더 행을 건너뛰는 옵션을 추가하여 \i 명령을 확장합니다.

파일 이름에 쉼표, 세미콜론 또는 공백이 포함된 경우 작은따옴표로 묶습니다. 또한 텍스트가 파일 이름 뒤에 오는 경우 따옴표로 텍스트를 묶습니다. UNIX에서 파일 이름은 대/소문자를 구분합니다. Windows에서 파일 이름은 대/소문자를 구분하지 않습니다.

다음 샘플은 명령의 출력을 보여줍니다.

```
(testcluster) user1@dev=# \! cat test.sql
select count(*) as lineitem_cnt from lineitem;
select count(*) as customer_cnt from customer;
select count(*) as orders_cnt from orders;

(testcluster) user1@dev=# \run file=test.sql
lineitem_cnt
-----
          4307207
(1 row)

customer_cnt
-----
        37796166
(1 row)

orders_cnt
-----
          0
(1 row)

(testcluster) user1@dev=# \run file=test.sql skip=2
2 records skipped in RUN file.
orders_cnt
-----
          0
(1 row)
```

\OS

\! 명령의 별칭입니다. \OS는 파라미터로 전달된 운영 체제 명령을 실행합니다. 명령이 실행되면 컨트롤이 Amazon Redshift RSQL로 돌아갑니다. 예를 들어, \os date 명령을 실행하여 현재 시스템 날짜 시간을 인쇄하고 RSQL 터미널로 돌아갈 수 있습니다.

```
(testcluster) user1@dev=# \os date
Tue Sep 7 20:47:54 UTC 2021
```

\GOTO

Amazon Redshift RSQL을 위한 새로운 명령입니다. \GOTO는 중간에 있는 모든 명령을 건너뛰고 지정된 \LABEL에서 처리를 재개합니다. \LABEL은 정방향 참조여야 합니다. 사전적으로 \GOTO 앞에 오는 \LABEL을 건너뛸 수 없습니다.

다음은 샘플 출력을 보여줍니다.

```
(testcluster) user1@dev=# \! cat test.sql
select count(*) as cnt from lineitem \gset
select :cnt as cnt;
\if :cnt > 100
    \goto LABELB
\endif

\label LABELA
\remark 'this is label LABELA'
\label LABELB
\remark 'this is label LABELB'
```

```
(testcluster) user1@dev=# \i test.sql
cnt
-----
4307207
(1 row)

\label LABELA ignored
\label LABELB processed
this is label LABELB
```

\LABEL

Amazon Redshift RSQL을 위한 새로운 명령입니다. \LABEL은 프로그램을 실행하기 위한 진입점을 \GOTO 명령의 대상으로 설정합니다.

다음은 명령의 샘플 출력을 보여줍니다.

```
(testcluster) user1@dev=# \! cat test.sql
select count(*) from lineitem limit 5;
\goto LABELB
\remark "this step was skipped by goto label";
```

```

\label LABELA
\remark 'this is label LABELA'
\label LABELB
\remark 'this is label LABELB'

(testcluster) user1@dev=# \i testgoto.sql
count
4307193
(1 row)

\label LABELA ignored
\label LABELB processed
this is label LABELB

```

\IF (\ELSEIF, \ELSE, \ENDIF)

\IF 및 관련 명령은 입력 스크립트의 일부를 조건부로 실행합니다. PSQL \if(\elif, \else, \endif) 명령의 확장입니다. \IF 및 \ELSEIF는 AND, OR 및 NOT 조건을 포함한 부울 표현식을 지원합니다.

다음은 명령의 샘플 출력을 보여줍니다.

```

(testcluster) user1@dev=# \! cat test.sql
SELECT query FROM stv_inflight LIMIT 1 \gset
select :query as query;
\if :query > 1000000
    \remark 'Query id is greater than 1000000'
\elseif :query = 1000000
    \remark 'Query id is equal than 1000000'
\else
    \remark 'Query id is less than 1000000'
\endif

(testcluster) user1@dev=# \i test.sql
query
-----
994803
(1 row)

Query id is less than 1000000

```

본기 로직에 ERRORCODE를 사용합니다.

```
\if :'ERRORCODE' = '00000'
    \remark 'The statement was executed without error'
\else
    \remark :LAST_ERROR_MESSAGE
\endif
```

\IF 블록 내에서 \GOTO를 사용하여 코드 실행 방법을 제어합니다.

Amazon Redshift RSQL 변수

RSQL에서 일부 키워드는 변수로 작동합니다. 각각을 특정 값으로 설정하거나 값을 재설정할 수 있습니다. 대부분은 대화형 모드와 배치 모드가 있는 \rset에 의해 설정됩니다. 소문자나 대문자로 명령을 정의할 수 있습니다.

ACTIVITYCOUNT

마지막으로 제출된 요청의 영향을 받는 행 수를 나타냅니다. 데이터 반환 요청의 경우 데이터베이스에서 RSQL로 반환되는 행 수입니다. 값은 0 또는 양의 정수입니다. 최댓값은 18,446,744,073,709,551,615입니다.

특수 처리되는 변수 ACTIVITYCOUNT는 ROW_COUNT 변수와 비슷합니다. 하지만 ROW_COUNT는 SELECT, COPY 또는 UNLOAD에 대한 명령 완료 시 영향을 받는 행 수를 클라이언트 애플리케이션에 보고하지 않습니다. 하지만 ACTIVITYCOUNT는 보고합니다.

activitycount_01.sql:

```
select viewname, schemaname
from pg_views
where schemaname = 'not_existing_schema';
\if :ACTIVITYCOUNT = 0
\remark 'views do not exist'
\endif
```

콘솔 출력:

```
viewname | schemaname
-----+-----
(0 rows)

views do not exist
```

ERRORLEVEL

오류에 심각도 수준을 할당합니다. 심각도 수준에 따라 일련의 조치를 결정할 수 있습니다. ERRORLEVEL 명령을 사용하지 않은 경우 해당 값은 기본적으로 ON입니다.

errorlevel_01.sql:

```
\rset errorlevel 42P01 severity 0

select * from tbl;

select 1 as col;

\echo exit
\quit
```

콘솔 출력:

```
Errorlevel is on.
rsql: ERROR: relation "tbl" does not exist
(1 row)

col
1

exit
```

HEADING 및 RTITLE

보고서 상단에 표시되는 헤더를 지정할 수 있습니다. RSET RTITLE 명령에서 지정한 헤더에는 클라이언트 컴퓨터의 현재 시스템 날짜를 자동으로 포함됩니다.

rset_heading_rtitle_02.rsqf 내용:

```
\remark Starting...
\rset rtitle "Marketing Department||Confidential//Third Quarter//Chicago"
\rset width 70
\rset rformat on
select * from rsql_test.tbl_currency order by id limit 2;
\exit
\remark Finishing...
```

콘솔 출력:

```
Starting...
Rtitle is set to: &DATE||Marketing Department||Confidential//Third Quarter//Chicago
(Changes will take effect after RFORMAT is
switched ON)
Target width is 70.
Rformat is on.
09/11/20      Marketing      Department Confidential
              Third Quarter
              Chicago
id | bankid | name |      start_date
100 |      1 | USD | 2020-09-11 10:51:39.106905
110 |      1 | EUR | 2020-09-11 10:51:39.106905
(2 rows)

Press any key to continue . . .
```

MAXERROR

RSQL이 해당 수준을 넘으면 작업 처리를 종료하는 최대 오류 심각도 수준을 지정합니다. 반환 코드는 각 작업 또는 태스크를 완료한 후 RSQL이 클라이언트 운영 체제에 반환하는 정수 값입니다. 반환 코드 값은 작업 또는 태스크의 완료 상태를 나타냅니다. 지정된 maxerror 값보다 큰 오류 심각도 수준을 생성하는 문이 스크립트에 포함된 경우 RSQL이 즉시 종료됩니다. 따라서 오류 심각도 수준 8에서 RSQL이 종료되도록 하려면 RSET MAXERROR 7을 사용합니다.

maxerror_01.sql 내용:

```
\rset maxerror 0

select 1 as col;

\quit
```

콘솔 출력:

```
Maxerror is default.
(1 row)

col
1
```

RFORMAT

형식 지정 명령에 대한 설정을 적용할지 여부를 지정할 수 있습니다.

rset_rformat.rsq1 내용:

```
\remark Starting...
\pset border 2
\pset format wrapped
\pset expanded on
\pset title 'Great Title'
select * from rsq1_test.tbl_long where id = 500;
\rset rformat
select * from rsq1_test.tbl_long where id = 500;
\rset rformat off
select * from rsq1_test.tbl_long where id = 500;
\rset rformat on
select * from rsq1_test.tbl_long where id = 500;
\exit
\remark Finishing...
```

콘솔 출력:

```
Starting...
Border style is 2. (Changes will take effect after RFORMAT is switched ON)
Output format is wrapped. (Changes will take effect after RFORMAT is switched ON)
Expanded display is on. (Changes will take effect after RFORMAT is switched ON)
Title is "Great Title". (Changes will take effect after RFORMAT is switched ON)
id | long_string
500 | In general, the higher the number the more borders and lines the tables will
    | have, but details depend on the particular
    | format.
(1 row)

Rformat is on.
Great Title
+-[ RECORD
  1 ]+-----+
-----+
| id          | 500
|
| long_string | In general, the higher the number the more borders and lines the tables
|             | will have, but details depend on the
```



```

particular format. |
+-----+
+-----+
+-----+

Rformat is off.
id | long_string
500 | In general, the higher the number the more borders and lines the tables will
    | have, but details depend on the particular format.
(1 row)

Rformat is on.
Great Title
+-[ RECORD
  1 ]+-----+
| id          | 500
|
| long_string | In general, the higher the number the more borders and lines the tables
|             | will have, but details depend on the
|             | particular format. |
+-----+
+-----+
+-----+
Press any key to continue . . .

```

ROW_COUNT

이전 쿼리의 영향을 받는 레코드 수를 가져옵니다. 일반적으로 다음 코드 조각에서처럼 결과를 확인하는 데 사용됩니다.

```

SET result = ROW_COUNT;

IF result = 0
...

```

TITLEDASHES

이 컨트롤을 사용하면 SQL 문에 대해 반환된 열 데이터 위에 대시 문자 행을 인쇄할지 여부를 지정할 수 있습니다.

예제:

```
\rset titledashes on
select dept_no, emp_no, salary from rsql_test.EMPLOYEE
where dept_no = 100;
\rset titledashes off
select dept_no, emp_no, salary from rsql_test.EMPLOYEE
where dept_no = 100;
```

콘솔 출력:

dept_no	emp_no	salary
100	1000346	1300.00
100	1000245	5000.00
100	1000262	2450.00

dept_no	emp_no	salary
100	1000346	1300.00
100	1000245	5000.00
100	1000262	2450.00

WIDTH

보고서의 출력 형식을 래핑으로 설정하고 각 행의 대상 너비를 지정합니다. 파라미터를 지정하지 않으면 형식과 대상 너비 모두 현재 설정이 반환됩니다.

rset_width_01.rsq1 내용:

```
\echo Starting...
\rset width
\rset width 50
\rset width
\quit
\echo Finishing...
```

콘솔 출력:

```
Starting...
Target width is 75.
Target width is 50.
Target width is 50.
Press any key to continue . . .
```

파라미터를 사용하는 예:

```
\echo Starting...
\rset rformat on
\pset format wrapped
select * from rsql_test.tbl_long where id = 500;
\rset width 50
select * from rsql_test.tbl_long where id = 500;
\quit
\echo Finishing...
```

콘솔 출력:

```
Starting...
Rformat is on.
Output format is wrapped.
id |                                long_string
500 | In general, the higher the number the more borders and lines the ta.
    | .bles will have, but details depend on the particular format.
(1 row)

Target width is 50.
id |                                long_string
500 | In general, the higher the number the more.
    | . borders and lines the tables will have, b.
    | .ut details depend on the particular format.
    | ..
(1 row)
Press any key to continue . . .
```

Amazon Redshift RSQL 오류 코드

성공 메시지, 경고 및 예외:

오류 코드	오류 클래스	조건 이름
00000	클래스 00 - 성공적인 완료	successful_completion
01000	클래스 01 - 경고	경고
0100C	클래스 01 - 경고	dynamic_result_sets_returned

오류 코드	오류 클래스	조건 이름
01008	클래스 01 - 경고	implicit_zero_bit_padding
01003	클래스 01 - 경고	null_value_eliminated_in_set_function
01007	클래스 01 - 경고	privilege_not_granted
01006	클래스 01 - 경고	privilege_not_revoked
01004	클래스 01 - 경고	string_data_right_truncation
01P01	클래스 01 - 경고	deprecated_feature
02000	클래스 02 - 데이터 없음	no_data
02001	클래스 02 - 데이터 없음	no_additional_dynamic_result_sets_returned
03000	클래스 03 - SQL 문이 아직 완료되지 않음	sql_statement_not_yet_complete
08000	클래스 08 - 연결 예외	connection_exception
08003	클래스 08 - 연결 예외	connection_does_not_exist
08006	클래스 08 - 연결 예외	connection_failure
08001	클래스 08 - 연결 예외	sqlclient_unable_to_establish_sqlconnection
08004	클래스 08 - 연결 예외	sqlserver_rejected_establishment_of_sqlconnection
08007	클래스 08 - 연결 예외	transaction_resolution_unknown
08P01	클래스 08 - 연결 예외	protocol_violation
09000	클래스 09 - 트리거된 작업 예외	triggered_action_exception
0A000	클래스 0A - 기능이 지원되지 않음	feature_not_supported

오류 코드	오류 클래스	조건 이름
0A000	클래스 0A - 기능이 지원되지 않음	feature_not_supported
0B000	클래스 0B - 잘못된 트랜잭션 시작	invalid_transaction_initiation
0F000	클래스 0F - 로케이터 예외	locator_exception
0F001	클래스 0F - 로케이터 예외	invalid_locator_specification
0L000	클래스 0L - 잘못된 부여자	invalid_grantor
0LP01	클래스 0L - 잘못된 부여자	invalid_grant_operation
0P000	클래스 0P - 잘못된 역할 지정	invalid_role_specification
0Z000	클래스 0Z - 진단 예외	diagnostics_exception
0Z002	클래스 0Z - 진단 예외	stacked_diagnostics_accessed_without_active_handler
20000	클래스 20 - 사례를 찾을 수 없음	case_not_found
21000	클래스 21 - 카디널리티 위반	cardinality_violation

데이터 예외:

오류 코드	오류 클래스	조건 이름
22000	클래스 22 - 데이터 예외	data_exception
2202E	클래스 22 - 데이터 예외	array_subscript_error
22021	클래스 22 - 데이터 예외	character_not_in_repertoire
22008	클래스 22 - 데이터 예외	datetime_field_overflow

오류 코드	오류 클래스	조건 이름
22012	클래스 22 - 데이터 예외	division_by_zero
22005	클래스 01 - 경고	error_in_assignment
2200B	클래스 01 - 경고	escape_character_conflict
22022	클래스 01 - 경고	indicator_overflow
22015	클래스 01 - 경고	interval_field_overflow
2201E	클래스 01 - 경고	invalid_argument_for_logarithm
2201F	클래스 01 - 경고	invalid_argument_for_power_function
2201G	클래스 01 - 경고	invalid_argument_for_width_bucket_function
22018	클래스 01 - 경고	invalid_character_value_for_cast
22007	클래스 01 - 경고	invalid_datetime_format
22019	클래스 01 - 경고	invalid_escape_character
2200D	클래스 01 - 경고	invalid_escape_octet
22025	클래스 01 - 경고	invalid_escape_sequence
22P06	클래스 01 - 경고	nonstandard_use_of_escape_character
22010	클래스 01 - 경고	invalid_indicator_parameter_value
22023	클래스 01 - 경고	invalid_parameter_value
2201B	클래스 01 - 경고	invalid_regular_expression
22009	클래스 01 - 경고	invalid_time_zone_displacement_value
2200C	클래스 01 - 경고	invalid_use_of_escape_character
2200G	클래스 01 - 경고	most_specific_type_mismatch

오류 코드	오류 클래스	조건 이름
22004	클래스 01 - 경고	null_value_not_allowed
22002	클래스 01 - 경고	null_value_no_indicator_parameter
22003	클래스 01 - 경고	numeric_value_out_of_range
22026	클래스 01 - 경고	string_data_length_mismatch
22001	클래스 01 - 경고	string_data_right_truncation
22011	클래스 01 - 경고	substring_error
22027	클래스 01 - 경고	trim_error
22024	클래스 01 - 경고	unterminated_c_string
2200F	클래스 01 - 경고	zero_length_character_string
22P01	클래스 01 - 경고	floating_point_exception
22P02	클래스 01 - 경고	invalid_text_representation
22P03	클래스 01 - 경고	invalid_binary_representation
22P04	클래스 01 - 경고	bad_copy_file_format
22P05	클래스 01 - 경고	untranslatable_character

무결성 제약 조건 위반:

오류 코드	오류 클래스	조건 이름
23000	클래스 23 - 무결성 제약 조건 위반	integrity_constraint_violation
23001	클래스 23 - 무결성 제약 조건 위반	restrict_violation

오류 코드	오류 클래스	조건 이름
23502	클래스 23 - 무결성 제약 조건 위반	not_null_violation
23503	클래스 23 - 무결성 제약 조건 위반	foreign_key_violation
23505	클래스 23 - 무결성 제약 조건 위반	unique_violation
23514	클래스 23 - 무결성 제약 조건 위반	check_violation
24000	클래스 24 - 잘못된 커서 상태	invalid_cursor_state
01004	클래스 01 - 경고	string_data_right_truncation
25000	클래스 25 - 잘못된 트랜잭션 상태	invalid_transaction_state
25001	클래스 25 - 잘못된 트랜잭션 상태	active_sql_transaction
25002	클래스 25 - 잘못된 트랜잭션 상태	invalid_transaction_state
25008	클래스 25 - 잘못된 트랜잭션 상태	held_cursor_requires_same_isolation_level
25003	클래스 25 - 잘못된 트랜잭션 상태	inappropriate_access_mode_for_branch_transaction
25004	클래스 25 - 잘못된 트랜잭션 상태	inappropriate_isolation_level_for_branch_transaction
25005	클래스 25 - 잘못된 트랜잭션 상태	no_active_sql_transaction_for_branch_transaction

오류 코드	오류 클래스	조건 이름
25006	클래스 25 - 잘못된 트랜잭션 상태	read_only_sql_transaction
25007	클래스 25 - 잘못된 트랜잭션 상태	no_active_sql_transaction_for_branch_transaction
25P01	클래스 25 - 잘못된 트랜잭션 상태	no_active_sql_transaction
25P02	클래스 25 - 잘못된 트랜잭션 상태	in_failed_sql_transaction
26000	클래스 26 - 잘못된 SQL 문 이름	invalid_sql_statement_name
28000	클래스 28 - 잘못된 인증 사양	invalid_authorization_specification
2B000	클래스 2B - 종속 권한 설명자가 여전히 존재	dependent_privilege_descriptors_still_exist
2BP01	클래스 2B - 종속 권한 설명자가 여전히 존재	dependent_objects_still_exist
2D000	클래스 2D - 잘못된 트랜잭션 종료	invalid_transaction_termination
2F000	클래스 2F - SQL 루틴 예외	sql_routine_exception
2F005	클래스 2F - SQL 루틴 예외	function_executed_no_return_statement
2F002	클래스 2F - SQL 루틴 예외	modifying_sql_data_not_permitted
2F003	클래스 2F - SQL 루틴 예외	prohibited_sql_statement_attempted
2F004	클래스 2F - SQL 루틴 예외	reading_sql_data_not_permitted
34000	클래스 34 - 잘못된 커서 이름	invalid_cursor_name
38000	클래스 38 - 외부 루틴 예외	external_routine_exception

오류 코드	오류 클래스	조건 이름
38001	클래스 38 - 외부 루틴 예외	containing_sql_not_permitted
38002	클래스 38 - 외부 루틴 예외	modifying_sql_data_not_permitted
38003	클래스 38 - 외부 루틴 예외	prohibited_sql_statement_attempted
38004	클래스 38 - 외부 루틴 예외	reading_sql_data_not_permitted
39000	클래스 39 - 외부 루틴 호출 예외	external_routine_invocation_exception
39001	클래스 39 - 외부 루틴 호출 예외	invalid_sqlstate_returned
39004	클래스 39 - 외부 루틴 호출 예외	null_value_not_allowed
39P01	클래스 39 - 외부 루틴 호출 예외	trigger_protocol_violated
39P02	클래스 39 - 외부 루틴 호출 예외	srf_protocol_violated
3D000	클래스 3D - 잘못된 카탈로그 이름	invalid_catalog_name
3F000	클래스 3F - 잘못된 스키마 이름	invalid_schema_name
42000	클래스 42 - 구문 오류 또는 액세스 규칙 위반	syntax_error_or_access_rule_violation
42601	클래스 42 - 구문 오류 또는 액세스 규칙 위반	syntax_error
42501	클래스 42 - 구문 오류 또는 액세스 규칙 위반	insufficient_privilege

오류 코드	오류 클래스	조건 이름
42846	클래스 42 - 구문 오류 또는 액세스 규칙 위반	cannot_coerce
42803	클래스 42 - 구문 오류 또는 액세스 규칙 위반	grouping_error
42830	클래스 42 - 구문 오류 또는 액세스 규칙 위반	invalid_foreign_key
42602	클래스 42 - 구문 오류 또는 액세스 규칙 위반	invalid_name
42622	클래스 42 - 구문 오류 또는 액세스 규칙 위반	name_too_long
42939	클래스 42 - 구문 오류 또는 액세스 규칙 위반	reserved_name
42804	클래스 42 - 구문 오류 또는 액세스 규칙 위반	datatype_mismatch
42P18	클래스 42 - 구문 오류 또는 액세스 규칙 위반	indeterminate_datatype
42809	클래스 42 - 구문 오류 또는 액세스 규칙 위반	wrong_object_type
42703	클래스 42 - 구문 오류 또는 액세스 규칙 위반	undefined_column
42883	클래스 42 - 구문 오류 또는 액세스 규칙 위반	undefined_function
42P01	클래스 42 - 구문 오류 또는 액세스 규칙 위반	undefined_table
42P02	클래스 42 - 구문 오류 또는 액세스 규칙 위반	undefined_parameter

오류 코드	오류 클래스	조건 이름
42704	클래스 42 - 구문 오류 또는 액세스 규칙 위반	undefined_object
42701	클래스 42 - 구문 오류 또는 액세스 규칙 위반	duplicate_column
42P03	클래스 42 - 구문 오류 또는 액세스 규칙 위반	duplicate_cursor
42P04	클래스 42 - 구문 오류 또는 액세스 규칙 위반	duplicate_database
42723	클래스 42 - 구문 오류 또는 액세스 규칙 위반	duplicate_function
42P05	클래스 42 - 구문 오류 또는 액세스 규칙 위반	duplicate_prepared_statement
42P06	클래스 42 - 구문 오류 또는 액세스 규칙 위반	duplicate_schema
42P07	클래스 42 - 구문 오류 또는 액세스 규칙 위반	duplicate_table
42712	클래스 42 - 구문 오류 또는 액세스 규칙 위반	duplicate_alias
42710	클래스 42 - 구문 오류 또는 액세스 규칙 위반	duplicate_object
42702	클래스 42 - 구문 오류 또는 액세스 규칙 위반	ambiguous_column
42725	클래스 42 - 구문 오류 또는 액세스 규칙 위반	ambiguous_function
42P08	클래스 42 - 구문 오류 또는 액세스 규칙 위반	ambiguous_parameter

오류 코드	오류 클래스	조건 이름
42P09	클래스 42 - 구문 오류 또는 액세스 규칙 위반	ambiguous_alias
42P10	클래스 42 - 구문 오류 또는 액세스 규칙 위반	invalid_column_reference
42611	클래스 42 - 구문 오류 또는 액세스 규칙 위반	invalid_column_definition
42P11	클래스 42 - 구문 오류 또는 액세스 규칙 위반	invalid_cursor_definition
42P12	클래스 42 - 구문 오류 또는 액세스 규칙 위반	invalid_database_definition
42P13	클래스 42 - 구문 오류 또는 액세스 규칙 위반	invalid_function_definition
42P14	클래스 42 - 구문 오류 또는 액세스 규칙 위반	invalid_prepared_statement_definition
42P15	클래스 42 - 구문 오류 또는 액세스 규칙 위반	invalid_schema_definition
42P16	클래스 42 - 구문 오류 또는 액세스 규칙 위반	invalid_table_definition
42P17	클래스 42 - 구문 오류 또는 액세스 규칙 위반	invalid_object_definition
44000	클래스 44 - WITH CHECK OPTION 위반	with_check_option_violation
53000	클래스 53 - 부족한 리소스	insufficient_resources
53100	클래스 53 - 부족한 리소스	disk_full
53200	클래스 53 - 부족한 리소스	out_of_memory

오류 코드	오류 클래스	조건 이름
53300	클래스 53 - 부족한 리소스	too_many_connections
54000	등급 54 - 프로그램 한도 초과	program_limit_exceeded
54001	등급 54 - 프로그램 한도 초과	statement_too_complex
54011	등급 54 - 프로그램 한도 초과	too_many_columns
54023	등급 54 - 프로그램 한도 초과	too_many_arguments
55000	클래스 55 - 객체가 사전 조건 상태가 아님	object_not_in_prerequisite_state
55006	클래스 55 - 객체가 사전 조건 상태가 아님	object_in_use
55P02	클래스 55 - 객체가 사전 조건 상태가 아님	cant_change_runtime_param
55P03	클래스 55 - 객체가 사전 조건 상태가 아님	lock_not_available
57000	클래스 57 - 운영자 개입	operator_intervention
57014	클래스 57 - 운영자 개입	query_canceled
57P01	클래스 57 - 운영자 개입	admin_shutdown
57P02	클래스 57 - 운영자 개입	crash_shutdown
57P03	클래스 57 - 운영자 개입	cannot_connect_now
58000	클래스 58 - 시스템 오류 (PostgreSQL 외부 오류)	system_error
58030	클래스 58 - 시스템 오류 (PostgreSQL 외부 오류)	io_error

오류 코드	오류 클래스	조건 이름
58P01	클래스 58 - 시스템 오류 (PostgreSQL 외부 오류)	undefined_file
58P02	클래스 58 - 시스템 오류 (PostgreSQL 외부 오류)	duplicate_file
F0000	클래스 F0 - 구성 파일 오류	duplicate_file
F0001	클래스 F0 - 구성 파일 오류	lock_file_exists
P0000	클래스 P0 - PL/pgSQL 오류	plpgsql_error
P0001	클래스 P0 - PL/pgSQL 오류	raise_exception
P0002	클래스 P0 - PL/pgSQL 오류	no_data_found
P0003	클래스 P0 - PL/pgSQL 오류	too_many_rows
XX000	클래스 XX - 내부 오류	internal_error
XX001	클래스 XX - 내부 오류	data_corrupted
XX002	클래스 XX - 내부 오류	index_corrupted

Amazon Redshift RSQL 환경 변수

Amazon Redshift RSQL은 환경 변수를 사용하여 기본 파라미터 값을 선택할 수 있습니다.

RSPASSWORD

Important

일부 운영 체제에서는 관리자 사용자가 아닌 사용자가 프로세스 환경 변수를 볼 수 있으므로 보안상의 이유로 이 환경 변수를 사용하지 않는 것이 좋습니다.

Amazon Redshift에 연결할 때 사용할 Amazon Redshift RSQL의 암호를 설정합니다. 이 환경 변수에는 Amazon Redshift RSQL 1.0.4 이상이 필요합니다.

RSPASSWORD가 설정된 경우 RSQL은 RSPASSWORD에 우선 순위를 둡니다. RSPASSWORD가 설정되지 않고 DSN을 사용해 연결하는 경우 RSQL은 DSN 파일의 파라미터에서 암호를 가져옵니다. 또한 RSPASSWORD가 설정되지 않고 DSN을 사용하지 않는 경우 RSQL은 연결을 시도한 후 암호 프롬프트를 제공합니다.

다음은 RSPASSWORD를 설정하는 예입니다.

```
export RSPASSWORD=TestPassw0rd
```

인증 프로파일을 사용하여 Amazon Redshift에 연결

Amazon Redshift에 대한 연결이 많은 경우 모든 연결에 대한 설정을 관리하기 어려울 수 있습니다. 각 JDBC 또는 ODBC 연결에서 특정 구성 옵션을 사용하는 경우가 많습니다. 인증 프로파일을 사용하여 연결 옵션을 함께 저장할 수 있습니다. 이렇게 하면 사용자가 연결할 프로파일을 선택할 수 있으며 개별 옵션에 대한 설정을 관리하지 않아도 됩니다. 다양한 시나리오와 사용자 유형에 프로파일을 적용할 수 있습니다.

인증 프로파일을 생성한 후 사용자가 연결 문자열에 즉시 사용 가능한 프로파일을 추가할 수 있습니다. 이렇게 하면 사용자는 각 역할 및 사용 사례에 적합한 설정을 사용하여 Amazon Redshift에 연결할 수 있습니다.

Amazon Redshift API 정보는 [CreateAuthenticationProfile](#)을 참조하세요.

인증 프로파일 생성

AWS CLI를 사용하여 create-authentication-profile 명령으로 인증 프로파일을 생성합니다. 여기서는 기존 Amazon Redshift 클러스터와 기존 데이터베이스가 있다고 가정합니다. 자격 증명에는 Amazon Redshift 데이터베이스에 연결할 수 있는 권한과 인증 프로파일을 가져올 수 있는 권한이 있어야 합니다. 구성 옵션을 JSON 문자열로 제공하거나 JSON 문자열이 포함된 파일을 참조합니다.

```
create-authentication-profile --authentication-profile-name<value: String> --
authentication-profile-content<value: String>
```

다음 예에서는 ExampleProfileName이라는 프로파일을 생성합니다. 여기에서 클러스터 이름 및 기타 옵션 설정을 정의하는 키와 값을 JSON 문자열로 추가할 수 있습니다.

```
create-authentication-profile --authentication-profile-name "ExampleProfileName"
--authentication-profile-content "{\"AllowDBUserOverride\":\"1\",\"Client_ID\":"
"\":\"ExampleClientID\",\"App_ID\":\"ExampleAppID\",\"AutoCreate\":false,"
"\":enableFetchRingBuffer\":true,\"databaseMetadataCurrentDbOnly\":true}"
```



```
}
```

이 명령은 지정된 JSON 설정으로 프로파일을 생성합니다. 프로파일이 생성되었음을 나타내는 다음이 반환됩니다.

```
{
  "AuthenticationProfileName": "ExampleProfileName",
  "AuthenticationProfileContent": "{
    \"AllowDBUserOverride\": \"1\",
    \"Client_ID\": \"ExampleClientID\",
    \"App_ID\": \"ExampleAppID\",
    \"AutoCreate\": false,
    \"enableFetchRingBuffer\": true,
    \"databaseMetadataCurrentDbOnly\": true
  }"
}
```

인증 프로파일 생성에 대한 제한 사항 및 할당량

각 고객에게는 10개의 인증 프로파일 할당량이 있습니다.

인증 프로파일에서 특정 오류가 발생할 수 있습니다. 예를 들어 기존 이름으로 새 프로파일을 생성하거나 프로파일 할당량을 초과하는 경우가 있습니다. 자세한 내용은 [CreateAuthenticationProfile](#)을 참조하세요.

인증 프로파일 스토어에 JDBC, ODBC 및 Python 연결 문자열에 대한 특정 옵션 키와 값을 저장할 수 없습니다.

- AccessKeyID
- access_key_id
- SecretAccessKey
- secret_access_key_id
- PWD
- Password
- password

JDBC 또는 ODBC 연결 문자열의 경우 프로파일 스토어에 키 또는 값 AuthProfile을 저장할 수 없습니다. Python 연결의 경우 auth_profile을 저장할 수 없습니다.

인증 프로파일은 Amazon DynamoDB에 저장되며 AWS에서 관리합니다.

인증 프로파일과 연결

인증 프로파일을 생성한 후 JDBC 버전 2.0 AuthProfile에 대한 연결 옵션으로 프로파일 이름을 포함할 수 있습니다. 이 연결 옵션을 사용하면 저장된 설정이 검색됩니다.

```
jdbc:redshift:iam://endpoint:port/database?AuthProfile=<Profile-Name>&AccessKeyID=<Caller-Access-Key>&SecretAccessKey=<Caller-Secret-Key>
```

다음은 JDBC URL 문자열의 예입니다.

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?AuthProfile="ExampleProfile"&AccessKeyID="AKIAIOSFODNN7EXAMPLE"&SecretAccessKey="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
```

인증 프로파일 이름과 함께 JDBC URL에 AccessKeyID와 SecretAccessKey를 모두 지정합니다.

로깅 옵션이 포함된 다음 예와 같이 세미콜론 구분 기호로 구성 옵션을 구분할 수도 있습니다.

```
jdbc:redshift:iam://my_redshift_end_point:5439/dev?LogLevel=6;LogPath=/tmp;AuthProfile=my_profile;AccessKeyID="AKIAIOSFODNN7EXAMPLE";SecretAccessKey="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
```

Note

인증 프로파일에 기밀 정보를 추가하지 않습니다. 예를 들어 인증 프로파일에 AccessKeyID 또는 SecretAccessKey 값을 저장하지 마세요. 인증 프로필 스토어에는 보안 암호 키의 저장을 금지하는 규칙이 있습니다. 민감한 정보와 관련된 키와 값을 저장하려고 하면 오류가 발생합니다.

인증 프로파일 가져오기

기존 인증 프로파일을 나열하려면 다음 명령을 호출합니다.

```
describe-authentication-profiles --authentication-profile-name <value: String>
```

다음 예에서는 2개의 검색된 프로파일을 보여줍니다. 프로파일 이름을 지정하지 않으면 모든 프로파일이 반환됩니다.

```
{ "AuthenticationProfiles": [ { "AuthenticationProfileName": "testProfile1", "AuthenticationProfileContent": "{ \"AllowDBUserOverride\": \"1\", \"Client_ID\": \"ExampleClientID\", \"App_ID\": \"ExampleAppID\", \"AutoCreate\": false, \"enableFetchRingBuffer\": true,
```

```
\ "databaseMetadataCurrentDbOnly\ ":true} " }, { "AuthenticationProfileName":
"testProfile2", "AuthenticationProfileContent": "{ \"AllowDBUserOverride
\ ": \"1\", \"Client_ID\ ": \"ExampleClientID\", \"App_ID\ ": \"ExampleAppID
\", \"AutoCreate\ ": false, \"enableFetchRingBuffer\ ": true,
\"databaseMetadataCurrentDbOnly\ ":true} " } ] }
```

Amazon Redshift 연결 문제 해결

SQL 클라이언트 도구에서 클러스터에 연결하는 데 문제가 발생하면 문제의 원인을 좁힐 수 있는 몇 가지 검사 방법이 있습니다. SSL 또는 서버 인증서를 사용할 때는 먼저 연결 문제를 해결하면서 이러한 복잡성을 제거했다가 이후 해결책을 발견했을 때 다시 추가하면 됩니다. 자세한 내용은 [연결을 위한 보안 옵션 구성](#) 섹션을 참조하세요.

애플리케이션에 영향을 미칠 수 있는 Amazon Redshift 기능의 동작 변경에 대한 자세한 내용은 [Amazon Redshift의 동작 변경 사항](#) 섹션을 참조하세요.

Important

Amazon Redshift는 SSL 인증서 관리 방법을 바꿨습니다. SSL을 사용하여 연결하는 데 문제가 있으면 현재 신뢰하는 루트 CA 인증서를 업데이트해야 할 수 있습니다. 자세한 내용은 [SSL 연결을 위해 ACM 인증서로 이전](#) 섹션을 참조하세요.

다음 섹션에서는 몇 가지 오류 메시지의 예와 가능한 연결 문제 해결책에 대해서 살펴보겠습니다. SQL 클라이언트 도구마다 오류 메시지가 다르기 때문에 아래와 같은 목록이 완벽할 수는 없지만 문제 해결을 위한 훌륭한 출발점이 될 것입니다.

Amazon EC2 외부에서 연결 및 방화벽 시간 제한 문제 발생

COPY 명령 같은 긴 쿼리를 실행할 때는 데이터베이스에 대한 클라이언트 연결이 멈추거나 제한 시간에 걸릴 수 있습니다. 이런 경우 Amazon Redshift 콘솔에서 쿼리의 완료 여부를 관찰할 수 있지만 클라이언트 도구에는 쿼리가 여전히 실행 중인 것으로 표시됩니다. 쿼리 결과는 연결 중단 시점에 따라 누락되거나 불완전할 수도 있습니다.

가능한 해결책

이 문제는 Amazon EC2 인스턴스가 아닌 다른 컴퓨터에서 Amazon Redshift에 연결할 때 발생합니다. 이 경우 유향 연결은 일정 시간 동안 사용하지 않으면 방화벽과 같은 중간 네트워크 구성 요소에 의해 종료됩니다. 이 동작은 가상 사설 네트워크(VPN) 또는 로컬 네트워크에서 로그인할 때 일반적입니다.

제한 시간 문제를 방지하려면 다음과 같이 변경하는 것이 좋습니다.

- TCP/IP 제한 시간을 처리하는 클라이언트 시스템 값을 높이세요. 단, 이러한 시스템 값은 클러스터에 연결할 때 사용하는 컴퓨터에서 변경해야 합니다. 또한 제한 시간은 클라이언트와 네트워크를 고려하여 조정해야 합니다. 자세한 내용은 [TCP/IP 제한 시간 설정 변경](#) 섹션을 참조하세요.
- 그 밖의 옵션으로 keepalive 동작을 DSN 수준으로 설정합니다. 자세한 내용은 [DSN 제한 시간 설정 변경](#) 섹션을 참조하세요.

TCP/IP 제한 시간 설정 변경

TCP/IP 제한 시간 설정을 변경하려면 클러스터에 연결할 때 사용하는 운영 체제에 따라 제한 시간 설정을 구성합니다.

- Linux - 클라이언트가 Linux에서 실행 중인 경우 루트 사용자로 다음 명령을 실행하여 현재 세션의 시간 제한 설정을 변경합니다.

```
/sbin/sysctl -w net.ipv4.tcp_keepalive_time=200 net.ipv4.tcp_keepalive_intvl=200
net.ipv4.tcp_keepalive_probes=5
```

설정을 계속해서 유지하려면 `/etc/sysctl.conf` 파일을 생성하거나 다음 값으로 수정한 후 시스템을 재부팅하세요.

```
net.ipv4.tcp_keepalive_time=200
net.ipv4.tcp_keepalive_intvl=200
net.ipv4.tcp_keepalive_probes=5
```

- Windows - 클라이언트가 Windows에서 실행되는 경우 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\`에서 다음 레지스트리 설정 값을 편집합니다.
 - KeepAliveTime: 30000
 - KeepAliveInterval: 1000
 - TcpMaxDataRetransmissions: 10

위 설정 값은 DWORD 데이터 형식을 사용합니다. 설정이 레지스트리 경로에 존재하지 않으면 설정을 직접 생성한 후 다음과 같은 권장 값을 지정할 수도 있습니다. Windows 레지스트리의 편집에 대한 자세한 내용은 Windows 설명서를 참조하세요.

값을 설정한 후에는 컴퓨터를 다시 시작해야 변경 사항이 적용됩니다.

- Mac - 클라이언트가 Mac에서 실행 중인 경우 다음 명령을 실행하여 현재 세션의 시간 제한 설정을 변경합니다.

```
sudo sysctl net.inet.tcp.keepintvl=200000
sudo sysctl net.inet.tcp.keepidle=200000
sudo sysctl net.inet.tcp.keepinit=200000
sudo sysctl net.inet.tcp.always_keepalive=1
```

설정을 계속해서 유지하려면 /etc/sysctl.conf 파일을 생성하거나 다음 값으로 수정합니다.

```
net.inet.tcp.keepidle=200000
net.inet.tcp.keepintvl=200000
net.inet.tcp.keepinit=200000
net.inet.tcp.always_keepalive=1
```

컴퓨터를 다시 시작한 후 다음 명령을 실행하여 값이 설정되었는지 확인합니다.

```
sysctl net.inet.tcp.keepidle
sysctl net.inet.tcp.keepintvl
sysctl net.inet.tcp.keepinit
sysctl net.inet.tcp.always_keepalive
```

DSN 제한 시간 설정 변경

원하는 경우 keepalive 동작을 DSN 수준으로 설정할 수 있습니다. odbc.ini 파일에서 다음 파라미터를 추가하거나 수정하면 가능합니다.

KeepAlivesCount

연결이 끊긴 것으로 간주할 때까지 손실될 수 있는 TCP keepalive 패킷의 수입니다.

KeepAlivesIdle

드라이버가 TCP keepalive 패킷을 전송할 때까지 아무런 작업 없이 대기하는 시간(초)입니다.

KeepAlivesInterval

TCP keepalive가 재전송되는 시간 간격(초)입니다.

이러한 파라미터가 존재하지 않거나 값이 0인 경우에는 시스템이 TCP/IP에 지정한 keepalive 파라미터를 사용하여 DSN keepalive 동작을 결정합니다. Windows에서는 레지스트리의 TCP/IP 파라미터를 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\에서 확인할 수 있습니다. 그리고 Linux 및 macOS에서는 TCP/IP 파라미터를 sysctl.conf 파일에서 찾아볼 수 있습니다.

연결 거부 또는 실패

연결이 거부되거나 실패하면 다음 중 하나와 유사한 오류가 발생할 수 있습니다.

- "Failed to establish a connection to *<endpoint>*."
- "Could not connect to server: Connection timed out. Is the server running on host '*<endpoint>*' and accepting TCP/IP connections on port '*<port>*'?"
- "Connection refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections."

가능한 해결책

일반적으로 연결 구성에 실패했다는 오류 메시지가 수신되면 클러스터에 대한 액세스 권한과 관련된 문제이거나 네트워크 트래픽이 클러스터에 도달하는 것과 관련된 문제입니다.

클러스터가 있는 네트워크 외부의 클라이언트 도구에서 클러스터에 연결하려면 클러스터의 보안 그룹에 인바운드 규칙을 추가합니다. 규칙 구성은 Amazon Redshift 클러스터가 Virtual Private Cloud(VPC)에 생성되었는지에 따라 달라집니다.

- Amazon VPC를 기반으로 Virtual Private Cloud(VPC)에서 Amazon Redshift 클러스터를 생성한 경우 Amazon VPC에서 클라이언트 CIDR/IP 주소를 지정하는 VPC 보안 그룹에 인바운드 규칙을 추가합니다. 클러스터의 VPC 보안 그룹 구성 및 퍼블릭 액세스 옵션에 대한 자세한 내용은 [VPC의 Redshift 리소스](#) 섹션을 참조하세요.
- VPC 외부에 Amazon Redshift 클러스터를 생성한 경우에는 Amazon Redshift의 클러스터 보안 그룹에 클라이언트 CIDR/IP 주소를 추가합니다. 클러스터 보안 그룹 구성에 대한 자세한 내용은 [Amazon Redshift 보안 그룹](#) 섹션을 참조하세요.

Amazon EC2 인스턴스에서 실행되는 클라이언트 도구에서 클러스터에 연결하려면 인바운드 규칙도 추가합니다. 이 경우 클러스터 보안 그룹에 규칙을 추가하세요. 규칙은 클라이언트 도구의 Amazon EC2 인스턴스와 연결된 Amazon EC2 보안 그룹을 지정해야 합니다.

경우에 따라 방화벽과 같이 클라이언트와 서버 간에 계층이 있을 수 있습니다. 이러한 경우 클러스터에 대해 구성된 포트를 통한 인바운드 연결을 방화벽에서 허용하는지 확인합니다.

클라이언트와 드라이버가 호환되지 않습니다

클라이언트와 드라이버가 호환되지 않는 경우 “지정된 DSN에 드라이버와 애플리케이션 간의 아키텍처 불일치가 포함되어 있습니다”라는 오류가 표시될 수 있습니다.

가능한 해결책

연결을 시도하다 아키텍처 불일치 오류가 발생하기도 합니다. 이는 클라이언트 도구와 드라이버가 호환되지 않는다는 것을 의미합니다. 이 오류는 시스템 아키텍처가 일치하지 않기 때문에 발생합니다. 예를 들어 클라이언트 도구가 32비트이지만 설치되어 있는 드라이버가 64비트 버전이면 이러한 불일치가 발생할 수 있습니다. 간혹 64비트 클라이언트 도구가 32비트 드라이버를 사용하는 경우가 있지만 64비트 드라이버에 32비트 애플리케이션을 사용할 수는 없습니다. 따라서 드라이버와 클라이언트 도구가 동일한 버전의 시스템 아키텍처를 사용하고 있는지 확인해야 합니다.

쿼리가 중단되거나, 간혹 클러스터까지 전송되지 않습니다

쿼리가 실행 중인 것으로 보이지만 SQL 클라이언트 도구에서는 중단되는 중 쿼리 완료에 대한 문제가 발생합니다. 간혹 시스템 테이블이나 Amazon Redshift 콘솔과 같은 클러스터에 쿼리가 표시되지 않기도 합니다.

가능한 해결책

이 문제는 패킷이 소실되어 발생할 수 있습니다. 이 경우 두 IP(Internet Protocol) 호스트 간 네트워크 경로에서 최대 전송 단위(MTU)의 크기 차이가 있습니다. 단일 이더넷 프레임으로 네트워크 연결을 통해 전송할 수 있는 패킷의 최대 크기(바이트)는 MTU의 크기에 따라 결정됩니다. AWS에서는 일부 Amazon EC2 인스턴스 유형이 1500MTU(이더넷 v2 프레임)를 지원하고, 그 밖에 9001MTU(TCP/IP 정보 프레임)를 지원하는 인스턴스 유형도 있습니다.

MTU 크기 차이에서 발생할 수 있는 문제에 대한 해결책으로서 다음 중 한 가지를 권장합니다.

- 클러스터가 EC2-VPC 플랫폼을 사용하는 경우에는 Destination Unreachable을 반환하는 사용자 정의 ICMP(Internet Control Message Protocol) 인바운드 규칙으로 Amazon VPC 보안 그룹을 구성합니다. 이 규칙은 전송 호스트에게 네트워크 경로를 따라 최저 MTU 크기를 사용하도록 지시합니다. 이러한 접근법에 대한 자세한 내용은 [ICMP "destination unreachable"을 허용하도록 보안 그룹 구성하기](#) 섹션을 참조하세요.

- 클러스터가 EC2-Classic 플랫폼을 사용하거나 ICMP 인바운드 규칙을 허용할 수 없는 경우 TCP/IP 점포 프레임을 비활성화하여 이더넷 v2 프레임을 사용하세요. 이러한 접근법에 대한 자세한 내용은 [인스턴스의 MTU 구성](#) 섹션을 참조하세요.

ICMP "destination unreachable"을 허용하도록 보안 그룹 구성하기

두 호스트 사이의 네트워크에서 MTU 크기에 차이가 있다면 먼저 네트워크 설정이 PMTUD(path MTU discovery)를 차단하지 않는지 확인해야 합니다. 수신 호스트가 ICMP 메시지 Destination Unreachable: fragmentation needed and DF set (ICMP Type 3, Code 4)과 함께 전송 호스트에 응답하기 위해서는 PMTUD가 차단되어서는 안 됩니다. 이 메시지는 전송 호스트에게 네트워크 경로를 따라 요청을 재전송하려면 최저 MTU 크기를 사용하라는 의미입니다. 이러한 협상이 없으면 수신 호스트가 허용할 수 없을 만큼 요청이 너무 많아져서 패킷 손실이 발생할 수 있습니다. 이 ICMP 메시지에 대한 자세한 내용은 IETF(Internet Engineering Task Force) 웹 사이트의 [RFC792](#) 섹션을 참조하세요.

Amazon VPC 보안 그룹에 이 ICMP 인바운드 규칙을 명시적으로 구성하지 않으면 PMTUD가 차단됩니다. AWS에서는 보안 그룹이 인바운드 및 아웃바운드 트래픽에 대한 규칙을 인스턴스로 지정하는 가상 방화벽입니다. Amazon Redshift 클러스터 보안 그룹에 대한 자세한 내용은 [Amazon Redshift 보안 그룹](#) 섹션을 참조하세요. EC2-VPC 플랫폼을 사용하는 클러스터의 경우에는 Amazon Redshift가 VPC 보안 그룹을 사용하여 클러스터로 전송되는 트래픽을 허용하거나 거부합니다. 기본적으로 보안 그룹은 잠겨있기 때문에 인바운드 트래픽을 모두 거부합니다. EC2-Classic 또는 EC2-VPC 인스턴스에 대한 인바운드 및 아웃바운드 규칙을 설정하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [EC2-Classic과 VPC의 인스턴스 간 차이점](#)을 참조하세요.

VPC 보안 그룹에 규칙을 추가하는 방법에 대한 자세한 내용은 [VPC 보안 그룹](#) 섹션을 참조하세요. 이 규칙에 필요한 특정 PMTUD 설정에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [경로 MTU 검색](#)을 참조하세요.

인스턴스의 MTU 구성

클러스터가 EC2-Classic 플랫폼을 사용하거나 인바운드 트래픽에 대한 사용자 지정 ICMP 규칙을 허용할 수 없는 경우가 있습니다. 이러한 경우 Amazon Redshift 클러스터에 연결하는 EC2 인스턴스의 네트워크 인터페이스(NIC)에서 MTU를 1500으로 조정하는 것이 좋습니다. 이렇게 조정하면 TCP/IP 점포 프레임을 비활성화하여 동일한 패킷 크기를 계속해서 사용할 수 있습니다. 하지만 Amazon Redshift 연결에 그치지 않고 인스턴스 전체의 최대 네트워크 처리 속도까지 떨어뜨리는 단점도 있습니다. 자세한 내용은 다음 절차를 참조하세요.

Microsoft Windows 운영 체제에서 MTU를 설정하려면

클라이언트가 Microsoft Windows 운영 체제 기반인 경우에는 netsh 명령을 사용하여 이더넷 어댑터의 MTU 값을 살펴보거나 설정할 수 있습니다.

1. 다음 명령을 실행하여 현재 MTU 값을 결정합니다.

```
netsh interface ipv4 show subinterfaces
```

2. 출력 화면에서 MTU 어댑터의 Ethernet 값을 살펴봅니다.
3. 이 값이 1500이 아닌 경우에는 다음 명령을 실행하여 설정합니다.

```
netsh interface ipv4 set subinterface "Ethernet" mtu=1500 store=persistent
```

이 값을 설정한 후에는 컴퓨터를 다시 시작해야 변경 사항이 적용됩니다.

Linux 운영 체제에서 MTU를 설정하려면

클라이언트가 Linux 운영 체제에서 작동하는 경우 ip 명령을 사용하여 MTU 값을 검토하고 설정할 수 있습니다.

1. 다음 명령을 실행하여 현재 MTU 값을 결정합니다.

```
$ ip link show eth0
```

2. 출력 결과에서 mtu 다음의 값을 검토합니다.
3. 이 값이 1500이 아닌 경우에는 다음 명령을 실행하여 설정합니다.

```
$ sudo ip link set dev eth0 mtu 1500
```

Mac 운영 체제에서 MTU를 설정하려면

- How to change the MTU for troubleshooting purposes에 대한 macOS 지원 사이트의 지침을 따릅니다. 자세한 내용은 [지원 사이트](#)를 참조하세요.

JDBC Fetch Size 파라미터 설정

기본적으로 Redshift JDBC 드라이버는 링 버퍼를 사용하여 메모리를 효율적으로 관리하고 메모리 부족 오류를 방지합니다. 가져오기 크기 파라미터는 링 버퍼가 명시적으로 비활성화된 경우에만 적용됩니다. 자세한 내용은 [링크](#)를 참조하세요. 이 구성에서는 가져오기 크기를 설정하여 각 배치에서 검색되는 행 수를 제어해야 합니다.

가져오기 크기는 언제 사용하나요?

다음과 같은 경우 가져오기 크기 파라미터를 사용합니다.

- 행 기반 일괄 처리를 세밀하게 제어해야 하는 경우
- 기존 가져오기 크기 동작이 필요한 레거시 애플리케이션 작업

가져오기 크기 설정

링 버퍼가 비활성화된 경우, JDBC 드라이버는 기본적으로 쿼리의 모든 결과를 한 번에 수집합니다. 큰 결과 집합을 반환하는 쿼리는 과도한 메모리를 소비할 수 있습니다. 결과 세트를 한 번에 모두 검색하는 대신 배치로 검색하려면 애플리케이션에서 JDBC 가져오기 크기 파라미터를 설정합니다.

Note

ODBC는 Fetch Size 파라미터가 지원되지 않습니다.

성능을 최적화하려면 메모리 부족 오류가 일어나지 않는 범위 내에서 페치 크기 값을 가장 높게 설정하세요. 페치 크기 값이 더 낮아지면 서버 전송이 늘어나서 실행 시간이 장기화될 수 있습니다. 서버는 클라이언트가 전체 결과 집합을 가져오거나 쿼리가 취소될 때까지 WLM 쿼리 슬롯이나 연결 메모리를 비롯한 리소스를 예약합니다. 이때 Fetch Size 값을 적절히 조정하면 이러한 리소스가 더욱 빠르게 해제되어 다른 쿼리에서도 사용할 수 있게 됩니다.

Note

대용량 데이터 집합을 추출해야 하는 경우에는 [UNLOAD](#) 문을 사용하여 데이터를 Amazon S3로 전송하는 것이 좋습니다. UNLOAD를 사용하면 컴퓨팅 노드가 병렬로 실행되어 데이터 전송 속도가 빨라집니다.

JDBC Fetch Size 파라미터 설정에 대한 자세한 내용은 PostgreSQL 설명서의 [Getting results based on a cursor](#)에서 확인할 수 있습니다.

Amazon Redshift Data API 사용

Amazon Redshift 데이터 API는 데이터베이스 드라이버, 연결, 네트워크 구성, 데이터 버퍼링, 자격 증명 등을 관리할 필요가 없도록 하여 Amazon Redshift 데이터 웨어하우스에 대한 액세스를 간소화합니다. AWS SDK에서 데이터 API 작업을 사용하여 SQL 문을 실행할 수 있습니다. 데이터 API 작업에 대한 자세한 내용은 [Amazon Redshift 데이터 API 참조](#)를 참조하세요.

데이터 API는 데이터베이스에 대한 지속적인 연결을 요구하지 않습니다. 대신에 AWS SDK와의 통합과 보안 HTTP 엔드포인트를 제공합니다. 연결을 관리하지 않고 엔드포인트를 사용하여 SQL 문을 실행할 수 있습니다. Data API에 대한 호출은 비동기식입니다. 데이터 API는 AWS Secrets Manager에 저장된 자격 증명 또는 임시 데이터베이스 자격 증명을 사용할 수 있습니다. 두 인증 방법 중 하나를 사용하여 API 호출에서 암호를 전달할 필요가 없습니다. AWS Secrets Manager에 대한 자세한 내용은 AWS Secrets Manager User Guide의 [What Is AWS Secrets Manager?](#)를 참조하세요. 권한 부여에 AWS IAM Identity Center를 사용할 수도 있습니다.

이 데이터 API를 사용하면 프로그래밍 방식으로 AWS Lambda, Amazon SageMaker AI 노트북, AWS Cloud9을 포함한 웹 서비스 기반 애플리케이션으로 Amazon Redshift 데이터에 액세스할 수 있습니다. 이러한 애플리케이션에 대한 자세한 내용은 [AWS Lambda](#), [Amazon SageMaker AI](#) 및 [AWS Cloud9](#) 섹션을 참조하세요.

데이터 API에 대해 자세히 알아보려면 AWS 빅 데이터 블로그에서 [Get started with the Amazon Redshift Data API](#)를 참조하세요.

Amazon Redshift Data API 작업

Amazon Redshift Data API를 사용하기 전에 다음 단계를 검토합니다.

1. Data API의 호출자로서 권한이 있는지 확인합니다. 권한 부여에 대한 자세한 내용은 [Amazon Redshift Data API에 대한 액세스 권한 부여](#) 섹션을 참조하세요.
2. Secrets Manager의 인증 자격 증명, 임시 자격 증명 또는 AWS IAM Identity Center를 사용하여 데이터 API를 직접적으로 호출할 계획인지 결정합니다. 자세한 내용은 [Amazon Redshift 데이터 API를 호출할 때 데이터베이스 인증 보안 인증 정보 선택](#) 섹션을 참조하세요.
3. 인증 자격 증명에 Secrets Manager 사용하는 경우 보안 암호를 설정합니다. 자세한 내용은 [AWS Secrets Manager에 데이터베이스 자격 증명 저장](#) 섹션을 참조하세요.

4. Data API를 호출할 때 고려 사항과 제한 사항을 검토합니다. 자세한 내용은 [Amazon Redshift Data API 호출 시 고려 사항](#) 섹션을 참조하세요.
5. AWS Command Line Interface(AWS CLI), 자체 코드 또는 Amazon Redshift 콘솔의 쿼리 편집기를 사용하여 Data API를 호출합니다. AWS CLI에서 호출 예는 [데이터 API 호출](#) 섹션을 참조하세요.

Amazon Redshift Data API 호출 시 고려 사항

Data API를 호출할 때는 다음 사항을 고려하세요.

- Amazon Redshift 데이터 API는 Amazon Redshift 프로비저닝 클러스터와 Redshift Serverless 작업 그룹의 데이터베이스에 액세스할 수 있습니다. Redshift 데이터 API를 사용할 수 있는 AWS 리전 목록은 Amazon Web Services 일반 참조의 [Redshift 데이터 API](#)에 나열된 엔드포인트를 참조하세요.
- 최대 쿼리 기간은 24시간입니다.
- Amazon Redshift 클러스터당 최대 활성 쿼리(STARTED 및 SUBMITTED 쿼리) 수는 500개입니다.
- 최대 쿼리 결과 크기는 500MB(gzip 압축 후)입니다. 직접 호출이 500MB를 초과하는 응답 데이터를 반환하면 직접 호출이 종료됩니다.
- 쿼리 결과의 최대 보존 시간은 24시간입니다.
- 최대 쿼리 문 크기는 100KB입니다.
- Data API는 다음 노드 유형의 단일 노드 및 다중 노드 클러스터를 쿼리하는 데 사용할 수 있습니다.
 - dc2.large
 - dc2.8xlarge
 - ra3.large
 - ra3.xlplus
 - ra3.4xlarge
 - ra3.16xlarge
- 클러스터는 Amazon VPC 서비스 기반의 Virtual Private Cloud(VPC)에 있어야 합니다.
- 기본적으로 ExecuteStatement 또는 BatchExecuteStatement API 작업의 실행자와 동일한 IAM 역할을 가진 사용자는 CancelStatement, DescribeStatement, GetStatementResult, GetStatementResultV2 및 ListStatements API 작업을 사용하여 동일한 문에 대해 작업을 수행할 수 있습니다. 다른 사용자의 동일한 SQL 문에 대해 작업을 수행하려면 사용자가 SQL 문을 실행한 사용자의 IAM 역할을 맡을 수 있어야 합니다. 역할 수임 방법에 대한 자세한 내용은 [Amazon Redshift Data API에 대한 액세스 권한 부여](#) 섹션을 참조하세요.

- BatchExecuteStatement API 작업의 Sqls 파라미터에 포함된 SQL 문은 단일 트랜잭션으로 실행됩니다. 배열 순서대로 순차 실행됩니다. 후순위 SQL 문은 배열 내 선순위 명령문이 완료될 때까지 시작되지 않습니다. SQL 문이 실패하면 하나의 트랜잭션으로 실행되므로 모든 작업이 롤백됩니다.
- ExecuteStatement 또는 BatchExecuteStatement API 작업에 사용되는 클라이언트 토큰의 최대 보존 시간은 8시간입니다.
- Amazon Redshift 프로비저닝된 클러스터와 Redshift Serverless 작업 그룹이 고객 관리형 키를 사용하여 암호화된 경우 Redshift는 Redshift 데이터 API가 작업에 키를 사용할 수 있도록 허용하는 권한을 생성합니다. 자세한 내용은 [Amazon Redshift 데이터 API와 함께 AWS KMS 사용](#) 단원을 참조하세요.
- Redshift 데이터 API의 각 API에는 요청을 제한하기 전에 초당 트랜잭션 할당량이 있습니다. 할당량에 대한 내용은 [Amazon Redshift 데이터 API의 할당량](#) 섹션을 참조하세요. 요청 비율이 할당량을 초과하면 HTTP 상태 코드가 400인 ThrottlingException 오류가 반환됩니다. 제한에 대응하려면 AWSSDK 및 도구 참조 안내서의 [재시도 동작](#)에 설명된 재시도 전략을 사용하세요. 이 전략은 일부 AWS SDK의 제한 오류에 대해 자동으로 구현됩니다.

Note

AWS Step Functions에서는 기본적으로 재시도가 활성화되지 않습니다. Step Functions 상태 머신에서 Redshift 데이터 API를 호출해야 하는 경우 Redshift 데이터 API 호출에 ClientToken 멍등성 파라미터를 포함하세요. ClientToken의 값은 재시도 시 지속되어야 합니다. ExecuteStatement API에 대한 다음 요청 스니펫 예시에서 States.ArrayGetItem(States.StringSplit(\$\$.Execution.Id, ':'), 7) 식은 내장 함수를 사용하여 \$\$.Execution.Id의 UUID 부분을 추출합니다. UUID 부분은 상태 머신을 실행할 때마다 고유합니다. 자세한 내용은 AWS Step Functions 개발자 안내서의 [내장 함수](#)를 참조하세요.

```
{
  "Database": "dev",
  "Sql": "select 1;",
  "ClusterIdentifier": "MyCluster",
  "ClientToken.$": "States.ArrayGetItem(States.StringSplit($$.Execution.Id,
    ':'), 7)"
}
```

Amazon Redshift 데이터 API를 호출할 때 데이터베이스 인증 보안 인증 정보 선택

Data API를 호출할 때 일부 API 작업에 대해 다음 인증 방법 중 하나를 사용합니다. 각 방법에는 서로 다른 파라미터 조합이 필요합니다.

AWS IAM Identity Center

데이터 API는 AWS IAM Identity Center에 등록된 Single Sign-On 사용자로 액세스할 수 있습니다. IAM Identity Center를 설정하는 단계에 대한 자세한 내용은 [신뢰할 수 있는 자격 증명 전파와 함께 데이터 API 사용](#) 섹션을 참조하세요.

AWS Secrets Manager

이 방법에서는 username 및 password가 있는 AWS Secrets Manager에 저장된 암호의 secret-arn을 입력합니다. 지정된 보안 암호에는 지정하는 database에 연결하기 위한 자격 증명이 포함되어 있습니다. 클러스터에 연결할 때는 데이터베이스 이름도 제공하며, 클러스터 식별자(dbClusterIdentifier)를 제공하는 경우 암호에 저장된 클러스터 식별자와 일치해야 합니다. 서버리스 작업 그룹에 연결할 때는 데이터베이스 이름도 제공합니다. 자세한 내용은 [AWS Secrets Manager에 데이터베이스 자격 증명 저장](#) 섹션을 참조하세요.

이 방법에서는 데이터가 위치한 AWS 리전을 지정하는 region 값을 제공할 수도 있습니다.

임시 자격 증명

이 방법에서는 다음 옵션 중 하나를 선택합니다.

- 서버리스 작업 그룹에 연결할 때는 작업 그룹 이름과 데이터베이스 이름을 지정합니다. 데이터베이스 사용자 이름은 IAM ID에서 파생됩니다. 예를 들어, arn:iam::123456789012:user:foo에는 데이터베이스 사용자 이름인 IAM:foo가 포함되어 있습니다. 또한 redshift-serverless:GetCredentials 작업을 호출할 수 있는 권한이 필요합니다.
- 클러스터에 IAM ID로 연결할 때는 클러스터 식별자 및 데이터베이스 이름을 지정합니다. 데이터베이스 사용자 이름은 IAM ID에서 파생됩니다. 예를 들어, arn:iam::123456789012:user:foo에는 데이터베이스 사용자 이름인 IAM:foo가 포함되어 있습니다. 또한 redshift:GetClusterCredentialsWithIAM 작업을 호출할 수 있는 권한이 필요합니다.
- 클러스터에 데이터베이스 사용자로 연결할 때는 클러스터 식별자, 데이터베이스 이름 및 데이터베이스 사용자 이름을 지정합니다. 또한 redshift:GetClusterCredentials 작업을 호출할

수 있는 권한이 필요합니다. 이 방법으로 연결할 때 데이터베이스 그룹에 가입하는 방법에 대한 자세한 내용은 [클러스터에 연결할 때 데이터베이스 그룹에 조인](#)을 참조하세요.

이 방법에서는 데이터가 위치한 AWS 리전을 지정하는 region 값을 제공할 수도 있습니다.

Amazon Redshift Data API를 호출할 때 JDBC 데이터 형식 매핑

다음 표는 JDBC(Java Database Connectivity) 데이터 유형을 데이터 API 호출에서 지정하는 데이터 형식에 매핑합니다.

JDBC 데이터 형식	데이터 API 데이터 형식
INTEGER, SMALLINT, BIGINT	LONG
FLOAT, REAL, DOUBLE	DOUBLE
DECIMAL	STRING
BOOLEAN, BIT	BOOLEAN
BLOB, BINARY, LONGVARBINARY	BLOB
VARBINARY	STRING
CLOB	STRING
다른 형식(날짜 및 시간과 관련된 형식 포함)	STRING

문자열 값은 Amazon Redshift 데이터베이스로 전달되고 암시적으로 데이터베이스 데이터 형식으로 변환됩니다.

Note

현재, Data API는 범용 고유 식별자(UUID) 배열을 지원하지 않습니다.

Amazon Redshift Data API를 호출할 때 파라미터로 SQL 문 실행

SQL 문의 일부에 대한 파라미터로 Data API 작업을 호출하여 데이터베이스 엔진에 제출된 SQL 텍스트를 제어할 수 있습니다. 명명된 파라미터는 SQL 텍스트에 하드코딩하지 않고 파라미터를 전달할 수 있는 유연한 방법을 제공합니다. SQL 텍스트를 재사용하고 SQL 삽입 문제를 방지하는 데 도움이 됩니다.

다음 예에서는 execute-statement AWS CLI 명령의 parameters 필드의 명명된 파라미터를 보여줍니다.

```
--parameters "[{\"name\": \"id\", \"value\": \"1\"},{\"name\": \"address\", \"value\": \"Seattle\"}]"
```

명명된 파라미터를 사용할 때는 다음 사항을 고려하세요.

- 명명된 파라미터는 SQL 문의 값을 대체하는 데만 사용할 수 있습니다.
- INSERT 문의 값을 대체할 수 있습니다(예: INSERT INTO mytable VALUES(:val1)).

명명된 파라미터는 임의의 순서로 지정할 수 있으며 SQL 텍스트에서 파라미터를 두 번 이상 사용할 수 있습니다. 이전 예에 표시된 파라미터 옵션에서 값 1 및 Seattle이 테이블 열 id 및 address에 삽입됩니다. SQL 텍스트에서 다음과 같이 명명된 파라미터를 지정합니다.

```
--sql "insert into mytable values (:id, :address)"
```

- 조건 절의 값(예: WHERE attr >= :val1, WHERE attr BETWEEN :val1 AND :val2 및 HAVING COUNT(attr) > :val)을 대체할 수 있습니다.
- SQL 문의 열 이름(예: SELECT column-name, ORDER BY column-name 또는 GROUP BY column-name)은 대체할 수 없습니다.

예를 들어, 다음 SELECT 문은 잘못된 구문으로 실패합니다.

```
--sql "SELECT :colname, FROM event" --parameters "[{\"name\": \"colname\", \"value\": \"eventname\"}]"
```

구문 오류가 있는 문을 설명(describe-statement 작업)하면 반환된 QueryString이 파라미터("QueryString": "SELECT :colname, FROM event")의 열 이름을 대체하지 않으며 오류가 보고됩니다(오류: "FROM" 또는 근처에 구문 오류가 있음\n 위치: 12).

- 집계 함수의 열 이름(예: COUNT(column-name), AVG(column-name) 또는 SUM(column-name))은 대체할 수 없습니다.
- JOIN 절의 열 이름은 대체할 수 없습니다.
- SQL 실행 시 암시적으로 데이터가 데이터 형식으로 캐스팅됩니다. 데이터 형식 캐스팅에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [데이터 형식](#)을 참조하세요.
- 값을 NULL로 설정할 수 없습니다. Data API는 이를 리터럴 문자열 NULL로 해석합니다. 다음 예에서는 id를 리터럴 문자열 null로 바꿉니다. SQL NULL 값이 아닙니다.

```
--parameters "[{\"name\": \"id\", \"value\": \"null\"}]"
```

- 길이가 0인 값을 설정할 수 없습니다. Data API SQL 문이 실패합니다. 다음 예에서는 id를 길이가 0인 값으로 설정하려고 하여 SQL 문이 실패합니다.

```
--parameters "[{\"name\": \"id\", \"value\": \"\"}]"
```

- 파라미터를 사용하여 SQL 문에서 테이블 이름을 설정할 수 없습니다. Data API는 JDBC PreparedStatement의 규칙을 따릅니다.
- describe-statement 작업의 출력은 SQL 문의 쿼리 파라미터를 반환합니다.
- execute-statement 작업만 파라미터가 있는 SQL 문을 지원합니다.

Amazon Redshift Data API를 호출할 때 멍등성 토큰으로 SQL 문 실행

변형 API 요청을 만들 때 요청은 일반적으로 작업의 비동기 워크플로가 완료되기 전에 결과를 반환합니다. 요청이 이미 결과를 반환했다라도 작업이 완료되기 전에 시간이 초과되거나 다른 서버 문제가 발생할 수도 있습니다. 이로 인해 요청의 성공 여부를 판단하기 어려울 수 있으며 작업이 성공적으로 완료되었는지 확인하기 위해 여러 번의 재시도가 발생할 수 있습니다. 그러나 원래 요청과 후속 재시도가 성공하면 작업이 여러 번 완료됩니다. 즉, 의도한 것보다 더 많은 리소스를 업데이트할 수 있습니다.

멍등성은 API 요청이 한 번만 완료되도록 합니다. 멍등성 요청을 사용하면 원래 요청이 성공적으로 완료되면 추가 작업을 수행하지 않고 후속 재시도가 성공적으로 완료됩니다. 데이터 API ExecuteStatement 및 BatchExecuteStatement 작업에는 선택적 ClientToken 멍등성 파라미터가 있습니다. ClientToken은 8시간 후에 만료됩니다.

Important

AWS SDK에서 ExecuteStatement 및 BatchExecuteStatement 작업을 호출하면 재시도 시 사용할 클라이언트 토큰이 자동으로 생성됩니다. 이 경우 ExecuteStatement 및

BatchExecuteStatement 작업과 함께 client-token 파라미터를 사용하지 않는 것이 좋습니다. ClientToken을 보려면 ClientToken을 참조하세요. CloudTrail 로그 예제는 [Amazon Redshift Data API 예제](#)를 참조하세요.

다음 execute-statement AWS CLI 명령은 멍등성을 위한 선택적 client-token 파라미터를 보여줍니다.

```
aws redshift-data execute-statement
  --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPwn
  --cluster-identifier mycluster-test
  --sql "select * from stl_query limit 1"
  --database dev
  --client-token b855dced-259b-444c-bc7b-d3e8e33f94g1
```

다음 테이블은 멍등성 API 요청에 대해 얻을 수 있는 몇 가지 일반적인 응답을 보여주고 재시도 권장 사항을 제공합니다.

응답	권장 사항	설명
200 OK	다시 시도하지 않음	원래 요청이 성공적으로 완료되었습니다. 이후의 모든 재시도는 성공적으로 반환됩니다.
400 시리즈 응답 코드	다시 시도하지 않음	다음 중에서는 요청에 문제가 있습니다. - 유효하지 않은 파라미터 또는 파라미터 조합이 포함되어 있습니다. - 권한이 없는 작업 또는 리소스를 사용합니다. - 상태를 변경하는 과정에 있는 리소스를 사용합니다. 요청에 상태 변경 프로세스에 있는 리소스가 포함된 경우 요청 재시도가 성공할 수 있습니다.

응답	권장 사항	설명
500 시리즈 응답 코드	재시도	이 오류는 AWS 서버 측 문제로 인해 발생하며 일반적으로 일시적입니다. 적절한 백오프 전략으로 요청을 반복합니다.

Amazon Redshift 응답 코드에 대한 자세한 내용은 Amazon Redshift API 참조의 [Common Errors](#)(일반 오류)를 참조하세요.

Amazon Redshift Data API를 직접 호출할 때 세션 재사용으로 SQL 문 실행

SQL 문을 실행하기 위해 API를 요청하면 일반적으로 SQL이 종료될 때 SQL이 실행되는 세션이 종료됩니다. 지정된 시간 동안 세션을 활성화하려면 Data API ExecuteStatement 및 BatchExecuteStatement 작업에는 옵션인 SessionKeepAliveSeconds 파라미터가 있습니다. SessionId 응답 필드에는 세션의 ID가 포함되어 있으며, 이는 후속 ExecuteStatement 및 BatchExecuteStatement 작업에 사용할 수 있습니다. 후속 직접 호출에서 유효 제한 시간을 변경하려면 다른 SessionKeepAliveSeconds를 지정할 수 있습니다. SessionKeepAliveSeconds를 변경하지 않으면 초기 유효 제한 시간 설정이 그대로 유지됩니다. 세션 재사용을 활용할 때는 다음을 고려하세요.

- SessionKeepAliveSeconds의 최댓값은 24시간입니다.
- 세션은 최대 24시간 동안 지속될 수 있습니다. 24시간이 지나면 세션이 강제로 닫히고 진행 중인 쿼리가 종료됩니다.
- Amazon Redshift 클러스터 또는 Redshift Serverless 작업 그룹당 최대 세션 수는 500개입니다.
- 세션에서 한 번에 하나의 쿼리만 실행할 수 있습니다. 동일한 세션에서 다음 쿼리를 실행하려면 쿼리가 완료될 때까지 기다려야 합니다. 즉, 제공된 세션에서는 쿼리를 병렬로 실행할 수 없습니다.
- Data API는 지정된 세션에 대해 쿼리를 대기열에 넣을 수 없습니다.

ExecuteStatement 및 BatchExecuteStatement 작업에 대한 직접 호출에서 사용되는 SessionId를 검색하려면 DescribeStatement 및 ListStatements 작업을 직접 호출합니다.

다음 예제는 세션을 활성화하고 재사용하기 위해 SessionKeepAliveSeconds 및 SessionId 파라미터를 사용하는 방법을 보여줍니다. 먼저 옵션인 session-keep-alive-seconds 파라미터를 2로 설정한 상태에서 execute-statement AWS CLI 명령을 직접 호출합니다.

```
aws redshift-data execute-statement
  --session-keep-alive-seconds 2
  --sql "select 1"
  --database dev
  --workgroup-name mywg
```

응답에는 세션 식별자가 포함됩니다.

```
{
  "WorkgroupName": "mywg",
  "CreatedAt": 1703022996.436,
  "Database": "dev",
  "DbUser": "awsuser",
  "Id": "07c5ffea-76d6-4786-b62c-4fe3ef529680",
  "SessionId": "5a254dc6-4fc2-4203-87a8-551155432ee4"
}
```

그런 다음 첫 번째 직접 호출에서 반환된 SessionId를 사용하여 execute-statement AWS CLI 명령을 직접 호출합니다. 또한, 필요에 따라 session-keep-alive-seconds 파라미터를 10으로 설정하여 유휴 제한 시간 값을 변경합니다.

```
aws redshift-data execute-statement
  --sql "select 1"
  --session-id 5a254dc6-4fc2-4203-87a8-551155432ee4
  --session-keep-alive-seconds 10
```

SQL 문의 결과 가져오기

결과 형식에 따라 다른 데이터 API 작업을 사용하여 SQL 결과를 가져옵니다. ExecuteStatement 및 BatchExecuteStatement 작업을 직접 호출할 때 결과를 JSON 또는 CSV 형식으로 지정할 수 있습니다. 지정하지 않으면 기본값은 JSON입니다. JSON 결과를 검색하려면 GetStatementResult 작업을 사용합니다. CSV 결과를 검색하려면 GetStatementResultV2 작업을 사용합니다.

JSON 형식으로 반환된 결과는 각 열에 대한 메타데이터를 포함하는 레코드입니다. 각 레코드는 JSON 형식입니다. 예를 들어, GetStatementResult의 응답은 다음과 유사합니다.

```
{
  "ColumnMetadata": [
    {
      "isCaseSensitive": false,
```

```

        "isCurrency": false,
        "isSigned": true,
        "label": "?column?",
        "name": "?column?",
        "nullable": 1,
        "precision": 10,
        "scale": 0,
        "schemaName": "",
        "tableName": "",
        "typeName": "int4",
        "length": 0
    }
],
"NextToken": "<token>",
"Records": [
    [
        {
            "longValue": 1
        }
    ]
],
"TotalNumRows": <number>
}

```

CSV 형식으로 반환된 결과는 각 열에 대한 메타데이터를 포함하는 레코드입니다. 결과는 1MB의 청크로 반환되며, 각 청크는 CSV 형식으로 행 수를 제한 없이 저장할 수 있습니다. 각 요청은 최대 15MB의 결과를 반환합니다. 결과가 15MB보다 크면 결과를 계속 검색하기 위해 다음 페이지 토큰이 반환됩니다. 예를 들어, GetStatementResultV2의 응답은 다음과 유사합니다.

```

{
  "ColumnMetadata": [
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": true,
      "label": "?column?",
      "name": "?column?",
      "nullable": 1,
      "precision": 10,
      "scale": 0,
      "schemaName": "",
      "tableName": "",
      "typeName": "int4",

```

```

        "length": 0
    },
    {
        "isCaseSensitive": false,
        "isCurrency": false,
        "isSigned": true,
        "label": "?column?",
        "name": "?column?",
        "nullable": 1,
        "precision": 10,
        "scale": 0,
        "schemaName": "",
        "tableName": "",
        "typeName": "int4",
        "length": 0
    },
    {
        "isCaseSensitive": false,
        "isCurrency": false,
        "isSigned": true,
        "label": "?column?",
        "name": "?column?",
        "nullable": 1,
        "precision": 10,
        "scale": 0,
        "schemaName": "",
        "tableName": "",
        "typeName": "int4",
        "length": 0
    }
],
"NextToken": "<token>",
"Records": [
    [
        {
            "CSVRecords": "1,2,3\r\n4,5,6\r\n7,8,9\r\n, .... 1MB" // First 1MB Chunk
        },
        {
            "CSVRecords": "1025,1026,1027\r\n1028,1029,1030\r\n....2MB" // Second
1MB chunk
        }
        ...
    ]
],

```

```

"ResultFormat" : "CSV",
"TotalNumRows": <number>
}

```

Amazon Redshift Data API에 대한 액세스 권한 부여

Data API에 액세스하려면 사용자에게 권한이 부여되어야 합니다. 해당 사용자에게 미리 정의된 AWS Identity and Access Management(IAM) 정책인 관리형 정책을 추가하여 Data API에 액세스할 수 있는 권한을 부여할 수 있습니다. 가장 좋은 방법은 권한 정책을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에 할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요. 관리형 정책에서 허용 및 거부하는 권한을 보려면 IAM 콘솔(<https://console.aws.amazon.com/iam/>)을 봅니다.

IAM 권한 구성

Amazon Redshift는 AmazonRedshiftDataFullAccess 관리형 정책을 제공합니다. 이 정책은 Amazon Redshift Data API 작업에 대한 전체 액세스 권한을 제공합니다. 이 정책은 또한 Amazon Redshift 클러스터 또는 Redshift Serverless 작업 그룹을 인증하고 이에 액세스하는 데 필요한 특정 Amazon Redshift, AWS Secrets Manager 및 IAM API 작업에 대한 범위 액세스를 허용합니다.

특정 리소스에 대한 액세스를 허용하는 자체 IAM 정책을 생성할 수도 있습니다. 정책을 생성하려면 AmazonRedshiftDataFullAccess 정책을 시작 템플릿으로 사용합니다. 정책을 생성한 후에는 해당 정책을 Data API에 액세스해야 하는 각 사용자에게 추가합니다.

사용자와 연결된 IAM 정책의 다음과 같은 요구 사항을 고려합니다.

- AWS Secrets Manager를 사용하여 인증하는 경우 정책이 `secretsmanager:GetSecretValue` 작업을 사용하여 `RedshiftDataFullAccess` 키로 태그가 지정된 비밀을 검색하도록 허용하는지 확인합니다.
- 임시 보안 인증 정보를 사용하여 클러스터에 인증하는 경우 정책이 클러스터의 모 든 데이터베이스에 대해 데이터베이스 사용자 이름 `redshift_data_api_user`에 `redshift:GetClusterCredentials` 작업을 사용하도록 허용하는지 확인합니다. 이 사용자 이름은 데이터베이스에 이미 생성되어 있어야 합니다.
- 임시 보안 인증 정보를 사용하여 서버리스 작업 그룹에 인증하는 경우 정책이 `RedshiftDataFullAccess` 키로 태깅된 작업 그룹을 가져오는 `redshift-serverless:GetCredentials` 작업의 사용을 허용하는지 확인합니다. 데이터베이스 사용자는 소스 AWS Identity and Access Management(IAM) 아이덴티티에 일대일로 매핑됩니다. 예를 들어 사용자 `sample_user`는 데이터베이스 사용자 IAM:`sample_user`로 매핑되고 IAM 역할 `sample_role`

은 IAMR:sample_role로 매핑됩니다. 다양한 IAM 아이덴티티에 대한 자세한 내용은 IAM 사용 설명서에서 [IAM 자격 증명\(사용자, 그룹 및 역할\)](#)을 참조하세요.

- IAM 작업 redshift-data:GetStatementResult는 GetStatementResult 및 GetStatementResultV2 API 작업 모두에 대한 액세스를 허용합니다.

다음 링크는 IAM User Guide에서 AWS Identity and Access Management에 대한 자세한 정보를 제공합니다.

- IAM 역할 생성에 대한 자세한 내용은 [IAM 역할 생성](#)을 참조하세요.
- IAM 정책 생성에 대한 자세한 내용은 [IAM 정책 생성](#)을 참조하세요.
- 사용자에게 IAM 정책 추가에 대한 자세한 내용은 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하세요.

다른 계정에서 소유하고 있는 클러스터에서 쿼리 실행

다른 계정이 소유한 클러스터에서 쿼리를 실행하려면 소유 계정이 호출 계정에서 Data API가 수임할 수 있는 IAM 역할을 제공해야 합니다. 예를 들어 계정 B가 계정 A가 액세스해야 하는 클러스터를 소유하고 있다고 가정합니다. 계정 B는 AWS 관리형 정책 AmazonRedshiftDataFullAccess를 계정 B의 IAM 역할에 연결할 수 있습니다. 그런 다음 계정 B는 다음과 같은 신뢰 정책을 사용하여 계정 A를 신뢰합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/someRoleA"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```


마지막으로 계정 A IAM 역할은 계정 B IAM 역할을 수임할 수 있어야 합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::111122223333:role/someRoleB"
  }
}
```

AWS 계정의 Redshift Serverless 작업 그룹 및 Amazon Redshift 클러스터로 리소스를 제한하는 IAM 역할 지정

ID 기반 정책에서 리소스 ARN을 지정하여 AWS 계정의 Redshift Serverless 작업 그룹 및 Amazon Redshift 클러스터에 대한 액세스 권한을 제어할 수 있습니다. 이 예제는 지정된 AWS 계정의 작업 그룹 및 클러스터에만 Data API에 액세스할 수 있는 정책을 만드는 방법을 보여줍니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift-data:CancelStatement",
        "redshift-data:DescribeStatement",
        "redshift-data:GetStatementResult",
        "redshift-data:ListStatements"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "redshift-data:*",
      "Resource": [
        "arn:aws:redshift:us-east-1:111122223333:workgroup/*",

```

```

        "arn:aws:redshift:us-east-1:111122223333:cluster:*"
    ]
}

```

SQL 문 정보에 대한 액세스 권한을 문 소유자로만 제한하는 IAM 정책 구성

기본적으로 Amazon Redshift Data API는 `ExecuteStatement` 및 `BatchExecuteStatement` 직접 호출 시 사용되는 IAM 역할을 SQL 문의 소유자로 취급합니다. 역할을 맡을 수 있는 사람은 누구나 결과를 포함하여 SQL 문에 대한 정보에 액세스할 수 있습니다. IAM 역할 세션에 대한 SQL 문 정보 액세스 권한을 특정 소유자로 제한하려면 `redshift-data:statement-owner-iam-userid: "${aws:userid}"` 조건을 추가하세요. 다음 IAM 정책은 액세스를 제한합니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift-data:CancelStatement",
        "redshift-data:DescribeStatement",
        "redshift-data:GetStatementResult",
        "redshift-data:ListStatements"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "redshift-data:statement-owner-iam-userid": "${aws:userid}"
        }
      }
    }
  ]
}

```

CancelStatement, DescribeStatement, GetStatementResult, ListStatements와 함께 statement-owner-iam-userid 조건을 사용할 수 있습니다. 자세한 내용은 [Amazon Redshift Data API에서 정의한 작업](#)을 참조하세요.

SQL 결과에 대한 액세스 권한을 세션 소유자로만 제한하는 IAM 정책 구성

기본적으로 Amazon Redshift Data API는 ExecuteStatement 및 BatchExecuteStatement 직접 호출 시 사용되는 IAM 역할을 SQL 문을 실행하는 데이터베이스 세션의 소유자로 취급합니다. 역할을 맡을 수 있는 사람은 누구나 데이터베이스 세션에 쿼리를 제출할 수 있습니다. 특정 소유자와의 IAM 역할 세션에 대한 세션 액세스를 제한하려면 redshift-data:session-owner-iam-userid: "\${aws:userid}" 조건을 추가하세요. 다음 IAM 정책은 액세스를 제한합니다.

다음 IAM 정책은 세션 소유자만 문 결과를 얻을 수 있도록 허용합니다. session-owner-iam-userid 조건은 리소스 액세스 권한을 지정된 userid로 제한하는 데 사용됩니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift-data:ExecuteStatement",
        "redshift-data:BatchExecuteStatement"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "redshift-data:session-owner-iam-userid": "${aws:userid}"
        }
      }
    }
  ]
}
```

ExecuteStatement, BatchExecuteStatement와 함께 session-owner-iam-userid 조건을 사용할 수 있습니다. 자세한 내용은 [Amazon Redshift Data API에서 정의한 작업](#)을 참조하세요.

AWS Secrets Manager에 데이터베이스 자격 증명 저장

Data API를 호출할 때 AWS Secrets Manager의 비밀을 사용하여 클러스터 또는 서버리스 작업 그룹에 대한 보안 인증 정보를 전달할 수 있습니다. 이 방식으로 자격 증명을 전달하려면 보안 암호의 이름 또는 보안 암호의 Amazon 리소스 이름(ARN)을 지정합니다.

Secrets Manager로 자격 증명을 저장하려면 SecretManagerReadWrite 관리형 정책 권한이 필요합니다. 최소 권한에 대한 자세한 내용은 AWS Secrets Manager User Guide의 [Creating and Managing Secrets with AWS Secrets Manager](#)를 참조하세요.

Amazon Redshift 클러스터의 보안 암호에 자격 증명을 저장하려면

1. AWS Secrets Manager 콘솔을 사용하여 클러스터의 보안 인증 정보를 포함하는 비밀을 생성합니다.
 - [새 보안 암호 저장(Store a new secret)]을 선택할 때 [Redshift 클러스터용 자격 증명(Credentials for Redshift cluster)]을 선택합니다.
 - 보안 암호에 [사용자 이름(User name)](데이터베이스 사용자), [암호>Password)] 및 [DB 클러스터(DB cluster)](클러스터 식별자) 값을 저장합니다.
 - 키 RedshiftDataFullAccess로 보안 암호에 태그를 지정합니다. AWS 관리형 정책 AmazonRedshiftDataFullAccess는 키 RedshiftDataFullAccess로 태그가 지정된 보안 암호에 대해서만 작업 secretsmanager:GetSecretValue를 허용합니다.

이에 관한 지침은 AWS Secrets Manager User Guide의 [Creating a Basic Secret](#)을 참조하세요.

2. AWS Secrets Manager 콘솔을 사용하여 생성한 보안 암호에 대한 세부 정보를 보거나 `aws secretsmanager describe-secret` AWS CLI 명령을 실행합니다.

보안 암호의 이름 및 ARN을 적어둡니다. 이러한 이름이나 ARN은 Data API 호출에서 사용할 수 있습니다.

서버리스 작업 그룹의 비밀에 보안 인증 정보를 저장하려면

1. 서버리스 작업 그룹에 대한 보안 인증 정보를 포함하는 비밀을 저장하려면 AWS Secrets Manager AWS CLI 명령을 사용합니다.
 - 파일에 보안 암호를 생성합니다(예: mycreds.json 이름의 JSON 파일) 파일에 사용자 이름(데이터베이스 사용자) 및 암호 값을 제공합니다.

```
{
  "username": "myusername",
  "password": "mypassword"
}
```

- 비밀에 값을 저장하고 RedshiftDataFullAccess 키를 사용하여 비밀에 태그를 지정합니다.

```
aws secretsmanager create-secret --name MyRedshiftSecret --tags
Key="RedshiftDataFullAccess",Value="serverless" --secret-string file://
mycreds.json
```

다음은 출력값을 보여줍니다.

```
{
  "ARN":
  "arn:aws:secretsmanager:region:accountId:secret:MyRedshiftSecret-mvLHxf",
  "Name": "MyRedshiftSecret",
  "VersionId": "a1603925-e8ea-4739-9ae9-e509eEXAMPLE"
}
```

자세한 내용은 [AWS CLI 사용 설명서](#)에서 AWS Secrets Manager로 기본 보안 암호 생성을 참조하세요.

2. AWS Secrets Manager 콘솔을 사용하여 생성한 보안 암호에 대한 세부 정보를 보거나 `aws secretsmanager describe-secret` AWS CLI 명령을 실행합니다.

보안 암호의 이름 및 ARN을 적어둡니다. 이러한 이름이나 ARN은 Data API 호출에서 사용할 수 있습니다.

Data API에 대한 Amazon VPC 엔드포인트(AWS PrivateLink) 생성

Amazon Virtual Private Cloud(Amazon VPC)를 사용하면 Amazon Redshift 클러스터 및 애플리케이션과 같은 AWS 리소스를 Virtual Private Cloud(VPC)에서 시작할 수 있습니다. AWS PrivateLink는 Amazon 네트워크에서 Virtual Private Cloud(VPC)와 AWS 서비스 간의 프라이빗 연결을 안전하게 제공합니다. AWS PrivateLink를 사용하면 VPC 엔드포인트를 생성하여 Amazon VPC 기반의 다른 계정 및 VPC에서 서비스에 연결할 수 있습니다. AWS PrivateLink에 대한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 [VPC 엔드포인트 서비스\(AWS PrivateLink\)](#)를 참조하세요.

Amazon VPC 엔드포인트를 사용하여 데이터 API를 호출할 수 있습니다. Amazon VPC 엔드포인트를 사용하면 퍼블릭 IP 주소를 사용하지 않고 Amazon VPC의 애플리케이션과 AWS 네트워크에서 데이터 API 간에 트래픽을 유지합니다. Amazon VPC 엔드포인트를 사용하면 퍼블릭 인터넷 연결 제한과 관련된 규정 준수 및 규정 요구 사항을 충족할 수 있습니다. 예를 들어 Amazon VPC 엔드포인트를 사용하는 경우 Amazon EC2 인스턴스에서 실행되는 애플리케이션과 해당 애플리케이션이 포함된 VPC의 데이터 API 간 트래픽을 유지할 수 있습니다.

Amazon VPC 엔드포인트를 생성한 후에는 애플리케이션에서 코드나 구성을 변경하지 않고 엔드포인트를 사용할 수 있습니다.

데이터 API에 대한 Amazon VPC 엔드포인트를 생성하려면

1. AWS Management Console에 로그인하고 <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 엔드포인트를 선택한 다음 엔드포인트 생성을 선택합니다.
3. 엔드포인트 생성 페이지에서 서비스 범주에 대해 AWS 서비스를 선택합니다. [서비스 이름 (Service Name)]에서 [redshift-data](com.amazonaws.*region*.redshift-data)를 선택합니다.
4. VPC의 경우 엔드포인트를 생성할 VPC를 선택합니다.

데이터 API를 호출하는 애플리케이션이 포함된 VPC를 선택합니다.

5. 서브넷의 경우 애플리케이션을 실행 중인 AWS 서비스에서 사용하는 각 가용 영역(AZ)의 서브넷을 선택합니다.

Amazon VPC 엔드포인트를 생성하려면 엔드포인트에 액세스할 수 있는 프라이빗 IP 주소 범위를 지정합니다. 이렇게 하려면 각 가용 영역에 대한 서브넷을 선택합니다. 이렇게 하면 VPC 엔드포인트가 각 가용 영역별 프라이빗 IP 주소 범위로 제한되고 각 가용 영역에 Amazon VPC 엔드포인트가 생성됩니다.

6. Enable DNS name(DNS 이름 활성화)에서 이 엔드포인트에 대해 활성화를 선택합니다.

프라이빗 DNS는 표준 데이터 API DNS 호스트 이름([https://redshift-data.*region*.amazonaws.com](https://redshift-data.<i>region</i>.amazonaws.com))을 Amazon VPC 엔드포인트에 특정한 DNS 호스트 이름과 연결된 프라이빗 IP 주소로 확인합니다. 따라서 데이터 API 엔드포인트 URL을 업데이트하기 위해 코드나 구성을 변경하지 않고도 AWS CLI 또는 AWS SDK를 사용하여 데이터 API VPC 엔드포인트에 액세스할 수 있습니다.

7. 보안 그룹의 경우 Amazon VPC 엔드포인트와 연결할 보안 그룹을 선택합니다.

애플리케이션을 실행 중인 AWS 서비스에 대한 액세스를 허용하는 보안 그룹을 선택합니다. 예를 들어 Amazon EC2 인스턴스가 애플리케이션을 실행 중인 경우 Amazon EC2 인스턴스에 대한 액세스를 허용하는 보안 그룹을 선택합니다. 보안 그룹을 사용하면 VPC의 리소스에서 Amazon VPC 엔드포인트로 가는 트래픽을 제어할 수 있습니다.

8. [Create endpoint]를 선택합니다.

엔드포인트를 생성한 후 AWS Management Console에서 링크를 선택하여 엔드포인트 세부 정보를 봅니다.

엔드포인트 세부 정보 탭에는 Amazon VPC 엔드포인트를 만드는 동안 생성된 DNS 호스트 이름이 표시됩니다.

표준 엔드포인트(redshift-data.**region**.amazonaws.com) 또는 VPC 관련 엔드포인트 중 하나를 사용하여 Amazon VPC에서 데이터 API를 호출할 수 있습니다. 표준 데이터 API 엔드포인트는 자동으로 Amazon VPC 엔드포인트로 라우팅됩니다. 이 라우팅은 Amazon VPC 엔드포인트를 생성할 때 프라이빗 DNS 호스트 이름을 활성화했기 때문에 발생합니다.

데이터 API 호출에서 Amazon VPC 엔드포인트를 사용하는 경우 애플리케이션과 데이터 API 간의 모든 트래픽은 해당 트래픽이 포함된 Amazon VPC에 남아 있습니다. 모든 유형의 데이터 API 호출에 Amazon VPC 엔드포인트를 사용할 수 있습니다. 데이터 API 호출에 대한 자세한 내용은 [Amazon Redshift Data API 호출 시 고려 사항](#) 섹션을 참조하세요.

클러스터에 연결할 때 데이터베이스 그룹에 조인

데이터베이스 그룹은 데이터베이스 사용자의 모음입니다. 그룹에 데이터베이스 권한을 부여할 수 있습니다. 관리자는 데이터 API로 SQL을 실행할 때 이러한 데이터베이스 그룹이 고려되도록 IAM 역할을 구성할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [그룹](#)을 참조하세요.

데이터 API가 클러스터에 연결할 때 호출에 지정된 데이터베이스 사용자가 데이터베이스 그룹에 조인하도록 데이터 API 호출자의 IAM 역할을 구성할 수 있습니다. 이 기능은 프로비저닝된 클러스터에 연결할 때만 지원됩니다. Redshift Serverless 작업 그룹에 연결할 때는 지원되지 않습니다. 데이터 API 호출자의 IAM 역할도 redshift:JoinGroup 작업을 허용해야 합니다.

IAM 역할에 태그를 추가하여 이를 구성합니다. 호출자의 IAM 역할의 관리자는 RedshiftDbGroups 키와 데이터베이스 그룹 목록의 키 값을 사용하여 태그를 추가합니다. 값은 콜론(:)으로 구분된 데이터베이스 그룹의 이름을 총 256자까지 나열한 목록입니다. 데이터베이스 그룹은 연결된 데이터베이스에 이전에 정의되어 있어

야 합니다. 지정된 그룹이 데이터베이스에서 발견되지 않으면 무시됩니다. 예를 들어 accounting 및 retail 데이터베이스 그룹의 경우 키-값은 accounting:retail입니다. 태그 키-값 쌍 {"Key": "RedshiftDbGroups", "Value": "accounting:retail"}은 데이터 API에서 데이터 API 호출에서 제공된 데이터베이스 사용자와 연관된 데이터베이스 그룹을 결정하는 데 사용됩니다.

데이터베이스 그룹에 조인하려면

1. AWS Management Console에 로그인하여 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 콘솔의 탐색 창에서 [역할(Roles)]을 선택한 다음 편집할 역할의 이름을 선택합니다.
3. 태그 탭을 선택한 후 태그 관리를 선택합니다.
4. 태그 추가를 선택한 다음 RedshiftDbGroups 키와 *database-groups-colon-separated* 목록인 값을 추가합니다.
5. 변경 사항 저장을 선택합니다.

이제 이 IAM 역할이 첨부된 IAM 주체가 Data API를 호출하면 지정된 데이터베이스 사용자가 IAM 역할에 지정된 데이터베이스 그룹에 조인합니다.

IAM 역할 및 IAM 사용자를 포함하여 보안 주체에 태그를 연결하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 리소스에 태깅](#)을 참조하세요.

신뢰할 수 있는 자격 증명 전파와 함께 데이터 API 사용

Amazon Redshift 계정 관리자는 Amazon Redshift 클러스터 또는 작업 그룹을 AWS IAM Identity Center와 통합할 수 있으므로 Single Sign-On으로 Amazon Redshift에 대한 인력 액세스를 관리하는 데 도움이 됩니다. 자세한 내용은 [Amazon Redshift와 AWS IAM Identity Center 통합 설정](#) 섹션을 참조하세요. Amazon Redshift 데이터 API는 IAM Identity Center 사용자 ID를 Amazon Redshift 클러스터 또는 작업 그룹 및 AWS Lake Formation과 같은 체인 아래 단계의 다른 서비스에 전파하는 것을 지원합니다. [신뢰할 수 있는 ID 전파를 사용하여 프로그래밍 방식으로 AWS 서비스 액세스](#)의 단계에 따라 데이터 API를 사용하여 설정하고 쿼리할 수 있습니다.

ID 강화 IAM 역할 세션에서 IAM Identity Center 사용자 ID를 사용하여 데이터 API를 직접적으로 호출하는 경우, 동일한 IAM Identity Center 사용자를 사용하여 결과 문 및 문 결과에만 액세스할 수 있습니다. 예를 들어 다음 AWS CLI 명령은 execute-statement 작업을 직접적으로 호출함으로써 신뢰할 수 있는 ID 전파를 사용하여 SQL 명령을 실행합니다.

```
aws redshift-data execute-statement
```



```
--sql "select current_user;"
--cluster-id mycluster
--database dev
```

다음 AWS CLI 명령은 batch-execute-statement 작업을 직접적으로 호출하여 두 개의 SQL 명령을 실행합니다.

```
aws redshift-data batch-execute-statement
--sqls "select current_user;" "select current_date;"
--cluster-id mycluster
--database dev
```

자격 증명 강화 IAM 역할 세션에서 제출한 cancel-statement, describe-statement, get-statement-result 및 get-statement-result-v2를 사용하여 문에 액세스하려면 IAM Identity Center 사용자 및 IAM 역할이 execute-statement 또는 batch-execute-statement를 실행하는데 사용된 자격 증명과 일치해야 합니다. 예를 들어 다음 AWS CLI 명령은 SQL 문의 결과를 가져옵니다.

```
aws redshift-data get-statement-result
--id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

문을 나열하려면 IAM Identity Center 사용자가 할당된 Amazon Redshift IAM Identity Center 애플리케이션에만 액세스할 수 있도록 cluster-identifier 또는 workgroup-name 파라미터를 제공해야 합니다. 예를 들어 다음 AWS CLI 명령은 특정 클러스터에 대한 문을 나열합니다.

```
aws redshift-data list-statements
--cluster-identifier mycluster
```

또한 신뢰할 수 있는 자격 증명 전파를 사용하여 클러스터 또는 작업 그룹의 데이터베이스 객체에 액세스하는 데이터 API 작업을 간접적으로 호출할 수 있습니다. 여기에는 list-databases, list-schemas, list-tables 및 describe-table 작업이 포함됩니다.

IAM Identity Center 사용자가 수행한 API 직접 호출은 AWS CloudTrail에서 추적할 수 있습니다. CloudTrail 이벤트의 onBehalfOf 섹션에는 IAM Identity Center 사용자 ID와 ID 스토어 ARN이 표시됩니다. 다음 예제는 IAM Identity Center 사용자 ID가 a1b2c3d4-5678-90ab-cdef-EXAMPLE11111이고 Identity 스토어 ARN이 arn:aws:identitystore::123456789012:identitystore/d-9067bc44d2인 onBehalfOf 섹션이 표시된 CloudTrail 이벤트의 스니펫을 보여줍니다.

```
{
    "eventVersion": "1.10",
    "userIdentity": {
        "type": "AssumedRole",
        ...
    },
    "onBehalfOf": {
        "userId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "identityStoreArn": "arn:aws:identitystore::123456789012:identitystore/d-9067bc44d2"
    },
    "eventTime": "2025-01-13T04:46:27Z",
    "eventSource": "redshift-data.amazonaws.com",
    "eventName": "ExecuteStatement",
    "awsRegion": "us-east-1"
}
```

다음 SQL 명령을 실행하여 IAM Identity Center 사용자가 제출한 쿼리를 확인할 수 있습니다. 이 예제에서 Identity Center에 등록된 이메일은 `username@example.com`입니다.

```
SELECT
    h.query_id,
    h.database_name,
    h.status,
    h.query_text,
    u.username,
    h.start_time,
    h.end_time
FROM
    sys_query_history h
LEFT JOIN
    pg_user u
ON
    h.user_id = u.usesysid
where u.username='awsidc:username@example.com'
ORDER BY
    h.start_time DESC;
```

데이터 API 호출

Data API 또는 AWS CLI를 호출하여 클러스터 또는 서버리스 작업 그룹에서 SQL 문을 실행할 수 있습니다. SQL 문을 실행하는 프라이머리 작업은 Amazon Redshift Data API 참조 내 [ExecuteStatement](#) 및 [BatchExecuteStatement](#)입니다. Data API는 AWS SDK에서 지원되는 프로그래밍 언어를 지원합니다. 이에 대한 자세한 내용은 [AWS 기반의 도구](#)를 참조하세요.

Data API 호출의 코드 예제를 보려면 GitHub에서 [Getting Started with Redshift Data API](#)를 참조하세요. 이 리포지토리에는 AWS Lambda를 사용하여 Amazon EC2, AWS Glue Data Catalog 및 Amazon SageMaker Runtime에서 Amazon Redshift 데이터에 액세스하는 예가 있습니다. 프로그래밍 언어의 예로는 Python, Go, Java 및 Javascript가 있습니다.

AWS CLI를 사용하여 Data API를 호출할 수 있습니다.

다음 예에서는 AWS CLI를 사용하여 Data API를 호출합니다. 예제를 실행하려면 환경에 맞게 파라미터 값을 편집합니다. 많은 예제에서 `cluster-identifier`가 클러스터에 대해 실행되도록 제공됩니다. 서버리스 작업 그룹에 대해 실행할 때는 `workgroup-name`을 대신 제공해야 합니다. 이 예에서는 몇 가지 Data API 작업을 보여줍니다. 자세한 내용은 AWS CLI 명령 참조를 참조하세요.

다음 예의 명령은 가독성을 위해 분할되고 형식이 지정되었습니다. 모든 파라미터와 응답이 모든 예시에 표시되는 것은 아닙니다. 전체 요청 구문, 요청 파라미터, 응답 구문 및 응답 요소의 API 정의는 [Amazon Redshift Data API Reference](#)를 참조하세요.

Amazon Redshift 데이터 웨어하우스에 SQL 스테이트먼트 전달

이 페이지의 예제에서는 데이터 웨어하우스에 SQL 스테이트먼트를 전달하는 다양한 방법을 다룹니다.

SQL 스테이트먼트 실행

SQL 문을 실행하려면 `aws redshift-data execute-statement` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 클러스터에 대해 SQL 문을 실행하고 결과를 가져올 식별자를 반환합니다. 이 예에서는 AWS Secrets Manager 인증 방법을 사용합니다.

```
aws redshift-data execute-statement
  --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPwn
  --cluster-identifier mycluster-test
  --sql "select * from stl_query limit 1"
```

```
--database dev
```

다음은 이 응답의 예입니다.

```
{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": 1598323175.823,
  "Database": "dev",
  "Id": "c016234e-5c6c-4bc5-bb16-2c5b8ff61814",
  "SecretArn": "arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-hKgPWn"
}
```

다음 AWS CLI 명령은 클러스터에 대해 SQL 문을 실행하고 결과를 가져올 식별자를 반환합니다. 이 예에서는 임시 자격 증명 인증 방법을 사용합니다.

```
aws redshift-data execute-statement
  --db-user myuser
  --cluster-identifier mycluster-test
  --database dev
  --sql "select * from stl_query limit 1"
```

다음은 이 응답의 예입니다.

```
{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": 1598306924.632,
  "Database": "dev",
  "DbUser": "myuser",
  "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

다음 AWS CLI 명령은 서버리스 작업 그룹에 대해 SQL 문을 실행하고 결과를 가져올 식별자를 반환합니다. 이 예에서는 임시 자격 증명 인증 방법을 사용합니다.

```
aws redshift-data execute-statement
  --database dev
  --workgroup-name myworkgroup
  --sql "select 1;"
```

다음은 이 응답의 예입니다.

```
{
  "CreatedAt": "2022-02-11T06:25:28.748000+00:00",
  "Database": "dev",
  "DbUser": "IAMR:RoleName",
  "Id": "89dd91f5-2d43-43d3-8461-f33aa093c41e",
  "WorkgroupName": "myworkgroup"
}
```

다음 AWS CLI 명령은 클러스터에 대해 SQL 문을 실행하고 결과를 가져올 식별자를 반환합니다. 이 예에서는 AWS Secrets Manager 인증 방법과 맥등성 토큰을 사용합니다.

```
aws redshift-data execute-statement
  --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
  --cluster-identifier mycluster-test
  --sql "select * from stl_query limit 1"
  --database dev
  --client-token b855dced-259b-444c-bc7b-d3e8e33f94g1
```

다음은 이 응답의 예입니다.

```
{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": 1598323175.823,
  "Database": "dev",
  "Id": "c016234e-5c6c-4bc5-bb16-2c5b8ff61814",
  "SecretArn": "arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn"
}
```

파라미터가 있는 SQL 스테이트먼트 실행

SQL 문을 실행하려면 `aws redshift-data execute-statement` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 클러스터에 대해 SQL 문을 실행하고 결과를 가져올 식별자를 반환합니다. 이 예에서는 AWS Secrets Manager 인증 방법을 사용합니다. SQL 텍스트에는 명명된 파라미터 `distance`가 있습니다. 이 경우, 조건자에 사용되는 거리는 5입니다. SELECT 문에서는 열 이름에 대해 명명된 파라미터를 조건자에서만 사용할 수 있습니다. SQL 문의 명명된 파라미터 값은 `parameters` 옵션에 지정됩니다.

```
aws redshift-data execute-statement
  --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPwn
  --cluster-identifier mycluster-test
  --sql "SELECT ratecode FROM demo_table WHERE trip_distance > :distance"
  --parameters "[{\"name\": \"distance\", \"value\": \"5\"}]"
  --database dev
```

다음은 이 응답의 예입니다.

```
{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": 1598323175.823,
  "Database": "dev",
  "Id": "c016234e-5c6c-4bc5-bb16-2c5b8ff61814",
  "SecretArn": "arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPwn"
}
```

다음 예에서는 샘플 데이터베이스의 EVENT 테이블을 사용합니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [EVENT 테이블](#)을 참조하세요.

데이터베이스에 EVENT 테이블이 없으면 다음과 같이 Data API를 사용하여 생성할 수 있습니다.

```
aws redshift-data execute-statement
  --database dev
  --cluster-id mycluster-test
  --db-user awsuser
  --sql "create table event( eventid integer not null distkey,
                           venueid smallint not null,
                           catid smallint not null,
                           dateid smallint not null sortkey,
                           eventname varchar(200),
                           starttime timestamp)"
```

다음 명령은 EVENT 테이블에 한 행을 삽입합니다.

```
aws redshift-data execute-statement
  --database dev
```

```
--cluster-id mycluster-test
--db-user awsuser
--sql "insert into event
values(:eventid, :venueid::smallint, :catid, :dateid, :eventname, :starttime)"
--parameters "[{"name\: \"eventid\", \"value\: \"1\"}, {"name\: \"venueid\",
\"value\: \"1\"},
                {\"name\: \"catid\", \"value\: \"1\"},
                {\"name\: \"dateid\", \"value\: \"1\"},
                {\"name\: \"eventname\", \"value\: \"event 1\"},
                {\"name\: \"starttime\", \"value\: \"2022-02-22\"}]"
```

다음 명령은 EVENT 테이블에 두 번째 행을 삽입합니다. 이 예제에서는 다음 작업을 설명합니다.

- id라는 파라미터가 SQL 텍스트에서 4번 사용됩니다.
- 파라미터 starttime을 삽입할 때 암시적 형식 변환이 자동으로 적용됩니다.
- venueid 열은 SMALLINT 데이터 형식으로 캐스팅됩니다.
- DATE 데이터 형식을 나타내는 문자열은 암시적으로 TIMESTAMP 데이터 형식으로 변환됩니다.
- 주석은 SQL 텍스트 내에서 사용할 수 있습니다.

```
aws redshift-data execute-statement
--database dev
--cluster-id mycluster-test
--db-user awsuser
--sql "insert into event values(:id, :id::smallint, :id, :id, :eventname, :starttime) /
*this is comment, and it won't apply parameterization for :id, :eventname or :starttime
here*/"
--parameters "[{"name\: \"eventname\", \"value\: \"event 2\"},
                {\"name\: \"starttime\", \"value\: \"2022-02-22\"},
                {\"name\: \"id\", \"value\: \"2\"}]"
```

다음은 삽입된 두 행을 보여줍니다.

eventid	venueid	catid	dateid	eventname	starttime
1	1	1	1	event 1	2022-02-22 00:00:00
2	2	2	2	event 2	2022-02-22 00:00:00

다음 명령은 WHERE 절에서 명명된 파라미터를 사용하여 행을 검색합니다. 여기서 eventid는 1입니다.

```
aws redshift-data execute-statement
--database dev
--cluster-id mycluster-test
--db-user awsuser
--sql "select * from event where eventid=:id"
--parameters "[{"name\": \"id\", \"value\": \"1\"}]"
```

다음 명령을 실행하여 이전 SQL 문의 SQL 결과를 가져옵니다.

```
aws redshift-data get-statement-result --id 7529ad05-b905-4d71-9ec6-8b333836eb5a
```

다음 결과를 제공합니다.

```
{
  "Records": [
    [
      {
        "longValue": 1
      },
      {
        "longValue": 1
      },
      {
        "longValue": 1
      },
      {
        "longValue": 1
      },
      {
        "stringValue": "event 1"
      },
      {
        "stringValue": "2022-02-22 00:00:00.0"
      }
    ]
  ]
}
```



```
    }
  ]
],
"ColumnMetadata": [
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "eventid",
    "length": 0,
    "name": "eventid",
    "nullable": 0,
    "precision": 10,
    "scale": 0,
    "schemaName": "public",
    "tableName": "event",
    "typeName": "int4"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "venueid",
    "length": 0,
    "name": "venueid",
    "nullable": 0,
    "precision": 5,
    "scale": 0,
    "schemaName": "public",
    "tableName": "event",
    "typeName": "int2"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "catid",
    "length": 0,
    "name": "catid",
    "nullable": 0,
    "precision": 5,
    "scale": 0,
    "schemaName": "public",
    "tableName": "event",
```

```
    "typeName": "int2"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "dateid",
    "length": 0,
    "name": "dateid",
    "nullable": 0,
    "precision": 5,
    "scale": 0,
    "schemaName": "public",
    "tableName": "event",
    "typeName": "int2"
  },
  {
    "isCaseSensitive": true,
    "isCurrency": false,
    "isSigned": false,
    "label": "eventname",
    "length": 0,
    "name": "eventname",
    "nullable": 1,
    "precision": 200,
    "scale": 0,
    "schemaName": "public",
    "tableName": "event",
    "typeName": "varchar"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": false,
    "label": "starttime",
    "length": 0,
    "name": "starttime",
    "nullable": 1,
    "precision": 29,
    "scale": 6,
    "schemaName": "public",
    "tableName": "event",
    "typeName": "timestamp"
  }
}
```

```
],
  "TotalNumRows": 1
}
```

여러 SQL 스테이트먼트 실행

하나의 명령으로 여러 SQL 문을 실행하려면 `aws redshift-data batch-execute-statement` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 클러스터에 대해 3개의 SQL 문을 실행하고 결과를 가져올 식별자를 반환합니다. 이 예에서는 임시 자격 증명 인증 방법을 사용합니다.

```
aws redshift-data batch-execute-statement
  --db-user myuser
  --cluster-identifier mycluster-test
  --database dev
  --sqls "set timezone to BST" "select * from mytable" "select * from another_table"
```

다음은 이 응답의 예입니다.

```
{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": 1598306924.632,
  "Database": "dev",
  "DbUser": "myuser",
  "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

SQL 스테이트먼트에 대한 메타데이터 나열

SQL 문에 대한 메타데이터를 나열하려면 `aws redshift-data list-statements` AWS CLI 명령을 사용합니다. 이 명령을 실행하기 위한 권한 부여는 호출자의 IAM 권한을 기반으로 합니다.

다음 AWS CLI 명령은 실행된 SQL 문을 나열합니다.

```
aws redshift-data list-statements
  --status ALL
```

다음은 이 응답의 예입니다.

```
{
  "Statements": [
    {
      "CreatedAt": 1598306924.632,
      "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766",
      "QueryString": "select * from stl_query limit 1",
      "Status": "FINISHED",
      "UpdatedAt": 1598306926.667
    },
    {
      "CreatedAt": 1598311717.437,
      "Id": "e0ebd578-58b3-46cc-8e52-8163fd7e01aa",
      "QueryString": "select * from stl_query limit 1",
      "Status": "FAILED",
      "UpdatedAt": 1598311719.008
    },
    {
      "CreatedAt": 1598313683.65,
      "Id": "c361d4f7-8c53-4343-8c45-6b2b1166330c",
      "QueryString": "select * from stl_query limit 1",
      "Status": "ABORTED",
      "UpdatedAt": 1598313685.495
    },
    {
      "CreatedAt": 1598306653.333,
      "Id": "a512b7bd-98c7-45d5-985b-a715f3cfde7f",
      "QueryString": "select 1",
      "Status": "FINISHED",
      "UpdatedAt": 1598306653.992
    }
  ]
}
```

SQL 스테이트먼트에 대한 메타데이터 설명

SQL 문에 대한 메타데이터에 대한 설명을 가져오려면 `aws redshift-data describe-statement` AWS CLI 명령을 사용합니다. 이 명령을 실행하기 위한 권한 부여는 호출자의 IAM 권한을 기반으로 합니다.

다음 AWS CLI 명령은 SQL 문을 설명합니다.

```
aws redshift-data describe-statement
```

```
--id d9b6c0c9-0747-4bf4-b142-e8883122f766
```

다음은 이 응답의 예입니다.

```
{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": 1598306924.632,
  "Duration": 1095981511,
  "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766",
  "QueryString": "select * from stl_query limit 1",
  "RedshiftPid": 20859,
  "RedshiftQueryId": 48879,
  "ResultRows": 1,
  "ResultSize": 4489,
  "Status": "FINISHED",
  "UpdatedAt": 1598306926.667
}
```

다음은 여러 SQL 문으로 batch-execute-statement 명령을 실행한 후 describe-statement 응답의 예입니다.

```
{
  "ClusterIdentifier": "mayo",
  "CreatedAt": 1623979777.126,
  "Duration": 6591877,
  "HasResultSet": true,
  "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652",
  "RedshiftPid": 31459,
  "RedshiftQueryId": 0,
  "ResultRows": 2,
  "ResultSize": 22,
  "Status": "FINISHED",
  "SubStatements": [
    {
      "CreatedAt": 1623979777.274,
      "Duration": 3396637,
      "HasResultSet": true,
      "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:1",
      "QueryString": "select 1;",
      "RedshiftQueryId": -1,
      "ResultRows": 1,
      "ResultSize": 11,
      "Status": "FINISHED",
    }
  ]
}
```

```

      "UpdatedAt": 1623979777.903
    },
    {
      "CreatedAt": 1623979777.274,
      "Duration": 3195240,
      "HasResultSet": true,
      "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2",
      "QueryString": "select 2;",
      "RedshiftQueryId": -1,
      "ResultRows": 1,
      "ResultSize": 11,
      "Status": "FINISHED",
      "UpdatedAt": 1623979778.076
    }
  ],
  "UpdatedAt": 1623979778.183
}

```

SQL 스테이트먼트 결과 가져오기

실행된 SQL 문에서 결과를 가져오려면 `redshift-data get-statement-result` 또는 `redshift-data get-statement-result-v2` AWS CLI 명령을 사용합니다. `get-statement-result`의 결과는 JSON 형식입니다. `get-statement-result-v2`의 결과는 CSV 형식입니다. `execute-statement` 또는 `batch-execute-statement`에 대한 응답으로 수신하는 Id를 제공할 수 있습니다. `batch-execute-statement`에 의해 실행된 SQL 문의 Id 값은 `describe-statement`의 결과에서 검색할 수 있으며 `b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2`와 같이 콜론과 시퀀스 번호가 접미사로 붙습니다. `batch-execute-statement`가 있는 여러 SQL 문을 실행하면 `describe-statement`와 같이 각 SQL 문에 Id 값이 있습니다. 이 명령을 실행하기 위한 권한 부여는 호출자의 IAM 권한을 기반으로 합니다.

다음 문은 `ResultFormat`이 JSON의 기본값을 갖게 하는 `execute-statement`에 의해 실행되는 SQL 문의 결과를 반환합니다. 결과를 검색하려면 `get-statement-result` 작업을 직접 호출합니다.

```

aws redshift-data get-statement-result
  --id d9b6c0c9-0747-4bf4-b142-e8883122f766

```

다음 문은 `batch-execute-statement`에 의해 실행되는 두 번째 SQL 문의 결과를 반환합니다.

```
aws redshift-data get-statement-result
--id b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2
```

다음은 SQL 결과가 응답 키 Records의 JSON 형식으로 반환되는 get-statement-result 직접 호출에 대한 응답의 예입니다.

```
{
  "ColumnMetadata": [
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": true,
      "label": "userid",
      "length": 0,
      "name": "userid",
      "nullable": 0,
      "precision": 10,
      "scale": 0,
      "schemaName": "",
      "tableName": "stll_query",
      "typeName": "int4"
    },
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": true,
      "label": "query",
      "length": 0,
      "name": "query",
      "nullable": 0,
      "precision": 10,
      "scale": 0,
      "schemaName": "",
      "tableName": "stll_query",
      "typeName": "int4"
    },
    {
      "isCaseSensitive": true,
      "isCurrency": false,
      "isSigned": false,
      "label": "label",
      "length": 0,
      "name": "label",
```

```
    "nullable": 0,
    "precision": 320,
    "scale": 0,
    "schemaName": "",
    "tableName": "stll_query",
    "typeName": "bpchar"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "xid",
    "length": 0,
    "name": "xid",
    "nullable": 0,
    "precision": 19,
    "scale": 0,
    "schemaName": "",
    "tableName": "stll_query",
    "typeName": "int8"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "pid",
    "length": 0,
    "name": "pid",
    "nullable": 0,
    "precision": 10,
    "scale": 0,
    "schemaName": "",
    "tableName": "stll_query",
    "typeName": "int4"
  },
  {
    "isCaseSensitive": true,
    "isCurrency": false,
    "isSigned": false,
    "label": "database",
    "length": 0,
    "name": "database",
    "nullable": 0,
    "precision": 32,
```



```
    "scale": 0,
    "schemaName": "",
    "tableName": "stll_query",
    "typeName": "bpchar"
  },
  {
    "isCaseSensitive": true,
    "isCurrency": false,
    "isSigned": false,
    "label": "querytxt",
    "length": 0,
    "name": "querytxt",
    "nullable": 0,
    "precision": 4000,
    "scale": 0,
    "schemaName": "",
    "tableName": "stll_query",
    "typeName": "bpchar"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": false,
    "label": "starttime",
    "length": 0,
    "name": "starttime",
    "nullable": 0,
    "precision": 29,
    "scale": 6,
    "schemaName": "",
    "tableName": "stll_query",
    "typeName": "timestamp"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": false,
    "label": "endtime",
    "length": 0,
    "name": "endtime",
    "nullable": 0,
    "precision": 29,
    "scale": 6,
    "schemaName": "",
```

```
    "tableName": "stll_query",
    "type": 93,
    "typeName": "timestamp"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "aborted",
    "length": 0,
    "name": "aborted",
    "nullable": 0,
    "precision": 10,
    "scale": 0,
    "schemaName": "",
    "tableName": "stll_query",
    "typeName": "int4"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "insert_pristine",
    "length": 0,
    "name": "insert_pristine",
    "nullable": 0,
    "precision": 10,
    "scale": 0,
    "schemaName": "",
    "tableName": "stll_query",
    "typeName": "int4"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "concurrency_scaling_status",
    "length": 0,
    "name": "concurrency_scaling_status",
    "nullable": 0,
    "precision": 10,
    "scale": 0,
    "schemaName": "",
    "tableName": "stll_query",
```

```
        "typeName": "int4"
    }
],
"Records": [
    [
        {
            "longValue": 1
        },
        {
            "longValue": 3
        },
        {
            "stringValue": "health"
        },
        {
            "longValue": 1023
        },
        {
            "longValue": 15279
        },
        {
            "stringValue": "dev"
        },
        {
            "stringValue": "select system_status from stv_gui_status;"
        },
        {
            "stringValue": "2020-08-21 17:33:51.88712"
        },
        {
            "stringValue": "2020-08-21 17:33:52.974306"
        },
        {
            "longValue": 0
        },
        {
            "longValue": 0
        },
        {
            "longValue": 6
        }
    ]
],
"TotalNumRows": 1
```

```
}
```

다음 예시에서는 결과를 JSON으로 반환하기 위해 `execute-statement`로 실행되는 SQL 문을 보여줍니다. 테이블 `testingtable`에는 3개의 정수 열(`col1`, `col2`, `col3`)이 있고, 값(1, 2, 3), (4, 5, 6), (7, 8, 9)이 있는 3개의 행이 있습니다.

```
aws redshift-data execute-statement
--database dev
--sql "SELECT col1, col2, col3 FROM testingtable"
--cluster-id mycluster-test
--result-format JSON
```

```
{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": "2024-04-02T16:45:25.144000+00:00",
  "Database": "dev",
  "DbUser": "IAMR:Administrator",
  "Id": "d468d942-6df9-4f85-8ae3-bac01a61aec3"
}
```

다음은 SQL 결과가 응답 키 `Records`의 JSON 형식으로 반환되는 `get-statement-result` 직접 호출에 대한 응답의 예입니다.

```
aws redshift-data get-statement-result
--id d468d942-6df9-4f85-8ae3-bac01a61aec3
```

```
{
  "Records": [
    [
      {
        "longValue": 1
      },
      {
        "longValue": 2
      },
      {
        "longValue": 3
      }
    ],
  ],
}
```

```
[
  {
    "longValue": 4
  },
  {
    "longValue": 5
  },
  {
    "longValue": 6
  }
],
[
  {
    "longValue": 7
  },
  {
    "longValue": 8
  },
  {
    "longValue": 9
  }
]
],
"ColumnMetadata": [
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "col1",
    "name": "col1",
    "nullable": 1,
    "precision": 10,
    "scale": 0,
    "schemaName": "public",
    "tableName": "testingtable",
    "typeName": "int4",
    "length": 0
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": true,
    "label": "col2",
    "name": "col2",
```

```

        "nullable": 1,
        "precision": 10,
        "scale": 0,
        "schemaName": "public",
        "tableName": "testingtable",
        "typeName": "int4",
        "length": 0
    },
    {
        "isCaseSensitive": false,
        "isCurrency": false,
        "isSigned": true,
        "label": "col3",
        "name": "col3",
        "nullable": 1,
        "precision": 10,
        "scale": 0,
        "schemaName": "public",
        "tableName": "testingtable",
        "typeName": "int4",
        "length": 0
    }
],
"TotalNumRows": 3
}

```

다음 예시에서는 결과를 CSV로 반환하기 위해 `execute-statement`로 실행되는 SQL 문을 보여줍니다. 테이블 `testingtable`에는 3개의 정수 열(`col1`, `col2`, `col3`)이 있고, 값(1, 2, 3), (4, 5, 6), (7, 8, 9)이 있는 3개의 행이 있습니다.

```

aws redshift-data execute-statement
--database dev
--sql "SELECT col1, col2, col3 FROM testingtable"
--cluster-id mycluster-test
--result-format CSV

```

```

{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": "2024-04-02T16:45:25.144000+00:00",
  "Database": "dev",
  "DbUser": "IAMR:Administrator",

```

```
"Id": "d468d942-6df9-4f85-8ae3-bac01a61aec3"
}
```

다음은 SQL 결과가 응답 키 Records의 CSV 형식으로 반환되는 get-statement-result-v2 직접 호출에 대한 응답의 예입니다. 행은 캐리지 리턴과 줄 바꿈(\r\n)으로 구분됩니다. Records에서 반환되는 첫 번째 행은 열 헤더입니다. CSV 형식으로 반환된 결과는 1MB로 반환되며, 각 청크는 최대 1MB까지 행 수를 저장할 수 있습니다.

```
aws redshift-data get-statement-result-v2
  --id d468d942-6df9-4f85-8ae3-bac01a61aec3
```

```
{
  "Records": [
    {
      "CSVRecords": "col1,col2,col3\r\n1,2,3\r\n4,5,6\r\n7,8,9\r\n"
    }
  ],
  "ColumnMetadata": [
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": true,
      "label": "col1",
      "name": "col1",
      "nullable": 1,
      "precision": 10,
      "scale": 0,
      "schemaName": "public",
      "tableName": "testingtable",
      "typeName": "int4",
      "length": 0
    },
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": true,
      "label": "col2",
      "name": "col2",
      "nullable": 1,
      "precision": 10,
      "scale": 0,
```

```

        "schemaName": "public",
        "tableName": "testingtable",
        "typeName": "int4",
        "length": 0
    },
    {
        "isCaseSensitive": false,
        "isCurrency": false,
        "isSigned": true,
        "label": "col3",
        "name": "col3",
        "nullable": 1,
        "precision": 10,
        "scale": 0,
        "schemaName": "public",
        "tableName": "testingtable",
        "typeName": "int4",
        "length": 0
    }
],
"TotalNumRows": 3,
"ResultFormat": "csv"
}

```

테이블 설명

테이블을 설명하는 메타데이터를 가져오려면 `aws redshift-data describe-table` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 클러스터에 대해 SQL 문을 실행하고 테이블을 설명하는 메타데이터를 반환합니다. 이 예에서는 AWS Secrets Manager 인증 방법을 사용합니다.

```

aws redshift-data describe-table
  --cluster-identifier mycluster-test
  --database dev
  --schema information_schema
  --table sql_features
  --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPwn

```

다음은 이 응답의 예입니다.


```
{
  "ColumnList": [
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": false,
      "length": 2147483647,
      "name": "feature_id",
      "nullable": 1,
      "precision": 2147483647,
      "scale": 0,
      "schemaName": "information_schema",
      "tableName": "sql_features",
      "typeName": "character_data"
    },
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": false,
      "length": 2147483647,
      "name": "feature_name",
      "nullable": 1,
      "precision": 2147483647,
      "scale": 0,
      "schemaName": "information_schema",
      "tableName": "sql_features",
      "typeName": "character_data"
    }
  ]
}
```

다음 AWS CLI 명령은 클러스터에 대해 테이블을 설명하는 SQL 문을 실행합니다. 이 예에서는 임시 자격 증명 인증 방법을 사용합니다.

```
aws redshift-data describe-table
  --db-user myuser
  --cluster-identifier mycluster-test
  --database dev
  --schema information_schema
  --table sql_features
```

다음은 이 응답의 예입니다.

```
{
  "ColumnList": [
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": false,
      "length": 2147483647,
      "name": "feature_id",
      "nullable": 1,
      "precision": 2147483647,
      "scale": 0,
      "schemaName": "information_schema",
      "tableName": "sql_features",
      "typeName": "character_data"
    },
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": false,
      "length": 2147483647,
      "name": "feature_name",
      "nullable": 1,
      "precision": 2147483647,
      "scale": 0,
      "schemaName": "information_schema",
      "tableName": "sql_features",
      "typeName": "character_data"
    },
    {
      "isCaseSensitive": false,
      "isCurrency": false,
      "isSigned": false,
      "length": 2147483647,
      "name": "sub_feature_id",
      "nullable": 1,
      "precision": 2147483647,
      "scale": 0,
      "schemaName": "information_schema",
      "tableName": "sql_features",
      "typeName": "character_data"
    },
    {
```

```
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": false,
    "length": 2147483647,
    "name": "sub_feature_name",
    "nullable": 1,
    "precision": 2147483647,
    "scale": 0,
    "schemaName": "information_schema",
    "tableName": "sql_features",
    "typeName": "character_data"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": false,
    "length": 2147483647,
    "name": "is_supported",
    "nullable": 1,
    "precision": 2147483647,
    "scale": 0,
    "schemaName": "information_schema",
    "tableName": "sql_features",
    "typeName": "character_data"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": false,
    "length": 2147483647,
    "name": "is_verified_by",
    "nullable": 1,
    "precision": 2147483647,
    "scale": 0,
    "schemaName": "information_schema",
    "tableName": "sql_features",
    "typeName": "character_data"
  },
  {
    "isCaseSensitive": false,
    "isCurrency": false,
    "isSigned": false,
    "length": 2147483647,
    "name": "comments",
```

```

        "nullable": 1,
        "precision": 2147483647,
        "scale": 0,
        "schemaName": "information_schema",
        "tableName": "sql_features",
        "typeName": "character_data"
    }
]
}

```

클러스터의 데이터베이스 나열

클러스터의 데이터베이스를 나열하려면 `aws redshift-data list-databases` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 클러스터에 대해 SQL 문을 실행하여 데이터베이스를 나열합니다. 이 예에서는 AWS Secrets Manager 인증 방법을 사용합니다.

```

aws redshift-data list-databases

--secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
--cluster-identifier mycluster-test
--database dev

```

다음은 이 응답의 예입니다.

```

{
  "Databases": [
    "dev"
  ]
}

```

다음 AWS CLI 명령은 클러스터에 대해 SQL 문을 실행하여 데이터베이스를 나열합니다. 이 예에서는 임시 자격 증명 인증 방법을 사용합니다.

```

aws redshift-data list-databases
--db-user myuser
--cluster-identifier mycluster-test

```

```
--database dev
```

다음은 이 응답의 예입니다.

```
{
  "Databases": [
    "dev"
  ]
}
```

데이터베이스의 스키마 나열

데이터베이스의 스키마를 나열하려면 `aws redshift-data list-schemas` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 클러스터에 대해 SQL 문을 실행하여 데이터베이스의 스키마를 나열합니다. 이 예에서는 AWS Secrets Manager 인증 방법을 사용합니다.

```
aws redshift-data list-schemas
  --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPwn
  --cluster-identifier mycluster-test
  --database dev
```

다음은 이 응답의 예입니다.

```
{
  "Schemas": [
    "information_schema",
    "pg_catalog",
    "pg_internal",
    "public"
  ]
}
```

다음 AWS CLI 명령은 클러스터에 대해 SQL 문을 실행하여 데이터베이스의 스키마를 나열합니다. 이 예에서는 임시 자격 증명 인증 방법을 사용합니다.

```
aws redshift-data list-schemas
--db-user mysuser
--cluster-identifier mycluster-test
--database dev
```

다음은 이 응답의 예입니다.

```
{
  "Schemas": [
    "information_schema",
    "pg_catalog",
    "pg_internal",
    "public"
  ]
}
```

데이터베이스의 테이블 나열

데이터베이스의 테이블을 나열하려면 `aws redshift-data list-tables` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 클러스터에 대해 SQL 문을 실행하여 데이터베이스의 테이블을 나열합니다. 이 예에서는 AWS Secrets Manager 인증 방법을 사용합니다.

```
aws redshift-data list-tables
--secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-hKgPwn
--cluster-identifier mycluster-test
--database dev
--schema information_schema
```

다음은 이 응답의 예입니다.

```
{
  "Tables": [
    {
      "name": "sql_features",
      "schema": "information_schema",
      "type": "SYSTEM TABLE"
    }
  ]
}
```

```

    },
    {
      "name": "sql_implementation_info",
      "schema": "information_schema",
      "type": "SYSTEM TABLE"
    }
  ]
}
```

다음 AWS CLI 명령은 클러스터에 대해 SQL 문을 실행하여 데이터베이스의 테이블을 나열합니다. 이 예에서는 임시 자격 증명 인증 방법을 사용합니다.

```

aws redshift-data list-tables

--db-user myuser
--cluster-identifier mycluster-test
--database dev
--schema information_schema
```

다음은 이 응답의 예입니다.

```

{
  "Tables": [
    {
      "name": "sql_features",
      "schema": "information_schema",
      "type": "SYSTEM TABLE"
    },
    {
      "name": "sql_implementation_info",
      "schema": "information_schema",
      "type": "SYSTEM TABLE"
    }
  ]
}
```

Amazon Redshift Data API 문제 해결

공통 오류 메시지를 소개하는 다음 섹션부터는 데이터 API를 사용하면서 발생하는 문제를 해결하는 데 유용합니다.

주제

- [쿼리 패킷이 너무 큼니다](#)
- [데이터베이스 응답이 크기 제한을 초과했습니다](#)

쿼리 패킷이 너무 큼니다

쿼리에 대한 패킷이 너무 크다는 오류가 표시되면 일반적으로 행에 대해 반환된 결과 집합이 너무 큼니다. 데이터 API는 데이터베이스에서 반환되는 결과 집합에서 각 행의 크기를 64KB로 제한합니다.

이 문제를 해결하려면 결과 집합의 각 행마다 크기를 64KB 이하로 유지해야 합니다.

데이터베이스 응답이 크기 제한을 초과했습니다

데이터베이스 응답이 크기 제한을 초과했음을 나타내는 오류가 표시되면 일반적으로 데이터베이스에서 반환된 결과 집합의 크기가 너무 큼니다. Data API는 데이터베이스에서 반환되는 결과 집합의 크기를 500MB로 제한합니다.

이러한 문제를 해결하려면 Data API 직접 호출 시 반환되는 데이터 크기를 500MB 이하로 유지해야 합니다. 반환해야 하는 결과 집합의 크기가 500MB를 넘으면 쿼리에서 LIMIT 절을 사용하여 여러 문의 직접 호출을 실행할 수 있습니다.

Amazon EventBridge로 Amazon Redshift Data API 작업 예약

선택된 이벤트와 일치할 경우 대상으로 라우팅하여 작업을 실행하는 규칙을 생성할 수 있습니다. 규칙을 사용하여 미리 결정된 일정에 따라 조치를 취할 수도 있습니다. 자세한 내용은 [Amazon EventBridge 사용 가이드](#)를 참조하세요.

EventBridge로 Data API 작업을 예약하려면 연결된 IAM 역할이 CloudWatch Events(events.amazonaws.com)의 보안 주체를 신뢰해야 합니다. 이 역할에는 관리형 정책 AmazonEventBridgeFullAccess에 해당하는 정책이 연결되어 있어야 합니다. Data API에 의해 관리되는 AmazonRedshiftDataFullAccess 정책 권한도 있어야 합니다. IAM 콘솔에서 이러한 권한으로 IAM 역할을 생성할 수 있습니다. IAM 콘솔에서 역할을 생성할 때 CloudWatch Events에 대해 AWS 서비스 신뢰할 수 있는 엔터티를 선택합니다. EventBridge 대상의 RoleArn JSON 값에 IAM 역할을 지정합니다. 서비스 역할 생성에 대한 자세한 내용은 IAM User Guide의 [Creating a Role for an AWS Service \(Console\)](#)를 참조하세요.

Amazon EventBridge에서 생성하는 규칙의 name은 RedshiftDataParameters의 StatementName과 일치해야 합니다.

다음 예제는 한 개 또는 여러 개의 SQL 문을 사용하고 Amazon Redshift 클러스터 또는 Amazon Redshift Serverless 작업 그룹을 데이터 웨어하우스로 사용하여 EventBridge 규칙을 생성하는 변형을 보여줍니다.

단일 SQL 문 및 클러스터를 사용한 호출

다음 예에서는 AWS CLI를 사용하여 Amazon Redshift 클러스터에 대해 SQL 문을 실행하는 데 사용되는 EventBridge 규칙을 생성합니다.

```
aws events put-rule
--name test-redshift-cluster-data
--schedule-expression "rate(1 minute)"
```

그런 다음 규칙에 지정된 일정에 따라 실행되도록 EventBridge 대상이 생성됩니다.

```
aws events put-targets
--cli-input-json file://data.json
```

입력 data.json 파일은 다음과 같습니다. Sql JSON 키는 단일 SQL 문이 있음을 나타냅니다. Arn JSON 값에는 클러스터 식별자가 포함됩니다. RoleArn JSON 값에는 앞서 설명한 대로 SQL을 실행하는 데 사용되는 IAM 역할이 포함됩니다.

```
{
  "Rule": "test-redshift-cluster-data",
  "EventBusName": "default",
  "Targets": [
    {
      "Id": "2",
      "Arn": "arn:aws:redshift:us-east-1:123456789012:cluster:mycluster",
      "RoleArn": "arn:aws:iam::123456789012:role/Administrator",
      "RedshiftDataParameters": {
        "Database": "dev",
        "DbUser": "root",
        "Sql": "select 1;",
        "StatementName": "test-redshift-cluster-data",
        "WithEvent": true
      }
    }
  ]
}
```

단일 SQL 문 및 작업 그룹을 사용한 호출

다음 예에서는 AWS CLI를 사용하여 Amazon Redshift Serverless 작업 그룹에 대해 SQL 문을 실행하는 데 사용되는 EventBridge 규칙을 생성합니다.

```
aws events put-rule
--name test-redshift-serverless-workgroup-data
--schedule-expression "rate(1 minute)"
```

그런 다음 규칙에 지정된 일정에 따라 실행되도록 EventBridge 대상이 생성됩니다.

```
aws events put-targets
--cli-input-json file://data.json
```

입력 data.json 파일은 다음과 같습니다. Sql JSON 키는 단일 SQL 문이 있음을 나타냅니다. Arn JSON 값에는 작업 그룹 이름이 포함됩니다. RoleArn JSON 값에는 앞서 설명한 대로 SQL을 실행하는 데 사용되는 IAM 역할이 포함됩니다.

```
{
  "Rule": "test-redshift-serverless-workgroup-data",
  "EventBusName": "default",
  "Targets": [
    {
      "Id": "2",
      "Arn": "arn:aws:redshift-serverless:us-east-1:123456789012:workgroup/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "RoleArn": "arn:aws:iam::123456789012:role/Administrator",
      "RedshiftDataParameters": {
        "Database": "dev",
        "Sql": "select 1;",
        "StatementName": "test-redshift-serverless-workgroup-data",
        "WithEvent": true
      }
    }
  ]
}
```

여러 SQL 문 및 클러스터를 사용한 호출

다음 예에서는 AWS CLI를 사용하여 Amazon Redshift 클러스터에 대해 여러 SQL 문을 실행하는 데 사용되는 EventBridge 규칙을 생성합니다.

```
aws events put-rule
--name test-redshift-cluster-data
--schedule-expression "rate(1 minute)"
```

그런 다음 규칙에 지정된 일정에 따라 실행되도록 EventBridge 대상이 생성됩니다.

```
aws events put-targets
--cli-input-json file:///data.json
```

입력 data.json 파일은 다음과 같습니다. Sqls JSON 키는 여러 SQL 문이 있음을 나타냅니다. Arn JSON 값에는 클러스터 식별자가 포함됩니다. RoleArn JSON 값에는 앞서 설명한 대로 SQL을 실행하는 데 사용되는 IAM 역할이 포함됩니다.

```
{
  "Rule": "test-redshift-cluster-data",
  "EventBusName": "default",
  "Targets": [
    {
      "Id": "2",
      "Arn": "arn:aws:redshift:us-east-1:123456789012:cluster:mycluster",
      "RoleArn": "arn:aws:iam::123456789012:role/Administrator",
      "RedshiftDataParameters": {
        "Database": "dev",
        "Sqls": ["select 1;", "select 2;", "select 3;"],
        "StatementName": "test-redshift-cluster-data",
        "WithEvent": true
      }
    }
  ]
}
```

여러 SQL 문 및 작업 그룹을 사용한 호출

다음 예에서는 AWS CLI를 사용하여 Amazon Redshift Serverless 작업 그룹에 대해 여러 SQL 문을 실행하는 데 사용되는 EventBridge 규칙을 생성합니다.

```
aws events put-rule
--name test-redshift-serverless-workgroup-data
--schedule-expression "rate(1 minute)"
```

그런 다음 규칙에 지정된 일정에 따라 실행되도록 EventBridge 대상이 생성됩니다.

```
aws events put-targets
--cli-input-json file://data.json
```

입력 data.json 파일은 다음과 같습니다. Sqls JSON 키는 여러 SQL 문이 있음을 나타냅니다. Arn JSON 값에는 작업 그룹 이름이 포함됩니다. RoleArn JSON 값에는 앞서 설명한 대로 SQL을 실행하는 데 사용되는 IAM 역할이 포함됩니다.

```
{
  "Rule": "test-redshift-serverless-workgroup-data",
  "EventBusName": "default",
  "Targets": [
    {
      "Id": "2",
      "Arn": "arn:aws:redshift-serverless:us-east-1:123456789012:workgroup/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "RoleArn": "arn:aws:iam::123456789012:role/Administrator",
      "RedshiftDataParameters": {
        "Database": "dev",
        "Sqls": ["select 1;", "select 2;", "select 3;"],
        "StatementName": "test-redshift-serverless-workgroup-data",
        "WithEvent": true
      }
    }
  ]
}
```

Data API 모니터링

Data API 및 다른 AWS 솔루션의 안정성, 가용성 및 성능을 유지하려면 모니터링이 중요합니다. AWS는 Data API를 모니터링하고, 이상이 있을 때 이를 보고하고, 필요한 경우 자동 조치를 취할 수 있도록 다음과 같은 모니터링 도구를 제공합니다.

- Amazon EventBridge를 사용하면 AWS 서비스를 자동화하고 애플리케이션 가용성 문제나 리소스 변경 같은 시스템 이벤트에 자동으로 대응할 수 있습니다. AWS 서비스의 이벤트는 거의 실시간으로 EventBridge로 전송됩니다. 원하는 이벤트만 표시하도록 간단한 규칙을 작성한 후 규칙과 일치하는 이벤트 발생 시 실행할 자동화 작업을 지정할 수 있습니다. 자세한 내용은 [Amazon EventBridge 사용 설명서](#)를 참조하세요.
- AWS CloudTrail은 직접 수행하거나 AWS 계정을 대신하여 수행한 API 호출 및 관련 이벤트를 캡처하고 지정한 Amazon S3 버킷에 로그 파일을 전송합니다. 어떤 사용자 및 계정이 AWS를 직접적으로 호출했는지, 어떤 소스 IP 주소에 직접 호출이 이루어졌는지, 언제 직접 호출이 발생했는지

확인할 수 있습니다. Amazon Redshift를 AWS CloudTrail에 통합하는 방법을 자세히 알아보려면 [CloudTrail을 사용한 로깅](#)을 참조하세요. CloudTrail에 대한 자세한 내용은 [AWS CloudTrail 사용자 안내서](#)를 참조하세요.

주제

- [Amazon EventBridge에서 Amazon Redshift Data API에 대한 이벤트 모니터링](#)

Amazon EventBridge에서 Amazon Redshift Data API에 대한 이벤트 모니터링

자체 애플리케이션, SaaS(Software-as-a-Service) 애플리케이션 및 AWS 서비스의 실시간 데이터 스트림을 제공하는 EventBridge의 Data API 이벤트를 모니터링할 수 있습니다. EventBridge는 해당 데이터를 AWS Lambda, Amazon SNS 등의 대상으로 라우팅합니다. 이러한 이벤트는 CloudWatch Events에 나타나는 이벤트와 동일하며, AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. Amazon Redshift 데이터베이스가 포함된 계정으로 이벤트가 전송됩니다. 예를 들어 다른 계정의 역할을 수임하면 이벤트가 해당 계정으로 전송됩니다. 자세한 내용은 Amazon EventBridge 사용 설명서의 [Amazon EventBridge 이벤트](#)를 참조하세요.

ExecuteStatement 또는 BatchExecuteStatement API 작업이 WithEvents 옵션을 true로 설정하면 Data API 이벤트가 전송됩니다. 이벤트의 state 필드에는 다음 값 중 하나가 포함됩니다.

- ABORTED – 쿼리 실행이 사용자에게 의해 중지되었습니다.
- [실패(FAILED)] – 쿼리 실행에 실패했습니다.
- [완료(FINISHED)] – 쿼리 실행이 완료되었습니다.

이벤트는 보장된 방식으로 전달됩니다. 자세한 내용은 Amazon EventBridge 사용 설명서의 [AWS 서비스 이벤트](#)를 참조하세요.

Data API 완료 이벤트의 예

다음 예에서는 ExecuteStatement API 작업이 완료될 때 Data API에 대한 이벤트를 보여줍니다. 이 예에서 test.testtable이라는 문이 실행을 완료했습니다.

```
{
  "version": "0",
  "id": "18e7079c-dd4b-dd64-caf9-e2a31640dab0",
  "detail-type": "Redshift Data Statement Status Change",
  "source": "aws.redshift-data",
  "account": "123456789012",
```

```

"time": "2020-10-01T21:14:26Z",
"region": "us-east-1",
"resources": [
    "arn:aws:redshift:us-east-1:123456789012:cluster:redshift-cluster-1"
],
"detail": {
    "principal": "arn:aws:iam::123456789012:user/myuser",
    "statementName": "test.testtable",
    "statementId": "dd2e1ec9-2ee3-49a0-819f-905fa7d75a4a",
    "redshiftQueryId": -1,
    "state": "FINISHED",
    "rows": 1,
    "expireAt": 1601673265
}
}

```

Amazon Redshift 데이터 API와 함께 AWS KMS 사용

Amazon Redshift 클러스터 또는 Redshift Serverless 작업 그룹을 고객 관리형 키로 암호화하는 경우 Amazon Redshift 데이터 API는 동일한 고객 관리형 키를 사용하여 쿼리와 결과를 저장하고 암호화합니다.

데이터 API는 기본적으로 데이터를 암호화하여 쿼리 텍스트 및 쿼리 결과와 같은 민감한 정보를 보호합니다. 이 보호를 위해 AWS에서 소유한 AWS KMS 암호화 키를 사용합니다.

저장 데이터의 기본 암호화는 민감한 데이터를 보호하는 데 수반되는 운영 오버헤드와 복잡성을 줄여 줍니다. 이 방법으로 엄격한 암호화 규정 준수 및 규제 요구 사항을 충족하는 안전한 애플리케이션을 구축할 수 있습니다.

AWS KMS에서 권한 부여 사용

데이터 API는 고객 관리형 키를 사용하려면 권한 부여가 필요합니다.

고객 관리형 키로 암호화된 클러스터에 대해 `ExecuteStatement` 또는 `BatchExecuteStatement`를 직접 호출하면 Amazon Redshift는 AWS KMS에 [CreateGrant](#) 요청을 전송하여 사용자를 대신하여 권한 부여를 생성합니다. AWS KMS는 권한 부여를 사용하여 계정의 KMS 키에 대한 데이터 API 액세스 권한을 부여합니다.

데이터 API는 다음 작업에 대해 고객 관리형 키를 사용할 수 있는 권한이 필요합니다.

- AWS KMS에 [Encrypt](#) 요청을 보내 고객 관리형 키로 쿼리 메타데이터를 암호화합니다.

- AWS KMS에 [GenerateDataKey](#) 요청을 보내 고객 관리 키로 암호화된 데이터 키를 생성합니다.
- 암호화된 데이터 키를 암호화할 수 있도록 암호화된 데이터 키의 암호를 해독하기 위한 [Decrypt](#) 요청을 AWS KMS에 보냅니다.

언제든지 권한 부여에 대한 액세스 권한을 취소하거나 고객 관리형 키에 대한 Amazon Redshift 액세스 권한을 제거할 수 있습니다. 그렇게 할 경우 API는 고객이 관리하는 키로 암호화된 데이터에 더 이상 접근할 수 없게 되어 해당 데이터에 의존하는 작업에 영향을 미칩니다. 예를 들어 권한 부여를 취소한 후 쿼리 결과를 검색하거나 쿼리 상태를 추적하려고 하면 데이터 API가 `AccessDeniedException`을 반환합니다.

고객 관리형 키에 대한 키 정책

키 정책은 고객 관리형 키에 대한 액세스를 제어합니다. 모든 고객 관리형 키에는 키를 사용할 수 있는 사람과 키를 사용하는 방법을 결정하는 문장이 포함된 정확히 하나의 키 정책이 있어야 합니다. 고객 관리형 키를 만들 때 키 정책을 지정할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 [고객 관리형 키](#)를 참조하세요.

데이터 API에서 고객 관리형 키를 사용하려면 먼저 Amazon Redshift에 대한 액세스를 허용해야 합니다. 키 정책에서 다음 API 작업을 허용해야 합니다.

- `kms:CreateGrant` – 고객 관리형 키에 권한 부여를 추가합니다. 권한 부여는 지정된 AWS KMS 키에 대한 액세스를 제어하여 Amazon Redshift에 필요한 권한 부여 작업에 대한 액세스를 허용합니다. 자세한 내용은 [AWS KMS에서 권한 부여 사용](#)을 참조하세요.

다음은 예제 키 정책입니다.

```
"Statement":[
  {
    "Sid":"Allow access to principals authorized to use Amazon Redshift",
    "Effect":"Allow",
    "Principal":{"
      "AWS":"*"
    },
    "Action":[
      "kms:DescribeKey",
      "kms:CreateGrant"
    ],
    "Resource":"*",
    "Condition":{"
```

```

    "StringEquals":{
      "kms:ViaService":"redshift.amazonaws.com",
      "kms:CallerAccount":"111122223333"
    }
  },
  {
    "Sid":"AllowKeyAdministratorsAccess",
    "Effect":"Allow",
    "Principal":{
      "AWS":"arn:aws:iam::111122223333:role/ExampleAdminRole"
    },
    "Action":"kms:*",
    "Resource":"*"
  },
  {
    "Sid":"AllowKeyUseForExampleRole",
    "Effect":"Allow",
    "Principal":{
      "AWS":"arn:aws:iam::111122223333:role/ExampleUserRole"
    },
    "Action":[
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource":"*"
  }
]

```

데이터 API 암호화 컨텍스트

암호화 컨텍스트는 데이터에 대한 추가 컨텍스트 정보를 포함하는 키-값 페어의 선택적 집합입니다. AWS KMS는 암호화 컨텍스트를 인증된 암호화를 지원하기 위한 추가 인증 데이터로 사용합니다. 데이터 암호화 요청에 암호화 컨텍스트를 포함하는 경우, AWS KMS는 암호화된 데이터에 암호화 컨텍스트를 바인딩합니다. 이 데이터를 해독하려면 요청에 동일한 암호화 컨텍스트를 포함해야 합니다.

데이터 API는 프로비저닝된 클러스터의 모든 AWS KMS 암호화 작업에서 동일한 세 개의 암호화 컨텍스트 키-값 페어를 사용합니다.

- `aws:redshift:arn` - 클러스터의 Amazon 리소스 이름(ARN)

- `aws:redshift:createtime` - 클러스터 생성을 요청한 시점의 타임스탬프
- `serviceName` – RedshiftDataAPI

```
"EncryptionContextSubset": {
  "aws:redshift:arn": "arn:aws:redshift:us-east-1:123456789012:cluster:redshift-
cluster",
  "aws:redshift:createtime": "20250815T0000Z",
  "serviceName": "RedshiftDataAPI",
}
```

데이터 API는 서버리스 작업 그룹의 모든 AWS KMS 암호화 작업에 두 개의 암호화 컨텍스트 키-값 페어를 사용합니다.

- `aws:redshift-serverless:arn` - 네임스페이스의 Amazon 리소스 이름(ARN)
- `serviceName` – RedshiftDataAPI

```
"EncryptionContextSubset": {
  "aws:redshift-serverless:arn": "arn:aws:redshift-serverless:us-
east-1:123456789012:namespace:12345678-1234-1234-1234-123456789012",
  "serviceName": "RedshiftDataAPI"
}
```

암호화에 대한 자세한 내용은 [AWS KMS의 암호화 세부 정보 소개](#)를 참조하세요. Amazon Redshift 및 AWS KMS 통합에 대한 자세한 내용은 [Amazon Redshift가 AWS KMS를 사용하는 방법](#)을 참조하세요.

Amazon SageMaker Unified Studio를 사용하여 Amazon Redshift 및 SageMaker Lakehouse의 데이터베이스 쿼리

Amazon SageMaker Unified Studio는 콘솔 외부 개발 환경을 제공하며 SageMaker Lakehouse, Amazon Redshift 및 Amazon Athena의 데이터에 대한 SQL 분석을 지원합니다. 관리자의 URL을 사용하여 Amazon SageMaker Unified Studio로 이동하고 SSO 또는 AWS 자격 증명을 사용하여 로그인합니다. 첫 번째 프로젝트를 설정하는 방법에 대한 자세한 내용은 Amazon SageMaker Unified Studio 사용자 가이드의 [시작하기](#)를 참조하세요.

Amazon SageMaker Unified Studio에서 [쿼리 편집기](#)로 Amazon Redshift 및 Amazon Athena를 실행하여 [SQL 분석](#)을 수행할 수 있습니다. 쿼리 편집기를 사용하여 쿼리를 작성 및 실행하고, 결과를 보고, 팀

과 작업을 공유할 수 있습니다. AWS 계정(동일 계정 및 다른 AWS 계정)의 Redshift 데이터 웨어하우스에 대해 쿼리를 실행하고, 동일한 인터페이스를 사용하여 Redshift와 Athena에 대한 SQL 쿼리를 작성하고, Amazon Managed Workflows for Apache Airflow를 사용하여 SQL 쿼리를 예약할 수 있습니다. Amazon Q 생성형 SQL을 사용하여 자연어에서 SQL을 생성할 수도 있습니다.

Amazon Redshift 파라미터 그룹

Amazon Redshift에서는 생성된 클러스터마다 파라미터 그룹을 연결합니다. 여기에서 파라미터 그룹이란 클러스터에서 생성하는 모든 데이터베이스에 적용되는 파라미터 모음을 말합니다. 이러한 파라미터가 쿼리 제한 시간이나 날짜 스타일 같은 데이터베이스 설정을 구성합니다. 클러스터를 시작할 때 파라미터 그룹과 연결해야 합니다. 나중에 파라미터 그룹을 변경하고 싶다면 클러스터를 수정하여 다른 파라미터 그룹을 선택할 수 있습니다.

파라미터 그룹은 각각 데이터베이스 설정을 구성하는 몇 가지 파라미터로 이루어집니다. 사용 가능한 파라미터 목록은 파라미터 그룹이 속하는 파라미터 그룹 패밀리에 따라 달라집니다. 기본 파라미터 그룹 패밀리는 `redshift-2.0`입니다.

Amazon Redshift는 파라미터 그룹 패밀리마다 기본 파라미터 그룹을 하나씩 제공합니다. 기본 파라미터 그룹은 각 파라미터마다 사전 설정된 값이 있으며, 이 값을 수정할 수도 없습니다. 기본 파라미터 그룹 이름의 형식은 `default.parameter_group_family`입니다. 예를 들어 `redshift-2.0` 파라미터 그룹 패밀리의 기본 파라미터 그룹은 `default.redshift-2.0`입니다.

기본 파라미터 그룹 외에 다른 파라미터 값을 사용하려면 사용자 지정 파라미터 그룹을 생성하여 클러스터와 연결해야 합니다. 처음에는 사용자 지정 파라미터 그룹의 파라미터 값이 기본 파라미터 그룹의 값과 동일합니다. 파라미터 값은 Amazon Redshift에서 사전 설정되기 때문에 모든 파라미터의 초기 `source`는 `engine-default`입니다. 이후 파라미터 값을 변경하면 `source`가 `user`로 바뀌면서 파라미터 값이 기본 값에서 수정되었다는 것을 나타냅니다.

Note

Amazon Redshift 콘솔에는 각 파라미터의 `source`가 표시되지 않습니다. `source`를 보려면 Amazon Redshift API, AWS CLI 또는 AWS SDK 중 하나를 사용해야 합니다.

사용자 지정 파라미터에서는 언제든지 파라미터 값을 수정하거나, 혹은 모든 파라미터 값을 기본값으로 재설정할 수 있습니다. 또한 다른 파라미터 그룹을 클러스터와 연결할 수도 있습니다. 경우에 따라 클러스터와 이미 연결된 파라미터 그룹의 파라미터 값을 수정하거나 다른 파라미터 그룹을 클러스터와 연결할 수 있습니다. 이러한 경우 업데이트된 파라미터 값이 적용되려면 클러스터를 다시 시작해야 할 수 있습니다. 클러스터가 실패하여 Amazon Redshift에서 다시 시작되면 변경 사항이 함께 적용됩니다. 유지 관리 중 클러스터가 다시 시작되면 변경 사항이 적용되지 않습니다. 자세한 내용은 [WLM 동적 및 정적 속성](#) 섹션을 참조하세요.

기본 파라미터 값

Note

2025년 1월 10일부터 `require_ssl` 파라미터의 기본값은 `true`입니다. 클러스터에 SSL이 필요하지 않게 하려면 클러스터를 생성할 때 사용자 지정 파라미터 그룹을 사용하거나 클러스터를 기본값으로 생성한 후 사용자 지정 파라미터 그룹과 연결하도록 클러스터를 수정할 수 있습니다.

다음 표는 각 파라미터에 대한 자세한 정보 링크와 함께 기본 파라미터 값을 한 눈에 볼 수 있도록 나타낸 것입니다. 나열된 값은 `redshift-2.0` 파라미터 그룹 패밀리의 기본값입니다.

파라미터 이름	값	추가 정보
<code>auto_analyze</code>	<code>true</code>	Amazon Redshift 데이터베이스 개발자 안내서의 auto_analyze
<code>auto_mv</code>	<code>true</code>	Amazon Redshift 데이터베이스 개발자 안내서의 자동화된 구체화된 뷰
<code>datestyle</code>	ISO, MDY	Amazon Redshift 데이터베이스 개발자 안내서의 datestyle
<code>enable_case_sensitive_identifier</code>	<code>false</code>	Amazon Redshift 데이터베이스 개발자 안내서의 enable_case_sensitive_identifier
<code>enable_user_activity_logging</code>	<code>false</code>	데이터베이스 감사 로깅 이 가이드의
<code>extra_float_digits</code>	0	Amazon Redshift 데이터베이스 개발자 안내서의 extra_float_digits
<code>max_concurrency_scaling_clusters</code>	1	Amazon Redshift 데이터베이스 개발자 안내서의 max_concurrency_scaling_clusters
<code>query_group</code>	기본값	Amazon Redshift 데이터베이스 개발자 안내서의 query_group

파라미터 이름	값	추가 정보
require_ssl	true	연결을 위한 보안 옵션 구성 이 가이드의
search_path	\$user, public	Amazon Redshift 데이터베이스 개발자 안내서의 search_path
statement_timeout	0	Amazon Redshift 데이터베이스 개발자 안내서의 statement_timeout
wlm_json_configuration	[{"auto_wlm":true}]	워크로드 관리 이 가이드의
use_fips_ssl	false	시스템에서 FIPS를 준수해야 하는 경우에만 FIPS 준수 SSL 모드를 활성화합니다.

Note

max_cursor_result_set_size 파라미터는 이제 사용되지 않습니다. 커서 결과 집합 크기
에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [커서 제약 조건](#)을 참조하세요.

파라미터는 데이터베이스에서 SET 명령을 사용하여 일시적으로 재정의할 수 있습니다. SET 명령이 현재 세션에 한해 파라미터를 재정의하기 때문입니다. 위 표에서 언급한 파라미터 외에도 데이터베이스에서 wlm_query_slot_count를 설정하면 슬롯 수를 임시로 조정할 수 있습니다. 파라미터 그룹을 구성할 때는 wlm_query_slot_count 파라미터가 제공되지 않습니다. 슬롯 수 조정에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [wlm_query_slot_count](#)를 참조하세요. 일시적으로 다른 파라미터 재정의에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [서버 구성 수정](#)을 참조하세요.

워크로드 관리

Amazon Redshift에서는 워크로드 관리(WLM)를 통해 사용 가능한 쿼리 대기열의 수와 쿼리가 처리를 위해 대기열로 라우팅되는 방식을 정의합니다. WLM은 파라미터 그룹 구성에 포함됩니다. 클러스터는 연결된 파라미터 그룹에서 지정하는 WLM 구성을 사용합니다.

파라미터 그룹을 생성할 때는 동시에 5개까지 쿼리를 실행할 수 있는 대기열 1개가 기본 WLM 구성에 포함됩니다. 여기에 대기열을 추가할 수 있으며, 쿼리 처리에 대한 통제 권한을 높이려면 각 대기열마다 WLM 속성을 구성하면 됩니다. 추가하는 대기열에는 속성을 구성할 때까지 마찬가지로 기본 WLM 구성이 적용됩니다.

대기열을 추가할 때는 마지막 대기열이 기본 대기열이 됩니다. WLM 구성 기준에 따라 다른 대기열로 라우팅되지 않는 쿼리는 기본 대기열에서 처리됩니다. 기본 대기열에 모드 및 동시성 레벨(쿼리 슬롯)을 지정할 수 있지만 기본 대기열에 사용자 그룹 또는 쿼리 그룹을 지정할 수는 없습니다.

다른 파라미터에서도 그렇지만 기본 파라미터 그룹의 WLM 구성은 수정할 수 없습니다. 기본 파라미터 그룹과 연결된 클러스터는 항상 기본 WLM 구성을 사용합니다. WLM 구성을 수정하려면 새 파라미터 그룹을 생성한 후 사용자 지정 WLM 구성이 필요한 모든 클러스터와 생성한 파라미터 그룹을 연결합니다.

WLM 동적 및 정적 속성

WLM 구성 속성은 동적이거나 정적입니다. 동적 속성은 클러스터 재부팅 없이도 데이터베이스에 적용할 수 있지만 정적 속성의 경우 변경 사항이 적용되려면 클러스터를 재부팅해야 합니다. 정적 및 동적 속성에 대한 자세한 내용은 [WLM 동적 및 정적 구성 속성](#)을 참조하십시오.

WLM 구성 파라미터의 속성

Amazon Redshift 콘솔, AWS CLI, Amazon Redshift API 또는 AWS SDK 중 하나를 사용하여 WLM을 구성할 수 있습니다. WLM 구성은 몇 가지 속성을 사용하여 대기열에 대한 메모리 할당, 대기열에서 동시에 실행 가능한 쿼리 수 등 대기열 동작을 정의합니다.

Note

다음 속성은 해당 Amazon Redshift 콘솔 이름과 함께 표시되며 해당 설명서 JSON 속성 이름이 표시됩니다.

다음 표는 속성이 자동 WLM에 적용되는지, 아니면 수동 WML에 적용되는지 여부를 요약한 것입니다.

WLM 속성	자동 WLM	수동 WLM
자동 WLM	예	예
단기 쿼리 가속화 활성화	예	예

WLM 속성	자동 WLM	수동 WLM
단기 쿼리의 최대 실행 시간	예	예
우선순위	예	아니요
대기열 유형	예	예
대기열 이름	예	예
동시성 확장 모드	예	예
동시성	아니요	예
사용자 그룹	예	예
사용자 그룹 와일드카드	예	예
쿼리 그룹	예	예
쿼리 그룹 와일드카드	예	예
사용자 역할	예	예
사용자 역할 와일드카드	예	예
제한 시간	아니요	사용되지 않음
Memory	아니요	예
쿼리 모니터링 규칙	예	예

다음은 구성할 수 있는 WLM 속성을 설명한 것입니다.

자동 WLM

Auto WLM(자동 WLM)을 true로 설정하면 자동 WLM이 활성화됩니다. 자동 WLM은 [기본 클러스터의 동시성(Concurrency on main)]과 [메모리(%) (Memory (%))의 값을 Auto로 설정합니다. Amazon Redshift는 쿼리 동시성과 메모리 할당을 관리합니다. 기본값은 true입니다.

JSON 속성: `auto_wlm`

단기 쿼리 가속화 활성화

단기 쿼리 가속화(SQA)는 선택한 단기 실행 쿼리를 장기 실행 쿼리보다 우선적으로 적용합니다. SQA 쿼리가 대기열에서 장기 쿼리 뒤에서 대기해야 하지 않도록 SQA는 전용 공간에서 단기 실행 쿼리를 실행합니다. SQA가 있으면 단기 실행 쿼리가 더 빠르게 실행하기 시작하며 사용자가 더 빨리 결과를 확인합니다. SQA를 활성화하면 단기 쿼리의 최대 실행 시간도 지정할 수 있습니다. SQA를 활성화하려면 `true`를 지정합니다. 기본값은 `false`입니다. 대기열이 아닌 각 파라미터 그룹에 적용되는 설정입니다.

JSON 속성: `short_query_queue`

단기 쿼리의 최대 실행 시간

SQA를 활성화하면 0을 지정하여 WLM에서 단기 쿼리의 최대 실행 시간을 동적으로 설정하게 할 수 있습니다. 또는 1~20초의 값을 밀리초 단위로 지정할 수 있습니다. 기본값은 0입니다.

JSON 속성: `max_execution_time`

우선순위

Priority를 통해 대기열에서 실행되는 쿼리의 우선 순위를 설정할 수 있습니다. 우선 순위를 설정하려면 WLM mode(WLM 모드)를 Auto WLM(자동 WLM)으로 설정해야 합니다. 즉 `auto_wlm`이 `true`이어야 합니다. 우선 순위 값은 `highest`, `high`, `normal`, `low` 및 `lowest`일 수 있습니다. 기본값은 `normal`입니다.

JSON 속성: `priority`

Queue type(대기열 유형)

Queue type(대기열 유형)에서는 Auto WLM(자동 WLM) 또는 Manual WLM(수동 WLM)으로 사용되는 대기열을 지정합니다. `queue_type`을 `auto` 또는 `manual`로 설정합니다. 지정하지 않은 경우 기본값은 `manual`입니다.

JSON 속성: `queue_type`

대기열 이름

대기열의 이름입니다. 비즈니스 요구에 따라 대기열 이름을 설정할 수 있습니다. 대기열 이름은 WLM 구성 내에서 고유해야 하고 최대 64자의 영숫자, 밑줄 또는 공백으로 구성되어야 하며 따옴표 포함할 수 없습니다. 예를 들어, ETL 쿼리에 대한 대기열이 있는 경우 이름을 `ETL queue`로 지정할 수 있습니다. 이 이름은 지표, 시스템 테이블 값 및 Amazon Redshift 콘솔에서 대기열을 식별하는 데 사용됩니다. 이러한 소스의 이름을 사용하는 쿼리 및 보고서는 이름 변경을 처리할 수 있어야 합니다. 이전에는 Amazon Redshift에서 대기열 이름을 생성했습니다. 대기열의 기본 이름은 `Queue 1`, `Queue 2` 등으로 지정되고 마지막 대기열은 `Default queue`입니다.

⚠ Important

대기열 이름을 변경하면 WLM 대기열 지표(예: WLMQueueLength, WLMQueueWaitTime, WLMQueriesCompletedPerSecond, WLMQueryDuration, WLMRunningQueries 등)의 QueueName 차원 값도 변경됩니다. 따라서 대기열의 이름을 변경하는 경우 설정한 CloudWatch 경보를 변경해야 할 수 있습니다.

JSON 속성: name

동시성 확장 모드

대기열에서 동시성 확장을 활성화하려면 Concurrency Scaling mode(동시성 확장 모드)를 auto로 설정합니다. 대기열로 라우팅된 쿼리의 수가 대기열의 구성된 동시성을 초과하면 적합한 쿼리가 확장 클러스터로 전송됩니다. 슬롯을 사용할 수 있게 되면 쿼리가 기본 클러스터에서 실행됩니다. 기본값은 off입니다.

JSON 속성: concurrency_scaling

동시성

수동 WLM 대기열에서 동시에 실행할 수 있는 쿼리의 수입니다. 이 속성은 수동 WLM에만 적용됩니다. 동시성 확장이 활성화하면 대기열이 동시성 레벨(쿼리 슬롯)에 도달할 때 적합한 쿼리가 확장 클러스터로 전송됩니다. 동시성 확장이 비활성화되면 슬롯을 사용할 수 있게 될 때까지 쿼리가 대기열에서 대기합니다. 범위는 1~50입니다.

JSON 속성: query_concurrency

사용자 그룹

사용자 그룹 이름이 쉼표로 구분된 목록입니다. 사용자 그룹 멤버가 데이터베이스에서 쿼리를 실행하면 사용자 그룹과 연결되어 있는 대기열로 쿼리가 라우팅됩니다.

JSON 속성: user_group

사용자 그룹 와일드카드

사용자 그룹의 와일드카드 활성화 여부를 나타내는 부울 값입니다. 값이 0이면 와일드카드가 비활성화되고, 값이 1이면 와일드카드가 활성화됩니다. 와일드카드가 활성화되면 쿼리를 실행할 때 "*" 또는 "?"를 사용하여 다수의 사용자 그룹을 지정할 수 있습니다. 자세한 내용은 [와일드카드](#)를 참조하십시오.

JSON 속성: user_group_wild_card

쿼리 그룹

썬표로 분리된 쿼리 그룹 목록입니다. 쿼리 그룹 멤버가 데이터베이스에서 쿼리를 실행하면 쿼리 그룹과 연결되어 있는 대기열로 쿼리가 라우팅됩니다.

JSON 속성: `query_group`

쿼리 그룹 와일드카드

쿼리 그룹의 와일드카드 활성화 여부를 나타내는 부울 값입니다. 값이 0이면 와일드카드가 비활성화되고, 값이 1이면 와일드카드가 활성화됩니다. 와일드카드가 활성화되면 쿼리를 실행할 때 "*" 또는 "?"를 사용하여 다수의 쿼리 그룹을 지정할 수 있습니다. 자세한 내용은 [와일드카드](#)를 참조하십시오.

JSON 속성: `query_group_wild_card`

사용자 역할

사용자 역할이 썬표로 구분된 목록입니다. 해당 사용자 역할을 가진 멤버가 데이터베이스에서 쿼리를 실행하면 사용자 역할과 연결되어 있는 대기열로 쿼리가 라우팅됩니다. 사용자 역할에 대한 자세한 내용은 [역할 기반 액세스 제어\(RBAC\)](#)를 참조하세요.

JSON 속성: `user_role`

사용자 역할 와일드카드

쿼리 그룹의 와일드카드 활성화 여부를 나타내는 부울 값입니다. 값이 0이면 와일드카드가 비활성화되고, 값이 1이면 와일드카드가 활성화됩니다. 와일드카드가 활성화되면 쿼리를 실행할 때 "*" 또는 "?"를 사용하여 다수의 쿼리 그룹을 지정할 수 있습니다. 자세한 내용은 [와일드카드](#)를 참조하십시오.

JSON 속성: `user_role_wild_card`

제한 시간(ms)

WLM 제한 시간(`max_execution_time`)은 더 이상 사용되지 않습니다. 자동 WLM을 사용 중인 경우에는 사용할 수 없습니다. 대신 `query_execution_time`을 통해 QMR(쿼리 모니터링 규칙)을 생성하여 쿼리 실행 경과 시간을 제한하세요. 자세한 내용은 [WLM 쿼리 모니터링 규칙](#)을 참조하십시오.

쿼리가 취소되기 전까지 최대한 실행할 수 있는 시간(ms)입니다. 경우에 따라 SELECT 문 등의 읽기 전용 쿼리는 WLM 제한 시간으로 인해 취소될 수 있습니다. 경우에 따라 WLM은 WLM 대기열 할당 규칙을 기반으로 다음에 일치하는 대기열에 쿼리를 라우팅하려고 합니다. 하지만 그 밖에 쿼리

와 일치하는 대기열 정의가 없으면 쿼리가 취소되고 기본 대기열로도 할당되지 않습니다. 자세한 내용은 [WLM 쿼리 대기열 건너뛰기](#)를 참조하십시오. returning 상태에 도달한 쿼리에는 WLM 제한 시간이 적용되지 않습니다. 쿼리 상태는 [STV_WLM_QUERY_STATE](#) 시스템 테이블에서 확인할 수 있습니다.

JSON 속성: max_execution_time

"Memory"

대기열에 할당되는 메모리 비율입니다. 대기열 중에서 최소 1개 이상의 대기열에 메모리 비율을 지정하는 경우에는 나머지 모든 대기열의 총 메모리 비율을 최대 100%까지만 지정해야 합니다. 모든 대기열에 할당되는 메모리 비율이 100% 미만인 경우에는 해당 서비스에서 미할당 메모리를 관리합니다. 이 서비스는 처리를 위해 추가 메모리가 필요한 대기열에 이 미할당 메모리를 임시로 제공할 수 있습니다.

JSON 속성: memory_percent_to_use

쿼리 모니터링 규칙

WLM 쿼리 모니터링 규칙은 지정하는 기준, 즉 조건자에 따라 WLM 쿼리 대기열을 연속해서 모니터링하는 데 사용됩니다. 예를 들어 시스템 리소스를 과도하게 소비하는 대기열을 모니터링하다가 쿼리가 지정한 성능 경계를 초과하면 지정한 작업을 시작할 수 있습니다.

Note

프로그래밍 방식으로 규칙을 생성하려면 콘솔을 사용하여 파라미터 그룹 정의에 추가할 JSON을 생성하는 것이 가장 좋습니다.

생성된 쿼리 모니터링 규칙을 원하는 쿼리 대기열과 연결합니다. 대기열 1개당 규칙 수는 최대 25개까지 가능하며 모든 대기열의 총 규칙 수도 25개로 제한됩니다.

JSON 속성: rules

JSON 속성 계층:

```
rules
  rule_name
  predicate
    metric_name
    operator
    value
```

action	value

각 규칙마다 지정되는 속성은 다음과 같습니다.

- **rule_name** - 규칙 이름은 WLM 구성 내에서 고유해야 합니다. 최대 32자의 영숫자 또는 밑줄로 구성되며, 공백이나 인용 부호는 포함될 수 없습니다.
- **predicate** - 규칙당 최대 3개의 조건자를 가질 수 있습니다. 각 조건자마다 지정되는 속성은 다음과 같습니다.
 - **metric_name** - 지표 목록은 Amazon Redshift 데이터베이스 개발자 안내서의 [쿼리 모니터링 링 지표](#)를 참조하세요.
 - **operator** - 연산은 =, < 및 >입니다.
 - **value** - 지정한 지표에서 작업이 트리거되는 임계값입니다.
- **action** - 각 규칙은 한 가지 작업과 연결됩니다. 유효 작업은 다음과 같습니다.
 - log
 - hop(수동 WLM에서만 사용할 수 있음)
 - abort
 - change_query_priority(자동 WLM에서만 사용할 수 있음)

다음은 rule_1이라는 이름의 WLM 쿼리 모니터링 규칙에 대한 JSON 속성을 나타낸 예로서, 조건자가 2개이고 작업은 hop입니다.

```
"rules": [
  {
    "rule_name": "rule_1",
    "predicate": [
      {
        "metric_name": "query_execution_time",
        "operator": ">",
        "value": 100000
      },
      {
        "metric_name": "query_blocks_read",
        "operator": ">",
        "value": 1000
      }
    ],
    "action": "hop"
  }
]
```

]

쿼리 대기열 구성을 위한 이러한 각 속성 및 전략에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [워크로드 관리 구현](#)을 참조하세요.

AWS CLI를 사용하여 WLM 파라미터 구성

WLM을 구성하려면 `wlm_json_configuration` 파라미터를 수정합니다.

`wlm_json_configuration` 속성 값의 최대 크기는 8000자입니다. 파라미터 값은 JSON(JavaScript Object Notation) 형식을 따릅니다. AWS CLI, Amazon Redshift API 또는 AWS SDK 중 하나를 사용하여 WLM을 구성하는 경우 이 섹션의 나머지 부분에서 `wlm_json_configuration` 파라미터에 맞춰 JSON 구조를 생성하는 방법에 대해 알아보세요.

Note

Amazon Redshift 콘솔을 사용해 WLM을 구성한다면 콘솔에서 쉽게 대기열을 추가하고 속성을 구성할 수 있기 때문에 JSON 형식에 대해 이해할 필요는 없습니다. 콘솔을 사용해 WLM을 구성하는 방법에 대한 자세한 내용은 [파라미터 그룹 수정](#) 단원을 참조하십시오.

예제

다음은 자동 WLM을 사용하는 대기열 하나를 정의하는 기본 WLM 구성 예입니다.

```
{
  "auto_wlm": true
}
```

예제

다음은 동시성 레벨(쿼리 슬롯)이 5인 수동 WLM 대기열 하나를 정의하는 사용자 지정 WLM 구성 예입니다.

```
{
  "query_concurrency":5
}
```

구문:

기본 WLM 구성은 대기열과 속성 하나로 이루어져 매우 간단합니다. 여기에 대기열을 추가하고, JSON 구조에 포함된 각 대기열마다 다수의 속성을 구성할 수 있습니다. 다음은 여러 속성과 함께 다수의 대기열을 구성하여 JSON 구조를 생성하는 구문입니다.

```
[
  {
    "ParameterName": "wlm_json_configuration", "ParameterValue":
      "[
        {
          "q1_first_property_name": "q1_first_property_value",
          "q1_second_property_name": "q1_second_property_value",
          ...
        },
        {
          "q2_first_property_name": "q2_first_property_value",
          "q2_second_property_name": "q2_second_property_value",
          ...
        }
      ]"
  }
]
```

위 예에서 q1으로 시작하는 대표 속성은 첫 번째 대기열의 배열에 속하는 객체입니다. 각 객체는 이름/값 페어입니다. 즉 name과 value 모두 첫 번째 대기열의 WLM 속성을 설정합니다. q2로 시작하는 대표 속성은 두 번째 대기열의 배열에 속하는 객체입니다. 대기열이 더 필요한 경우에는 추가되는 대기열마다 배열도 하나씩 더한 후 각 객체 속성을 설정합니다.

WLM 구성을 수정할 때는 대기열의 속성 하나만 변경하더라도 대기열 구조 전체를 입력해야 합니다. 이는 전체 JSON 구조가 wlm_json_configuration 파라미터 값처럼 하나의 문자열로 전달되기 때문입니다.

AWS CLI 명령 형식 지정

wlm_json_configuration를 사용할 때는 AWS CLI 파라미터에 특정 형식이 필요합니다. 사용하는 형식은 클라이언트 운영 체제에 따라 다릅니다. 운영 체제마다 JSON 구조를 묶는(enclose) 방법이 다르기 때문에 명령줄에서 정확히 입력해야 합니다. Linux, Mac OS X 및 Windows 운영 체제에서 명령을 알맞게 작성하는 방법에 대한 자세한 내용은 이후 단원들을 참조하십시오. 일반적으로 AWS CLI로 JSON 데이터 구조를 묶는 차이점에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 [인용 문자열](#)을 참조하세요.

예제

다음은 example-parameter-group이라는 파라미터 그룹에 대해 수동 WLM을 구성하는 명령의 예입니다. 구성을 사용하면 단기 쿼리의 가속의 최대 실행 시간이 0으로 설정된 단기 쿼리 가속이 활성화되어 WLM에서 이 값을 동적으로 설정하도록 지시합니다. ApplyType 설정은 dynamic입니다. 즉, 다른 정적 속성 변경이 없는 한 파라미터의 동적 속성에 대한 변경 사항이 바로 적용됩니다. 구성은 다음과 같이 대기열 3개를 정의합니다.

- 첫 번째 대기열에서는 쿼리가 대기열로 라우팅될 수 있도록 사용자가 report를 쿼리 레이블로 지정할 수 있습니다(query_group 속성에서 지정). report* 레이블에는 와일드카드 검색이 활성화되어 있기 때문에 레이블이 정확하지 않더라도 쿼리가 대기열에 라우팅되는 데 아무런 문제가 없습니다. reports와 reporting이 모두 이 쿼리 그룹과 일치하는 경우를 예로 들어 보겠습니다. 모든 대기열 중에서 이 대기열에 할당되는 메모리는 총 메모리의 25%이며, 동시에 실행할 수 있는 최대 쿼리 수는 4개입니다. 쿼리의 최대 시간은 20000밀리초(ms)로 제한됩니다. 모드는 auto(자동)로 설정되는데, 따라서 대기열의 쿼리 슬롯이 꽉 찬 경우 적합한 쿼리가 확장 클러스터로 전송됩니다.
- 두 번째 대기열에서는 데이터베이스의 admin 또는 dba 그룹 멤버인 사용자가 처리를 위해 쿼리를 대기열로 라우팅할 수 있습니다. 사용자 그룹에서는 와일드카드 검색이 비활성화되어 있기 때문에 사용자가 데이터베이스의 그룹과 정확히 일치해야만 쿼리가 대기열로 라우팅됩니다. 모든 대기열 중에서 이 대기열에 할당되는 메모리는 총 메모리의 40%이며, 동시에 실행할 수 있는 최대 쿼리 수는 5개입니다. 모드는 off(해제)로 설정되므로 관리자 또는 dba 그룹의 멤버가 보낸 모든 쿼리는 기본 클러스터에서 실행됩니다.
- 구성에서 마지막 대기열은 기본 대기열입니다. 모든 대기열 중에서 이 대기열에 할당되는 메모리는 총 메모리의 35%이며, 동시에 처리할 수 있는 최대 쿼리 수는 5개입니다. 모드는 auto(자동)로 설정됩니다.

Note

다음 예에서는 설명을 위해 여러 줄로 표시하였지만 실제 명령에서는 줄 바꿈을 사용하면 안 됩니다.

```
aws redshift modify-cluster-parameter-group
--parameter-group-name example-parameter-group
--parameters
'[
{
  "query_concurrency": 4,
```

```

    "max_execution_time": 20000,
    "memory_percent_to_use": 25,
    "query_group": ["report"],
    "query_group_wild_card": 1,
    "user_group": [],
    "user_group_wild_card": 0,
    "user_role": [],
    "user_role_wild_card": 0,
    "concurrency_scaling": "auto",
    "queue_type": "manual"
  },
  {
    "query_concurrency": 5,
    "memory_percent_to_use": 40,
    "query_group": [],
    "query_group_wild_card": 0,
    "user_group": [
      "admin",
      "dba"
    ],
    "user_group_wild_card": 0,
    "user_role": [],
    "user_role_wild_card": 0,
    "concurrency_scaling": "off",
    "queue_type": "manual"
  },
  {
    "query_concurrency": 5,
    "query_group": [],
    "query_group_wild_card": 0,
    "user_group": [],
    "user_group_wild_card": 0,
    "user_role": [],
    "user_role_wild_card": 0,
    "concurrency_scaling": "auto",
    "queue_type": "manual"
  },
  {"short_query_queue": true}
]'

```

다음은 자동 WLM 구성에 대해 WLM 쿼리 모니터링 규칙을 구성하는 예입니다. 다음 예에서는 `example-monitoring-rules`라는 로그 그룹을 생성합니다. 구성에서는 위 예와 마찬가지로 3개의

대기열을 정의합니다. 그러나 `query_concurrency` 및 `memory_percent_to_use`는 더 이상 지정되지 않습니다. 또한 구성에서는 다음과 같은 규칙 및 쿼리 우선 순위를 추가합니다.

- 첫 번째 대기열에서는 `rule_1`이라는 이름의 규칙을 정의합니다. 이 규칙에는 조건자가 `query_cpu_time > 100000000`과 `query_blocks_read > 1000`, 2개입니다. 규칙 작업은 `log`입니다. 이 대기열의 우선 순위는 `Normal`입니다.
- 두 번째 대기열에서는 `rule_2`라는 이름의 규칙을 정의합니다. 이 규칙에는 조건자가 `query_execution_time > 6000000000`과 `scan_row_count > 10000000000`, 2개입니다. 규칙 작업은 `abort`입니다. 이 대기열의 우선 순위는 `Highest`입니다.
- 구성에서 마지막 대기열은 기본 대기열입니다. 이 대기열의 우선 순위는 `Low`입니다.

Note

다음 예에서는 설명을 위해 여러 줄로 표시하였지만 실제 명령에서는 줄 바꿈을 사용하면 안 됩니다.

```
aws redshift modify-cluster-parameter-group
--parameter-group-name example-monitoring-rules
--parameters
'[ {
  "query_group" : [ "report" ],
  "query_group_wild_card" : 1,
  "user_group" : [ ],
  "user_group_wild_card" : 0,
  "user_role": [ ],
  "user_role_wild_card": 0,
  "concurrency_scaling" : "auto",
  "rules" : [{
    "rule_name": "rule_1",
    "predicate": [{
      "metric_name": "query_cpu_time",
      "operator": ">",
      "value": 10000000 },
      { "metric_name": "query_blocks_read",
        "operator": ">",
        "value": 1000
      } ],
    "action" : "log"
  } ],
}
```

```

    "priority": "normal",
    "queue_type": "auto"
  }, {
    "query_group" : [ ],
    "query_group_wild_card" : 0,
    "user_group" : [ "admin", "dba" ],
    "user_group_wild_card" : 0,
    "user_role": [ ],
    "user_role_wild_card": 0,
    "concurrency_scaling" : "off",
    "rules" : [ {
      "rule_name": "rule_2",
      "predicate": [
        {"metric_name": "query_execution_time",
         "operator": ">",
         "value": 6000000000},
        {"metric_name": "scan_row_count",
         "operator": ">",
         "value": 10000000000}],
      "action": "abort"}],
    "priority": "high",
    "queue_type": "auto"
  }, {
    "query_group" : [ ],
    "query_group_wild_card" : 0,
    "user_group" : [ ],
    "user_group_wild_card" : 0,
    "user_role": [ ],
    "user_role_wild_card": 0,
    "concurrency_scaling" : "auto",
    "priority": "low",
    "queue_type": "auto",
    "auto_wlm": true
  }, {
    "short_query_queue" : true
  } ]'

```

명령줄에서 JSON 파일과 함께 AWS CLI를 사용하여 WLM 구성

wlm_json_configuration를 사용하여 AWS CLI 파라미터를 수정하고 parameters 인수 값을 JSON 파일로 전달할 수 있습니다.

```
aws redshift modify-cluster-parameter-group --parameter-group-name
myclusterparaametergroup --parameters file://modify_pg.json
```

--parameters의 인수는 modify_pg.json 파일에 저장됩니다. 파일 위치는 해당 운영 체제에 맞는 형식으로 지정됩니다. 자세한 내용은 [파일에서 파라미터 로드](#)를 참조하세요. 다음은 modify_pg.json JSON 파일의 내용 예를 보여줍니다.

```
[
  {
    "ParameterName": "wlm_json_configuration",
    "ParameterValue": "[{\"user_group\": \"example_user_group1\", \"query_group\": \"example_query_group1\", \"query_concurrency\": 7}, {\"query_concurrency\": 5}]"
  }
]
```

```
[
  {
    "ParameterName": "wlm_json_configuration",
    "ParameterValue": "[{\"query_group\": [\"reports\"], \"query_group_wild_card\": 0, \"query_concurrency\": 4, \"max_execution_time\": 20000, \"memory_percent_to_use\": 25}, {\"user_group\": [\"admin\", \"dba\"], \"user_group_wild_card\": 1, \"query_concurrency\": 5, \"memory_percent_to_use\": 40}, {\"query_concurrency\": 5, \"memory_percent_to_use\": 35}, {\"short_query_queue\": true, \"max_execution_time\": 5000 }]",
    "ApplyType": "dynamic"
  }
]
```

Linux 및 macOS X 운영 체제의 명령줄에서 AWS CLI를 사용하여 WLM을 구성하는 규칙

다음 규칙에 따라 파라미터가 한 줄에 있는 AWS CLI 명령을 실행합니다.

- 전체 JSON 구조는 작은 따옴표(')와 대괄호([])로 묶여야 합니다.
- 모든 파라미터 이름과 파라미터 값은 큰 따옴표(")로 묶여야 합니다.
- ParameterValue 값에서는 중첩되는 전체 구조를 큰 따옴표(")와 대괄호([])로 묶어야 합니다.
- 중첩 구조에서는 각 대기열의 속성과 값을 일일이 중괄호({ })로 묶어야 합니다.
- 중첩 구조에서는 큰 따옴표(")가 나올 때마다 그 앞에 백슬래시(\) 이스케이프 문자를 사용해야 합니다.

- 이름/값 페어에서는 콜론(:)을 사용하여 각 속성과 값을 구분합니다.
- 이름/값 페어는 각각 쉼표(,)를 사용해 서로 구분합니다.
- 다수의 대기열을 추가할 때는 앞쪽 대기열의 후행 중괄호(})와 뒤쪽 대기열의 선행 중괄호({) 사이에 쉼표(,)를 찍어 서로 구분합니다.

Microsoft Windows 운영 체제의 Windows PowerShell에서 AWS CLI를 사용하여 WLM을 구성하는 규칙

다음 규칙에 따라 파라미터가 한 줄에 있는 AWS CLI 명령을 실행합니다.

- 전체 JSON 구조는 작은 따옴표(')와 대괄호([])로 묶어야 합니다.
- 모든 파라미터 이름과 파라미터 값은 큰 따옴표(")로 묶어야 합니다.
- ParameterValue 값에서는 중첩되는 전체 구조를 큰 따옴표(")와 대괄호([])로 묶어야 합니다.
- 중첩 구조에서는 각 대기열의 속성과 값을 일일이 중괄호({ })로 묶어야 합니다.
- 중첩 구조에서는 큰따옴표(")와 백슬래시(\) 이스케이프 문자가 나올 때마다 그 앞에 백슬래시(\) 이스케이프 문자를 사용해야 합니다. 즉, 속성이 올바르게 전달되려면 백슬래시 3개와 큰따옴표 1개를 사용해야 합니다(\\").
- 이름/값 페어에서는 콜론(:)을 사용하여 각 속성과 값을 구분합니다.
- 이름/값 페어는 각각 쉼표(,)를 사용해 서로 구분합니다.
- 다수의 대기열을 추가할 때는 앞쪽 대기열의 후행 중괄호(})와 뒤쪽 대기열의 선행 중괄호({) 사이에 쉼표(,)를 찍어 서로 구분합니다.

Windows 운영 체제에서 명령 프롬프트를 사용하여 WLM를 구성하는 규칙

다음 규칙에 따라 파라미터가 한 줄에 있는 AWS CLI 명령을 실행합니다.

- 전체 JSON 구조는 큰따옴표(")와 대괄호([])로 묶어야 합니다.
- 모든 파라미터 이름과 파라미터 값은 큰 따옴표(")로 묶어야 합니다.
- ParameterValue 값에서는 중첩되는 전체 구조를 큰 따옴표(")와 대괄호([])로 묶어야 합니다.
- 중첩 구조에서는 각 대기열의 속성과 값을 일일이 중괄호({ })로 묶어야 합니다.
- 중첩 구조에서는 큰따옴표(")와 백슬래시(\) 이스케이프 문자가 나올 때마다 그 앞에 백슬래시(\) 이스케이프 문자를 사용해야 합니다. 즉, 속성이 올바르게 전달되려면 백슬래시 3개와 큰따옴표 1개를 사용해야 합니다(\\").
- 이름/값 페어에서는 콜론(:)을 사용하여 각 속성과 값을 구분합니다.

- 이름/값 페어는 각각 쉼표(,)를 사용해 서로 구분합니다.
- 다수의 대기열을 추가할 때는 앞쪽 대기열의 후행 중괄호()와 뒤쪽 대기열의 선행 중괄호({) 사이에 쉼표(,)를 찍어 서로 구분합니다.

파라미터 그룹 생성

파라미터 값을 기본 파라미터 그룹과 다르게 설정하고 싶다면 고유한 파라미터 그룹을 생성할 수 있습니다.

파라미터 그룹을 생성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 구성(Configurations) 및 워크로드 관리(Workload management)를 차례로 선택하여 워크로드 관리(Workload management) 페이지를 표시합니다.
3. 생성을 선택하여 파라미터 그룹 생성 창을 표시합니다.
4. 파라미터 그룹 이름 및 설명에 값을 입력합니다.
5. 파라미터 그룹을 생성하려면 생성을 선택합니다.

파라미터 그룹 수정

파라미터 그룹을 각각 확인하면 파라미터 값과 워크로드 관리(WLM) 구성을 간략하게 살펴볼 수 있습니다. 파라미터를 수정하여 파라미터 설정과 WLM 구성 속성을 변경할 수 있습니다.

Note

기본 파라미터 그룹은 수정할 수 없습니다.

AWS Management Console

콘솔에서 그룹 파라미터가 파라미터 탭에 표시되고 워크로드 대기열이 워크로드 관리 탭에 표시됩니다.

파라미터 그룹을 수정하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 구성(Configurations) 및 워크로드 관리(Workload management)를 차례로 선택하여 워크로드 관리(Workload management) 페이지를 표시합니다.
3. 수정하려는 파라미터 그룹을 선택하여 파라미터 및 워크로드 관리 탭이 있는 세부 정보 페이지를 표시합니다.
4. 현재 파라미터 설정을 확인하려면 파라미터 탭을 선택합니다.
5. 이러한 파라미터 설정을 변경하려면 파라미터 편집을 선택합니다.

- auto_analyze
- auto_mv
- datestyle
- enable_case_sensitive_identifier
- enable_user_activity_logging
- extra_float_digits
- max_concurrency_scaling_clusters
- max_cursor_result_set_size
- query_group
- require_ssl
- search_path
- statement_timeout
- use_fips_ssl

이런 파라미터에 대한 자세한 내용은 [Amazon Redshift 파라미터 그룹](#) 섹션을 참조하세요.

6. 변경 사항을 입력한 후 저장을 선택하여 파라미터 그룹을 업데이트합니다.

파라미터 그룹의 WLM 구성을 수정하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.

2. 탐색 메뉴에서 구성(Configurations) 및 워크로드 관리(Workload management)를 차례로 선택하여 워크로드 관리(Workload management) 페이지를 표시합니다.
3. 수정하려는 파라미터 그룹을 선택하여 파라미터 및 워크로드 관리 탭이 있는 세부 정보 페이지를 표시합니다.
4. 현재 WLM 구성을 보려면 워크로드 관리 탭을 선택합니다.
5. WLM 구성을 편집하려면 워크로드 대기열 편집을 선택합니다.
6. (선택 사항) 단기 쿼리 가속화(SQA)를 활성화하려면 단기 쿼리 가속화 활성화를 선택합니다.

SQA를 활성화하면, 단기 쿼리의 최대 런타임(1~20초)이 동적으로 기본 설정됩니다. 최대 실행 시간을 고정 값으로 설정하려면 1~20 범위에서 값을 선택합니다.

7. 다음 중 한 가지 이상을 실행하여 대기열 구성을 수정합니다.
 - Switch WLM mode(WLM 모드 전환)를 선택하여 Automatic WLM(자동 WLM)과 Manual WLM(수동 WLM) 중에 선택합니다.
- Automatic WLM(자동 WLM)에서는 메모리 및 Concurrency on main(기본의 동시성) 값이 auto로 설정됩니다.
- 대기열을 생성하려면 Edit workload queues(워크로드 대기열 편집)를 선택한 다음 대기열 추가를 선택합니다.
 - 대기열을 수정하려면 테이블에서 속성 값을 변경합니다. 대기열 유형에 따라 속성에 다음이 포함될 수 있습니다.
 - 대기열 이름은 변경할 수 있습니다.

- "Memory"

- 기본 클러스터의 동시성
- 동시성 확장 모드는 off 또는 auto일 수 있습니다.
- 제한 시간(ms)
- 사용자 그룹
- 쿼리 그룹
- 사용자 역할

이러한 속성에 대한 자세한 내용은 [WLM 구성 파라미터의 속성](#) 단원을 참조하십시오.

⚠ Important

대기열 이름을 변경하면 WLM 대기열 지표(예: WLMQueueLength, WLMQueueWaitTime, WLMQueriesCompletedPerSecond, WLMQueryDuration, WLMRunningQueries 등)의 QueueName 차원 값도 변경됩니다. 따라서 대기열의 이름을 변경하는 경우 설정한 CloudWatch 경보를 변경해야 할 수 있습니다.

- 대기열 순서를 변경하려면 위쪽 및 아래쪽 화살표 버튼을 선택합니다.
 - 대기열을 삭제하려면 테이블에서 삭제할 대기열 행에 있는 삭제 버튼을 선택합니다.
8. (선택 사항) 다음 재부팅 이후 변경 사항을 클러스터에 적용하려면 재부팅 후 동적 변경 사항 적용을 선택합니다.

i Note

일부 변경 사항은 이 설정과 상관없이 클러스터 재부팅이 필요합니다. 자세한 내용은 [WLM 동적 및 정적 속성](#) 섹션을 참조하세요.

9. 저장을 선택합니다.

AWS CLI

AWS CLI를 사용하여 Amazon Redshift 파라미터를 구성할 때는 특정 파라미터 그룹에 대해 [modify-cluster-parameter-group](#) 명령을 사용합니다. 수정할 파라미터 그룹은 parameter-group-name에서 지정합니다. parameters 파라미터(modify-cluster-parameter-group 명령일 때)는 파라미터 그룹에서 수정할 파라미터마다 이름/값 페어를 지정하는 데 사용됩니다.

i Note

wlm_json_configuration를 사용하여 AWS CLI 파라미터를 구성할 때는 특히 고려해야 할 사항이 있습니다. 이번 단원에서 다루는 예는 wlm_json_configuration을 제외한 모든 파라미터에 적용됩니다. wlm_json_configuration을 사용한 AWS CLI 구성 방법에 대한 자세한 내용은 [워크로드 관리](#) 단원을 참조하십시오.

파라미터 값을 수정한 후에는 변경된 파라미터 그룹과 연결된 모든 클러스터를 재부팅해야 합니다. 변경 값이 적용되는 동안에는 클러스터 상태가 applying for ParameterApplyStatus로 표시되고, 값이 적용된 후에는 pending-reboot로 바뀝니다. 재부팅 후에는 클러스터의 데이터베이스가

새로운 파라미터 값을 사용하기 시작합니다. 클러스터 생성에 대한 자세한 내용은 [클러스터 재부팅 단원](#)을 참조하십시오.

Note

wlm_json_configuration 파라미터에는 동적으로 바뀌는 몇 가지 속성이 포함되어 있기 때문에 변경 사항을 적용하기 위해 연결된 클러스터를 재부팅할 필요가 없습니다. 동적 라우팅과 고정 라우팅의 비교 정보는 [WLM 동적 및 정적 속성](#) 단원을 참조하십시오.

다음은 modify-cluster-parameter-group 명령을 사용하여 파라미터를 구성하는 방법을 나타낸 구문입니다. 먼저 *parameter_group_name*을 지정하고, *parameter_name*과 *parameter_value*을 실제로 수정할 파라미터와 파라미터 값으로 치환합니다. 한 번에 다수의 파라미터를 수정할 때는 각 파라미터 및 값 집합을 서로 공백으로 구분해야 합니다.

```
aws redshift modify-cluster-parameter-group --parameter-group-
name parameter_group_name --parameters
ParameterName=parameter_name,ParameterValue=parameter_value
```

다음은 statement_timeout 파라미터 그룹에서 enable_user_activity_logging 및 myclusterparametergroup 파라미터를 구성하는 예입니다.

Note

가독성을 위해 예를 몇 줄로 표시하였지만 실제 AWS CLI에서는 한 줄로 표시됩니다.

```
aws redshift modify-cluster-parameter-group
--parameter-group-name myclusterparametergroup
--parameters ParameterName=statement_timeout,ParameterValue=20000
ParameterName=enable_user_activity_logging,ParameterValue=true
```

쿼리 모니터링 규칙 생성

Amazon Redshift 콘솔을 사용하여 WLM 쿼리 모니터링 규칙을 생성하거나 수정할 수 있습니다. 쿼리 모니터링 규칙은 파라미터 그룹의 WLM 구성 파라미터 중 일부입니다. 쿼리 모니터링 규칙(QMR)을 수정하면 클러스터를 수정할 필요 없이 자동으로 변경됩니다. 자세한 내용은 [WLM 쿼리 모니터링 규칙](#)을 참조하십시오.

규칙을 생성할 때는 규칙 이름과 1개 이상의 조건자, 그리고 작업을 정의합니다.

규칙이 포함된 WLM 구성을 저장하면 규칙 정의에 대한 JSON 코드를 WLM 구성 파라미터의 JSON에서 확인할 수 있습니다.

쿼리 모니터링 규칙을 생성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 구성(Configurations) 및 워크로드 관리(Workload management)를 차례로 선택하여 워크로드 관리(Workload management) 페이지를 표시합니다.
3. 수정하려는 파라미터 그룹을 선택하여 파라미터 및 워크로드 관리 탭이 있는 세부 정보 페이지를 표시합니다.
4. 워크로드 관리 탭을 선택하고 워크로드 대기열 편집을 선택하여 WLM 구성을 편집합니다.
5. 사전 정의된 템플릿을 사용하거나 처음부터 새 규칙을 추가합니다.

사전 정의된 템플릿을 사용하려면 다음을 수행하십시오.

1. [쿼리 모니터링 규칙(Query monitoring rules)] 그룹에서 [템플릿에서 규칙 추가(Add rule from template)]를 선택합니다. 규칙 템플릿 목록이 표시됩니다.
2. 정책 템플릿을 하나 이상 선택합니다. 저장을 선택하면 WLM이 선택하는 템플릿마다 규칙을 1개씩 생성합니다.
3. 규칙 이름, 조건자 및 작업을 포함하여 규칙의 값을 입력하거나 확인합니다.
4. 저장을 선택합니다.

처음부터 새 규칙을 추가하려면 다음을 수행하십시오.

1. 조건자를 추가하려면 Add predicate(조건자 추가)를 선택합니다. 규칙당 최대 3개의 조건자를 가질 수 있습니다. 모든 조건자가 충족되면 WLM이 연결되어 있는 작업을 시작합니다.
2. 작업을 선택합니다. 규칙마다 작업이 하나씩 있습니다.
3. 저장을 선택합니다.

Amazon Redshift는 WLM 구성 파라미터를 JSON 형식으로 생성하여 JSON 섹션에 표시합니다.

파라미터 그룹 삭제

더 이상 필요가 없거나 연결되어 있는 클러스터가 없는 파라미터 그룹은 삭제할 수 있습니다. 단, 사용자 지정 파라미터 그룹만 삭제가 가능합니다.

파라미터 그룹을 삭제하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 구성(Configurations) 및 워크로드 관리(Workload management)를 차례로 선택하여 워크로드 관리(Workload management) 페이지를 표시합니다.
3. 파라미터 그룹에서 수정할 파라미터 그룹을 선택합니다.

Note

기본 파라미터 그룹은 삭제할 수 없습니다.

4. 삭제를 선택한 다음 파라미터 그룹 삭제를 확인합니다.

AWS 파트너와 Amazon Redshift 통합

Amazon Redshift와 협력하면 Amazon Redshift 콘솔에서 AWS 파트너와 통합할 수 있습니다. [클러스터 세부 정보(Cluster details)] 페이지에서 AWS 파트너 애플리케이션을 사용하여 Amazon Redshift 데이터 웨어하우스로 데이터 온보딩 속도를 높일 수 있습니다. 또한 클러스터의 기존 데이터와 함께 다른 소스의 데이터를 조인하고 분석할 수 있습니다. Informatica와 완전히 통합하기 전에 파트너의 IP 주소를 인바운드 트래픽 허용 목록에 추가해야 합니다. 다음 AWS 파트너는 Amazon Redshift와 통합 가능합니다.

- [Datacoral](#)
- [Etleap](#)
- [Fivetran](#)
- [SnapLogic](#)
- [Stitch](#)
- [Upsolver](#)
- [Matillion\(미리 보기\)](#)
- [Sisense\(미리 보기\)](#)
- [Thoughtspot](#)

AWS 파트너는 AWS CLI 또는 Amazon Redshift API 작업을 사용하여 Amazon Redshift와 통합할 수 있습니다. 자세한 내용은 AWS CLI Command Reference 또는 Amazon Redshift API Reference를 참조하세요.

클러스터를 AWS 파트너와 통합하려면 다음 절차를 따르세요.

Amazon Redshift 클러스터를 AWS 파트너와 통합하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다.
3. 통합할 클러스터를 선택합니다.
4. [파트너 통합 추가(Add partner integration)]를 선택합니다. 사용 가능한 AWS 파트너에 대한 세부 정보가 포함된 [파트너 선택(Choose partner)] 페이지가 열립니다.
5. AWS 파트너를 선택하고 [다음(Next)]을 선택합니다.

선택한 AWS 파트너에 대한 세부 정보와 통합하려는 클러스터에 대한 세부 정보가 나타납니다. 클러스터 세부 정보 섹션에는 클러스터 식별자, 엔드포인트, 데이터베이스 이름 및 사용자 이름(데이터베이스 사용자 이름)과 같이 AWS 파트너 웹 사이트에서 제공하는 정보가 포함됩니다. 이 정보는 선택한 파트너에게 전송됩니다.

6. [파트너 추가(Add partner)]를 선택하여 AWS 파트너의 웹 사이트를 엽니다.
7. 파트너의 웹 사이트에서 Amazon Redshift 클러스터와의 통합을 구성합니다. 파트너의 웹 사이트에서 Amazon Redshift 클러스터에 로드되는 데이터 원본을 선택하고 구성할 수 있습니다. 또한 추가 추출, 로드 및 변환(ELT) 변환을 정의하여 비즈니스 데이터를 처리하고, 다른 데이터 집합과 조인하고, 분석 및 보고를 위한 통합 뷰를 구축할 수 있습니다.

클러스터 세부 정보 [속성(Properties)] 탭에서 AWS 파트너 통합을 보고 관리할 수 있습니다. 통합 섹션에는 AWS 파트너 웹 사이트에 연결하는 데 사용할 수 있는 파트너 이름, 통합의 상태, 데이터를 수신하는 데이터베이스 및 클러스터를 업데이트했을 수 있는 마지막으로 성공한 연결이 나열됩니다.

가능한 상태 값은 다음과 같습니다.

- [활성(Active)] - AWS 파트너가 클러스터에 연결하고 구성된 태스크를 완료할 수 있습니다.
- [비활성(Inactive)] - AWS 파트너 통합이 존재하지 않습니다.
- [런타임 실패(Runtime failure)] - AWS 파트너가 클러스터에 연결할 수 있지만 구성된 태스크를 완료할 수 없습니다.
- [연결 실패(Connection failure)] - AWS 파트너가 클러스터에 연결할 수 없습니다.

Amazon Redshift에서 AWS 파트너 통합을 삭제한 후에도 데이터는 클러스터로 계속 유입됩니다. 파트너의 웹 사이트에서 삭제를 완료합니다.

AWS 파트너를 통한 데이터 로드

파트너를 Amazon Redshift 클러스터와 통합하는 것 외에도 파트너의 데이터 로드 도구를 사용하여 30 개 이상의 소스에서 Amazon Redshift 클러스터로 데이터를 이동할 수도 있습니다. 그렇게 하려면 먼저 파트너의 IP 주소(아래 참조)를 인바운드 규칙의 허용 목록에 추가해야 합니다. Amazon EC2 보안 그룹에 규칙 추가에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스의 인바운드 트래픽 권한 부여](#)를 참조하세요. Informatica Data Loader 도구는 무료로 사용할 수 있지만 선택한 데이터 소스 및 대상에 따라 데이터 수신 요금이 부과될 수 있습니다.

데이터는 다음 파트너에서 로드할 수 있습니다.

- [Informatica](#)

파트너를 통해 Amazon Redshift 클러스터에 데이터를 로드하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 AWS Partner Integration(Amazon 파트너 통합)을 선택한 다음 클러스터를 통합하려는 파트너를 선택합니다.
3. Complete <partner-name> integration(<파트너-이름> 통합 완료)을 선택합니다. 파트너의 통합 사이트로 리디렉션됩니다.
4. 파트너 사이트에 필요한 세부 정보를 입력하고 통합을 완료합니다.

예약 노드

AWS에서 Amazon Redshift 사용으로 발생하는 요금은 컴퓨팅 노드를 기준으로 합니다. 각각의 컴퓨팅 노드는 시간당 요금으로 과금됩니다. 시간당 요금은 리전, 노드 유형 및 노드에 온디맨드 요금이 적용되는지 또는 예약 노드 요금이 적용되는지 등의 요인에 따라 달라집니다.

온디맨드 노드 요금은 Amazon Redshift에서 가장 비싸지만 가장 유연한 옵션입니다. 온디맨드 요금의 경우, 실행 중인 클러스터에 있는 컴퓨팅 노드에 대해서만 요금이 청구됩니다. 클러스터를 종료하거나 삭제하면 해당 클러스터에 있었던 컴퓨팅 노드에 대해 더 이상 요금이 청구되지 않습니다. 사용하는 컴퓨팅 노드에 대해서만 과금되며, 그 이상의 비용은 들지 않습니다. 각각의 컴퓨팅 노드에 대해 청구되는 시간당 요금은 리전, 노드 유형 등의 요인에 따라 달라집니다.

예약 노드 요금은 할인된 시간당 요금이 컴퓨팅 노드에 과금되기 때문에 온디맨드 요금보다 저렴합니다. 하지만 이 할인 요금을 적용받으려면 예약 노드 상품을 구매해야 합니다. 예약은 상품 구매 시 하게 됩니다. 예약을 통해 예약 기간 동안 예약하는 각각의 노드에 대해 할인율이 설정됩니다. 할인율은 리전, 노드 유형, 기간, 결제 방식 등의 요인에 따라 달라집니다.

Amazon Redshift 콘솔에서 `PurchaseReservedNodeOffering` API 작업을 호출하거나 예약 노드 구매를 선택하여 예약 노드를 노드로 지정할 수도 있습니다. 예약 노드를 구매할 때, 해당하는 예약 노드 유형에 대한 AWS 리전, 노드 유형, 기간, 노드의 수량, 제공 유형을 지정해야 합니다. 예약 노드는 지정된 AWS 리전에서만 사용할 수 있습니다.

이 주제에서는 예약 노드 상품이 무엇인지 살펴보고 Amazon Redshift 클러스터 실행 비용을 줄이기 위한 구매 방법을 알아봅니다. 이 주제에서는 요금제 개념과 요금제가 과금에 미치는 영향을 이해할 수 있도록 온디맨드 요금 또는 할인 요금을 개괄적으로 설명합니다. 구체적인 요금에 대한 자세한 내용은 [Amazon Redshift 요금](#) 섹션을 참조하세요.

예약 노드

장기간 Amazon Redshift 클러스터를 계속 실행할 계획이라면 예약 노드 상품 구매를 고려해 봐야 합니다. 이 상품은 온디맨드 요금에 비해 비용을 크게 절감할 수 있지만 컴퓨팅 노드를 예약하고 1년 또는 3년 동안 이러한 노드에 대한 요금을 지불하기로 약속해야 합니다.

예약 노드는 노드에 대해 청구되는 요금을 결정하는 데만 사용되는 과금 개념입니다. 노드를 예약해도 실제로 노드가 생성되지는 않습니다. 예약된 노드에 대해서는 사용 여부에 상관없이 요금이 청구됩니다. 즉, 실행 중인 클러스터에 할인 요금이 적용되는 노드가 있는지 여부에 상관없이 예약 기간 동안 예약하는 각 노드에 대해 요금을 지불해야 합니다.

프로젝트 평가 단계에 있거나 개념 증명을 개발 중일 때 온디맨드 요금제는 종량제 요금을 지불하거나, 사용한 만큼만 요금을 지불하거나, 클러스터를 종료하거나 삭제하여 언제든지 지불을 중단할 수 있는 유연성을 제공합니다. 프로덕션 환경의 요구가 확정되고 실행 단계가 시작되면 하나 이상의 상품을 구매하여 컴퓨팅 노드를 예약하는 것을 고려해 봐야 합니다.

상품은 하나 이상의 컴퓨팅 노드에 적용될 수 있습니다. 상품을 구매할 때 예약할 컴퓨팅 노드 수를 지정합니다. 여러 컴퓨팅 노드에 하나의 상품을 구매할 수도 있고 여러 개의 상품을 구매해 각 상품마다 일정 수의 컴퓨팅 노드를 지정할 수도 있습니다.

예를 들어 3개의 컴퓨팅 노드를 위한 다음 구매 방법 중 어느 것을 사용해도 됩니다.

- 하나의 상품을 구매해 3개의 컴퓨팅 노드를 지정합니다.
- 2개의 상품을 구매하고 첫 상품에 1개의 컴퓨팅 노드를 지정하고 두 번째 상품에 2개의 컴퓨팅 노드를 지정합니다.
- 3개의 상품을 구매해 각 상품마다 1개의 컴퓨팅 노드를 지정합니다.

예약 노드 간 요금 비교

Amazon Redshift는 몇 가지 결제 옵션을 제공합니다. 선택하는 결제 방식은 결제 일정과 예약에 대해 청구되는 할인 요금에 영향을 미칩니다. 예약에 대해 선불로 결제하는 액수가 클수록 전체적인 비용 절감에 좋습니다.

상품에 사용할 수 있는 결제 방식은 다음과 같습니다. 온디맨드 요금제에 비해 절감폭이 가장 적은 상품부터 가장 큰 상품 순서로 나열되어 있습니다.

Note

지정된 예약 기간 동안은 예약 노드 사용 여부에 상관없이 매시간 적용되는 시간당 요금이 청구됩니다. 결제 방식은 결제 빈도 및 적용 할인율만을 결정합니다. 자세한 내용은 [예약 노드](#) 섹션을 참조하세요.

결제 옵션	결제 일정	상대적 절감액	지속 시간	선불 요금	월 기본 요금
선수금 없음	예약 기간 동안 월 할부. 선결제 없음.	온디맨드 요금에 비해 약 20퍼센트 할인.	1년 또는 3년	없음	예

결제 옵션	결제 일정	상대적 절감액	지속 시간	선불 요금	월 기본 요금
부분 선결제	일부 선결제 및 예약 기간 동안 월 할부.	기간에 따라 최대 41~73퍼센트 할인.	1년 또는 3년	예	예
전체 선결제	예약에 대해 전액 선결제. 월별 요금 없음.	기간에 따라 최대 42~76퍼센트 할인.	1년 또는 3년	예	없음

특정 옵션 및 기간은 이용 가능 여부에 따라 달라질 수 있습니다.

Note

이전에 Amazon Redshift용 Heavy 사용률 서비스를 구매한 경우 이에 상응하는 서비스는 부분 선결제 서비스입니다.

예약 노드의 작동 방식

예약 노드 상품의 경우, 이전 단원에서 설명한 결제 조건에 따라 결제합니다. 이미 실행 중인 클러스터가 있는 경우나 예약 후에 클러스터를 시작하는 경우나 상관없이 이 방식으로 결제합니다.

상품을 구매하면 예약은 처리될 때까지 결제 보류 중 상태가 됩니다. 예약 처리에 실패하면 결제 실패 상태로 표시되며 프로세스를 다시 시도할 수 있습니다. 예약이 성공적으로 처리되면 상태가 활성으로 변경됩니다. 상태가 활성으로 바뀔 때까지는 예약에 적용되는 할인율이 청구서에 적용되지 않습니다. 예약 기간이 경과하면 상태가 만료로 바뀌지만 이력 조회를 목적으로 예약에 관한 정보에 계속 액세스할 수 있습니다. 예약이 만료 상태일 때 클러스터는 계속 실행되지만 노드에 할인 요금을 적용하는 다른 예약이 없다면 온디맨드 요금이 과금될 수 있습니다.

예약 노드는 상품을 구매하는 리전 전용입니다. Amazon Redshift 콘솔을 사용하여 상품을 구매하는 경우, 상품을 구매하려는 AWS 리전을 선택한 다음 예약 프로세스를 완료합니다. 프로그래밍 방식으로 상품을 구매하는 경우, 연결하는 Amazon Redshift 엔드포인트에 의해 리전이 결정됩니다. Amazon Redshift 리전에 대한 자세한 내용은 Amazon Web Services 일반 참조의 [리전 및 엔드포인트](#)를 참조하세요.

클러스터를 시작할 때 모든 노드에 할인율이 적용되려면 선택한 리전, 노드 유형, 노드 수가 하나 이상의 활성 예약과 일치해야 합니다. 그렇지 않으면, 활성 예약과 일치하지 않는 노드의 요금이 온디맨드 요금으로 청구됩니다.

실행 중인 클러스터에서 예약한 노드 수를 초과하는 경우, 이 추가적인 노드에 대해서는 온디맨드 요금이 발생하기 시작합니다. 다시 말해 예약한 노드 수가 얼마냐에 따라 같은 클러스터에 있는 노드에 청구되는 요금이 달라질 수 있습니다. 이러한 추가적 노드에 대해 다른 상품을 구매하면 예약 상태가 활성으로 바뀐 후 남은 기간 동안 이러한 노드에 대해 할인율이 적용됩니다.

클러스터 크기를 다른 노드 유형으로 조정하고 이러한 노드 유형을 예약하지 않은 경우, 온디맨드 요금이 청구됩니다. 크기를 조정된 클러스터에 대해 할인 요금을 적용받으려면 새 노드 유형의 다른 상품을 구매할 수 있습니다. 다만 원래 예약에 대해서는 기간이 경과하기 전까지 계속 요금을 지불해야 합니다. 기간이 만료되기 전에 예약을 변경해야 하는 경우 [AWS Console](#)을 사용하여 지원 사례를 생성합니다.

Note

콘솔에는 사용된 예약 노드와 사용되지 않은 예약 노드의 수가 표시됩니다. 하지만 콘솔에는 현재 사용자 계정이 사용하는 노드 수만 사용된 것으로 표시됩니다. 동일한 지급인 계정의 다른 사용자 계정이 노드를 사용하는 경우 콘솔에 해당 노드가 사용되지 않은 것으로 표시됩니다.

예제

- 지급인 계정에서 20개의 노드를 예약합니다.
- 현재 사용자 계정이 6개의 노드를 사용합니다.
- 동일한 지급인 계정의 다른 사용자 계정도 6개의 노드를 사용합니다.

이 예에서 콘솔에는 사용된 노드가 6개, 사용되지 않은 노드가 14개로 표시됩니다.

예약 노드 및 통합 결제

구매 계정이 단일 통합 결제 지급인 계정으로 과금되는 계정의 일부인 경우, 예약 노드의 요금 혜택이 공유됩니다. 모든 하위 계정에서 발생한 시간당 요금 또한 매월 지급인 계정으로 합산됩니다. 이 방식은 일반적으로 직무가 서로 다른 팀이나 그룹이 있는 회사에서 유용하며, 정상적인 예약 노드 규칙에 따라 요금이 계산됩니다. 자세한 내용은 AWS Billing User Guide의 [Consolidated Billing](#)을 참조하세요.

예약 노드 예제

이 단원의 시나리오는 다음과 같은 예약 세부 정보를 사용하여 온디맨드 요금과 할인 요금을 기준으로 노드에 어떻게 요금이 발생하는지 보여 줍니다.

- 리전: 미국 서부(오레곤)
- 노드 유형: ra3.xlplus
- 결제 방식: 선결제 없음
- 기간: 1년
- 예약 노드 수: 16

예 1

미국 서부(오레곤) 리전에 노드가 20개인 클러스터가 1개 있는 경우

이 시나리오에서 16개 노드는 예약 시부터 할인 요금이 적용되지만 클러스터에 있는 추가적인 4개 노드에는 온디맨드 요금이 과금됩니다.

예제 2

미국 서부(오레곤) 리전에 노드가 12개인 클러스터가 1개 있는 경우

이 시나리오에서는 클러스터의 12개 노드 모두 예약 시부터 할인 요금이 적용됩니다. 하지만 현재 할인이 적용되는 실행 중인 클러스터가 없더라도 예약한 나머지 예약 노드에 대해 요금을 지불해야 합니다.

예 3

미국 서부(오레곤) 리전에 노드가 12개인 클러스터가 1개 있는 경우 이 구성으로 몇 개월 동안 클러스터를 실행한 다음 클러스터에 노드를 추가해야 합니다. 클러스터 크기를 조정하고 같은 노드 유형을 선택해 총 16개의 노드를 지정합니다.

이 시나리오에서는 16개 노드에 대해 할인된 요금이 청구됩니다. 클러스터에 있는 노드 수와 예약한 노드 수가 동일하기 때문에 1년 전체 기간 동안 요금은 그대로입니다.

예 4

미국 서부(오레곤) 리전에 노드가 16개인 클러스터가 1개 있는 경우 이 구성으로 몇 개월 동안 클러스터를 실행한 다음 노드를 추가해야 합니다. 클러스터 크기를 조정하고 같은 노드 유형을 선택해 총 20개의 노드를 지정합니다.

이 시나리오에서는 크기 조정 이전의 모든 노드에 대해 할인된 요금이 청구됩니다. 크기 조정 후에는 1년 기간 중 남은 기간 동안 16개 노드에 대해 할인된 요금이 청구되며, 클러스터에 추가한 4개 노드에 대해서는 온디맨드 요금이 청구됩니다.

예 5

미국 서부(오레곤) 리전에 클러스터가 2개 있는 경우 클러스터 하나에는 노드 6개, 다른 하나에는 노드 10개가 있습니다.

이 시나리오에서는 두 클러스터에 있는 전체 노드 수가 예약한 노드 수와 동일하기 때문에 모든 노드에 대해 할인된 요금이 청구됩니다.

예 6

미국 서부(오레곤) 리전에 클러스터가 2개 있는 경우 클러스터 하나에는 노드 4개, 다른 하나에는 노드 6개가 있습니다.

이 시나리오에서는 실행 중인 클러스터에 있는 10개 노드에 대해 할인된 요금이 청구되며, 현재 할인이 적용되는 실행 중인 클러스터가 없더라도 예약한 추가 6개의 노드에 대해서도 할인된 요금을 지불합니다.

예약 노드 구입

AWS Management Console 또는 AWS CLI를 사용하여 예약된 노드 상품을 구매하고 현재 및 과거 예약을 봅니다.

AWS Management Console

예약 노드를 구매하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 다음 예약 노드(Reserved nodes)를 선택하여 예약 노드 목록을 표시합니다.
3. 예약 노드 구매를 선택하여 구매하려는 노드의 속성을 선택할 수 있는 페이지를 표시합니다.
4. 노드의 속성을 입력한 후 예약 노드 구매를 선택합니다.

노드 구매를 마치면 예약 노드 목록에 예약을 비롯해 노드 유형, 노드 수, 예약 상태 등 각 예약에 대한 세부 정보가 표시됩니다. 보안 정책에 대한 자세한 내용은 [예약 노드의 작동 방식](#) 단원을 참조하십시오.

예약 노드를 업그레이드하려면 AWS CLI를 사용합니다.

모든 노드 유형을 예약 노드로 전환할 수는 없으며 기존 예약 노드를 갱신할 수 없을 수도 있습니다. 이는 노드 유형이 단종되었기 때문일 수 있습니다. 단종된 노드 유형을 갱신하려면 고객 지원에 문의하십시오.

AWS CLI

AWS CLI를 사용하여 예약 노드 예약을 업그레이드하려면

1. 결제 유형, 기간, 요금 등에 대한 요구사항을 충족하는 상품의 ReservedNodeOfferingID 목록을 가져옵니다. 다음 예는 이 단계를 보여 줍니다.

```
aws redshift get-reserved-node-exchange-offerings --reserved-node-id xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
{
  "ReservedNodeOfferings": [
    {
      "Duration": 31536000,
      "ReservedNodeOfferingId": "yyyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyyy",
      "UsagePrice": 0.0,
      "NodeType": "dc2.large",
      "RecurringCharges": [
        {
          "RecurringChargeFrequency": "Hourly",
          "RecurringChargeAmount": 0.2
        }
      ],
      "CurrencyCode": "USD",
      "OfferingType": "No Upfront",
      "ReservedNodeOfferingType": "Regular",
      "FixedPrice": 0.0
    }
  ]
}
```

2. `accept-reserved-node-exchange`를 호출하고 교환하려는 DC1 예약 노드의 ID를 이전 단계에서 얻은 ReservedNodeOfferingID와 함께 제공합니다.

다음 예는 이 단계를 보여 줍니다.

```
aws redshift accept-reserved-node-exchange --reserved-node-id xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxxxxx --target-reserved-node-offering-id yyyyyyyy-yyyy-yyyy-
yyyy-yyyyyyyyyyyyyyyy
{
  "ExchangedReservedNode": {
    "UsagePrice": 0.0,
    "OfferingType": "No Upfront",
    "State": "exchanging",
    "FixedPrice": 0.0,
    "CurrencyCode": "USD",
    "ReservedNodeId": "zzzzzzzz-zzzz-zzzz-zzzz-zzzzzzzzzzzzz",
    "NodeType": "dc2.large",
    "NodeCount": 1,
    "RecurringCharges": [
      {
        "RecurringChargeFrequency": "Hourly",
        "RecurringChargeAmount": 0.2
      }
    ],
    "ReservedNodeOfferingType": "Regular",
    "StartTime": "2018-06-27T18:02:58Z",
    "ReservedNodeOfferingId": "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyyyyyy",
    "Duration": 31536000
  }
}
```

[describe-reserved-nodes](#)를 호출하고 Node type에 대한 값을 확인하여 교환이 완료되었는지 확인할 수 있습니다.

Amazon Redshift의 보안

AWS은 클라우드 보안을 가장 중요하게 생각합니다. AWS 고객으로서 여러분은 가장 높은 보안 요구 사항을 충족하기 위해 설계된 데이터 센터 및 네트워크 아키텍처의 혜택을 받게 됩니다.

보안은 AWS와 사용자의 공동 책임입니다. [공동 책임 모델](#)은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드의 보안 - AWS는 AWS 클라우드 내에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 제3자 감사자는 정기적으로 [AWS 규정 준수 프로그램](#)의 일환으로 보안 효과를 테스트하고 검증합니다. Amazon Redshift에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 [AWS 규정 준수 프로그램 제공 범위 내 서비스](#)를 참조하세요.
- 클라우드 내 보안 - 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 데이터의 민감도, 조직의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

Amazon Redshift 리소스에 대한 액세스는 4가지 수준에서 제어됩니다.

- 클러스터 관리 - 클러스터의 생성, 구성 및 삭제는 AWS 보안 자격 증명과 연결되어 있는 사용자 또는 계정에 부여된 권한을 통해 제어됩니다. 적합한 권한을 가진 사용자라면 누구나 AWS Management Console, AWS Command Line Interface(CLI) 또는 Amazon Redshift 애플리케이션 프로그램 인터페이스(API)를 사용하여 클러스터를 관리할 수 있습니다. 이러한 액세스는 IAM 정책을 통해 관리됩니다.

Important

Amazon Redshift에는 권한, ID 및 보안 액세스를 관리하기 위한 모범 사례 모음이 포함되어 있습니다. Amazon Redshift를 처음 사용하면서 이러한 내용을 숙지하는 것이 좋습니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#) 섹션을 참조하세요.

- 클러스터 연결 - Amazon Redshift 보안 그룹이 CIDR(Classless Inter-Domain Routing) 형식으로 Amazon Redshift 클러스터 연결 권한을 부여할 AWS 인스턴스를 지정합니다. Amazon Redshift, Amazon EC2 및 Amazon VPC 보안 그룹을 생성하고 클러스터와 연결하는 방법에 대한 자세한 내용은 [Amazon Redshift 보안 그룹](#) 섹션을 참조하세요.
- 데이터베이스 액세스 - 테이블이나 뷰 같은 데이터베이스 객체에 대한 액세스는 Amazon Redshift 데이터베이스의 사용자 계정에서 제어합니다. 사용자는 자신의 계정에 액세스 권한이 부여된 경우에 한해 데이터베이스 리소스에 액세스할 수 있습니다. 이러한 Amazon Redshift 사용자 계정과 관리

권한은 [CREATE USER](#), [CREATE GROUP](#), [GRANT](#) 및 [REVOKE](#) SQL 문을 사용하여 생성합니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [데이터베이스 보안 관리](#)를 참조하세요.

- 임시 데이터 자격 증명과 통합 인증 – CREATE USER 및 ALTER USER 등과 같은 SQL 명령으로 데이터베이스 사용자를 생성하고 관리하는 일 외에도, 사용자 정의 Amazon Redshift JDBC 또는 ODBC 드라이버로 SQL 클라이언트를 구성할 수 있습니다. 이러한 드라이버는 데이터베이스 로그인 프로세스의 일환으로 데이터베이스 사용자 및 임시 암호 생성 프로세스를 관리합니다.

드라이버는 AWS Identity and Access Management(IAM) 인증을 바탕으로 데이터베이스 사용자를 인증합니다. 이미 AWS 밖에서 사용자 자격 증명을 관리하고 있는 경우 SAML 2.0을 준수하는 자격 증명 공급자(IdP)를 통해 Amazon Redshift 리소스에 대한 액세스를 관리할 수 있습니다. IAM 역할을 사용해 IdP 및 AWS를 구성하고 페더레이션 사용자가 임시 데이터베이스 자격 증명을 새로 만들고 Amazon Redshift 데이터베이스에 로그인하도록 허용할 수 있습니다. 자세한 내용은 [IAM 인증을 이용한 데이터베이스 사용자 자격 증명 생성](#) 섹션을 참조하세요.

이 설명서는 Amazon Redshift를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목적에 맞게 Amazon Redshift를 구성하는 방법을 보여줍니다. 또한 Amazon Redshift 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 배우게 됩니다.

주제

- [Amazon Redshift에서 데이터 보호](#)
- [Amazon Redshift의 Identity and Access Management](#)
- [AWS Secrets Manager를 사용하여 Amazon Redshift 관리자 암호를 관리하는 방법](#)
- [Amazon Redshift의 로깅 및 모니터링](#)
- [Amazon Redshift의 규정 준수 확인](#)
- [Amazon Redshift의 복원성](#)
- [Amazon Redshift의 인프라 보안](#)
- [Amazon Redshift의 구성 및 취약성 분석](#)

Amazon Redshift에서 데이터 보호

AWS [공동 책임 모델](#)은 Amazon Redshift의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 사용자는 이 인프라

라에 호스팅되는 콘텐츠에 대한 통제 권한을 유지할 책임이 있습니다. 사용하는 AWS 서비스의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 관한 자세한 내용은 [데이터 프라이버시 FAQ](#)를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 [AWS 공동 책임 모델 및 GDPR](#) 블로그 게시물을 참조하세요.

데이터를 보호하려면 AWS 계정자격 증명을 보호하고 AWS IAM Identity Center 또는 AWS Identity and Access Management(IAM)를 통해 개별 사용자 계정을 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정합니다. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용자 안내서의 [CloudTrail 추적 작업](#)을 참조하세요.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 제어와 함께 사용합니다.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-3 검증된 암호화 모듈이 필요한 경우, FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 [연방 정보 처리 표준\(FIPS\) 140-3](#)을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 Amazon Redshift 또는 기타 AWS 서비스에서 콘솔, API, AWS CLI 또는 AWS를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명을 URL에 포함해서는 안 됩니다.

데이터 암호화

데이터 보호란 전송 중인 데이터(Amazon Redshift 안팎으로 데이터가 이동 중)와 저장된 데이터(Amazon Redshift 데이터 센터의 디스크에 데이터가 저장됨)를 보호하는 것을 말합니다. SSL을 사용하거나 클라이언트 측 암호화를 사용하여 전송 중에도 데이터를 보호할 수 있습니다. Amazon Redshift에서 저장된 데이터를 보호하는 방법은 다음과 같습니다.

- 서버 측 암호화 사용 – 데이터 센터의 디스크에 저장하기 전에 데이터를 암호화하고 객체를 다운로드할 때 이를 복호화하도록 Amazon Redshift에 요청합니다.

- 클라이언트 측 암호화 사용 – 클라이언트 측에서 데이터를 암호화하여 암호화된 데이터를 Amazon Redshift에 업로드할 수 있습니다. 이 경우 사용자가 암호화 프로세스, 암호화 키 및 관련 도구를 관리합니다.

저장 시 암호화

서버 측 암호화는 저장 시의 데이터 암호화에 관한 것으로 즉, Amazon Redshift에서는 필요에 따라 데이터 센터에 기록할 때 데이터를 암호화하고 해당 데이터에 액세스할 때 이를 복호화합니다. 요청을 인증하기만 하면 액세스 권한을 갖게 되며, 데이터의 암호화 여부와 관계없이 액세스 방식에는 차이가 없습니다.

Amazon Redshift는 암호화를 통해 저장된 데이터를 보호합니다. 필요한 경우 고급 암호화 표준 (AES-256)으로 클러스터 내 디스크 상에 저장된 모든 데이터와 Amazon S3의 모든 백업을 보호할 수 있습니다.

Amazon Redshift 리소스의 암호화 및 복호화에 사용되는 키를 관리하려면 [AWS Key Management Service\(AWS KMS\)](#)를 사용합니다. AWS KMS는 클라우드에 맞게 조정된 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 AWS CloudTrail를 지원하므로 키가 적절하게 사용되고 있는지 확인하기 위해 키 사용을 감사할 수 있습니다. AWS KMS 키는 Amazon Redshift 및 지원되는 AWS 서비스와 함께 사용할 수 있습니다. AWS KMS를 지원하는 서비스 목록을 보려면 AWS Key Management Service Developer Guide의 [How AWS Services Use AWS KMS](#)를 참조하세요.

AWS Secrets Manager를 사용하여 프로비저닝된 클러스터 또는 서버리스 네임스페이스의 관리자 암호를 관리하기로 선택한 경우 Amazon Redshift는 AWS Secrets Manager가 보안 인증 정보를 암호화하는 데 사용하는 추가 AWS KMS 키도 허용합니다. 이 추가 키는 AWS Secrets Manager에서 자동으로 생성된 키이거나 사용자가 제공하는 사용자 지정 키일 수 있습니다.

Amazon Redshift 쿼리 편집기 v2는 쿼리 편집기에 입력한 정보를 다음과 같이 안전하게 저장합니다.

- 쿼리 편집기 v2 데이터를 암호화하는 데 사용되는 KMS 키의 Amazon 리소스 이름(ARN).
- 데이터베이스 연결 정보.
- 파일 및 폴더의 이름 및 내용.

Amazon Redshift 쿼리 편집기 v2는 KMS 키 또는 서비스 계정 KMS 키로 블록 수준 암호화를 사용하여 정보를 암호화합니다. Amazon Redshift 데이터의 암호화는 Amazon Redshift 클러스터 속성에 의해 제어됩니다.

주제

- [Amazon Redshift 데이터베이스 암호화](#)

Amazon Redshift 데이터베이스 암호화

Amazon Redshift에서는 저장 데이터를 보호하기 위해 데이터베이스가 기본적으로 암호화됩니다. 데이터베이스 암호화는 클러스터뿐만 아니라 해당 스냅샷에도 적용됩니다.

AWS Key Management Service(AWS KMS) 암호화를 사용하도록 암호화되지 않은 클러스터를 수정할 수 있습니다. 이렇게 하려면 AWS 소유 키나 고객 관리형 키를 사용합니다. AWS KMS 암호화를 사용하도록 클러스터를 수정하면 Amazon Redshift에서 암호화된 새 클러스터로 데이터를 자동으로 마이그레이션합니다. 암호화된 클러스터에서 생성한 스냅샷 역시 암호화됩니다. 클러스터를 수정하고 데이터베이스 암호화(Encrypt database) 옵션을 변경해 암호화된 클러스터를 암호화되지 않은 클러스터로 마이그레이션할 수도 있습니다. 자세한 내용은 [클러스터 암호화 변경](#) 섹션을 참조하세요.

클러스터를 생성한 후에도 기본 암호화된 클러스터를 암호화되지 않은 상태로 변환할 수 있지만 민감한 데이터가 포함된 클러스터는 암호화된 상태로 유지하는 것이 좋습니다. 또한 데이터 통합 관리 지침 또는 규정에 따라 암호화를 사용해야 하는 경우도 있습니다. 예를 들어 미국 신용카드협회 데이터 보안 표준(PCI DSS), 사베인스-옥슬리 법(SOX), 건강보험 이전 및 책임법(HIPAA) 및 기타 관련 규정 등은 특정 유형의 데이터 취급에 대한 지침 준수를 요구합니다.

Amazon Redshift는 암호화 키 계층을 통해 데이터베이스를 암호화합니다. 이 계층에서 최상위 암호화 키는 AWS Key Management Service(AWS KMS) 또는 하드웨어 보안 모듈(HSM)을 사용하여 관리할 수 있습니다. Amazon Redshift가 암호화에 사용하는 프로세스는 키 관리 방식에 따라 다릅니다. Amazon Redshift는 AWS KMS와 자동으로 통합되지만 HSM과는 통합되지 않습니다. HSM을 사용할 때는 클라이언트 및 서버 인증서를 통해 Amazon Redshift와 HSM 사이에 신뢰할 수 있는 연결을 구성해야 합니다.

Important

고객 관리형 KMS 키를 비활성화하면 Amazon Redshift가 프로비저닝된 클러스터 또는 서버리스 네임스페이스의 KMS 키에 액세스하지 못할 수 있습니다. 이러한 경우 Amazon Redshift는 Amazon Redshift 데이터 웨어하우스의 백업을 가져와 14일 동안 `inaccessible-kms-key` 상태로 둡니다. 해당 기간 내에 KMS 키를 복원하면 Amazon Redshift의 액세스가 복원되고 웨어하우스가 정상적으로 작동합니다. KMS 키를 복원하지 않고 14일 기간이 종료될 경우 Amazon Redshift는 데이터 웨어하우스를 삭제합니다. 웨어하우스가 `inaccessible-kms-key` 상태인 동안에는 다음과 같은 특성이 있습니다.

- 데이터 웨어하우스에서 쿼리를 실행할 수 없습니다.

- 데이터 웨어하우스가 데이터 공유의 생산자 웨어하우스인 경우 소비자 웨어하우스에서 그에 대해 데이터 공유 쿼리를 실행할 수 없습니다.
- 교차 리전 스냅샷 복사본을 생성할 수 없습니다.

비활성화된 KMS 키 복원에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [키 활성화 및 비활성화](#)를 참조하세요. 웨어하우스의 KMS 키가 삭제된 경우 `inaccessible-kms-key` 상태 웨어하우스가 삭제되기 전에 백업을 사용하여 새 데이터 웨어하우스를 생성할 수 있습니다.

성능 및 가용성 향상을 위해 암호화 프로세스 개선

RA3 노드를 사용한 암호화

RA3 노드의 암호화 프로세스가 업데이트되어 경험이 훨씬 개선되었습니다. 암호화로 인한 성능 영향을 줄이면서 프로세스 중에 읽기 및 쓰기 쿼리를 모두 실행할 수 있습니다. 또한 암호화가 훨씬 빠르게 완료됩니다. 업데이트된 프로세스 단계에는 복원 작업과 클러스터 메타데이터를 타깃 클러스터로 마이그레이션하는 작업이 포함됩니다. 향상된 경험은 예를 들어 AWS KMS와 같은 암호화 유형에 적용됩니다. 페타바이트 규모의 데이터 볼륨이 있는 경우 작업 시간이 몇 주에서 며칠로 단축되었습니다.

클러스터를 암호화하기 전에 데이터베이스 워크로드를 계속 실행하려는 경우 탄력적 크기 조정이 가능한 노드를 추가하여 성능을 개선하고 프로세스 속도를 높일 수 있습니다. 암호화가 진행 중일 때는 탄력적 크기 조정을 사용할 수 없으므로 암호화하기 전에 사용하세요. 일반적으로 노드를 추가하면 비용이 더 많이 든다는 점을 참고하세요.

다른 노드 유형과의 암호화

DC2 노드로 클러스터를 암호화하면 RA3 노드에서와 같이 쓰기 쿼리를 실행할 수 없습니다. 읽기 쿼리만 실행할 수 있습니다.

RA3 노드를 사용한 암호화에 대한 사용 참고 사항

다음 인사이트와 리소스는 암호화를 준비하고 프로세스를 모니터링하는 데 도움이 됩니다.

- 암호화 시작 후 쿼리 실행 - 암호화가 시작된 후 약 15분 이내에 읽기 및 쓰기가 가능합니다. 전체 암호화 프로세스를 완료하는 데 걸리는 시간은 클러스터 데이터의 양과 워크로드 수준에 따라 달라집니다.
- 암호화에는 시간이 얼마나 걸리나요? - 데이터를 암호화하는 데 걸리는 시간은 실행 중인 워크로드 수, 사용 중인 컴퓨팅 리소스, 노드 수, 노드 유형 등 여러 요인에 따라 달라집니다. 처음에는 테스트

환경에서 암호화를 수행하는 것이 좋습니다. 일반적으로 페타바이트 단위의 데이터 볼륨으로 작업하는 경우 암호화가 완료되는 데 1~3일이 걸릴 수 있습니다.

- 암호화가 완료되었는지 어떻게 알 수 있나요? - 암호화를 활성화한 후 첫 번째 스냅샷이 완료되면 암호화가 완료된 것입니다.
- 암호화 롤백 - 암호화 작업을 롤백해야 하는 경우 가장 좋은 방법은 암호화가 시작되기 전에 만든 가장 최근의 백업에서 복원하는 것입니다. 마지막 백업 이후의 새 업데이트(업데이트/삭제/삽입)은 다시 적용해야 합니다.
- 테이블 복원 수행 - 암호화되지 않은 클러스터에서 암호화된 클러스터로는 테이블을 복원할 수 없습니다.
- 단일 노드 클러스터 암호화 - 단일 노드 클러스터를 암호화하면 성능 제한이 있습니다. 다중 노드 클러스터 암호화보다 시간이 오래 걸립니다.
- 암호화 후 백업 생성 - 클러스터의 데이터를 암호화하면 클러스터가 완전히 암호화될 때까지 백업이 생성되지 않습니다. 소요 시간은 다를 수 있습니다. 백업에 소요되는 시간은 클러스터 크기에 따라 몇 시간에서 며칠이 될 수 있습니다. 암호화가 완료된 후 백업을 생성하기까지 지연이 발생할 수 있습니다.

참고로, 암호화 프로세스 중에 백업 및 복원 작업이 이루어지기 때문에 BACKUP NO에서 생성한 테이블 또는 구체화된 뷰는 보존되지 않습니다. 자세한 내용은 [테이블 생성](#) 또는 [구체화된 뷰 생성](#)을 참조하십시오.

주제

- [AWS KMS로 암호화](#)
- [하드웨어 보안 모듈을 사용한 암호화](#)
- [키 교체 암호화](#)
- [클러스터 암호화 변경](#)
- [HSM 암호화 클러스터로 마이그레이션](#)
- [암호화 키 교체](#)

AWS KMS로 암호화

Amazon Redshift에서 AWS KMS를 선택하여 키를 관리할 때는 4개 티어의 암호화 키 계층으로 구성됩니다. 이들 키는 계층 순서에 따라 루트 키, 클러스터 암호화 키(CEK), 데이터베이스 암호화 키(DEK) 및 데이터 암호화 키입니다.

클러스터를 시작하면 Amazon Redshift가 Amazon Redshift 또는 AWS 계정이 AWS KMS에서 생성했거나 사용 권한이 있는 AWS KMS keys 목록을 반환합니다. 암호화 계층에서 루트 키로 사용할 KMS 키를 선택합니다.

기본적으로 Amazon Redshift는 자동으로 생성된 AWS 소유 키를 Amazon Redshift에서 사용할 AWS 계정의 루트 키로 선택합니다.

기본 키를 사용하지 않으려면 Amazon Redshift에서 클러스터를 시작하기 전에 AWS KMS에서 고객 관리형 KMS 키를 별도로 가지고 있거나 생성해야 합니다. 고객 관리형 키에는 액세스 제어 권한에 대한 생성, 순환, 비활성화 및 정의 기능을 비롯해 데이터 보호에 사용되는 암호화 키에 대한 감사 기능까지 포함되어 유연성을 높여주는 효과가 있습니다. KMS 키 생성에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [키 생성](#)을 참조하세요.

다른 AWS 계정에서 AWS KMS 키를 사용하려면 먼저 키에 대한 사용 권한이 있어야 하며, 또한 Amazon Redshift에서 Amazon 리소스 이름(ARN)을 지정해야 합니다. AWS KMS의 키 액세스에 대한 자세한 내용은 AWS Key Management Service Developer Guide의 [Controlling Access to Your Keys](#)를 참조하세요.

루트 키를 선택하면 Amazon Redshift가 AWS KMS에 데이터 키를 생성한 후 선택한 루트 키를 사용해 데이터 키를 암호화하도록 요청합니다. 이 데이터 키는 Amazon Redshift에서 CEK로 사용됩니다. AWS KMS가 암호화된 CEK를 Amazon Redshift로 내보내면 여기에서 CEK의 암호화 컨텍스트와 KMS 키에 대한 권한이 부여되어 클러스터와 분리되어 있는 네트워크 디스크에 저장됩니다. 암호화된 CEK만 Amazon Redshift로 내보내집니다. KMS 키는 AWS KMS에 남아 있습니다. 또한 Amazon Redshift는 암호화된 CEK를 보안 채널을 통해 클러스터로 전달하고 이를 메모리에 로드합니다. 그런 다음 Amazon Redshift는 AWS KMS를 호출하여 CEK를 복호화한 후 복호화된 CEK를 메모리에 로드합니다. 권한 부여, 암호화 컨텍스트 및 기타 AWS KMS 관련 개념에 대한 자세한 내용은 AWS Key Management Service Developer Guide의 [Concepts](#) 섹션을 참조하세요.

그런 다음 Amazon Redshift가 DEK로 사용할 키를 무작위로 생성하여 클러스터의 메모리에 로드합니다. 복호화된 CEK는 DEK를 암호화하는 데 사용되며, 이후 암호화된 DEK는 클러스터에서 보안 채널을 통해 전송되어 Amazon Redshift에서 내부적으로 클러스터와 분리되어 있는 네트워크 디스크에 저장됩니다. CEK와 마찬가지로 암호화 버전과 복호화 버전의 DEK 모두 클러스터의 메모리에 로드됩니다. 이후 복호화 버전의 DEK는 데이터베이스의 데이터 블록마다 무작위로 생성되는 개별 암호화 키를 암호화하는 데 사용됩니다.

클러스터가 재부팅되면 Amazon Redshift가 내부에 저장된 암호화 버전의 CEK와 DEK를 통해 시작되면서 두 키를 메모리에 다시 로드합니다. 그런 다음 AWS KMS를 호출하여 다시 KMS 키를 사용해 메모리에 로드할 수 있도록 CEK를 복호화합니다. 이후 복호화된 CEK가 다시 DEK를 복호화하는 데 사

용되고, 이렇게 복호화된 DEK는 메모리에 로드되어 필요할 때마다 데이터 블록 키를 암호화하거나 복호화하는 역할을 합니다.

AWS KMS 키를 사용하여 암호화되는 Amazon Redshift 클러스터 생성에 대한 자세한 정보는 [클러스터 생성](#) 섹션을 참조하세요.

AWS KMS로 암호화된 스냅샷을 다른 AWS 리전으로 복사

AWS KMS 키는 AWS 리전마다 고유합니다. Amazon Redshift 스냅샷을 암호화된 소스 클러스터에서 다른 AWS 리전으로 복사하려 하지만 대상의 스냅샷에 자체 AWS KMS 키를 사용하려는 경우, Amazon Redshift가 대상 AWS 리전의 사용자 계정에서 루트 키를 사용할 수 있는 권한 부여를 구성해야 합니다. 이 권한을 부여하면 Amazon Redshift가 대상 AWS 리전의 스냅샷을 암호화할 수 있습니다. 대상의 스냅샷을 AWS 리전 소유 키로 암호화하려는 경우 대상 AWS 리전에서 권한 부여를 구성할 필요가 없습니다. 리전 간 스냅샷 복사에 대한 자세한 내용은 [다른 AWS 리전에 스냅샷 복사](#) 단원을 참조하십시오.

Note

암호화된 클러스터의 스냅샷 복사 기능을 활성화하고 루트 키로 AWS KMS를 사용하는 경우에는 클러스터 이름이 암호화 컨텍스트에 포함되어 클러스터 이름을 변경할 수 없습니다. 이때 클러스터 이름을 변경해야 한다면 원본 AWS 리전의 스냅샷 복사 기능을 사용 중지하여 클러스터 이름을 변경한 후 스냅샷 복사 기능을 다시 구성하여 사용하는 방법이 있습니다.

스냅샷 복사 권한의 구성 프로세스는 다음과 같습니다.

1. 대상 AWS 리전에서 다음과 같은 방법으로 스냅샷 복사 권한을 생성합니다.
 - 사용할 AWS KMS 키가 아직 없으면 생성합니다. AWS KMS 키 생성에 대한 자세한 내용은 AWS Key Management Service Developer Guide의 [Creating Keys](#)를 참조하세요.
 - 스냅샷 복사 권한 이름을 지정합니다. 이름은 AWS 계정이 속한 AWS 리전에서 고유해야 합니다.
 - 권한을 생성할 AWS KMS 키 ID를 지정합니다. 키 ID를 지정하지 않으면 권한이 기본 키에 적용됩니다.
2. 원본 AWS 리전에서 스냅샷 복사 기능을 사용한 후 대상 AWS 리전에서 생성한 스냅샷 복사 권한 이름을 지정합니다.

위의 선행 프로세스는 AWS CLI, Amazon Redshift API 또는 SDK를 사용하여 스냅샷 복사 기능을 사용하는 경우에만 필요합니다. 콘솔을 사용하면 리전 간 스냅샷 복사 기능을 사용할 때 Amazon Redshift

가 권한을 구성하기 위한 적절한 워크플로를 제공합니다. AWS KMS로 암호화된 클러스터에서 콘솔을 사용해 리전 간 스냅샷 복사 기능을 구성하는 방법에 대한 자세한 내용은 [AWS KMS 암호화 클러스터에 대한 리전 간 스냅샷 복사 구성](#) 단원을 참조하십시오.

스냅샷이 대상 AWS 리전에 복사되기 전에 Amazon Redshift는 원본 AWS 리전의 루트 키를 사용하여 스냅샷을 복호화하고 Amazon Redshift가 내부적으로 관리하는 무작위로 생성된 RSA 키를 사용하여 임시로 다시 암호화합니다. 그런 다음 Amazon Redshift는 보안 채널을 통해 대상 AWS 리전으로 스냅샷을 복사하고 내부 관리형 RSA 키를 사용하여 스냅샷을 복호화한 다음 대상 AWS 리전의 루트 키를 사용하여 스냅샷을 다시 암호화합니다.

하드웨어 보안 모듈을 사용한 암호화

키 관리에 AWS KMS를 사용하지 않는 경우에는 Amazon Redshift에서 하드웨어 보안 모듈(HSM)을 키 관리에 사용할 수 있습니다.

Important

HSM 암호화는 DC2 및 RA3 노드 유형에 대해 지원되지 않습니다.

HSM이란 키 생성 및 관리를 직접 제어하는 디바이스를 말합니다. 이 디바이스들은 키 관리를 애플리케이션 및 데이터베이스 레이어와 분리함으로써 보안을 강화하는 역할을 합니다. Amazon Redshift는 키 관리를 위해 AWS CloudHSM Classic을 지원합니다. AWS KMS 대신 HSM을 사용하여 암호화 키를 관리할 때는 암호화 프로세스가 다릅니다.

Important

Amazon Redshift는 AWS CloudHSM Classic만 지원합니다. 새로운 AWS CloudHSM 서비스는 지원되지 않습니다.

AWS CloudHSM 신규 고객은 Classic을 이용할 수 없습니다. 자세한 내용은 [CloudHSM Classic 요금](#)을 참조하세요. AWS CloudHSM Classic은 일부 AWS 리전에서는 사용할 수 없습니다. 사용 가능한 AWS 리전에 대한 자세한 내용은 [AWS 리전 표](#)를 참조하세요.

HSM을 사용하도록 클러스터를 구성하면 Amazon Redshift가 CEK로 사용할 키를 생성하여 저장하라는 요청을 HSM으로 보냅니다. 하지만 AWS KMS와 달리 HSM은 CEK를 Amazon Redshift로 내보내지 않습니다. 대신에 Amazon Redshift가 클러스터에서 무작위로 DEK를 생성한 후 CEK로 암호화할 수 있도록 HSM으로 전송합니다. HSM은 암호화된 DEK를 Amazon Redshift로 반환하며, 여기서 무작위로 생성된 내부 루트 키를 사용하여 추가로 암호화되고 클러스터와 별도의 네트워크에 있는 디스크에

내부적으로 저장됩니다. Amazon Redshift는 또한 DEK의 복호화된 버전을 클러스터의 메모리에 로드하므로 DEK를 사용하여 데이터 블록의 개별 키를 암호화하고 복호화할 수 있습니다.

클러스터가 재부팅되는 경우에는 Amazon Redshift가 내부 루트 키를 사용해 내부적으로 저장되어 있는 이중 암호화 DEK를 복호화하여 내부에 저장되어 있는 DEK를 CEK 암호화 상태로 반환합니다. 그런 다음 CEK 암호화 상태의 DEK가 HSM으로 전송되어 복호화된 후 다시 Amazon Redshift로 보내집니다. 여기에서 개별 데이터 블록 키에 사용할 수 있도록 다시 메모리에 로드됩니다.

Amazon Redshift와 HSM 사이에 신뢰할 수 있는 연결 구성

클러스터 키 관리에 HSM을 사용하는 경우에는 Amazon Redshift와 HSM 사이에 신뢰할 수 있는 네트워크 연결을 구성해야 합니다. 이를 위해서는 클라이언트 및 서버 인증서의 구성이 필요합니다. 신뢰할 수 있는 연결은 암호화 및 복호화 작업 도중 HSM과 Amazon Redshift 사이에 암호화 키를 전송하는 데 사용됩니다.

Amazon Redshift는 무작위로 생성되는 비공개/공개 키 페어에서 퍼블릭 클라이언트 인증서를 생성합니다. 이 두 키는 암호화를 통해 내부에 저장됩니다. 퍼블릭 클라이언트 인증서는 다운로드하여 HSM에서 등록한 후 해당하는 HSM 파티션에 할당합니다.

Amazon Redshift에 HSM IP 주소, HSM 파티션 이름, HSM 파티션 암호 및 내부 루트 키를 사용하여 암호화된 퍼블릭 HSM 서버 인증서를 제공합니다. Amazon Redshift가 구성 프로세스를 완료하고 HSM에 연결할 수 있는지 확인합니다. 이때 연결이 되지 않으면 클러스터가 INCOMPATIBLE_HSM 상태로 바뀌면서 생성되지 않습니다. 이러한 경우에는 불완전한 클러스터를 삭제한 후 다시 시도해야 합니다.

Important

클러스터를 수정하여 다른 HSM 파티션을 사용할 때는 Amazon Redshift가 새로운 파티션에 대한 연결 가능성을 확인하지만 유효한 암호화 키의 존재 유무까지 확인하지는 않습니다. 따라서 새로운 파티션을 사용하기 전에 반드시 암호화 키를 새로운 파티션으로 복제해야 합니다. 클러스터가 다시 시작될 때 Amazon Redshift가 유효한 키를 찾지 못하면 재시작이 중단됩니다. 자세한 내용은 [HSM에 키 복제](#) 단원을 참조하십시오.

초기 구성 이후 Amazon Redshift가 HSM에 연결하지 못하면 이벤트가 기록됩니다. 이러한 이벤트에 대한 자세한 내용은 [Amazon Redshift 이벤트 알림](#)을 참조하세요.

키 교체 암호화

Amazon Redshift에서는 암호화된 클러스터에 대한 암호화 키를 교체할 수 있습니다. 키 교체 프로세스를 시작하면 Amazon Redshift가 지정된 클러스터와 클러스터의 자동 또는 수동 스냅샷에 대해 CEK

를 교체합니다. Amazon Redshift는 지정된 클러스터에 대한 DEK도 교체하지만 스냅샷이 Amazon Simple Storage Service(Amazon S3)에 내부적으로 저장되고 기존 DEK를 사용하여 암호화되는 동안에는 스냅샷에 대한 DEK를 교체할 수 없습니다.

교체가 진행 중일 때는 클러스터가 ROTATING_KEYS 상태로 바뀌고, 교체가 끝나면 클러스터가 AVAILABLE 상태로 돌아옵니다. Amazon Redshift는 키 교체 프로세스 중에 복호화 및 재암호화를 처리합니다.

Note

원본 클러스터가 없는 스냅샷에서는 키를 순환시키며 사용할 수 없습니다. 따라서 클러스터를 삭제하려면 먼저 스냅샷이 키 순환을 이용하는지 확인해야 합니다.

키 순환 프로세스에서는 클러스터를 잠시 사용할 수 없기 때문에 데이터 요건에 따라 필요하거나, 혹은 키의 손상 여부가 의심될 때만 가끔씩 키를 순환시켜야 합니다. 가장 좋은 방법은 저장되는 데이터 유형을 먼저 살펴본 후 이를 기준으로 데이터의 암호화 키 순환 주기를 계획하는 것입니다. 키 순환 주기는 기업의 데이터 보안 정책이나 민감한 데이터 및 규정 준수에 대한 업계 표준에 따라 달라질 수 있습니다. 하지만 보안 요건과 클러스터의 가용성의 밸런스를 맞춰서 계획을 세우는 것이 중요합니다.

키 교체에 대한 자세한 내용은 [암호화 키 교체](#) 섹션을 참조하세요.

클러스터 암호화 변경

AWS 소유 키 또는 고객 관리형 키를 사용하여 AWS Key Management Service(AWS KMS) 암호화를 사용하도록 암호화되지 않은 클러스터를 수정할 수 있습니다. AWS KMS 암호화를 사용하도록 클러스터를 수정하면 Amazon Redshift에서 암호화된 새 클러스터로 데이터를 자동으로 마이그레이션합니다. 또한 AWS Management Console이 아닌 AWS CLI를 통해 클러스터를 수정하여 암호화되지 않은 클러스터를 암호화된 클러스터로 마이그레이션할 수도 있습니다.

마이그레이션 작업 중에는 클러스터를 읽기 전용 모드로 사용할 수 있으며 클러스터 상태는 크기 조정 중으로 표시됩니다.

교차 AWS 리전 스냅샷 복사가 가능하도록 클러스터가 구성된 경우 암호화를 변경하기 전에 이 옵션을 사용 중지해야 합니다. 자세한 내용은 [다른 AWS 리전에 스냅샷 복사](#) 및 [AWS KMS 암호화 클러스터에 대한 리전 간 스냅샷 복사 구성](#) 섹션을 참조하세요. 클러스터를 수정해 하드웨어 보안 모듈(HSM)을 활성화할 수는 없습니다. 대신에 새 HSM 암호화 클러스터를 생성해 데이터를 새 클러스터로 마이그레이션합니다. 자세한 내용은 [HSM 암호화 클러스터로 마이그레이션](#) 섹션을 참조하세요.

Amazon Redshift console

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 암호화를 수정할 클러스터를 선택합니다.
3. 속성(Properties)을 선택합니다.
4. [데이터베이스 구성(Database configurations)] 섹션에서 [편집(Edit)]을 선택하고 [암호화 편집(Edit encryption)]을 선택합니다.
5. 암호화 옵션 중 하나를 선택하고 [변경 사항 저장(Save changes)]을 선택합니다.

AWS CLI

AWS KMS를 사용하도록 암호화되지 않은 클러스터를 수정하려면 `modify-cluster` CLI 명령을 실행하고 다음에 표시된 것처럼 `--encrypted`를 지정합니다. 기본적으로 기본 KMS 키가 사용됩니다. 고객 관리형 키를 지정하려면 `--kms-key-id` 옵션을 포함합니다.

```
aws redshift modify-cluster --cluster-identifier <value> --encrypted --kms-key-id <value>
```

클러스터에서 암호화를 제거하려면 다음 CLI 명령을 실행합니다.

```
aws redshift modify-cluster --cluster-identifier <value> --no-encrypted
```

HSM 암호화 클러스터로 마이그레이션

하드웨어 보안 모듈(HSM)로 암호화된 클러스터로 암호화되지 않은 클러스터를 마이그레이션하려면 암호화된 새 클러스터를 생성해 데이터를 새 클러스터로 이전합니다. 클러스터를 수정하여 HSM 암호화 클러스터로 마이그레이션할 수는 없습니다.

암호화되지 않은 클러스터에서 HSM 암호화 클러스터로 마이그레이션하려면 먼저 데이터를 기존 원본 클러스터에서 언로드합니다. 그런 다음 데이터를 선택한 암호화 설정과 함께 새로운 대상 클러스터에 다시 로드합니다. 암호화된 클러스터 시작에 대한 자세한 내용은 [Amazon Redshift 데이터베이스 암호화](#) 단원을 참조하십시오.

마이그레이션 프로세스에서 원본 클러스터는 마지막 단계까지 읽기 전용 쿼리에 사용할 수 있습니다. 마지막 단계는 대상 및 원본 클러스터의 이름을 변경하는 것입니다. 그러면 모든 트래픽이 새로운 대상

클러스터로 라우팅되도록 엔드포인트가 전환됩니다. 대상 클러스터는 이름 변경 이후 재부팅할 때까지 사용할 수 없습니다. 데이터가 전송되는 동안 원본 클러스터에서 모든 데이터 로드 및 기타 쓰기 작업을 일시 중지합니다.

마이그레이션을 준비하려면

1. Amazon Redshift와 상호 작용하는 종속 시스템(예: 비즈니스 인텔리전스(BI) 도구)을 모두 식별하여 시스템을 추출, 변환, 로드(ETL)합니다.
2. 마이그레이션을 테스트할 검증 쿼리를 식별합니다.

예를 들어 다음 쿼리를 사용하여 사용자 정의 테이블의 수를 확인할 수 있습니다.

```
select count(*)
from pg_table_def
where schemaname != 'pg_catalog';
```

다음 쿼리는 모든 사용자 정의 테이블의 목록과 각 테이블의 행 수를 반환합니다.

```
select "table", tbl_rows
from svv_table_info;
```

3. 마이그레이션에 적합한 시간을 선택합니다. 클러스터 사용량이 가장 낮은 시간대를 찾으려면 CPU 사용량, 데이터베이스 연결 수와 같은 클러스터 지표를 모니터링합니다. 자세한 내용은 [클러스터 성능 데이터 보기](#) 섹션을 참조하세요.
4. 사용되지 않는 테이블을 삭제합니다.

테이블 목록 및 각 테이블이 쿼리된 횟수를 생성하려면 다음 쿼리를 실행합니다.

```
select database,
schema,
table_id,
"table",
round(size::float/(1024*1024)::float,2) as size,
sortkey1,
nvl(s.num_qs,0) num_qs
from svv_table_info t
left join (select tbl,
perm_table_name,
count(distinct query) num_qs
from stl_scan s
where s.userid > 1
```

```
and s.perm_table_name not in ('Internal worktable','S3')
group by tbl,
perm_table_name) s on s.tbl = t.table_id
where t."schema" not in ('pg_internal');
```

5. 새로운 암호화된 클러스터를 시작합니다.

원본 클러스터에 대상 클러스터와 동일한 포트 번호를 사용합니다. 암호화된 클러스터 시작에 대한 자세한 내용은 [Amazon Redshift 데이터베이스 암호화](#) 단원을 참조하십시오.

6. 언로드 및 로드 프로세스를 설정합니다.

클러스터 간 데이터 마이그레이션에 [Amazon Redshift 언로드/복사 유틸리티](#)를 사용할 수 있습니다. 이 유틸리티는 원본 클러스터의 데이터를 Amazon S3 상의 한 위치로 내보냅니다. 데이터는 AWS KMS를 사용하여 암호화됩니다. 그런 다음 이 유틸리티가 자동으로 데이터를 대상으로 가져옵니다. 선택적으로, 마이그레이션 완료 후 유틸리티를 사용하여 Amazon S3를 정리할 수 있습니다.

7. 테스트를 실행하여 프로세스를 확인하고 쓰기 작업을 일시 중지해야 하는 시간을 추정합니다.

언로드 및 로드 작업 도중 데이터 로드 및 기타 쓰기 작업을 일시 중지하여 데이터 일관성을 유지합니다. 가장 대규모 테이블 중 하나를 사용하여 언로드 및 로드 프로세스를 실행하면 시간을 추정하는 데 도움이 됩니다.

8. 스키마, 뷰, 테이블과 같은 데이터베이스 객체를 생성합니다. 간편하게 필요한 데이터 정의 언어 (DDL) 문을 생성하려면 AWS GitHub 리포지토리의 [AdminViews](#) 내 스크립트를 사용할 수 있습니다.

클러스터를 마이그레이션하려면

1. 원본 클러스터에서 모든 ETL 프로세스를 중단합니다.

진행 중인 쓰기 작업이 없는지 확인하려면 Amazon Redshift 관리 콘솔을 사용하여 쓰기 IOPS를 모니터링합니다. 자세한 내용은 [클러스터 성능 데이터 보기](#) 섹션을 참조하세요.

2. 앞서 식별한 검증 쿼리를 실행하여 마이그레이션 전에 암호화된 원본 클러스터에 대한 정보를 수집합니다.
3. (선택 사항) 원본 및 대상 클러스터 모두에서 최대 가용 리소스를 사용하려면 워크로드 관리(WLM) 대기열을 하나 생성합니다. 예를 들어 data_migrate라는 이름의 대기열을 생성하고 메모리 95%, 동시성 4를 사용하여 대기열을 구성합니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [사용자 그룹 및 쿼리 그룹에 따른 쿼리의 대기열 라우팅](#)을 참조하세요.
4. data_migrate 대기열을 사용하여 UnloadCopyUtility를 실행합니다.

Amazon Redshift 콘솔을 사용하여 UNLOAD 및 COPY 프로세서를 모니터링합니다.

5. 검증 쿼리를 다시 실행하여 결과가 원본 클러스터의 결과와 일치하는지 확인합니다.
6. 원본 클러스터 및 대상 클러스터의 이름을 변경하여 엔드포인트를 스왑합니다. 중단을 방지하려면 업무 시간을 피해 이 작업을 수행하십시오.
7. 모든 SQL 클라이언트(예: ETL) 및 보고 도구를 사용하여 대상 클라이언트에 연결할 수 있는지 확인합니다.
8. 암호화되지 않은 원본 클러스터를 종료합니다.

암호화 키 교체

다음 절차를 통해 Amazon Redshift 콘솔을 사용하여 암호화 키를 교체할 수 있습니다.

클러스터의 암호화 키를 교체하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 암호화 키를 업데이트할 클러스터를 선택합니다.
3. 작업에서 Rotate encryption(암호화 교체)을 선택하여 암호화 키 교체 페이지를 표시합니다.
4. 암호화 키 교체 페이지에서 암호화 키 교체를 선택합니다.

전송 중 암호화

전송 중 데이터의 기밀성과 무결성을 보호하도록 환경을 구성할 수 있습니다.

JDBC/ODBC를 통해 Amazon Redshift 클러스터와 SQL 클라이언트 간에 전송 중인 데이터의 암호화에는 다음 사항이 적용됩니다.

- JDBC(Java Database Connectivity) 및 ODBC(Open Database Connectivity) 연결을 통해 SQL 클라이언트 도구에서 Amazon Redshift 클러스터로 연결할 수 있습니다.
- Amazon Redshift는 데이터와 서버 인증서 암호화를 위한 보안 소켓 계층(SSL) 연결을 지원하여 클라이언트에서 연결할 서버 인증서를 검증합니다. 클라이언트는 Amazon Redshift 클러스터의 리더 노드에 연결됩니다. 자세한 내용은 [연결을 위한 보안 옵션 구성](#) 섹션을 참조하세요.
- SSL 연결을 지원하기 위해 Amazon Redshift는 AWS Certificate Manager(ACM)가 발급한 인증서를 생성해 각 클러스터에 설치합니다. 자세한 내용은 [SSL 연결을 위해 ACM 인증서로 이전](#) 섹션을 참조하세요.

- AWS 클라우드 내에서 전송 중인 데이터를 보호하기 위해 Amazon Redshift는 COPY, UNLOAD, 백업 및 복원 작업을 위해 하드웨어 가속 SSL을 사용하여 Amazon S3 또는 Amazon DynamoDB와 통신합니다.

Amazon Redshift 클러스터와 Amazon S3 또는 DynamoDB 간에 전송 중인 데이터의 암호화에는 다음 사항이 적용됩니다.

- Amazon Redshift는 하드웨어 가속화 SSL을 사용하여 COPY, UNLOAD, 백업 및 복원 작업을 위해 Amazon S3 또는 DynamoDB와 통신합니다.
- Redshift Spectrum은 AWS Key Management Service(KMS)에서 관리되는 계정 기본 키를 사용한 Amazon S3 서버 측 암호화(SSE)를 지원합니다.
- Amazon S3 및 AWS KMS로 Amazon Redshift 로드를 암호화할 수 있습니다. 자세한 내용은 [Amazon S3 및 AWS KMS로 Amazon Redshift 로드 암호화](#)를 참조하세요.

AWS CLI, SDK 또는 API 클라이언트와 Amazon Redshift 엔드포인트 간에 전송 중인 데이터의 암호화와 서명에는 다음 사항이 적용됩니다.

- Amazon Redshift는 전송 중인 데이터를 암호화하기 위한 HTTPS 엔드포인트를 제공합니다.
- Amazon Redshift에 대한 API 요청의 무결성을 보호하려면 호출자가 API 호출에 서명해야 합니다. 호출은 서명 버전 4 서명 프로세스(Sigv4)에 따라 X.509 인증서 또는 고객의 AWS 비밀 액세스 키로 서명됩니다. 자세한 정보는 AWS 일반 참조의 [서명 버전 4 서명 프로세스](#)를 참조하세요.
- AWS CLI 또는 AWS SDK 중 하나를 사용하여 AWS에 요청합니다. 이들 도구는 도구를 구성할 때 지정한 액세스 키를 사용하여 사용자의 요청에 자동으로 서명합니다.

Amazon Redshift 클러스터와 Amazon Redshift Query Editor V2 간에 전송 중인 데이터의 암호화에는 다음 사항이 적용됩니다.

- 데이터는 TLS 암호화 채널을 통해 쿼리 편집기 v2와 Amazon Redshift 클러스터 간에 전송됩니다.

Amazon Redshift를 사용한 VPC 암호화 제어

Amazon Redshift는 리전의 VPC 내부 및 VPC 간 모든 트래픽에 대해 전송 중 암호화를 적용하는 데 도움이 되는 보안 기능인 [VPC 암호화 제어](#)를 지원합니다. 이 문서에서는 Amazon Redshift 클러스터 및 서버리스 작업 그룹에서 VPC 암호화 제어를 사용하는 방법을 설명합니다.

VPC 암호화 제어는 중앙 집중식 제어를 제공하여 VPC 내에서 전송 중 암호화를 모니터링하고 적용합니다. 적용 모드에서 활성화하면 모든 네트워크 트래픽이 하드웨어 계층(AWS Nitro 시스템 사용) 또는 애플리케이션 계층(TLS/SSL 사용)에서 암호화됩니다.

Amazon Redshift는 VPC 암호화 제어와 통합되어 의료(HIPAA), 정부(FedRAMP) 및 금융(PCI DSS)과 같은 산업의 규정 준수 요구 사항을 충족하는 데 도움이 됩니다.

Amazon Redshift로 VPC 암호화를 제어하는 방식

VPC 암호화 제어는 다음 두 가지 모드로 작동합니다.

- 모니터링 모드: 트래픽 흐름의 암호화 상태에 대한 가시성을 제공하고 암호화되지 않은 트래픽을 허용하는 리소스를 식별하는 데 도움이 됩니다.
- 적용 모드: VPC 내에서 암호화되지 않은 트래픽을 허용하는 리소스의 생성 또는 사용을 방지합니다. 모든 트래픽은 하드웨어 계층(Nitro 기반 인스턴스) 또는 애플리케이션 계층(TLS/SSL)에서 암호화되어야 합니다.

VPC 암호화 제어 사용 요구 사항

인스턴스 유형 요구 사항

Amazon Redshift를 사용하려면 VPC 암호화 제어를 지원하는 Nitro 기반 인스턴스가 필요합니다. 모든 최신 Redshift 인스턴스 유형은 필요한 암호화 기능을 지원합니다.

SSL/TLS 요구 사항

적용 모드에서 VPC 암호화 제어가 활성화된 경우 `require_ssl` 파라미터를 `true`로 설정해야 하며 비활성화할 수 없습니다. 이렇게 하면 모든 클라이언트 연결이 암호화된 TLS 연결을 사용할 수 있습니다.

VPC 암호화 제어로 마이그레이션

기존 클러스터 및 작업 그룹의 경우

기존 Redshift 클러스터 또는 서버리스 작업 그룹이 포함된 VPC에서는 적용 모드에서 VPC 암호화 제어를 활성화할 수 없습니다. 기존 클러스터 또는 작업 그룹이 있는 경우 암호화 제어를 사용하려면 다음 단계를 참조하세요.

1. 기존 클러스터 또는 네임스페이스의 스냅샷 생성
2. 적용 모드에서 VPC 암호화 제어가 활성화된 새 VPC 생성

3. 다음 작업 중 하나를 사용하여 스냅샷에서 새 VPC로 복원합니다.

- 프로비저닝된 클러스터: `restore-from-cluster-snapshot` 작업 사용
- 서버리스: 작업 그룹에서 `restore-from-snapshot` 작업 사용

암호화 제어가 활성화된 VPC에서 새 클러스터 또는 작업 그룹을 생성할 때는 `require_ssl` 파라미터를 `true`로 설정해야 합니다.

Amazon Redshift를 사용하려면 VPC 암호화 제어를 지원하는 Nitro 기반 인스턴스가 필요합니다. 모든 최신 Redshift 인스턴스 유형은 필요한 암호화 기능을 지원합니다.

SSL/TLS 요구 사항

적용 모드에서 VPC 암호화 제어가 활성화된 경우 `require_ssl` 파라미터를 `true`로 설정해야 하며 비활성화할 수 없습니다. 이렇게 하면 모든 클라이언트 연결이 암호화된 TLS 연결을 사용할 수 있습니다.

고려 사항 및 제한

Amazon Redshift에서 VPC 암호화 제어를 사용할 때는 다음 사항을 고려하세요.

VPC 상태 제한

- VPC 암호화 제어가 `enforce-in-progress` 상태이면 클러스터 및 작업 그룹 생성이 차단됩니다.
- 새 리소스를 생성하려면 먼저 VPC가 `enforce` 모드가 될 때까지 기다려야 합니다.

SSL 구성

- `require_ssl` 파라미터: 암호화가 적용된 VPC에서 생성된 클러스터 및 작업 그룹의 경우 항상 `true`여야 합니다.
- 암호화가 적용된 VPC에서 클러스터 또는 작업 그룹이 생성되면 수명 동안 `require_ssl`을 비활성화할 수 없습니다.

리전 가용성

다음 리전에서는 Amazon Redshift Serverless의 적용 모드에서 이 기능을 사용할 수 없습니다.

- 남아메리카(상파울루)
- 유럽(취리히)

키 관리

키를 통해 데이터를 보호하도록 환경을 구성할 수 있습니다.

- Amazon Redshift는 키 관리를 위해 AWS Key Management Service(AWS KMS)와 자동으로 통합됩니다. AWS KMS는 봉투 암호화를 사용합니다. 자세한 내용은 [봉투 암호화](#)를 참조하십시오.
- AWS KMS에서 암호화 키가 관리될 때 Amazon Redshift는 암호화를 위해 4단계 키 기반 아키텍처를 사용합니다. 이 아키텍처는 임의로 생성된 AES-256 데이터 암호화 키, 데이터베이스 키, 클러스터 키, 루트 키로 구성됩니다. 자세한 내용은 [Amazon Redshift가 AWS KMS를 사용하는 방법](#)을 참조하세요.
- AWS KMS에서 자체 고객 관리형 키를 생성할 수 있습니다. 자세한 내용은 [키 생성](#)을 참조하십시오.
- 새 AWS KMS keys에 대한 자체 키 구성 요소를 가져올 수도 있습니다. 자세한 내용은 [AWS Key Management Service\(AWS KMS\)에서 키 구성 요소 가져오기](#)를 참조하세요.
- Amazon Redshift는 외부 하드웨어 보안 모듈(HSM)에서 암호화 키 관리를 지원합니다. HSM은 온프레미스 모듈일 수도 있고, AWS CloudHSM일 수도 있습니다. HSM을 사용할 때는 클라이언트 및 서버 인증서를 통해 Amazon Redshift와 HSM 사이에 신뢰할 수 있는 연결을 구성해야 합니다. Amazon Redshift는 키 관리를 위해 AWS CloudHSM Classic만 지원합니다. 자세한 내용은 [하드웨어 보안 모듈을 사용한 암호화](#) 섹션을 참조하세요. AWS CloudHSM에 대한 자세한 내용은 [AWS CloudHSM이란 무엇입니까?](#)를 참조하세요.
- 클러스터 암호화를 위해 암호화 키를 순환 사용할 수 있습니다. 자세한 내용은 [키 교체 암호화](#) 섹션을 참조하세요.

데이터 토큰화

토큰화는 데이터 보안을 위해 실제 값을 불투명한 값으로 바꾸는 프로세스입니다. 보안에 민감한 애플리케이션은 토큰화로 개인 식별 정보(PII) 또는 보호 대상 건강 정보(PHI)와 같은 민감한 데이터를 토큰으로 대체하여 보안 위험을 줄입니다. 역토큰화는 적절한 보안 정책을 사용하여 권한이 부여된 사용자에게 실제 값으로 토큰을 되돌립니다.

서드 파티 토큰화 서비스와의 통합을 위해 [AWS Lambda](#)를 사용하여 생성한 Amazon Redshift 사용자 정의 함수(UDF)를 사용할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Lambda 사용자 정의 함수](#)를 참조하세요. 예를 들어 [Protegrity](#)를 참조하세요.

Amazon Redshift는 REST API 또는 사전 정의된 엔드포인트를 통해 액세스되는 토큰화 서버에 토큰화 요청을 전송합니다. 2개 이상의 무료 Lambda 함수가 토큰화 및 역토큰화 요청을 처리합니다. 이 처리를 위해 서드 파티 토큰화 공급자가 제공하는 Lambda 함수를 사용할 수 있습니다. Amazon Redshift에서 Lambda UDF로 등록한 Lambda 함수를 사용할 수도 있습니다.

예를 들어 열에서 토큰화 또는 역토큰화 UDF를 호출하는 쿼리가 제출되었다고 가정합니다. Amazon Redshift 클러스터는 적용 가능한 인수 행을 스푼링하고 이러한 행을 일괄적으로 Lambda 함수에 병렬로 전송합니다. 클라이언트가 액세스할 수 없는 별도의 격리된 네트워크 연결에서 Amazon Redshift 컴퓨팅 노드와 Lambda 간에 데이터가 전송됩니다. Lambda 함수는 데이터를 토큰화 서버 엔드포인트로 전달합니다. 토큰화 서버는 필요에 따라 데이터를 토큰화하거나 역토큰화하여 반환합니다. 그런 다음 Lambda 함수는 필요한 경우 추가 처리를 위해 결과를 Amazon Redshift 클러스터로 전송한 다음 쿼리 결과를 반환합니다.

Amazon Redshift에서 인터넷워크 트래픽 라우팅

알려진 사설 네트워크 경로를 통해 Amazon Redshift에서 트래픽을 라우팅할 수 있습니다. 이 페이지에서는 회사 네트워크에서 트래픽을 라우팅하는 방법과 동일한 AWS 리전의 리소스 간에 트래픽을 라우팅하는 방법을 다룹니다.

Amazon Redshift와 기업 네트워크의 클라이언트 및 애플리케이션 간에 트래픽을 라우팅하는 방법은 다음과 같습니다.

- Virtual Private Cloud(VPC)와 회사 네트워크 간에 프라이빗 연결을 설정합니다. 인터넷을 통해 IPsec VPN 연결을 설정하거나 Direct Connect 연결을 사용하여 프라이빗 물리적 연결을 설정합니다. Direct Connect를 사용하여 온프레미스 네트워크에서 Amazon VPC로 직접 연결되는 프라이빗 가상 인터페이스를 설정함으로써 네트워크와 VPC 간에 고대역폭의 사설 네트워크 연결을 제공할 수 있습니다. 여러 개의 가상 인터페이스를 사용하면 네트워크 격리를 유지하면서 여러 VPC로 프라이빗 연결을 설정할 수 있습니다. 자세한 내용은 [AWS Site-to-Site VPN이란 무엇입니까?](#) 및 [Direct Connect란 무엇입니까?](#)를 참조하세요.

VPC의 Amazon Redshift 클러스터와 같은 AWS 리전의 Amazon S3 버킷 간에 트래픽을 라우팅하는 방법은 다음과 같습니다.

- ETL 로드 또는 언로드로부터 Amazon S3 데이터에 개인적으로 액세스할 수 있도록 Amazon S3 프라이빗 VPC 엔드포인트를 설정합니다. 자세한 내용은 [Amazon S3용 엔드포인트](#)를 참조하세요.
- Amazon Redshift 클러스터에 대한 “Enhanced VPC routing”을 사용하여 목표하는 Amazon S3 VPC 엔드포인트를 지정합니다. Amazon Redshift COPY, UNLOAD 또는 CREATE LIBRARY 명령을 통해 생성된 트래픽은 프라이빗 엔드포인트를 통해 라우팅됩니다. 자세한 내용은 [향상된 VPC 라우팅 커기](#) 섹션을 참조하세요.

Amazon Redshift의 Identity and Access Management

Amazon Redshift에 액세스하려면 AWS가 요청을 인증하는 데 사용할 수 있는 자격 증명이 필요합니다. 이 자격 증명에는 Amazon Redshift 클러스터와 같은 AWS 리소스에 액세스할 수 있는 권한이 있어야 합니다. 다음 섹션에서는 리소스에 액세스할 수 있는지 대상을 제어하여 리소스를 보호할 수 있도록 [AWS Identity and Access Management\(IAM\)](#) 및 Amazon Redshift를 사용하는 방법에 대한 세부 정보를 제공합니다.

- [ID를 사용한 인증](#)
- [액세스 관리](#)

Important

이 주제에는 권한, ID 및 보안 액세스를 관리하기 위한 모범 사례 모음이 포함되어 있습니다. Amazon Redshift에서 IAM을 사용하기 위한 모범 사례를 숙지하는 것이 좋습니다. 여기에는 권한 적용을 위한 IAM 역할 사용이 포함됩니다. 이러한 섹션을 잘 이해하면 보다 안전한 Amazon Redshift 데이터 웨어하우스를 유지 관리하는 데 도움이 됩니다.

ID를 사용한 인증

인증은 ID 자격 증명을 사용하여 AWS에 로그인하는 방식입니다. AWS 계정 루트 사용자인지 IAM 사용자인지, 또는 IAM 역할을 수임하여 인증(에 로그인)받아야 합니다.

AWS IAM Identity Center(IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. 로그인하는 방법에 대한 자세한 내용은 AWS 로그인사용 설명서의 [AWS 계정에 로그인하는 방법](#) 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해 AWS는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 제공합니다. 자세한 내용은 IAM 사용 설명서의 [API 요청용 AWS Signature Version 4](#) 섹션을 참조하세요.

AWS 계정 루트 사용자

AWS 계정을 생성하는 경우에는 모든 AWS 서비스 서비스와 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 루트 사용자라는 단일 로그인 ID로 시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자가 필요한 작업 목록은 IAM 사용자 설명서의 [루트 사용자 자격 증명에 필요한 작업](#)을 참조하세요.

IAM 사용자 및 그룹

[IAM 사용자](#)는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명이 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서에서 [임시 자격 증명을 사용하여 AWS에 액세스하려면 인간 사용자가 ID 제공업체와의 페더레이션을 사용하도록 요구](#)를 참조하세요.

[IAM 그룹](#)은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 [IAM 사용자 사용 사례](#) 섹션을 참조하세요.

IAM 역할

[IAM 역할](#)은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. [사용자에서 IAM 역할\(콘솔\)로 전환](#)하거나 AWS CLI 또는 AWS API 작업을 직접적으로 호출하여 역할을 수입할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [역할 수입 방법](#)을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 [교차 계정 리소스 액세스](#)를 참조하세요.

Note

Redshift의 액세스 전달 세션(FAS)은 12시간 동안만 유효합니다. 이 기간이 지나면 다른 서비스와 통합하기 위해 FAS를 사용하는 모든 연결 세션은 다시 설정해야 합니다.

교차 서비스 혼동된 대리인 방지

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. AWS에서는 교차 서비스 가장으로 인해 혼동된 대리자 문제가 발생할 수 있습니다. 교차 서비스 가장은 한 서비스(직접 호출하는 서비스)가 다른 서비스(직접 호출되는 서비스)를 직접 호출할 때 발생할 수 있습니다. 직접 호출하는 서비스는 다른 고객의 리소스에 대해 액세스 권한이 없는 방식으로 작동하게 권한을 사용하도록 조작될 수 있습니다. 이를 방지하기 위해 AWS에서는 계정의 리소스에 대한 액세스 권한이 부여된 서비스 위탁자를 사용하여 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다. 자세한 내용은 IAM 사용 설명서의 [서비스 간 혼동된 대리자 방지](#)를 참조하세요.

Amazon Redshift가 리소스에 다른 서비스를 제공하는 권한을 제한하려면 리소스 기반 정책에서 [aws:SourceArn](#) 및 [aws:SourceAccount](#) 글로벌 조건 컨텍스트 키를 사용하는 것이 좋습니다. 만

약 `aws:SourceArn` 값에 Amazon S3 버킷 ARN과 같은 계정 ID가 포함되어 있지 않은 경우 권한을 제한하려면 두 전역 조건 컨텍스트 키를 모두 사용해야 합니다. 두 전역 조건 컨텍스트 키와 계정을 포함한 `aws:SourceArn` 값을 모두 사용하는 경우, `aws:SourceAccount` 값 및 `aws:SourceArn` 값의 계정은 동일한 정책 명령문에서 사용할 경우 반드시 동일한 계정 ID를 사용해야 합니다.

다음 예제에서는 Amazon Redshift의 혼동된 대리자 문제를 제한하기 위해 적용할 수 있는 정책을 보여줍니다.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionForRedshift",
    "Effect": "Allow",
    "Principal": {
      "Service": "redshift.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:redshift:us-east-1:123456789012:cluster:my-cluster"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

액세스 관리

요청을 인증하는 데 유효한 자격 증명이 있더라도 권한이 없다면 Amazon Redshift 리소스를 생성하거나 액세스할 수 없습니다. 예를 들어 Amazon Redshift 클러스터를 생성하거나, 스냅샷을 생성하거나, 이벤트 구독을 추가할 수 있는 권한이 있어야 합니다.

다음 섹션에서는 Amazon Redshift에 대한 권한을 관리하는 방법을 설명합니다. 먼저 개요를 읽어보면 도움이 됩니다.

- [Amazon Redshift 리소스에 대한 액세스 권한 관리 개요](#)
- [Amazon Redshift에 대한 자격 증명 기반 정책\(IAM 정책\) 사용](#)

Amazon Redshift 리소스에 대한 액세스 권한 관리 개요

모든 AWS 리소스는 AWS 계정의 소유이고 리소스 생성 또는 액세스 권한은 권한 정책에 따라 결정됩니다. 계정 관리자는 IAM ID(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 설명서의 [IAM 모범 사례](#)를 참조하세요.

권한을 부여하려면 권한을 부여 받을 사용자, 권한 대상이 되는 리소스, 해당 리소스에 허용되는 특정 작업을 결정합니다.

Amazon Redshift 리소스 및 작업

Amazon Redshift에서는 IAM 권한 정책에 사용할 수 있는 서비스별 리소스, 작업 및 조건 컨텍스트 키를 제공합니다.

Amazon Redshift, Amazon Redshift Serverless, Amazon Redshift Data API 및 Amazon Redshift 쿼리 에디터 v2 액세스 권한

[액세스 관리](#)를 설정하면 IAM 자격 증명에 연결할 수 있는 권한 정책(자격 증명 기반 정책)을 작성할 수 있습니다. 자세한 참조 정보는 서비스 권한 부여 참조의 다음 주제를 참조하세요.

- Amazon Redshift의 경우 `redshift:` 접두사를 사용하며 [Amazon Redshift에 사용되는 작업, 리소스 및 조건 키](#)를 참조하세요.
- Amazon Redshift Serverless의 경우 `redshift-serverless:` 접두사를 사용하며 [Amazon Redshift Serverless에 사용되는 작업, 리소스 및 조건 키](#)를 참조하세요.
- Amazon Redshift Data API의 경우 `redshift-data:` 접두사를 사용하며 [Amazon Redshift Data API에 사용되는 작업, 리소스 및 조건 키](#)를 참조하세요.
- Amazon Redshift 쿼리 에디터 v2의 경우 `sqlworkbench:` 접두사를 사용하며 [AWS SQL Workbench\(Amazon Redshift 쿼리 에디터 v2\)에 사용되는 작업, 리소스 및 조건 키](#)를 참조하세요.

쿼리 에디터 v2에는 API 작업에 직접 해당하지 않는 권한 전용 작업이 포함되어 있습니다. 이러한 작업은 서비스 승인 참조에 [permission only]로 나와 있습니다.

서비스 승인 참조에는 IAM 정책에서 사용할 수 있는 API 작업에 대한 정보가 포함되어 있습니다. 또한 권한을 부여할 수 있는 AWS 리소스와 세분화된 액세스 제어에 포함할 수 있는 조건 키가 포함되어 있습니다. 조건에 대한 자세한 내용은 [IAM 정책 조건을 사용하여 세분화된 액세스 제어 구현](#) 섹션을 참조하세요.

정책의 Action 필드에서 작업을 지정하고, Resource 필드에서 리소스 값을 지정하고, Condition 필드에서 조건을 지정합니다. Amazon Redshift에 대한 작업을 지정하려면 `redshift:` 접두사 다음에 API 작업 이름을 사용합니다(예: `redshift:CreateCluster`).

리소스 소유권 이해

리소스 소유자는 리소스를 만든 AWS 계정입니다. 즉, 리소스 소유자는 리소스를 생성하는 요청을 인증하는 AWS 보안 주체 엔터티(루트 계정, IAM 사용자 또는 IAM 역할)의 계정입니다. 다음 예제에서는 이러한 작동 방법을 설명합니다.

- AWS 계정의 루트 계정 자격 증명을 사용하여 DB 클러스터를 생성하는 경우, AWS 계정이 Amazon Redshift 리소스 소유자가 됩니다.
- AWS 계정에서 Amazon Redshift 리소스를 생성할 권한이 있는 IAM 역할을 만드는 경우, 해당 역할을 담당할 수 있는 사람은 누구나 Amazon Redshift 리소스를 생성할 수 있습니다. 이 경우 역할이 속한 AWS 계정이 Amazon Redshift 리소스를 소유합니다.
- AWS 계정에서 IAM 사용자를 생성하고 Amazon Redshift 리소스 생성 권한을 부여하면 해당 사용자는 Amazon Redshift 리소스를 생성할 수 있습니다. 하지만 해당 사용자가 속한 AWS 계정이 Amazon Redshift 리소스를 소유합니다. 대부분의 경우 이 방법은 권장되지 않습니다. IAM 역할을 생성하고 역할에 권한을 연결한 다음 사용자에게 역할을 할당하는 것이 좋습니다.

리소스 액세스 관리

권한 정책은 누가 무엇에 액세스 할 수 있는지를 나타냅니다. 다음 섹션에서는 권한 정책을 만드는 데 사용 가능한 옵션에 대해 설명합니다.

Note

이 섹션에서는 Amazon Redshift의 맥락에서 IAM을 사용하는 방법에 대해 설명하며, IAM 서비스에 대한 자세한 내용은 다루지 않습니다. IAM 설명서 전체 내용은 IAM 사용 설명서의 [IAM이란 무엇입니까?](#)를 참조하세요. IAM 정책 구문과 설명에 대한 자세한 내용은 [IAM User Guide](#)의 AWS IAM policy reference 섹션을 참조하세요.

IAM 자격 증명에 연결된 정책을 자격 증명 기반 정책(IAM 정책)이라 하고, 리소스에 연결된 정책을 리소스 기반 정책이라고 합니다. Amazon Redshift는 자격 증명 기반 정책(IAM 정책)만 지원합니다.

자격 증명 기반 정책(IAM 정책)

정책을 IAM 역할에 연결한 다음 해당 역할을 사용자 또는 그룹에 할당하여 권한을 할당할 수 있습니다. 다음은 사용자에게 AWS 계정의 Amazon Redshift 클러스터를 생성, 삭제, 수정 및 재부팅하도록 허용하는 권한을 포함한 예시 정책입니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowManageClusters",
      "Effect": "Allow",
      "Action": [
        "redshift:CreateCluster",
        "redshift>DeleteCluster",
        "redshift:ModifyCluster",
        "redshift:RebootCluster"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon Redshift에서 자격 증명 기반 정책 사용에 대한 자세한 내용은 [Amazon Redshift에 대한 자격 증명 기반 정책\(IAM 정책\) 사용](#) 섹션을 참조하세요. 사용자, 그룹, 역할 및 권한에 대한 자세한 내용은 IAM User Guide의 [Identities \(users, groups, and roles\)](#)를 참조하세요.

리소스 기반 정책

Amazon S3과 같은 다른 서비스도 리소스 기반 권한 정책을 지원합니다. 예를 들어, 정책을 S3 버킷에 연결하여 해당 버킷에 대한 액세스 권한을 관리할 수 있습니다. Amazon Redshift에서는 리소스 기반 정책을 지원하지 않습니다.

정책 요소 지정: 작업, 효과, 리소스, 보안 주체

각 Amazon Redshift 리소스([Amazon Redshift 리소스 및 작업](#) 참조)에 대해 서비스는 API 작업을 정의합니다([작업](#) 참조). 이러한 API 작업에 대한 권한을 부여하기 위해 Amazon Redshift에서는 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 Redshift 리소스 및 작업](#) 섹션을 참조하세요.
- 작업 – 작업 키워드를 사용하여 허용 또는 거부할 리소스 작업을 식별합니다. 예를 들어 `redshift:DescribeClusters` 권한은 사용자에게 Amazon Redshift `DescribeClusters` 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 묵시적으로 거부됩니다. 다른 정책에서 액세스 권한을 부여하는 경우라도 사용자가 해당 리소스에 액세스할 수 없도록 하기 위해 리소스에 대한 액세스 권한을 명시적으로 거부할 수도 있습니다.
- 보안 주체 – ID 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 Redshift에서는 리소스 기반 정책을 지원하지 않습니다.

IAM 정책 구문과 설명에 대한 자세한 내용은 IAM User Guide의 [AWS IAM policy reference](#)를 참조하세요.

모든 Amazon Redshift API 작업과 해당 작업이 적용되는 리소스를 보여주는 표는 [Amazon Redshift, Amazon Redshift Serverless, Amazon Redshift Data API 및 Amazon Redshift 쿼리 에디터 v2 액세스 권한](#) 섹션을 참조하세요.

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. 액세스 정책 언어에서 조건 지정에 대한 자세한 내용은 IAM User Guide의 [IAM JSON policy elements: Condition](#)을 참조하세요.

권한 정책이 적용되는 조건을 식별하려면 Condition 요소를 IAM 권한 정책에 포함시킵니다. 예를 들어, 사용자가 `redshift>CreateCluster` 작업을 사용하여 DB 인스턴스를 만들 수 있도록 허용하는 정책을 만들 수 있으며 Condition 요소를 추가하여 해당 사용자가 MySQL 데이터베이스 엔진을 사

용하여 DB 인스턴스만 만들 수 있도록 제한할 수 있습니다. 자세한 내용은 [IAM 정책 조건을 사용하여 세분화된 액세스 제어 구현](#)을 참조하세요. 모든 조건 키 값과 이 값이 적용되는 Amazon Redshift 작업 및 리소스를 보여주는 목록은 [Amazon Redshift, Amazon Redshift Serverless, Amazon Redshift Data API 및 Amazon Redshift 쿼리 에디터 v2 액세스 권한](#) 섹션을 참조하세요.

IAM 정책 조건을 사용하여 세분화된 액세스 제어 구현

Amazon Redshift에서는 리소스 태그에 따라 조건 키를 사용하여 리소스에 대한 액세스를 제한할 수 있습니다. 다음은 공통의 Amazon Redshift 조건 키입니다.

조건 키	설명
<code>aws:RequestTag</code>	사용자에게 리소스를 생성할 때마다 태그 키(이름)와 값을 추가하도록 요구합니다. 자세한 내용은 IAM User Guide의 aws:RequestTag 를 참조하세요.
<code>aws:ResourceTag</code>	특정 태그 키 및 값에 따라 리소스에 대한 사용자 액세스를 제한합니다. 자세한 내용은 IAM User Guide의 aws:ResourceTag 를 참조하세요.
<code>aws:TagKeys</code>	이 키를 사용하여 요청의 태그 키를 정책에서 지정한 키와 비교합니다. 자세한 내용은 IAM User Guide의 aws:TagKeys 를 참조하세요.

태그를 생성하는 방법에 대한 자세한 내용은 [Amazon Redshift에서 리소스 태그](#) 섹션을 참조하세요.

`redshift:RequestTag` 및 `redshift:ResourceTag` 조건 키를 지원하는 API 작업 목록은 [Amazon Redshift, Amazon Redshift Serverless, Amazon Redshift Data API 및 Amazon Redshift 쿼리 에디터 v2 액세스 권한](#) 섹션을 참조하세요.

다음 조건 키는 Amazon Redshift `GetClusterCredentials` 작업과 함께 사용할 수 있습니다.

조건 키	설명
<code>redshift:DurationSeconds</code>	기간에 대해 지정할 수 있는 시간(초)을 제한합니다.
<code>redshift:DbName</code>	지정할 수 있는 데이터베이스 이름을 제한합니다.
<code>redshift:DbUser</code>	지정할 수 있는 데이터베이스 사용자 이름을 제한합니다.

예 1: aws:ResourceTag 조건 키를 사용한 액세스 제한

다음 IAM 정책을 사용하여 us-west-2 리전에서 이름이 environment인 태그와 값이 test인 태그가 있는 특정 AWS 계정에 한해 사용자에게 Amazon Redshift 클러스터를 수정하도록 허용할 수 있습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "AllowModifyTestCluster",
    "Effect": "Allow",
    "Action": "redshift:ModifyCluster",
    "Resource": "arn:aws:redshift:us-west-2:123456789012:cluster:*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/environment": "test"
      }
    }
  }
}
```

예 2: aws:RequestTag 조건 키를 사용한 액세스 제한

다음 IAM 정책을 사용하여 클러스터 생성 명령에 이름이 usage인 태그와 값이 production인 태그를 추가한 경우에 한해 사용자에게 Amazon Redshift 클러스터를 생성하도록 허용할 수 있습니다. aws:TagKeys 및 ForAllValues 한정자가 있는 조건은 요청에서 costcenter 및 usage 키만 지정할 수 있도록 지정합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "AllowCreateProductionCluster",
    "Effect": "Allow",
    "Action": [
      "redshift:CreateCluster",
      "redshift:CreateTags"
    ]
  }
}
```

```

    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/usage": "production"
        },
        "ForAllValues:StringEquals": {
            "aws:TagKeys": [
                "costcenter",
                "usage"
            ]
        }
    }
}
}
}

```

Amazon Redshift에 대한 자격 증명 기반 정책(IAM 정책) 사용

이 주제에서는 계정 관리자가 IAM ID(사용자, 그룹, 역할)에 권한 정책을 연결할 수 있는 ID 기반 정책의 예제를 제공합니다.

Important

Amazon Redshift 리소스에 대한 액세스 관리를 위해 제공되는 기본 개념과 옵션 설명에 대한 소개 주제 부분을 먼저 읽어 보는 것이 좋습니다. 자세한 내용은 [Amazon Redshift 리소스에 대한 액세스 권한 관리 개요](#) 섹션을 참조하세요.

다음은 권한 정책의 예입니다. 이 정책은 사용자에게 모든 클러스터를 생성, 삭제, 수정 및 재부팅할 수 있도록 허용한 후 AWS 리전 us-west-2 및 AWS 계정 123456789012에서 클러스터 식별자가 production으로 시작되는 모든 클러스터를 삭제 또는 수정하는 권한을 거부합니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowClusterManagement",

```

```

    "Action": [
        "redshift:CreateCluster",
        "redshift>DeleteCluster",
        "redshift:ModifyCluster",
        "redshift:RebootCluster"
    ],
    "Resource": [
        "*"
    ],
    "Effect": "Allow"
},
{
    "Sid": "DenyDeleteModifyProtected",
    "Action": [
        "redshift>DeleteCluster",
        "redshift:ModifyCluster"
    ],
    "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:cluster:production*"
    ],
    "Effect": "Deny"
}
]
}

```

이 정책에는 두 명령문이 있습니다:

- 첫 번째 문은 사용자에게 클러스터를 생성, 삭제, 수정 및 재부팅할 수 있는 권한을 부여합니다. 여기에서 와일드카드 문자(*)를 Resource 값으로 지정하면 루트 AWS 계정에 속한 모든 Amazon Redshift 리소스에게 정책이 적용됩니다.
- 두 번째 문은 클러스터의 삭제 또는 수정 권한을 거부합니다. 여기에서는 와일드카드 문자(*)가 포함된 Resource 값에 클러스터 Amazon 리소스 이름(ARN)을 지정합니다. 그러면 루트 AWS 계정에서 클러스터 식별자가 production으로 시작하는 모든 Amazon Redshift 클러스터에 이 문이 적용됩니다.

Amazon Redshift에 대한 AWS 관리형 정책

AWS는 AWS에서 생성하고 관리하는 독립형 IAM 정책을 제공하여 많은 일반 사용 사례를 처리합니다. 관리형 정책은 필요한 권한을 사용자가 조사할 필요가 없도록 일반 사용 사례에 필요한 권한을 부여합니다. 자세한 내용은 IAM User Guide의 [AWS managed policies](#)를 참조하세요.

Amazon Redshift API 작업 및 리소스에 대한 권한을 허용하는 고유의 사용자 정의 IAM 정책을 생성할 수도 있습니다. 해당 권한이 필요한 IAM 역할 또는 그룹에 이러한 사용자 지정 정책을 연결할 수 있습니다.

다음 섹션에서는 계정의 사용자에게 연결할 수 있고 Amazon Redshift에 고유한 AWS 관리형 정책입니다.

AWS 관리형 정책에 대한 Amazon Redshift 업데이트

이 서비스가 이러한 변경 내용을 추적하기 시작한 이후부터 Amazon Redshift의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받아보려면 Amazon Redshift 문서 기록 페이지에서 RSS 피드를 구독하세요.

변경	설명	날짜
AmazonRedshiftFederatedAuthorization - 새 정책	Amazon Redshift 페더레이션 권한 부여를 사용하여 쿼리를 실행하기 위한 새로운 간편 사용 정책 정책이 Amazon Redshift에 추가되었습니다.	2025년 11월 21일
AmazonRedshiftServiceLinkedRolePolicy - 기존 정책 업데이트	관리형 정책에 작업 lakeformation:GetDataAccess 의 권한이 추가됩니다. 이를 추가하면 AWS Lake Formation에서 페더레이션된 카탈로그 정보를 가져올 수 있는 권한이 부여됩니다. glue:GetCatalog 및 glue:GetCatalogs 작업에 대한 추가 조건이 관리형 정책에 추가됩니다.	2025년 3월 13일
AmazonRedshiftServiceLinkedRolePolicy - 기존 정책 업데이트	glue:GetCatalog 및 glue:GetCatalogs 작업에 대한 권한이 관리형 정책에 추	2024년 12월 3일

변경	설명	날짜
	가됩니다. 이를 추가하면 AWS Glue에서 카탈로그 정보를 가져올 수 있는 권한이 부여됩니다.	
AmazonRedshiftServiceLinkedRolePolicy - 기존 정책 업데이트	관리형 정책에 작업 <code>servicequotas:GetServiceQuota</code> 의 권한이 추가됩니다. 이렇게 하면 할당량 또는 제한에 액세스할 수 있는 권한이 부여됩니다.	2024년 3월 8일
AmazonRedshiftQueryEditorV2FullAccess - 기존 정책 업데이트	<code>redshift-serverless:ListNamespaces</code> 및 <code>redshift-serverless:ListWorkgroups</code> 작업에 대한 권한이 관리형 정책에 추가됩니다. 추가하면 Amazon Redshift 데이터 웨어하우스의 서버리스 네임스페이스와 서버리스 작업 그룹을 나열할 권한이 부여됩니다.	2024년 2월 21일
AmazonRedshiftQueryEditorV2NoSharing - 기존 정책 업데이트	<code>redshift-serverless:ListNamespaces</code> 및 <code>redshift-serverless:ListWorkgroups</code> 작업에 대한 권한이 관리형 정책에 추가됩니다. 추가하면 Amazon Redshift 데이터 웨어하우스의 서버리스 네임스페이스와 서버리스 작업 그룹을 나열할 권한이 부여됩니다.	2024년 2월 21일

변경	설명	날짜
AmazonRedshiftQueryEditorV2ReadSharing - 기존 정책 업데이트	redshift-serverless:ListNamespaces 및 redshift-serverless:ListWorkgroups 작업에 대한 권한이 관리형 정책에 추가됩니다. 추가하면 Amazon Redshift 데이터 웨어하우스의 서버리스 네임스페이스와 서버리스 작업 그룹을 나열할 권한이 부여됩니다.	2024년 2월 21일
AmazonRedshiftQueryEditorV2ReadWriteSharing - 기존 정책 업데이트	redshift-serverless:ListNamespaces 및 redshift-serverless:ListWorkgroups 작업에 대한 권한이 관리형 정책에 추가됩니다. 추가하면 Amazon Redshift 데이터 웨어하우스의 서버리스 네임스페이스와 서버리스 작업 그룹을 나열할 권한이 부여됩니다.	2024년 2월 21일
AmazonRedshiftReadOnlyAccess - 기존 정책 업데이트	관리형 정책에 작업 redshift:ListRecommendations 의 권한이 추가됩니다. 이렇게 하면 Amazon Redshift Advisor 권장사항을 나열할 수 있는 권한이 부여됩니다.	2024년 2월 7일

변경	설명	날짜
AmazonRedshiftServiceLinkedRolePolicy - 기존 정책 업데이트	ec2:AssignIpv6Addresses 및 ec2:UnassignIpv6Addresses 작업에 대한 권한이 관리형 정책에 추가됩니다. 추가하면 IP 주소를 할당하고 할당 취소할 수 있는 권한이 부여됩니다.	2023년 10월 31일
AmazonRedshiftQueryEditorV2NoSharing - 기존 정책 업데이트	sqlworkbench:GetAutocompletionMetadata 및 sqlworkbench:GetAutocompletionResource 작업에 대한 권한이 관리형 정책에 추가됩니다. 이 기능을 추가하면 쿼리를 편집하는 동안 SQL 자동 완성을 위한 데이터베이스 정보를 생성하고 검색할 수 있는 권한이 부여됩니다.	2023년 8월 16일
AmazonRedshiftQueryEditorV2ReadSharing - 기존 정책 업데이트	sqlworkbench:GetAutocompletionMetadata 및 sqlworkbench:GetAutocompletionResource 작업에 대한 권한이 관리형 정책에 추가됩니다. 이 기능을 추가하면 쿼리를 편집하는 동안 SQL 자동 완성을 위한 데이터베이스 정보를 생성하고 검색할 수 있는 권한이 부여됩니다.	2023년 8월 16일

변경	설명	날짜
AmazonRedshiftQueryEditorV2ReadWriteSharing - 기존 정책 업데이트	sqlworkbench:GetAutocompletionMetadata 및 sqlworkbench:GetAutocompletionResource 작업에 대한 권한이 관리형 정책에 추가됩니다. 이 기능을 추가하면 쿼리를 편집하는 동안 SQL 자동 완성을 위한 데이터베이스 정보를 생성하고 검색할 수 있는 권한이 부여됩니다.	2023년 8월 16일
AmazonRedshiftServiceLinkedRolePolicy - 기존 정책 업데이트	보안 암호 생성 및 관리 작업을 위한 AWS Secrets Manager 작업 권한이 관리형 정책에 추가됩니다. 추가된 권한은 다음과 같습니다. • secretsmanager:GetRandomPassword • secretsmanager:DescribeSecret • secretsmanager:PutSecretValue • secretsmanager:UpdateSecret • secretsmanager:UpdateSecretVersionStage • secretsmanager:RotateSecret • secretsmanager>DeleteSecret	2023년 8월 14일

변경	설명	날짜
AmazonRedshiftServiceLinkedRolePolicy - 기존 정책 업데이트	Amazon EC2에서 보안 그룹 및 라우팅 규칙을 생성하고 관리하기 위한 작업 권한이 관리형 정책에서 제거됩니다. 이러한 권한은 서브넷 및 VPC 생성과 관련이 있습니다. 제거되는 권한은 다음과 같습니다. • <code>ec2:AuthorizeSecurityGroupEgress</code> • <code>ec2:AuthorizeSecurityGroupIngress</code> • <code>ec2:UpdateSecurityGroupRuleDescriptionsEgress</code> • <code>ec2:ReplaceRouteTableAssociation</code> • <code>ec2:CreateRouteTable</code> • <code>ec2:AttachInternetGateway</code> • <code>ec2:UpdateSecurityGroupRuleDescriptionsIngress</code> • <code>ec2:AssociateRouteTable</code> • <code>ec2:RevokeSecurityGroupIngress</code> • <code>ec2:CreateRoute</code> • <code>ec2:CreateSecurityGroup</code>	2023년 5월 8일

변경	설명	날짜
	• ec2:RevokeSecurityGroupEgress • ec2:ModifyVpcAttribute • ec2:CreateSubnet • ec2:CreateInternetGateway • ec2:CreateVpc 이들은 Purpose:RedshiftMigrateToVpc 리소스 태그와 연결되었습니다. 이 태그는 Amazon EC2 Classic에서 Amazon EC2 VPC로 마이그레이션하는 작업에 대한 권한 범위를 제한했습니다. 리소스 태그에 대한 자세한 내용은 태그를 사용한 AWS 리소스 액세스 제어를 참조하세요.	
AmazonRedshiftDataFullAccess - 기존 정책 업데이트	관리형 정책에 작업 redshift:GetClusterCredentialsWithIAM 의 권한이 추가됩니다. 추가를 통해 지정된 AWS 계정에서 Amazon Redshift 데이터베이스에 액세스하기 위한 임시 강화 보안 인증 정보를 가져올 권한이 부여됩니다.	2023년 4월 7일

변경	설명	날짜
AmazonRedshiftServiceLinkedRolePolicy - 기존 정책 업데이트	Amazon EC2에서 보안 그룹 규칙을 생성하고 관리하기 위한 작업 권한이 관리형 정책에 추가됩니다. 이러한 보안 그룹 및 규칙은 특히 Amazon Redshift <code>aws:RequestTag/Redshift</code> 리소스 태그와 관련이 있습니다. 이렇게 하면 권한 범위가 특정 Amazon Redshift 리소스로 제한됩니다.	2023년 4월 6일
AmazonRedshiftQueryEditorV2NoSharing - 기존 정책 업데이트	관리형 정책에 작업 <code>sqlworkbench:GetSchemaInference</code> 의 권한이 추가됩니다. 추가를 통해 파일에서 추론된 열 및 데이터 유형을 가져올 수 있는 권한이 부여됩니다.	2023년 3월 21일
AmazonRedshiftQueryEditorV2ReadSharing - 기존 정책 업데이트	관리형 정책에 작업 <code>sqlworkbench:GetSchemaInference</code> 의 권한이 추가됩니다. 추가를 통해 파일에서 추론된 열 및 데이터 유형을 가져올 수 있는 권한이 부여됩니다.	2023년 3월 21일
AmazonRedshiftQueryEditorV2ReadWriteSharing - 기존 정책 업데이트	관리형 정책에 작업 <code>sqlworkbench:GetSchemaInference</code> 의 권한이 추가됩니다. 추가를 통해 파일에서 추론된 열 및 데이터 유형을 가져올 수 있는 권한이 부여됩니다.	2023년 3월 21일

변경	설명	날짜
AmazonRedshiftQueryEditorV2NoSharing - 기존 정책 업데이트	관리형 정책에 작업 sqlworkbench:AssociateNotebookWithTab 의 권한이 추가됩니다. 이 권한을 추가하면 사용자의 자체 노트북에 연결되는 탭을 만들고 업데이트할 수 있는 권한이 부여됩니다.	2023년 2월 2일
AmazonRedshiftQueryEditorV2ReadSharing - 기존 정책 업데이트	관리형 정책에 작업 sqlworkbench:AssociateNotebookWithTab 의 권한이 추가됩니다. 이 권한을 추가하면 사용자의 자체 노트북이나 공유하는 노트북에 연결되는 탭을 만들고 업데이트할 수 있는 권한이 부여됩니다.	2023년 2월 2일
AmazonRedshiftQueryEditorV2ReadWriteSharing - 기존 정책 업데이트	관리형 정책에 작업 sqlworkbench:AssociateNotebookWithTab 의 권한이 추가됩니다. 이 권한을 추가하면 사용자의 자체 노트북이나 공유하는 노트북에 연결되는 탭을 만들고 업데이트할 수 있는 권한이 부여됩니다.	2023년 2월 2일

변경	설명	날짜
AmazonRedshiftQueryEditorV2NoSharing - 기존 정책 업데이트	노트북 사용 권한을 부여하기 위해 Amazon Redshift는 다음 작업에 대한 권한을 추가했습니다. • sqlworkbench:ListNotebooks • sqlworkbench:CreateNotebook • sqlworkbench:DuplicateNotebook • sqlworkbench:CreateNotebookFromVersion • sqlworkbench:ImportNotebook • sqlworkbench:GetNotebook • sqlworkbench:UpdateNotebook • sqlworkbench>DeleteNotebook • sqlworkbench:CreateNotebookCell • sqlworkbench>DeleteNotebookCell • sqlworkbench:UpdateNotebookCellContent • sqlworkbench:UpdateNotebookCellLayout	2022년 10월 17일

변경	설명	날짜
	• sqlworkbench:BatchGetNotebookCell • sqlworkbench:ListNotebookVersions • sqlworkbench:CreateNotebookVersion • sqlworkbench:GetNotebookVersion • sqlworkbench>DeleteNotebookVersion • sqlworkbench:RestoreNotebookVersion • sqlworkbench:ExportNotebook	

변경	설명	날짜
AmazonRedshiftQueryEditorV2ReadSharing - 기존 정책 업데이트	노트북 사용 권한을 부여하기 위해 Amazon Redshift는 다음 작업에 대한 권한을 추가했습니다. • sqlworkbench:ListNotebooks • sqlworkbench:CreateNotebook • sqlworkbench:DuplicateNotebook • sqlworkbench:CreateNotebookFromVersion • sqlworkbench:ImportNotebook • sqlworkbench:GetNotebook • sqlworkbench:UpdateNotebook • sqlworkbench>DeleteNotebook • sqlworkbench:CreateNotebookCell • sqlworkbench>DeleteNotebookCell • sqlworkbench:UpdateNotebookCellContent • sqlworkbench:UpdateNotebookCellLayout	2022년 10월 17일

변경	설명	날짜
	• sqlworkbench:BatchGetNotebookCell • sqlworkbench:ListNotebookVersions • sqlworkbench:CreateNotebookVersion • sqlworkbench:GetNotebookVersion • sqlworkbench>DeleteNotebookVersion • sqlworkbench:RestoreNotebookVersion • sqlworkbench:ExportNotebook	

변경	설명	날짜
AmazonRedshiftQueryEditorV2ReadWriteSharing - 기존 정책 업데이트	노트북 사용 권한을 부여하기 위해 Amazon Redshift는 다음 작업에 대한 권한을 추가했습니다. • sqlworkbench:ListNotebooks • sqlworkbench:CreateNotebook • sqlworkbench:DuplicateNotebook • sqlworkbench:CreateNotebookFromVersion • sqlworkbench:ImportNotebook • sqlworkbench:GetNotebook • sqlworkbench:UpdateNotebook • sqlworkbench>DeleteNotebook • sqlworkbench:CreateNotebookCell • sqlworkbench>DeleteNotebookCell • sqlworkbench:UpdateNotebookCellContent • sqlworkbench:UpdateNotebookCellLayout	2022년 10월 17일

변경	설명	날짜
	• sqlworkbench:BatchGetNotebookCell • sqlworkbench:ListNotebookVersions • sqlworkbench:CreateNotebookVersion • sqlworkbench:GetNotebookVersion • sqlworkbench>DeleteNotebookVersion • sqlworkbench:RestoreNotebookVersion • sqlworkbench:ExportNotebook	
AmazonRedshiftServiceLinkedRolePolicy - 기존 정책 업데이트	Amazon Redshift에서 CloudWatch에 지표를 게시할 수 있도록 네임스페이스 AWS/Redshift 가 추가되었습니다.	2022년 9월 7일
AmazonRedshiftQueryEditorV2NoSharing - 기존 정책 업데이트	Amazon Redshift가 작업 sqlworkbench:ListQueryExecutionHistory 및sqlworkbench:GetQueryExecutionHistory 에 대한 권한을 추가했습니다. 쿼리 기록을 볼 수 있는 권한을 부여합니다.	2022년 8월 30일

변경	설명	날짜
AmazonRedshiftQueryEditorV2ReadSharing - 기존 정책 업데이트	Amazon Redshift가 작업 sqlworkbench:ListQueryExecutionHistory 및sqlworkbench:GetQueryExecutionHistory 에 대한 권한을 추가했습니다. 쿼리 기록을 볼 수 있는 권한을 부여합니다.	2022년 8월 30일
AmazonRedshiftQueryEditorV2ReadWriteSharing - 기존 정책 업데이트	Amazon Redshift가 작업 sqlworkbench:ListQueryExecutionHistory 및sqlworkbench:GetQueryExecutionHistory 에 대한 권한을 추가했습니다. 쿼리 기록을 볼 수 있는 권한을 부여합니다.	2022년 8월 30일
AmazonRedshiftFullAccess - 기존 정책 업데이트	Amazon Redshift Serverless에 대한 권한을 기존 AmazonRedshiftFullAccess 관리형 정책에 추가했습니다.	2022년 7월 22일

변경	설명	날짜
AmazonRedshiftDataFullAccess - 기존 정책 업데이트	Amazon Redshift에서 태그 키 RedshiftDataFullAccess 와 태그 값으로 태깅된 리소스에 대한 액세스 권한을 부여하기 위해 태그 aws:ResourceTag/RedshiftDataFullAccess 권한의 redshift-serverless:GetCredentials 기본 범위 지정 조건을 StringEquals 에서 StringLike 로 업데이트했습니다.	2022년 7월 11일
AmazonRedshiftDataFullAccess - 기존 정책 업데이트	Amazon Redshift에서 Amazon Redshift Serverless에 대한 임시 보안 인증을 위해 redshift-serverless:GetCredentials 를 허용하도록 새 권한을 추가했습니다.	2022년 7월 8일
AmazonRedshiftQueryEditorV2NoSharing - 기존 정책 업데이트	Amazon Redshift가 sqlworkbench:GetAccountSettings 작업에 대한 권한을 추가했습니다. 계정 설정을 가져오는 권한을 부여합니다.	2022년 6월 15일
AmazonRedshiftQueryEditorV2ReadSharing - 기존 정책 업데이트	Amazon Redshift가 sqlworkbench:GetAccountSettings 작업에 대한 권한을 추가했습니다. 계정 설정을 가져오는 권한을 부여합니다.	2022년 6월 15일

변경	설명	날짜
AmazonRedshiftQueryEditorV2ReadWriteSharing - 기존 정책 업데이트	Amazon Redshift가 <code>sqlworkbench:GetAccountSettings</code> 작업에 대한 권한을 추가했습니다. 계정 설정을 가져오는 권한을 부여합니다.	2022년 6월 15일
AmazonRedshiftServiceLinkedRolePolicy - 기존 정책 업데이트	새로운 Amazon Redshift Serverless 엔드포인트에 대한 퍼블릭 액세스를 사용하기 위해 Amazon Redshift는 탄력적 IP 주소를 고객 계정에 있는 VPC 엔드포인트의 탄력적 네트워크 인터페이스에 할당하고 연결합니다. 이 작업은 서비스 연결 역할을 통해 제공된 권한을 사용하여 수행됩니다. 이 사용 사례를 지원하기 위해 탄력적 IP 주소를 할당 및 해제하는 작업이 Amazon Redshift Serverless 서비스 연결 역할에 추가되었습니다.	2022년 5월 26일
AmazonRedshiftQueryEditorV2FullAccess - 기존 정책 업데이트	작업 <code>sqlworkbench:ListTaggedResources</code> 에 대한 권한. Amazon Redshift 쿼리 편집기 v2 리소스로 범위가 지정되었습니다. 이 정책 업데이트에서는 쿼리 편집기 v2를 통해서만 <code>tag:GetResources</code> 를 호출할 수 있는 권한을 부여합니다.	2022년 2월 22일

변경	설명	날짜
AmazonRedshiftQueryEditorV2NoSharing - 기존 정책 업데이트	작업 <code>sqlworkbench:ListTaggedResources</code> 에 대한 권한. Amazon Redshift 쿼리 편집기 v2 리소스로 범위가 지정되었습니다. 이 정책 업데이트에서는 쿼리 편집기 v2를 통해서만 <code>tag:GetResources</code> 를 호출할 수 있는 권한을 부여합니다.	2022년 2월 22일
AmazonRedshiftQueryEditorV2ReadSharing - 기존 정책 업데이트	작업 <code>sqlworkbench:ListTaggedResources</code> 에 대한 권한. Amazon Redshift 쿼리 편집기 v2 리소스로 범위가 지정되었습니다. 이 정책 업데이트에서는 쿼리 편집기 v2를 통해서만 <code>tag:GetResources</code> 를 호출할 수 있는 권한을 부여합니다.	2022년 2월 22일
AmazonRedshiftQueryEditorV2ReadWriteSharing - 기존 정책 업데이트	작업 <code>sqlworkbench:ListTaggedResources</code> 에 대한 권한. Amazon Redshift 쿼리 편집기 v2 리소스로 범위가 지정되었습니다. 이 정책 업데이트에서는 쿼리 편집기 v2를 통해서만 <code>tag:GetResources</code> 를 호출할 수 있는 권한을 부여합니다.	2022년 2월 22일

변경	설명	날짜
AmazonRedshiftQueryEditorV2ReadSharing - 기존 정책 업데이트	관리형 정책에 작업 <code>sqlworkbench:AssociateQueryWithTab</code> 의 권한이 추가됩니다. 이것을 추가하면 고객이 공유되는 쿼리에 연결된 편집기 탭을 만들 수 있습니다.	2022년 2월 22일
AmazonRedshiftServiceLinkedRolePolicy - 기존 정책 업데이트	Amazon Redshift는 Amazon Redshift 네트워크 및 VPC 리소스를 관리할 수 있도록 새로운 작업에 대한 권한을 추가했습니다.	2021년 11월 22일
AmazonRedshiftAllCommandsFullAccess - 새 정책	Amazon Redshift는 Amazon Redshift 콘솔에서 생성된 IAM 역할을 사용하도록 허용하고 클러스터가 Amazon S3, UNLOAD, CREATE EXTERNAL SCHEMA, CREATE EXTERNAL FUNCTION, CREATE MODEL 또는 CREATE LIBRARY 명령에서 COPY를 실행하기 위한 기본값으로 설정할 수 있도록 새 정책을 추가했습니다.	2021년 11월 18일
AmazonRedshiftServiceLinkedRolePolicy - 기존 정책 업데이트	Amazon Redshift는 감사 로그 내보내기를 포함하여 Amazon Redshift CloudWatch 로그 그룹 및 로그 스트림을 관리할 수 있도록 새로운 작업에 대한 권한을 추가했습니다.	2021년 11월 15일

변경	설명	날짜
AmazonRedshiftFullAccess - 기존 정책 업데이트	Amazon Redshift는 모델 설명, DynamoDB, Redshift Spectrum 및 Amazon RDS 페더레이션을 허용하기 위한 새로운 권한을 추가했습니다.	2021년 10월 7일
AmazonRedshiftQueryEditorV2FullAccess - 새 정책	Amazon Redshift는 Amazon Redshift 쿼리 편집기 v2에 대한 전체 액세스를 허용하기 위한 새로운 정책을 추가했습니다.	2021년 9월 24일
AmazonRedshiftQueryEditorV2NoSharing - 새 정책	Amazon Redshift는 리소스를 공유하지 않고 Amazon Redshift 쿼리 편집기 v2 사용을 허용하기 위한 새로운 정책을 추가했습니다.	2021년 9월 24일
AmazonRedshiftQueryEditorV2ReadSharing - 새 정책	Amazon Redshift는 Amazon Redshift 쿼리 편집기 v2 내에서 읽기 공유를 허용하기 위한 새로운 정책을 추가했습니다.	2021년 9월 24일
AmazonRedshiftQueryEditorV2ReadWriteSharing - 새 정책	Amazon Redshift는 Amazon Redshift 쿼리 편집기 v2 내에서 읽기 및 업데이트 공유를 허용하기 위한 새로운 정책을 추가했습니다.	2021년 9월 24일
AmazonRedshiftFullAccess - 기존 정책 업데이트	Amazon Redshift는 <code>sagemaker:*Job*</code> 를 허용하는 새로운 권한을 추가했습니다.	2021년 8월 18일

변경	설명	날짜
AmazonRedshiftDataFullAccess - 기존 정책 업데이트	Amazon Redshift는 AuthorizeDataShare 를 허용하는 새로운 권한을 추가했습니다.	2021년 8월 12일
AmazonRedshiftDataFullAccess - 기존 정책 업데이트	Amazon Redshift는 BatchExecuteStatement 를 허용하는 새로운 권한을 추가했습니다.	2021년 7월 27일
Amazon Redshift에서 변경 사항 추적 시작	Amazon Redshift가 AWS 관리형 정책에 대한 변경 내용을 추적했습니다.	2021년 7월 27일

AmazonRedshiftReadOnlyAccess

AWS 계정의 모든 Amazon Redshift 리소스에 대한 읽기 전용 액세스 권한을 부여합니다.

[AmazonRedshiftReadOnlyAccess](#) 정책은 IAM 콘솔 및 AWS 관리형 정책 참조 안내서의 [AmazonRedshiftReadOnlyAccess](#)에서 찾을 수 있습니다.

AmazonRedshiftFullAccess

AWS 계정의 모든 Amazon Redshift 리소스에 대한 전체 액세스 권한을 부여합니다. 또한 이 정책은 모든 Amazon Redshift Serverless 리소스에 대한 모든 액세스 권한을 부여합니다.

[AmazonRedshiftFullAccess](#) 정책은 IAM 콘솔 및 AWS 관리형 정책 참조 안내서의 [AmazonRedshiftFullAccess](#)에서 찾을 수 있습니다.

AmazonRedshiftQueryEditor

Amazon Redshift 콘솔의 쿼리 편집기에 대한 전체 액세스 권한을 부여합니다.

[AmazonRedshiftQueryEditor](#) 정책은 IAM 콘솔 및 AWS 관리형 정책 참조 안내서의 [AmazonRedshiftQueryEditor](#)에서 찾을 수 있습니다.

AmazonRedshiftDataFullAccess

AWS 계정의 Amazon Redshift Data API 작업 및 리소스에 전체 액세스 권한을 부여합니다.

[AmazonRedshiftDataFullAccess](#) 정책은 IAM 콘솔 및 AWS 관리형 정책 참조 안내서의 [AmazonRedshiftDataFullAccess](#)에서 찾을 수 있습니다.

AmazonRedshiftQueryEditorV2FullAccess

Amazon Redshift 쿼리 편집기 v2 작업 및 리소스에 대한 전체 액세스 권한을 부여합니다. 이 정책은 다른 필수 서비스에 대한 액세스 권한도 부여합니다.

[AmazonRedshiftQueryEditorV2FullAccess](#) 정책은 IAM 콘솔 및 AWS 관리형 정책 참조 안내서의 [AmazonRedshiftQueryEditorV2FullAccess](#)에서 찾을 수 있습니다.

AmazonRedshiftQueryEditorV2NoSharing

리소스를 공유하지 않고 Amazon Redshift 쿼리 편집기 v2 관련 작업을 수행할 수 있는 권한을 부여합니다. 이 정책은 다른 필수 서비스에 대한 액세스 권한도 부여합니다. 이 정책을 사용하는 보안 주체는 리소스(예: 쿼리)에 태그를 지정하여 동일한 AWS 계정의 다른 보안 주체와 리소스를 공유할 수 없습니다.

[AmazonRedshiftQueryEditorV2NoSharing](#) 정책은 IAM 콘솔 및 AWS 관리형 정책 참조 안내서의 [AmazonRedshiftQueryEditorV2NoSharing](#)에서 찾을 수 있습니다.

AmazonRedshiftQueryEditorV2ReadSharing

제한된 리소스 공유로 Amazon Redshift 쿼리 편집기 v2 관련 작업을 수행할 수 있는 권한을 부여합니다. 이 정책은 다른 필수 서비스에 대한 액세스 권한도 부여합니다. 이 정책을 사용하는 보안 주체는 리소스(예: 쿼리)에 태그를 지정하여 동일한 AWS 계정의 다른 보안 주체와 리소스를 공유할 수 없습니다. 부여된 보안 주체는 팀과 공유된 리소스를 읽을 수 있지만 업데이트할 수는 없습니다.

[AmazonRedshiftQueryEditorV2ReadSharing](#) 정책은 IAM 콘솔 및 AWS 관리형 정책 참조 안내서의 [AmazonRedshiftQueryEditorV2ReadSharing](#)에서 찾을 수 있습니다.

AmazonRedshiftQueryEditorV2ReadWriteSharing

리소스 공유와 함께 Amazon Redshift 쿼리 편집기 v2 관련 작업을 수행할 수 있는 권한을 부여합니다. 이 정책은 다른 필수 서비스에 대한 액세스 권한도 부여합니다. 이 정책을 사용하는 보안 주체는 리소

스(예: 쿼리)에 태그를 지정하여 동일한 AWS 계정의 다른 보안 주체와 리소스를 공유할 수 없습니다. 부여된 보안 주체는 팀과 공유하는 리소스를 읽고 업데이트할 수 있습니다.

[AmazonRedshiftQueryEditorV2ReadWriteSharing](#) 정책은 IAM 콘솔 및 AWS 관리형 정책 참조 안내서의 [AmazonRedshiftQueryEditorV2ReadWriteSharing](#)에서 찾을 수 있습니다.

AmazonRedshiftServiceLinkedRolePolicy

AmazonRedshiftServiceLinkedRolePolicy를 IAM 엔터티에 연결할 수 없습니다. 이 정책은 Amazon Redshift가 계정 리소스에 액세스할 수 있도록 허용하는 서비스 연결 역할에 연결됩니다. 자세한 내용은 [Amazon Redshift에 대해 서비스 연결 역할 사용](#)을 참조하세요.

[AmazonRedshiftServiceLinkedRolePolicy](#) 정책은 IAM 콘솔 및 AWS 관리형 정책 참조 안내서의 [AmazonRedshiftServiceLinkedRolePolicy](#)에서 찾을 수 있습니다.

AmazonRedshiftAllCommandsFullAccess

Amazon Redshift 콘솔에서 생성된 IAM 역할을 사용하고 클러스터가 Amazon S3, UNLOAD, CREATE EXTERNAL SCHEMA, CREATE EXTERNAL FUNCTION 및 CREATE MODEL 명령에서 COPY를 실행하기 위한 기본값으로 설정할 수 있는 권한을 부여합니다. 이 정책은 Amazon S3, CloudWatch Logs, Amazon SageMaker AI 또는 AWS Glue와 같은 관련 서비스에 대해 SELECT 문을 실행할 수 있는 권한도 부여합니다.

[AmazonRedshiftAllCommandsFullAccess](#) 정책은 IAM 콘솔 및 AWS 관리형 정책 참조 안내서의 [AmazonRedshiftAllCommandsFullAccess](#)에서 찾을 수 있습니다.

AmazonRedshiftFederatedAuthorization

이 정책은 Amazon Redshift 페더레이션 권한으로 Glue 데이터 카탈로그 데이터베이스에 대해 쿼리를 실행하는 데 필요한 IAM 작업을 통합합니다. 이러한 쿼리는 AWS Glue를 통과하므로 객체를 검색하려면 카탈로그 객체에 대한 작업 가져오기가 필요하고 객체를 수정하려면 생성, 업데이트, 이름 바꾸기 및 삭제 작업이 필요합니다. 리소스는 Amazon Redshift에서 관리하므로 보안 주체는 쿼리를 완료하는 데 Redshift 권한도 필요합니다. glue:FederateAuthorization 작업을 통해 AWS Glue는 카탈로그 객체에 대한 권한 부여 결정을 Amazon Redshift에 위임할 수 있습니다.

이 정책은 보안 주체가 Amazon Redshift 페더레이션 권한을 사용하여 카탈로그에 대해 쿼리를 실행하도록 허용하지만 Amazon Redshift 네임스페이스를 AWS Glue에 등록 및 등록 취소하는 것은 허용하지 않습니다. Amazon Redshift 페더레이션 권한 설정에 대한 IAM 정책 요구 사항에 대한 설명서를 참조하세요.

[AmazonRedshiftFederatedAuthorization](#) 정책은 IAM 콘솔 및 AWS 관리형 정책 참조 가이드의 [AmazonRedshiftFederatedAuthorization](#)에서 확인할 수 있습니다.

Amazon Redshift API 작업 및 리소스에 대한 권한을 허용하는 고유의 사용자 정의 IAM 정책을 생성할 수도 있습니다. 해당 권한이 필요한 IAM 역할 또는 그룹에 이러한 사용자 지정 정책을 연결할 수 있습니다.

Redshift Spectrum을 사용하는 데 필요한 권한

Amazon Redshift Spectrum에서 리소스에 액세스하려면 다른 AWS 서비스에 대한 권한이 필요합니다. Redshift Spectrum에 대한 IAM 정책의 권한에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Amazon Redshift Spectrum에 대한 IAM 정책](#)을 참조하세요.

Amazon Redshift 콘솔 사용에 필요한 권한

사용자가 Amazon Redshift 콘솔로 작업하려면 AWS 계정에 대한 Amazon Redshift 리소스를 입력하도록 허용하는 최소 권한이 있어야 합니다. 이러한 권한이 있어야만 사용자가 Amazon EC2 보안, Amazon CloudWatch, Amazon SNS 및 네트워크 정보 등 다른 관련 정보를 입력할 수 있습니다.

최소 필수 권한보다 더 제한적인 IAM 정책을 만들면 콘솔은 해당 IAM 정책에 연결된 사용자에게 의도대로 작동하지 않습니다. 이 사용자가 Amazon Redshift 콘솔을 사용할 수 있도록 하려면 AmazonRedshiftReadOnlyAccess 관리형 정책을 사용자에게 연결합니다. 이 작업을 수행하는 방법은 [Amazon Redshift에 대한 AWS 관리형 정책](#)에 설명되어 있습니다.

사용자에게 Amazon Redshift 콘솔의 쿼리 편집기에 대한 액세스 권한을 부여하는 방법에 대한 내용은 [Amazon Redshift 콘솔 쿼리 편집기 사용에 필요한 권한](#) 섹션을 참조하세요.

AWS CLI 또는 Amazon Redshift API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요가 없습니다.

Amazon Redshift 콘솔 쿼리 편집기 사용에 필요한 권한

사용자가 Amazon Redshift 쿼리 편집기로 작업하려면 해당 사용자에게 Amazon Redshift 및 Amazon Redshift Data API 작업에 대한 최소 권한 집합이 있어야 합니다. 보안 암호를 사용하여 데이터베이스에 연결하려면 Secrets Manager 권한도 있어야 합니다.

사용자가 Amazon Redshift 콘솔에서 쿼리 편집기에 액세스할 수 있는 권한을 제공하려면 AmazonRedshiftQueryEditor 및 AmazonRedshiftReadOnlyAccess AWS 관리형 정책을 연결

합니다. AmazonRedshiftQueryEditor 정책은 사용자 권한이 자신의 SQL 문의 결과만 검색하도록 허용합니다. 즉, AmazonRedshiftQueryEditor AWS 관리형 정책의 이 섹션에 표시된 것과 동일한 `aws:userid`로 제출된 문입니다.

```
{
  "Sid": "DataAPIIAMStatementPermissionsRestriction",
  "Action": [
    "redshift-data:GetStatementResult",
    "redshift-data:CancelStatement",
    "redshift-data:DescribeStatement",
    "redshift-data:ListStatements"
  ],
  "Effect": "Allow",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "redshift-data:statement-owner-iam-userid": "${aws:userid}"
    }
  }
}
```

사용자가 동일한 IAM 역할에 있는 다른 사용자의 SQL 문의 결과를 검색할 수 있도록 허용하려면 현재 사용자에게 대한 액세스를 제한하는 조건 없이 자체 정책을 생성합니다. 또한 정책을 변경할 수 있는 액세스 권한을 관리자로서 제한합니다.

쿼리 편집기 v2를 사용하는 데 필요한 권한

사용자가 Amazon Redshift 쿼리 편집기 v2를 사용하려면 해당 사용자에게 Amazon Redshift, 쿼리 편집기 v2 작업 및 AWS Key Management Service, AWS Secrets Manager 및 태깅 서비스와 같은 기타 AWS 서비스에 대한 최소 권한 집합이 있어야 합니다.

사용자에게 쿼리 편집기 v2에 대한 전체 액세스 권한을 부여하려면

AmazonRedshiftQueryEditorV2FullAccess AWS 관리형 정책을 연결합니다.

AmazonRedshiftQueryEditorV2FullAccess 정책은 사용자 권한이 쿼리와 같은 쿼리 편집기 v2 리소스를 같은 팀의 다른 사람들과 공유하도록 허용합니다. 쿼리 편집기 v2 리소스에 대한 액세스를 제어하는 방법에 대한 자세한 내용은 IAM 콘솔에서 쿼리 편집기 v2에 대한 특정 관리형 정책 정의를 참조하세요.

일부 Amazon Redshift 쿼리 편집기 v2 AWS 관리형 정책은 조건 내에서 AWS 태그를 사용하여 리소스에 대한 액세스 범위를 지정합니다. 쿼리 에디터 v2 내에서 쿼리 공유는 보안 주체(IAM 역할)에 연결된 IAM 정책의 태그 키 및 값 `"aws:ResourceTag/sqlworkbench-team"`:

"\${aws:PrincipalTag/sqlworkbench-team}"을 기반으로 합니다. 동일한 태그 값(예: accounting-team)이 있는 동일한 AWS 계정의 보안 주체는 쿼리 편집기 v2에서 동일한 팀에 있습니다. 한 번에 한 팀과만 연결할 수 있습니다. 관리 권한이 있는 사용자는 모든 팀원에게 sqlworkbench-team 태그에 대해 동일한 값을 부여하여 IAM 콘솔에서 팀을 설정할 수 있습니다. IAM 사용자 또는 IAM 역할에 대해 sqlworkbench-team의 태그 값이 변경되면 변경 사항이 공유 리소스에 반영될 때까지 지연이 있을 수 있습니다. 리소스(예: 쿼리)의 태그 값이 변경되면 변경 사항이 반영될 때까지 다시 지연이 있을 수 있습니다. 팀원에게도 tag:GetResources 권한이 있어야 공유할 수 있습니다.

예: IAM 역할에 대한 **accounting-team** 태그를 추가하려면

1. AWS Management Console에 로그인하여 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 콘솔의 탐색 창에서 [역할(Roles)]을 선택한 다음 편집할 역할의 이름을 선택합니다.
3. 태그(Tags) 탭을 선택한 후 태그 추가(Add tags)를 선택합니다.
4. 태그 키 sqlworkbench-team과 값 accounting-team을 추가합니다.
5. 변경 사항 저장을 선택합니다.

이제 IAM 보안 주체(이 IAM 역할이 연결됨)가 팀과 쿼리를 공유하면 동일한 accounting-team 태그 값을 가진 다른 보안 주체가 쿼리를 볼 수 있습니다.

IAM 역할 및 IAM 사용자를 포함하여 보안 주체에 태그를 연결하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 리소스에 태깅](#)을 참조하세요.

자격 증명 공급자(IdP)를 사용하여 세션 수준에서 팀을 설정할 수도 있습니다. 이를 통해 동일한 IAM 역할을 사용하는 여러 사용자가 서로 다른 팀을 가질 수 있습니다. IAM 역할 신뢰 정책은 sts:TagSession 작업을 허용해야 합니다. 자세한 내용은 IAM 사용 설명서의 [세션 태그를 추가하는 데 필요한 권한](#)을 참조하세요. IdP에서 제공한 SAML 어설션에 보안 주체 태그 속성을 추가합니다.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:sqlworkbench-team">
  <AttributeValue>accounting-team</AttributeValue>
</Attribute>
```

자격 증명 공급자(IdP)의 지침에 따라 SAML 속성을 디렉터리의 콘텐츠로 채웁니다. 자격 증명 공급자(IdP) 및 Amazon Redshift에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 인증을 이용한 데이터베이스 사용자 자격 증명 생성](#)과 [자격 증명 공급자 및 페더레이션](#)을 참조하세요.

은 노트북 셀의 현재 콘텐츠를 가져오고 계정에서 노트북 버전을 생성할 수 있는 권한을 부여합니다. 즉, 버전 생성 당시 노트북의 현재 내용은 버전의 내용과 동일합니다. 나중에 버전의 셀 내용은 현재 노트북이 업데이트된 것과 동일하게 유지됩니다. `sqlworkbench:GetNotebookVersion`은 노트북 버전을 가져올 수 있는 권한을 부여합니다. `sqlworkbench:BatchGetNotebookCell` 권한은 없지만 노트북에 대한 `sqlworkbench>CreateNotebookVersion` 및 `sqlworkbench:GetNotebookVersion` 권한이 있는 사용자는 해당 버전의 노트북 셀에 액세스할 수 있습니다. `sqlworkbench:BatchGetNotebookCell` 권한이 없는 이 사용자는 먼저 버전을 생성한 다음 이 생성된 버전을 가져오는 방식으로 여전히 Notebook 셀의 콘텐츠를 검색할 수 있습니다.

Amazon Redshift 스케줄러 사용에 필요한 권한

Amazon Redshift 스케줄러를 사용하는 경우 스케줄러가 사용자를 대신하여 권한을 맡을 수 있도록 Amazon Redshift 스케줄러(**`scheduler.redshift.amazonaws.com`**)와 신뢰 관계가 있는 IAM 역할을 설정합니다. 또한 예약하려는 Amazon Redshift API 작업에 대한 역할에 정책(권한)을 연결합니다.

다음 예에서는 Amazon Redshift 스케줄러 및 Amazon Redshift와 신뢰 관계를 설정하기 위한 정책 문서를 JSON 형식으로 보여줍니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "scheduler.redshift.amazonaws.com",
          "redshift.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

신뢰 엔터티에 대한 자세한 내용은 IAM User Guide의 [Creating a role to delegate permissions to an AWS service](#)를 참조하세요.

또한 예약하려는 Amazon Redshift 작업에 대한 권한을 추가해야 합니다.

스케줄러가 `ResizeCluster` 작업을 사용하려면 IAM 정책에 다음과 유사한 권한을 추가하세요. 환경에 따라 정책을 보다 제한적으로 만들 수 있습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "redshift:ResizeCluster",
      "Resource": "*"
    }
  ]
}
```

Amazon Redshift 스케줄러에 대한 역할을 생성하는 단계는 IAM User Guide의 [Creating a role for an AWS service \(console\)](#)를 참조하세요. IAM 콘솔에서 역할을 생성할 때 다음을 선택합니다.

- Choose the service that will use this role(이 역할을 사용할 서비스 선택): Redshift를 선택합니다.
- Select your use case(사용 사례 선택): Redshift - Scheduler(Redshift - 스케줄러)를 선택합니다.
- Amazon Redshift 작업을 예약할 수 있는 정책을 생성하거나 해당 역할에 연결합니다. 정책 생성을 선택하거나 역할을 수정하여 정책을 연결합니다. 예약할 작업에 대한 JSON 정책을 입력합니다.
- 역할을 생성한 후 서비스 `redshift.amazonaws.com`을 포함하도록 IAM 역할의 신뢰 관계를 편집합니다.

생성한 IAM 역할에는 `scheduler.redshift.amazonaws.com` 및 `redshift.amazonaws.com`의 신뢰할 수 있는 개체가 있습니다. 또한 지원되는 Amazon Redshift API 작업(예: `"redshift:ResizeCluster"`)을 허용하는 정책이 연결되어 있습니다.

Amazon EventBridge 스케줄러 사용에 필요한 권한

Amazon EventBridge 스케줄러를 사용하는 경우 스케줄러가 사용자를 대신하여 권한을 맡을 수 있도록 Amazon EventBridge 스케줄러(**`events.amazonaws.com`**)와 신뢰 관계가 있는 IAM 역할을 설정

합니다. 또한 예약하려는 Amazon Redshift Data API 작업에 대한 역할과 Amazon EventBridge 작업에 대한 정책에 정책(권한)을 연결합니다.

콘솔에서 Amazon Redshift 쿼리 편집기를 사용하여 예약된 쿼리를 생성할 때 EventBridge 스케줄러를 사용합니다.

IAM 콘솔에서 예약된 쿼리를 실행하는 IAM 역할을 생성할 수 있습니다. 이 IAM 역할에서 AmazonEventBridgeFullAccess와 AmazonRedshiftDataFullAccess를 연결합니다.

다음 예에서는 EventBridge 스케줄러와 신뢰 관계를 설정하기 위한 정책 문서를 JSON 형식으로 보여줍니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "events.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

신뢰 엔터티에 대한 자세한 내용은 IAM User Guide의 [Creating a role to delegate permissions to an AWS service](#)를 참조하세요.

EventBridge 스케줄러에 대한 역할을 생성하는 단계는 IAM User Guide의 [Creating a role for an AWS service \(console\)](#)를 참조하세요. IAM 콘솔에서 역할을 생성할 때 다음을 선택합니다.

- [이 역할을 사용할 서비스 선택(Choose the service that will use this role)]에서 [CloudWatch Events]를 선택합니다.
- [사용 사례 선택(Select your use case):]에서 [CloudWatch Events]를 선택합니다.
- 권한 정책 AmazonEventBridgeFullAccess 및 AmazonRedshiftDataFullAccess를 연결합니다.

생성한 IAM 역할에는 `events.amazonaws.com`의 신뢰할 수 있는 엔터티가 있습니다. 또한 지원되는 Amazon Redshift Data API 작업(예: `"redshift-data:*"`)을 허용하는 정책이 연결되어 있습니다.

Amazon Redshift 기계 학습(ML)을 사용하는 데 필요한 권한

다음에서 다양한 사용 사례에 대해 Amazon Redshift 기계 학습(ML)을 사용하는 데 필요한 권한에 대한 설명을 찾을 수 있습니다.

사용자가 Amazon SageMaker AI와 함께 Amazon Redshift 기계 학습을 사용하려면 기본값보다 더 제한적인 정책으로 IAM 역할을 생성합니다. 다음 정책을 사용할 수 있습니다. 필요에 맞게 이 정책을 수정할 수도 있습니다.

다음 정책은 Amazon Redshift의 모델 설명 기능을 사용하여 SageMaker AI Autopilot을 실행하는 데 필요한 권한을 보여줍니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateTrainingJob",
        "sagemaker:CreateAutoMLJob",
        "sagemaker:CreateCompilationJob",
        "sagemaker:CreateEndpoint",
        "sagemaker:DescribeAutoMLJob",
        "sagemaker:DescribeTrainingJob",
        "sagemaker:DescribeCompilationJob",
        "sagemaker:DescribeProcessingJob",
        "sagemaker:DescribeTransformJob",
        "sagemaker:ListCandidatesForAutoMLJob",
        "sagemaker:StopAutoMLJob",
        "sagemaker:StopCompilationJob",
        "sagemaker:StopTrainingJob",
        "sagemaker:DescribeEndpoint",
        "sagemaker:InvokeEndpoint",
        "sagemaker:StopProcessingJob",
        "sagemaker:CreateModel",
        "sagemaker:CreateProcessingJob"
      ],
    }
  ],
}
```

```

        "Resource": [
            "arn:aws:sagemaker::*:model/*redshift*",
            "arn:aws:sagemaker::*:training-job/*redshift*",
            "arn:aws:sagemaker::*:automl-job/*redshift*",
            "arn:aws:sagemaker::*:compilation-job/*redshift*",
            "arn:aws:sagemaker::*:processing-job/*redshift*",
            "arn:aws:sagemaker::*:transform-job/*redshift*",
            "arn:aws:sagemaker::*:endpoint/*redshift*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "logs:CreateLogGroup",
            "logs:CreateLogStream",
            "logs:DescribeLogStreams",
            "logs:PutLogEvents"
        ],
        "Resource": [
            "arn:aws:logs::*:log-group:/aws/sagemaker/Endpoints/*redshift*",
            "arn:aws:logs::*:log-group:/aws/sagemaker/ProcessingJobs/
*redshift*",
            "arn:aws:logs::*:log-group:/aws/sagemaker/TrainingJobs/
*redshift*",
            "arn:aws:logs::*:log-group:/aws/sagemaker/TransformJobs/
*redshift*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "cloudwatch:PutMetricData"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "cloudwatch:namespace": [
                    "SageMaker",
                    "/aws/sagemaker/Endpoints",
                    "/aws/sagemaker/ProcessingJobs",
                    "/aws/sagemaker/TrainingJobs",
                    "/aws/sagemaker/TransformJobs"
                ]
            }
        }
    }
}

```

```

    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ecr:BatchCheckLayerAvailability",
      "ecr:BatchGetImage",
      "ecr:GetAuthorizationToken",
      "ecr:GetDownloadUrlForLayer"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:GetBucketAcl",
      "s3:GetBucketCors",
      "s3:GetEncryptionConfiguration",
      "s3:GetBucketLocation",
      "s3:ListBucket",
      "s3:ListAllMyBuckets",
      "s3:ListMultipartUploadParts",
      "s3:ListBucketMultipartUploads",
      "s3:PutObject",
      "s3:PutBucketAcl",
      "s3:PutBucketCors",
      "s3:DeleteObject",
      "s3:AbortMultipartUpload",
      "s3:CreateBucket"
    ],
    "Resource": [
      "arn:aws:s3:::redshift-downloads",
      "arn:aws:s3:::redshift-downloads/*",
      "arn:aws:s3:::*redshift*",
      "arn:aws:s3:::*redshift/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:GetBucketAcl",
      "s3:GetBucketCors",

```

```

        "s3:GetEncryptionConfiguration",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListAllMyBuckets",
        "s3:ListMultipartUploadParts",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject",
        "s3:PutBucketAcl",
        "s3:PutBucketCors",
        "s3:DeleteObject",
        "s3:AbortMultipartUpload",
        "s3:CreateBucket"
    ],
    "Resource": "*",
    "Condition": {
        "StringEqualsIgnoreCase": {
            "s3:ExistingObjectTag/Redshift": "true"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::*:role/*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [
                "redshift.amazonaws.com",
                "sagemaker.amazonaws.com"
            ]
        }
    }
}
]
}

```

다음 정책은 Amazon DynamoDB, Redshift Spectrum 및 Amazon RDS 페더레이션에 대한 액세스를 허용하기 위한 최소한의 전체 권한을 보여줍니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateTrainingJob",
        "sagemaker:CreateAutoMLJob",
        "sagemaker:CreateCompilationJob",
        "sagemaker:CreateEndpoint",
        "sagemaker:DescribeAutoMLJob",
        "sagemaker:DescribeTrainingJob",
        "sagemaker:DescribeCompilationJob",
        "sagemaker:DescribeProcessingJob",
        "sagemaker:DescribeTransformJob",
        "sagemaker:ListCandidatesForAutoMLJob",
        "sagemaker:StopAutoMLJob",
        "sagemaker:StopCompilationJob",
        "sagemaker:StopTrainingJob",
        "sagemaker:DescribeEndpoint",
        "sagemaker:InvokeEndpoint",
        "sagemaker:StopProcessingJob",
        "sagemaker:CreateModel",
        "sagemaker:CreateProcessingJob"
      ],
      "Resource": [
        "arn:aws:sagemaker::*:model/*redshift*",
        "arn:aws:sagemaker::*:training-job/*redshift*",
        "arn:aws:sagemaker::*:automl-job/*redshift*",
        "arn:aws:sagemaker::*:compilation-job/*redshift*",
        "arn:aws:sagemaker::*:processing-job/*redshift*",
        "arn:aws:sagemaker::*:transform-job/*redshift*",
        "arn:aws:sagemaker::*:endpoint/*redshift*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",

```

```

        "logs:PutLogEvents"
    ],
    "Resource": [
        "arn:aws:logs:*:*:log-group:/aws/sagemaker/Endpoints/*redshift*",
        "arn:aws:logs:*:*:log-group:/aws/sagemaker/ProcessingJobs/
*redshift*",
        "arn:aws:logs:*:*:log-group:/aws/sagemaker/TrainingJobs/
*redshift*",
        "arn:aws:logs:*:*:log-group:/aws/sagemaker/TransformJobs/
*redshift*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:PutMetricData"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "cloudwatch:namespace": [
                "SageMaker",
                "/aws/sagemaker/Endpoints",
                "/aws/sagemaker/ProcessingJobs",
                "/aws/sagemaker/TrainingJobs",
                "/aws/sagemaker/TransformJobs"
            ]
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:BatchGetImage",
        "ecr:GetAuthorizationToken",
        "ecr:GetDownloadUrlForLayer"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "s3:GetObject",

```

```

        "s3:GetBucketAcl",
        "s3:GetBucketCors",
        "s3:GetEncryptionConfiguration",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListAllMyBuckets",
        "s3:ListMultipartUploadParts",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject",
        "s3:PutBucketAcl",
        "s3:PutBucketCors",
        "s3:DeleteObject",
        "s3:AbortMultipartUpload",
        "s3:CreateBucket"
    ],
    "Resource": [
        "arn:aws:s3:::redshift-downloads",
        "arn:aws:s3:::redshift-downloads/*",
        "arn:aws:s3:::*redshift*",
        "arn:aws:s3:::*redshift/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "s3:GetObject",
        "s3:GetBucketAcl",
        "s3:GetBucketCors",
        "s3:GetEncryptionConfiguration",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListAllMyBuckets",
        "s3:ListMultipartUploadParts",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject",
        "s3:PutBucketAcl",
        "s3:PutBucketCors",
        "s3:DeleteObject",
        "s3:AbortMultipartUpload",
        "s3:CreateBucket"
    ],
    "Resource": "*",
    "Condition": {
        "StringEqualsIgnoreCase": {

```

```

        "s3:ExistingObjectTag/Redshift": "true"
    }
}
},
{
    "Effect": "Allow",
    "Action": [
        "dynamodb:Scan",
        "dynamodb:DescribeTable",
        "dynamodb:Getitem"
    ],
    "Resource": [
        "arn:aws:dynamodb:*:*:table/*redshift*",
        "arn:aws:dynamodb:*:*:table/*redshift*/index/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "elasticmapreduce:ListInstances"
    ],
    "Resource": [
        "arn:aws:elasticmapreduce:*:*:cluster/*redshift*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "elasticmapreduce:ListInstances"
    ],
    "Resource": "*",
    "Condition": {
        "StringEqualsIgnoreCase": {
            "elasticmapreduce:ResourceTag/Redshift": "true"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "lambda:InvokeFunction"
    ],
    "Resource": "arn:aws:lambda:*:*:function:*redshift*"
},

```

```

{
  "Effect": "Allow",
  "Action": [
    "glue:CreateDatabase",
    "glue>DeleteDatabase",
    "glue:GetDatabase",
    "glue:GetDatabases",
    "glue:UpdateDatabase",
    "glue:CreateTable",
    "glue>DeleteTable",
    "glue:BatchDeleteTable",
    "glue:UpdateTable",
    "glue:GetTable",
    "glue:GetTables",
    "glue:BatchCreatePartition",
    "glue:CreatePartition",
    "glue>DeletePartition",
    "glue:BatchDeletePartition",
    "glue:UpdatePartition",
    "glue:GetPartition",
    "glue:GetPartitions",
    "glue:BatchGetPartition"
  ],
  "Resource": [
    "arn:aws:glue:*:table/*redshift*/",
    "arn:aws:glue:*:catalog",
    "arn:aws:glue:*:database/*redshift*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "secretsmanager:GetResourcePolicy",
    "secretsmanager:GetSecretValue",
    "secretsmanager:DescribeSecret",
    "secretsmanager:ListSecretVersionIds"
  ],
  "Resource": [
    "arn:aws:secretsmanager:*:secret:*redshift*"
  ]
},
{
  "Effect": "Allow",
  "Action": [

```

```

        "secretsmanager:GetRandomPassword",
        "secretsmanager:ListSecrets"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "secretsmanager:ResourceTag/Redshift": "true"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::*:role/*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [
                "redshift.amazonaws.com",
                "glue.amazonaws.com",
                "sagemaker.amazonaws.com",
                "athena.amazonaws.com"
            ]
        }
    }
}
]
}

```

선택적으로 암호화에 AWS KMS 키를 사용하려면 정책에 다음 권한을 추가합니다.

```

{
    "Effect": "Allow",
    "Action": [
        "kms:CreateGrant",
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": [

```

```

    "arn:aws:kms:<your-region>:<your-account-id>:key/<your-kms-key>"
  ]
}

```

Amazon Redshift와 SageMaker AI가 다른 서비스와 상호 작용하는 이전 IAM 역할을 수입하도록 허용하려면 역할에 다음 신뢰 정책을 추가합니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "redshift.amazonaws.com",
          "sagemaker.amazonaws.com",
          "forecast.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

앞의 경우 Amazon S3 버킷 `redshift-downloads/redshift-m1/`은 다른 단계와 예제에 사용되는 샘플 데이터가 저장되는 위치입니다. Amazon S3에서 데이터를 로드할 필요가 없으면 이 버킷을 제거할 수 있습니다. 또는 Amazon Redshift로 데이터를 로드하는 데 사용하는 다른 Amazon S3 버킷으로 바꿉니다.

your-account-id, **your-role** 및 **your-s3-bucket** 값은 CREATE MODEL 명령에서 지정하는 계정 ID, 역할 및 버킷입니다.

필요에 따라 Amazon Redshift 기계 학습에 사용할 AWS KMS 키를 지정하는 경우 샘플 정책의 AWS KMS 키 섹션을 사용할 수 있습니다. **your-kms-key** 값은 CREATE MODEL 명령의 일부로 사용하는 키입니다.

하이퍼파라미터 튜닝 작업에 대해 Virtual Private Cloud(VPC)를 지정할 때 다음 권한을 추가합니다.

```
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DeleteNetworkInterface",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
    ]
}
```

모델 설명으로 작업하려면 SageMaker AI API 작업을 직접 호출할 수 있는 권한이 있는지 확인해야 합니다. AmazonSageMakerFullAccess 관리형 정책을 사용하는 것이 좋습니다. 보다 제한적인 정책으로 IAM 역할을 생성하려는 경우 다음 정책을 사용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateEndpoint",
        "sagemaker:CreateEndpointConfig",
        "sagemaker:DeleteEndpoint",
        "sagemaker:DeleteEndpointConfig",
        "sagemaker:DescribeEndpoint",
        "sagemaker:DescribeEndpointConfig",
        "sagemaker:DescribeModel",
        "sagemaker:InvokeEndpoint",
        "sagemaker:ListTags"
      ],
      "Resource": "*"
    }
  ]
}
```


AmazonSageMakerFullAccess 관리형 정책에 대한 자세한 내용은 Amazon SageMaker AI Developer Guide의 [AmazonSageMakerFullAccess](#)를 참조하세요.

예측 모델을 만들려면 AmazonForecastFullAccess 관리형 정책을 사용하는 것이 좋습니다. 보다 제한적인 정책을 사용하려면 IAM 역할에 다음 정책을 추가합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "forecast:CreateAutoPredictor",
        "forecast:CreateDataset",
        "forecast:CreateDatasetGroup",
        "forecast:CreateDatasetImportJob",
        "forecast:CreateForecast",
        "forecast:CreateForecastExportJob",
        "forecast>DeleteResourceTree",
        "forecast:DescribeAutoPredictor",
        "forecast:DescribeDataset",
        "forecast:DescribeDatasetGroup",
        "forecast:DescribeDatasetImportJob",
        "forecast:DescribeForecast",
        "forecast:DescribeForecastExportJob",
        "forecast:StopResource",
        "forecast:TagResource",
        "forecast:UpdateDatasetGroup"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon Bedrock 모델을 만들려면 AmazonBedrockFullAccess 관리형 정책을 사용하는 것이 좋습니다. 보다 제한적인 정책을 사용하려면 IAM 역할에 다음 정책을 추가합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "bedrock:InvokeModel",
      "Resource": [
        "*",
        "arn:aws:bedrock:us-east-1::foundation-model/*"
      ]
    }
  ]
}
```

Amazon Redshift ML에 관한 자세한 내용은 [Amazon Redshift에서 기계 학습 사용](#) 또는 [CREATE MODEL](#), [CREATE EXTERNAL MODEL](#)을 참조하시기 바랍니다.

스트리밍 수집 권한

스트리밍 수집은 두 가지 서비스에서 작동합니다. 바로 Kinesis Data Streams와 Amazon MSK입니다.

Kinesis Data Streams에서 스트리밍 수집을 사용하는 데 필요한 권한

관리형 정책 예시가 포함된 절차는 [Amazon Kinesis Data Streams에서 스트리밍 수집 시작하기](#)를 참조하세요.

Amazon MSK에서 스트리밍 수집을 사용하는 데 필요한 권한

관리형 정책 예시가 포함된 절차는 [Amazon Managed Streaming for Apache Kafka에서 스트리밍 수집 시작하기](#)를 참조하세요.

데이터 공유 API 작업을 사용하는 데 필요한 권한

데이터 공유 API 작업에 대한 액세스를 제어하려면 IAM 작업 기반 정책을 사용합니다. IAM 정책을 관리하는 방법에 대한 자세한 내용은 IAM User Guide의 [Managing IAM policies](#) 섹션을 참조하세요.

특히, 생산자 클러스터 관리자가 AuthorizeDataShare 호출을 사용하여 AWS 계정 외부의 datashare에 대한 송신 권한을 부여해야 한다고 가정합니다. 이 경우 이 권한을 부여하도록 IAM 작업 기반 정책을 설정합니다. DeauthorizeDataShare 호출을 사용하여 송신을 취소합니다.

IAM 작업 기반 정책을 사용할 때 DataShareARN과 같이 정책에서 IAM 리소스를 지정할 수도 있습니다. 다음은 DataShareARN의 형식과 예를 보여줍니다.

```
arn:aws:redshift:region:account-id:datashare:namespace-guid/datashare-name
arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-e2e24359e9a8/
SalesShare
```

IAM 정책에서 datashare 이름을 지정하여 특정 datashare에 대한 AuthorizeDataShare 액세스를 제한할 수 있습니다.

```
{
  "Statement": [
    {
      "Action": [
        "redshift:AuthorizeDataShare",
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-
e2e24359e9a8/SalesShare"
      ],
      "Effect": "Deny"
    }
  ]
}
```

특정 생산자 클러스터가 소유한 모든 datashare로 IAM 정책을 제한할 수도 있습니다. 이렇게 하려면 정책의 **datashare-name** 값을 와일드카드나 별표로 바꿉니다. 클러스터의 namespace-guid 값을 유지합니다.

```
arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-e2e24359e9a8/
*
```

다음은 특정 생산자 클러스터가 소유한 datashare에서 엔터티가 AuthorizeDataShare를 호출하지 못하도록 하는 IAM 정책입니다.

```
{
  "Statement": [
    {
      "Action": [
        "redshift:AuthorizeDataShare",
      ],
```

```

    "Resource": [
      "arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-
e2e24359e9a8/*"
    ],
    "Effect": "Deny"
  }
]
}

```

DataShareARN은 datashare 이름과 소유 클러스터의 네임스페이스에 대한 GUID(Globally Unique ID)를 기반으로 액세스를 제한합니다. 이름을 별표로 지정하여 이 작업을 수행합니다.

GetClusterCredentials에 대한 리소스 정책

IAM 데이터베이스 자격 증명이 있는 JDBC 또는 ODBC 연결을 이용해 클러스터 데이터베이스에 연결하려거나 GetClusterCredentials 작업을 프로그래밍 방식으로 직접 호출하려면 dbuser 리소스에 액세스하여 redshift:GetClusterCredentials 작업을 직접 호출할 수 있는 권한이 필요합니다.

server 및 port 대신 JDBC 또는 ODBC 연결을 사용하는 경우 cluster_id 및 region을 지정할 수 있습니다. 단, 그렇게 하려면 사용자 정책이 redshift:DescribeClusters 작업을 허용하고 cluster 리소스에 액세스 권한이 있어야 합니다.

선택적 파라미터 Autocreate, DbGroups 및 DbName을 사용하여 GetClusterCredentials를 호출하는 경우 그러한 작업을 허용하고, 다음 표에 나열된 리소스에 대한 액세스를 허용해야 합니다.

GetClusterCredentials 파라미터	작업	리소스
Autocreate	redshift dbuser CreateClusterUser	
DbGroups	redshift dbgroup JoinGroup	
DbName	NA	dbname

리소스에 대한 자세한 내용은 [Amazon Redshift 리소스 및 작업](#) 섹션을 참조하세요.

정책에 다음의 조건도 포함할 수 있습니다.

- `redshift:DurationSeconds`
- `redshift:DbName`
- `redshift:DbUser`

Important

SAML SSO 통합의 경우 `${redshift:DbUser}` 변수를 사용하여 IAM 정책을 지정해야 할 수 있습니다. 이러한 경우 호출자가 AWS 사용자 자격 증명과 일치하지 않는 사용자의 자격 증명을 얻을 수 없도록 하는 조건문을 사용하는 것이 좋습니다. 예를 들면 `"StringEquals":{"aws:userid":"AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}"}`가 있습니다.

[예제 8: GetClusterCredentials 사용을 위한 IAM 정책 예제](#)을(를) 참조하세요. 조건에 대한 자세한 내용은 [정책에서 조건 지정](#) 단원을 참조하십시오.

고객 관리형 정책 예제

이 섹션에서는 다양한 Amazon Redshift 작업에 대한 권한을 부여하는 사용자 정책의 예를 제공합니다. 이러한 정책은 Amazon Redshift API, AWS SDK 또는 AWS CLI를 사용하는 경우에 적용됩니다.

Note

모든 예에서는 미국 서부(오리건) 리전(us-west-2)을 사용하며 가상의 계정 ID를 포함합니다.

예 1: 모든 Amazon Redshift 작업 및 리소스에 대한 전체 액세스 권한 허용

다음 정책은 모든 리소스를 사용하는 Amazon Redshift 작업 모두에 대한 액세스 권한을 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Sid": "AllowRedshift",
  "Action": [
    "redshift:*"
  ],
  "Effect": "Allow",
  "Resource": "*"
}
```

Action 요소에서 값 `redshift:*`는 Amazon Redshift의 모든 작업을 의미합니다.

예 2: 한 Amazon Redshift 작업 집합에 대한 사용자 액세스 거부

기본적으로 모든 권한을 거부합니다. 하지만 간혹 특정 작업 또는 작업 집합에 대한 액세스를 명시적으로 거부해야 하는 경우도 있습니다. 이때는 다음 정책을 통해 모든 Amazon Redshift 작업에 대한 액세스를 허용하면서, 이름이 `Delete`로 시작하는 Amazon Redshift 작업에 대해서는 액세스를 명시적으로 거부할 수 있습니다. 이 정책은 `us-west-2`에 속한 모든 Amazon Redshift 리소스에 적용됩니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUSWest2Region",
      "Action": [
        "redshift:*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:redshift:us-east-1:111122223333:*"
    },
    {
      "Sid": "DenyDeleteUSWest2Region",
      "Action": [
        "redshift:Delete*"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:redshift:us-east-1:111122223333:*"
    }
  ]
}
```

```
]
}
```

예 3: 사용자에게 클러스터 관리 허용

다음 정책은 사용자에게 모든 클러스터를 생성, 삭제, 수정 및 재부팅할 수 있도록 허용한 후 클러스터 이름이 `protected`으로 시작되는 모든 클러스터를 삭제하는 권한을 거부합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowClusterManagement",
      "Action": [
        "redshift:CreateCluster",
        "redshift>DeleteCluster",
        "redshift:ModifyCluster",
        "redshift:RebootCluster"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Sid": "DenyDeleteProtected",
      "Action": [
        "redshift>DeleteCluster"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:cluster:protected*"
      ],
      "Effect": "Deny"
    }
  ]
}
```

예 4: 사용자가 스냅샷 액세스 권한을 부여하거나 취소하도록 허용

다음 정책은 사용자(사용자 A)에게 아래와 같은 작업을 허용합니다.

- shared라는 이름의 클러스터에서 생성된 모든 스냅샷에 대한 액세스 권한을 부여합니다.
- shared 클러스터에서 생성된 스냅샷 중에서 이름이 revokable로 시작하는 모든 스냅샷에 대한 액세스 권한을 취소합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSharedSnapshots",
      "Action": [
        "redshift:AuthorizeSnapshotAccess"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:shared/*"
      ],
      "Effect": "Allow"
    },
    {
      "Sid": "AllowRevokableSnapshot",
      "Action": [
        "redshift:RevokeSnapshotAccess"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:snapshot:*/revokable*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

사용자 A가 사용자 B에게 스냅샷에 대한 액세스를 허용한 경우에는 다음과 같이 사용자 B에게 스냅샷에서 클러스터를 복원할 수 있는 정책이 할당되어야 합니다. 다음 정책은 사용자 B에게 스냅샷에 대해 설명한 후 이를 통해 복원하거나, 클러스터를 생성할 수 있는 권한을 허용합니다. 클러스터의 이름은 from-other-account로 시작해야 합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDescribeSnapshots",
      "Action": [
        "redshift:DescribeClusterSnapshots"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Sid": "AllowUserRestoreFromSnapshot",
      "Action": [
        "redshift:RestoreFromClusterSnapshot"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:snapshot:*/*",
        "arn:aws:redshift:us-west-2:444455556666:cluster:from-other-account*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

예 5: 사용자가 클러스터 스냅샷을 복사한 후 스냅샷에서 클러스터를 복원하도록 허용

다음 정책은 사용자에게 big-cluster-1이라는 이름의 클러스터에서 생성된 모든 스냅샷을 복사한 후 그 중에서 이름이 snapshot-for-restore로 시작하는 스냅샷을 복원하도록 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Sid": "AllowCopyClusterSnapshot",
    "Action": [
        "redshift:CopyClusterSnapshot"
    ],
    "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:snapshot:big-cluster-1/*"
    ],
    "Effect": "Allow"
  },
  {
    "Sid": "AllowRestoreFromClusterSnapshot",
    "Action": [
        "redshift:RestoreFromClusterSnapshot"
    ],
    "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:snapshot:*/snapshot-for-restore*",
        "arn:aws:redshift:us-west-2:123456789012:cluster:*"
    ],
    "Effect": "Allow"
  }
]
}

```

예 6: Amazon Redshift 및 관련 AWS 서비스의 공통 작업과 리소스에 대한 사용자 액세스 허용

다음 예제 정책은 Amazon Redshift, Amazon Simple Notification Service(Amazon SNS) 및 Amazon CloudWatch에 대한 모든 작업과 리소스에 대한 액세스를 허용합니다. 또한 해당 계정에서 관련된 모든 Amazon EC2 리소스에 대한 지정된 작업을 허용합니다.

Note

이 정책 예에서 지정하는 Amazon EC2 작업은 리소스 수준 권한이 지원되지 않습니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```
"Sid":"AllowRedshift",
"Effect": "Allow",
"Action": [
    "redshift:*"
],
"Resource": [
    "*"
]
},
{
    "Sid":"AllowSNS",
    "Effect": "Allow",
    "Action": [
        "sns:*"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid":"AllowCloudWatch",
    "Effect": "Allow",
    "Action": [
        "cloudwatch:*"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid":"AllowEC2Actions",
    "Effect": "Allow",
    "Action": [
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AttachNetworkInterface",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAddresses",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs"
    ],
}
```

```

    "Resource": [
        "*"
    ]
}
]
}

```

예제 7: 사용자가 Amazon Redshift 콘솔로 리소스에 태그를 지정할 수 있도록 허용

다음 예제 정책에서는 사용자가 AWS Resource Groups를 사용하여 Amazon Redshift 콘솔로 리소스에 태그를 지정할 수 있도록 허용합니다. 새 Amazon Redshift 콘솔이나 원래 Amazon Redshift 콘솔을 호출하는 사용자 역할에 이 정책을 연결할 수 있습니다. 태그 지정에 대한 자세한 내용은 [Amazon Redshift에서 리소스 태그](#) 섹션을 참조하세요.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "TaggingPermissions",
      "Effect": "Allow",
      "Action": [
        "redshift:DeleteTags",
        "redshift:CreateTags",
        "redshift:DescribeTags",
        "tag:UntagResources",
        "tag:TagResources"
      ],
      "Resource": "*"
    }
  ]
}

```

예제 8: GetClusterCredentials 사용을 위한 IAM 정책 예제

다음 정책은 다음과 같은 샘플 파라미터 값을 사용합니다.

- 리전: us-west-2

- AWS 계정: 123456789012
- 클러스터 이름: examplecluster

다음 정책은 GetCredentials, CreateClusterUser, JoinGroup 작업을 허용합니다. 이 정책은 조건 키를 사용하여 GetClusterCredentials 및 CreateClusterUser 작업을 허용합니다. 이것은 AWS 사용자 ID가 "AIDIO0R4TAW7CSEXAMPLE:\${redshift:DbUser}@yourdomain.com"과 일치할 때만 해당됩니다. IAM 액세스는 "testdb" 데이터베이스에 대해서만 요청됩니다. 또한 이 정책은 사용자가 "common_group"이라는 그룹을 조인할 수 있도록 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GetClusterCredsStatement",
      "Effect": "Allow",
      "Action": [
        "redshift:GetClusterCredentials"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/${redshift:DbUser}",
        "arn:aws:redshift:us-west-2:123456789012:dbname:examplecluster/testdb",
        "arn:aws:redshift:us-west-2:123456789012:dbgroup:examplecluster/common_group"
      ],
      "Condition": {
        "StringEquals": {
          "aws:userid": "AIDIO0R4TAW7CSEXAMPLE:${redshift:DbUser}@yourdomain.com"
        }
      }
    },
    {
      "Sid": "CreateClusterUserStatement",
      "Effect": "Allow",
      "Action": [
        "redshift:CreateClusterUser"
      ],
      "Resource": [
```

```

    "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/
    ${redshift:DbUser}"
  ],
  "Condition": {
    "StringEquals": {
      "aws:userid": "AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}@yourdomain.com"
    }
  },
  {
    "Sid": "RedshiftJoinGroupStatement",
    "Effect": "Allow",
    "Action": [
      "redshift:JoinGroup"
    ],
    "Resource": [
      "arn:aws:redshift:us-west-2:123456789012:dbgroup:examplecluster/
      common_group"
    ]
  }
]
}

```

다음 예는 IAM 역할이 `GetClusterCredentials` 작업을 호출하도록 허용하는 정책입니다. Amazon Redshift `dbuser` 리소스를 지정하면 데이터베이스 사용자 이름 `temp_creds_user(examplecluster라는 클러스터에서)`에 역할의 액세스를 허용합니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "redshift:GetClusterCredentials",
    "Resource": "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/
    temp_creds_user"
  }
}

```

와일드카드(*)를 사용하여 클러스터 이름, 사용자 이름, 데이터베이스 그룹 이름의 일부 또는 전부를 대체할 수 있습니다. 다음은 지정 계정에서 아무 클러스터가 있는 temp_로 시작하는 모든 사용자 이름을 허용하는 예입니다.

Important

다음 예의 구문은 리소스의 값의 일부로 와일드카드 문자(*)를 지정하여 정책이 해당 문자로 시작하는 모든 리소스를 허용하게 합니다. IAM 정책에 와일드카드 문자를 사용하면 너무 허용적일 수 있습니다. 가장 좋은 방법은 해당 업무 용도에 맞는 정책 중 가장 제한적인 정책을 사용하는 것입니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "redshift:GetClusterCredentials",
    "Resource": "arn:aws:redshift:us-west-2:123456789012:dbuser:*/temp_*"
  }
}
```

다음 예는 IAM 역할이 새 사용자를 자동으로 생성하고 사용자 로그인 시 조인하는 그룹을 지정하는 옵션으로 GetClusterCredentials 작업의 호출을 허용하는 정책입니다. "Resource": "*" 절은 클러스터, 데이터베이스 사용자, 또는 사용자 그룹을 포함한 모든 리소스에 역할 액세스 권한을 허용합니다.

Important

다음 예제의 문은 와일드카드 문자(*)를 지정된 작업에 대한 리소스로 지정하므로, 정책은 모든 클러스터 및 데이터베이스 사용자에 대한 액세스를 허용하고 모든 사용자를 생성할 수 있습니다. IAM 정책에 와일드카드 문자를 사용하면 너무 허용적일 수 있습니다. 가장 좋은 방법은 해당 업무 용도에 맞는 정책 중 가장 제한적인 정책을 사용하는 것입니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "redshift:GetClusterCredentials",
      "redshift:CreateClusterUser",
      "redshift:JoinGroup"
    ],
    "Resource": "*"
  }
}
```

자세한 내용은 [Amazon Redshift ARN 구문](#)을 참조하십시오.

Amazon Redshift용 네이티브 자격 증명 공급자(IdP) 페더레이션

네이티브 자격 증명 공급자 페더레이션을 사용하면 기존 자격 증명 공급자를 활용하여 인증 및 권한 관리를 간소화하기 때문에 Amazon Redshift에 대한 자격 증명 및 권한 관리가 더 쉬워집니다. 이는 자격 증명 공급자로부터 Redshift로 자격 증명 메타데이터를 공유할 수 있기 때문에 가능합니다. 이 기능을 처음 사용 시 지원되는 자격 증명 공급자는 [Microsoft Azure Active Directory\(Azure AD\)](#)입니다.

서드 파티 자격 증명 공급자의 자격 증명을 인증할 수 있도록 Amazon Redshift를 구성하려면 Amazon Redshift에 자격 증명 공급자를 등록합니다. 이렇게 하면 Redshift는 자격 증명 공급자가 정의한 사용자와 역할을 인증할 수 있습니다. 따라서 자격 증명 정보가 공유되므로 서드 파티 자격 증명 공급자와 Amazon Redshift 모두에서 세분화된 자격 증명 관리를 수행하지 않아도 됩니다.

ID 제공업체(iDP) 그룹에서 전송된 세션 역할을 사용하는 방법에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [PG_GET_SESSION_ROLES](#)를 참조하세요.

네이티브 ID 제공업체(idP) 페더레이션

자격 증명 공급자와 Amazon Redshift 간의 예비 설정을 완료하려면 몇 가지 단계를 수행합니다. 먼저 Amazon Redshift를 자격 증명 공급자에 서드 파티 애플리케이션으로 등록하고 필요한 API 권한을 요청합니다. 그런 다음 자격 증명 공급자에서 사용자 및 그룹을 생성합니다. 마지막으로 자격 증명 공급자에 고유한 인증 파라미터를 설정하는 SQL 문을 사용하여 Amazon Redshift에 자격 증명 공급자를 등

록합니다. 자격 증명 공급자를 Redshift에 등록하는 과정에서 네임스페이스를 할당하여 사용자와 역할이 올바르게 그룹화되도록 합니다.

Amazon Redshift에 자격 증명 공급자가 등록되면 Redshift와 자격 증명 공급자 간에 통신이 설정됩니다. 그런 다음 클라이언트는 토큰을 전달하고 Redshift에 자격 증명 공급자 엔터티로 인증할 수 있습니다. Amazon Redshift는 IdP 그룹 멤버십 정보를 사용하여 Redshift 역할에 매핑합니다. 사용자가 이전에 Redshift에 존재하지 않는 경우 사용자가 생성됩니다. 자격 증명 공급자 그룹에 매핑되는 역할이 존재하지 않는 경우 새로 생성됩니다. Amazon Redshift 관리자는 역할에 권한을 부여하며 사용자는 쿼리를 실행하고 다른 데이터베이스 작업을 수행할 수 있습니다.

다음 단계에서는 사용자가 로그인할 때 네이티브 자격 증명 공급자 페더레이션이 작동하는 방식을 간략하게 설명합니다.

1. 사용자가 네이티브 IdP 옵션을 사용하여 로그인하면 자격 증명 공급자 토큰이 클라이언트에서 드라이버로 전송됩니다.
2. 사용자가 인증됩니다. 사용자가 Amazon Redshift에 아직 존재하지 않으면 새 사용자가 생성됩니다. Redshift는 사용자의 자격 증명 공급자 그룹을 Redshift 역할에 매핑합니다.
3. 사용자의 Redshift 역할에 따라 권한이 할당됩니다. 권한은 관리자가 사용자 및 역할에 부여합니다.
4. 사용자가 Redshift를 쿼리할 수 있습니다.

데스크톱 클라이언트 도구

네이티브 자격 증명 공급자 페더레이션을 사용하여 Power BI로 Amazon Redshift에 연결하는 방법에 대한 지침은 블로그 게시물 [Integrate Amazon Redshift native IdP federation with Microsoft Azure Active Directory \(AD\) and Power BI](#)(Microsoft Azure Active Directory(AD) 및 Power BI와 Amazon Redshift 네이티브 IdP 페더레이션 통합)를 참조하세요. 이 블로그에서는 Azure AD를 사용한 Amazon Redshift 네이티브 IdP 설정의 단계별 구현을 설명합니다. Power BI Desktop 또는 Power BI 서비스에 대한 클라이언트 연결을 설정하는 단계를 자세히 설명합니다. 이 단계에는 애플리케이션 등록, 권한 구성 및 보안 인증 구성이 포함됩니다.

Power BI Desktop 및 JDBC Client-SQL Workbench/J를 사용하여 Amazon Redshift 기본 IdP(아이덴티티 제공업체) 페더레이션을 Azure AD와 통합하는 방법을 알아보려면 다음 비디오를 시청하세요.

네이티브 자격 증명 공급자 페더레이션을 사용하여 SQL 클라이언트, 특히 DBeaver 또는 SQL Workbench/J로 Amazon Redshift에 연결하는 방법에 대한 지침은 [Integrate Amazon Redshift native IdP federation with Microsoft Azure AD using a SQL client](#)(SQL 클라이언트를 사용하여 Microsoft Azure AD와 Amazon Redshift 네이티브 IdP 페더레이션 통합)를 참조하세요.

제한 사항

이러한 제한 사항이 적용됩니다.

- Amazon Redshift 드라이버는 다음 버전부터 BrowserIdcAuthPlugin을 지원합니다.
 - Amazon Redshift JDBC 드라이버 v2.1.0.30
 - Amazon Redshift ODBC 드라이버 v2.1.3
 - Amazon Redshift Python 드라이버 v2.1.3
- Amazon Redshift 드라이버는 다음 버전부터 IdpTokenAuthPlugin을 지원합니다.
 - Amazon Redshift JDBC 드라이버 v2.1.0.19
 - Amazon Redshift ODBC 드라이버 v2.0.0.9
 - Amazon Redshift Python 드라이버 v2.0.914
- 향상된 VPC 지원 안 함 - AWS IAM Identity Center를 사용하여 Redshift 신뢰할 수 있는 ID 전파를 구성할 때는 향상된 VPC가 지원되지 않습니다. 향상된 VPC에 대한 자세한 내용은 [Amazon Redshift의 Enhanced VPC Routing](#)을 참조하세요.
- AWS IAM Identity Center 캐싱 - AWS IAM Identity Center는 세션 정보를 캐싱합니다. 이 경우 Redshift 쿼리 에디터 v2를 통해 Redshift 쿼리 데이터베이스에 연결하려고 할 때 예기치 않은 액세스 문제가 발생할 수 있습니다. 데이터베이스 사용자가 AWS 콘솔에서 로그아웃한 경우에도 쿼리 에디터 v2의 연결된 AWS IAM Identity Center 세션이 유효하기 때문입니다. 캐시는 1시간 후에 만료되므로 일반적으로 모든 문제가 해결됩니다.

Amazon Redshift Redshift에서 자격 증명 공급자 설정

이 섹션에서는 네이티브 자격 증명 공급자 페더레이션을 위한 통신을 설정하도록 자격 증명 공급자와 Amazon Redshift를 구성하는 단계를 보여줍니다. 자격 증명 공급자의 활성 계정이 필요합니다. Amazon Redshift를 구성하기 전에 Redshift를 자격 증명 공급자에 애플리케이션으로 등록하고 관리자 동의를 얻습니다.

Amazon Redshift에서 다음 단계를 완료합니다.

1. SQL 문을 실행하여 Azure 애플리케이션 메타데이터에 대한 설명과 함께 자격 증명 공급자를 등록합니다. Amazon Redshift에서 자격 증명 공급자를 생성하려면 파라미터 값 issuer, client_id, client_secret 및 audience를 바꾼 후 다음 명령을 실행합니다. 이러한 파라미터는 Microsoft Azure AD에만 적용됩니다. 자격 증명 공급자 이름을 선택한 이름으로 바꾸고 네임스페이스를 자격 증명 공급자 디렉터리의 사용자와 역할을 포함하는 고유한 이름으로 바꿉니다.

```
CREATE IDENTITY PROVIDER oauth_standard TYPE azure
NAMESPACE 'aad'
PARAMETERS '{
  "issuer": "https://sts.windows.net/2sdfdsf-d475-420d-b5ac-667adad7c702/",
  "client_id": "<client_id>",
  "client_secret": "BUAH~ewrqewrqwerUUY^%tHe1oNZShoiU7",
  "audience": ["https://analysis.windows.net/powerbi/connector/AmazonRedshift"]
}'
```

azure 유형은 공급자가 특히 Microsoft Azure AD와의 통신을 용이하게 함을 나타냅니다. 현재 유일하게 지원되는 서드 파티 자격 증명 공급자입니다.

- issuer - 토큰을 수신할 때 신뢰할 수 있는 발급자 ID입니다. tenant_id의 고유 식별자가 발급자에 추가됩니다.
- client_id - 자격 증명 공급자에 등록된 애플리케이션의 고유한 퍼블릭 식별자입니다. 애플리케이션 ID라고도 합니다.
- client_secret - 자격 증명 공급자와 등록된 애플리케이션에게만 알려진 비밀 식별자 또는 암호입니다.
- audience - Azure의 애플리케이션에 할당된 애플리케이션 ID입니다.

공유 클라이언트 비밀을 사용하는 대신, 아이덴티티 제공업체를 생성할 때 파라미터를 설정하여 인증서, 프라이빗 키 및 프라이빗 키 암호를 지정할 수 있습니다.

```
CREATE IDENTITY PROVIDER example_idp TYPE azure
NAMESPACE 'example_aad'
PARAMETERS '{"issuer": "https://sts.windows.net/2sdfdsf-d475-420d-b5ac-667adad7c702/",
  "client_id": "<client_id>",
  "audience": ["https://analysis.windows.net/powerbi/connector/AmazonRedshift"],
  "client_x5t": "<certificate thumbprint>",
  "client_pk_base64": "<private key in base64 encoding>",
  "client_pk_password": "test_password"}';
```

프라이빗 키 암호 client_pk_password는 선택 사항입니다.

2. 선택 사항: Amazon Redshift에서 SQL 명령을 실행하여 사용자와 역할을 미리 생성합니다. 이를 통해 사전에 권한을 쉽게 부여할 수 있습니다. Amazon Redshift에서 역할 이름의 형식은 <Namespace>: <GroupName on Azure AD>입니다. 예를 들어 이름이 rsgroup인 Microsoft

Azure AD 그룹과 이름이 aad인 네임스페이스를 생성한 경우 역할 이름은 aad:rsgroup입니다. Amazon Redshift의 사용자 및 역할 이름은 ID 제공업체 네임스페이스에서 이 사용자 이름 및 그룹 멤버십으로 정의됩니다.

역할 및 사용자의 매핑 작업에는 external_id 값을 통해 최신 상태인지 확인하는 것이 포함됩니다. 외부 ID가 ID 공급자의 그룹 또는 사용자의 식별자에 매핑됩니다. 예를 들어 역할의 외부 ID는 해당 Azure AD 그룹 ID에 매핑됩니다. 마찬가지로 각 사용자의 외부 ID는 ID 공급자의 ID에 매핑됩니다.

```
create role "aad:rsgroup";
```

- 요구 사항에 따라 역할에 관련 권한을 부여합니다. 예를 들면 다음과 같습니다.

```
GRANT SELECT on all tables in schema public to role "aad:rsgroup";
```

- 특정 사용자에게 권한을 부여할 수도 있습니다.

```
GRANT SELECT on table foo to aad:alice@example.com
```

페더레이션된 외부 사용자의 역할 멤버십은 해당 사용자의 세션에서만 사용할 수 있다는 점에 유의하세요. 이는 데이터베이스 객체 생성에 영향을 미칩니다. 예를 들어 페더레이션된 외부 사용자가 뷰 또는 저장 프로시저를 만드는 경우 동일한 사용자는 해당 개체의 권한을 다른 사용자 및 역할에 위임할 수 없습니다.

네임스페이스에 대한 설명

네임스페이스는 사용자 또는 역할을 특정 자격 증명 공급자에 매핑합니다. 예를 들어, AWS IAM에서 생성한 사용자의 접두사는 iam:입니다. 이 접두사는 사용자 이름 충돌을 방지하고 여러 ID 저장소를 지원할 수 있게 합니다. aad 네임스페이스에 등록된 자격 증명 소스의 alice@example.com 사용자가 로그인할 때 aad:alice@example.com 사용자가 존재하지 않는 경우 Redshift에서 사용자가 생성됩니다. 사용자 및 역할 네임스페이스는 클러스터와 연결된 고유 식별자인 Amazon Redshift 클러스터 네임스페이스와 기능이 다릅니다.

ID 제공업체에 대한 Amazon Redshift 역할 자동 생성

이 기능을 사용하면 ID 제공업체(idP)의 그룹 멤버십을 기반으로 Redshift에서 역할을 자동으로 생성할 수 있습니다. 자동 생성 역할은 기본 IdP 통합을 통해 Azure Active Directory를 지원합니다.

역할을 자동으로 생성하면 몇 가지 이점이 있습니다. 역할을 자동 생성할 때 Redshift는 IdP에 그룹 멤버십이 있는 역할을 생성하므로 번거로운 수동 역할 생성 및 유지 관리를 피할 수 있습니다. Redshift 역할에 매핑되는 그룹을 필터링할 수도 있습니다.

작동 방식

IdP 사용자로서 Redshift에 로그인하면 다음과 같은 이벤트 시퀀스가 발생합니다.

1. Redshift는 IdP에서 그룹 멤버십을 검색합니다.
2. Redshift는 역할 형식 *idp_namespace:rolename*을 사용하여 해당 그룹에 대한 역할 매핑을 자동으로 생성합니다.
3. Redshift는 매핑된 역할에 대한 권한을 부여합니다.

각 사용자 로그인 시 카탈로그에는 없지만 사용자가 속한 각 그룹이 자동으로 생성됩니다. 선택적으로 필터 포함 및 제외를 설정하여 Redshift 역할이 생성된 IdP 그룹을 제어할 수 있습니다.

역할 자동 생성 구성

CREATE IDENTITY PROVIDER 및 ALTER IDENTITY PROVIDER 명령을 사용하여 자동 역할 생성을 활성화하고 구성합니다.

```
-- Create a new IdP with auto role creation enabled
CREATE IDENTITY PROVIDER <idp_name> TYPE azure
  NAMESPACE '<namespace>'
  APPLICATION_ARN '<app_arn>'
  IAM_ROLE '<role_arn>'
  AUTO_CREATE_ROLES TRUE;

-- Enable on existing IdP
ALTER IDENTITY PROVIDER <idp_name>
  AUTO_CREATE_ROLES TRUE;

-- Disable
ALTER IDENTITY PROVIDER <idp_name>
  AUTO_CREATE_ROLES FALSE;
```

그룹 필터링

선택적으로 INCLUDE 및 EXCLUDE 패턴을 사용하여 Redshift 역할에 매핑되는 IdP 그룹을 필터링할 수 있습니다. 패턴이 충돌하면 EXCLUDE가 INCLUDE보다 우선합니다.

```
-- Only create roles for groups with 'dev'
CREATE IDENTITY PROVIDER <idp_name> TYPE azure
...
  AUTO_CREATE_ROLES TRUE
  INCLUDE GROUPS LIKE '%dev%';

-- Exclude 'test' groups
ALTER IDENTITY PROVIDER <idp_name>
  AUTO_CREATE_ROLES TRUE
  EXCLUDE GROUPS LIKE '%test%';
```

예제

다음 예제에서는 필터링 없이 역할 자동 생성을 활성화하는 방법을 보여줍니다.

```
CREATE IDENTITY PROVIDER prod_idc TYPE azure ...
  AUTO_CREATE_ROLES TRUE;
```

다음 예제에는 개발 그룹이 포함되며 테스트 그룹은 제외됩니다.

```
ALTER IDENTITY PROVIDER prod_idc
  AUTO_CREATE_ROLES TRUE
  INCLUDE GROUPS LIKE '%dev%'
  EXCLUDE GROUPS LIKE '%test%';
```

모범 사례

역할에 대해 자동 생성을 활성화할 때 다음 모범 사례를 고려하세요.

- INCLUDE 및 EXCLUDE 필터를 사용하여 역할을 가져오는 그룹을 제어합니다.
- 역할을 정기적으로 감사하고 사용하지 않는 역할을 정리합니다.
- Redshift 역할 계층 구조를 활용하여 권한 관리를 간소화합니다.

Single Sign-On 경험을 위해 Redshift와 AWS IAM Identity Center 연결

신뢰할 수 있는 ID 전파를 통해 Amazon Redshift 데이터 웨어하우스에 대한 사용자 및 그룹 액세스를 관리할 수 있습니다.

[신뢰할 수 있는 ID 전파](#)는 연결된 AWS 서비스의 관리자가 서비스 데이터에 대한 액세스 권한을 부여하고 감사하는 데 사용할 수 있는 AWS IAM Identity Center 기능입니다. 이 데이터에 대한 액세스는 그

그룹 연결과 같은 사용자 속성을 기반으로 합니다. 신뢰할 수 있는 ID 전파를 설정하려면 연결된 AWS 서비스의 관리자와 IAM Identity Center 관리자 간의 협업이 필요합니다. 자세한 내용은 [사전 조건 및 고려 사항](#)을 참조하세요.

한 가지 엔드 투 엔드 사례를 설명하자면, Amazon Quick 대시보드 또는 Amazon Redshift 쿼리 에디터 v2를 사용하여 Redshift에 액세스할 수 있습니다. 이 경우 액세스는 AWS IAM Identity Center 그룹을 기반으로 합니다. Redshift는 사용자의 신원과 그룹 멤버십을 확인할 수 있습니다. AWS IAM Identity Center에서는 Okta 또는 PingOne과 같은 서드 파티 ID 제공업체(idP)를 통해 ID를 연결하고 관리할 수 있습니다.

관리자가 Redshift와 AWS IAM Identity Center 간의 연결을 설정한 후에는 ID 제공업체 그룹을 기반으로 세분화된 액세스를 구성하여 데이터에 대한 사용자 액세스 권한을 부여할 수 있습니다.

Important

AWS IAM Identity Center 또는 연결된 ID 제공업체(idP) 디렉터리에서 사용자를 삭제하는 경우 Amazon Redshift 카탈로그에서 사용자가 자동으로 삭제되지는 않습니다. Amazon Redshift 카탈로그에서 사용자를 수동으로 삭제하려면 DROP USER 명령을 실행하여 AWS IAM Identity Center 또는 idP에서 제거된 사용자를 완전히 삭제합니다. 사용자 제거에 관한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서에 나와 있는 [DROP USER](#)를 참고하시기 바랍니다.

Redshift와 AWS IAM Identity Center를 통합하여 얻을 수 있는 이점

Redshift와 함께 AWS IAM Identity Center를 사용하면 조직이 다음과 같은 이점을 얻을 수 있습니다.

- Amazon Quick의 대시보드 작성자는 암호를 다시 입력하거나 관리자가 복잡한 권한으로 IAM 역할을 설정할 필요 없이 Redshift 데이터 소스에 연결할 수 있습니다.
- AWS IAM Identity Center는 AWS의 직원 사용자를 위한 중앙 위치를 제공합니다. AWS IAM Identity Center에서 직접 사용자 및 그룹을 만들거나 Okta, PingOne 또는 Microsoft Entra ID(Azure AD)와 같은 표준 기반 ID 제공업체에서 관리하는 기존 사용자 및 그룹을 연결할 수 있습니다. AWS IAM Identity Center는 사용자 및 그룹을 위해 선택한 정보 소스로 인증을 지시하고 Redshift에서 액세스할 수 있는 사용자 및 그룹 디렉터리를 유지 관리합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [ID 소스 관리](#) 및 [지원되는 ID 제공업체](#)를 참조하세요.
- 간단한 자동 검색 및 연결 기능을 사용하여 하나의 AWS IAM Identity Center 인스턴스를 여러 Redshift 클러스터 및 작업 그룹과 공유할 수 있습니다. 따라서 각 클러스터에 대해 AWS IAM Identity Center 연결을 별도로 구성하지 않고도 클러스터를 빠르게 추가할 수 있으며, 모든 클러스

터와 작업 그룹이 사용자, 사용자 속성 및 그룹을 일관되게 볼 수 있습니다. 단, 조직의 AWS IAM Identity Center 인스턴스는 연결하려는 Redshift 데이터 공유와 동일한 리전에 있어야 합니다.

- 사용자 ID는 데이터 액세스와 함께 알려지고 로깅되므로 AWS CloudTrail에서 사용자 액세스 감사를 통해 규정을 더 쉽게 충족할 수 있습니다.

애플리케이션 연결을 위한 관리자 페르소나

다음은 분석 애플리케이션을 Redshift용 AWS IAM Identity Center 관리형 애플리케이션에 연결하는 데 있어 핵심이 되는 페르소나입니다.

- 애플리케이션 관리자 - 애플리케이션을 생성하고 ID-토큰 교환을 가능하게 할 서비스를 구성합니다. 또한 이 관리자는 애플리케이션에 액세스할 수 있는 사용자 또는 그룹을 지정합니다.
- 데이터 관리자 - 데이터에 대한 세분화된 액세스를 구성합니다. AWS IAM Identity Center의 사용자 및 그룹은 특정 권한에 매핑될 수 있습니다.

Amazon Quick를 통해 AWS IAM Identity Center와 연결된 Amazon Redshift에 연결

[Quick에서 Amazon Redshift 클러스터로의 연결 승인](#)에서는 Redshift가 AWS IAM Identity Center에 연결되고 IAM Identity Center를 통해 액세스가 관리되는 경우 Quick을 사용하여 Redshift를 인증하는 방법을 보여 줍니다. 이 단계는 Amazon Redshift Serverless에도 적용됩니다.

Amazon Redshift 쿼리 에디터 v2를 통해 AWS IAM Identity Center와 연결된 Amazon Redshift에 연결

Redshift와 AWS IAM Identity Center 연결을 설정하는 단계를 완료하면 사용자는 AWS IAM Identity Center 기반의 네임스페이스 접두사가 붙은 ID를 통해 데이터베이스 및 데이터베이스의 적절한 객체에 액세스할 수 있습니다. 쿼리 에디터 v2 로그인으로 Redshift 데이터베이스에 연결하는 방법에 대한 자세한 내용은 [Query Editor V2를 사용하여 데이터베이스 쿼리](#) 섹션을 참조하세요.

여러 AWS 리전에서 AWS IAM Identity Center 사용

Amazon Redshift는 여러 AWS 리전에서 AWS IAM Identity Center를 지원합니다. AWS IAM Identity Center를 기본 AWS 리전에서 추가 리전으로 확장하여 사용자와의 근접성과 신뢰성을 통해 성능을 개선할 수 있습니다. AWS IAM Identity Center에 새 리전이 추가되면 기본 리전에서 ID를 복제하지 않고도 새 리전에서 Redshift IAM Identity Center 애플리케이션을 생성할 수 있습니다. 행 수준, 열 수준 및 마스킹 제어를 활성화할 수 있는 새 리전에서 AWS IAM Identity Center를 사용하여 Amazon Redshift 페더레이션 권한을 설정할 수 있습니다. 여러 리전에서 AWS IAM Identity Center를 시작하는 방법에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [여러 AWS 리전에서 AWS IAM Identity Center 관리](#)를 참조하세요.

AWS IAM Identity Center를 사용하여 Amazon Redshift에 연결 시 제한 사항

AWS IAM Identity Center Single Sign-On을 사용할 때는 다음 제한 사항을 고려하세요.

- 향상된 VPC 지원 없음 - Amazon Redshift에 AWS IAM Identity Center Single Sign-On을 사용하는 경우 향상된 VPC가 지원되지 않습니다. 향상된 VPC에 대한 자세한 내용은 [Amazon Redshift의 Enhanced VPC Routing](#)을 참조하세요.

Amazon Redshift와 AWS IAM Identity Center 통합 설정

Amazon Redshift 클러스터 관리자 또는 Amazon Redshift Serverless 관리자는 여러 단계를 수행하여 Redshift를 AWS IAM Identity Center 지원 애플리케이션으로 구성해야 합니다. 이렇게 하면 Redshift는 AWS IAM Identity Center를 자동으로 검색하고 연결하여 로그인 및 사용자 디렉터리 서비스를 받을 수 있습니다. 그런 다음 Redshift 관리자가 클러스터 또는 작업 그룹을 생성할 때 새 데이터 웨어하우스에서 AWS IAM Identity Center를 사용하여 데이터베이스 액세스를 관리하도록 할 수 있습니다.

Redshift를 AWS IAM Identity Center 관리형 애플리케이션으로 활성화하는 이유는 AWS IAM Identity Center 내에서 또는 통합된 타사 ID 제공업체로부터 사용자 및 그룹 권한을 제어할 수 있도록 하기 위해서입니다. 데이터베이스 사용자(예: 분석가 또는 데이터 과학자)가 Redshift 데이터베이스에 로그인 하면 AWS IAM Identity Center의 해당 그룹이 Redshift의 역할 이름과 일치하는지 확인합니다. 이러한 방식으로 Redshift 데이터베이스 역할의 이름을 정의하는 그룹은 예를 들어 판매 분석을 위한 테이블 세트에 액세스할 수 있습니다. 다음 섹션에서는 설정 방법을 안내합니다.

사전 조건

AWS IAM Identity Center를 Amazon Redshift와 통합하기 위한 사전 조건은 다음과 같습니다.

- 계정 구성 - 계정 간 사용 사례를 계획하거나 동일한 AWS IAM Identity Center 인스턴스로 서로 다른 계정에서 Redshift 클러스터를 사용하는 경우 AWS 조직의 관리 계정에서 AWS IAM Identity Center를 구성해야 합니다. 여기에는 ID 소스 구성이 포함됩니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [시작하기](#), [작업 인력 ID](#) 및 [지원되는 ID 제공업체](#)를 참조하세요. Redshift의 데이터에 할당하려면 먼저 AWS IAM Identity Center에서 사용자 또는 그룹을 생성했거나 ID 소스에서 사용자 및 그룹을 동기화해야 합니다.

Note

Redshift와 AWS IAM Identity Center가 동일한 계정에 있는 경우 AWS IAM Identity Center의 계정 인스턴스를 사용할 수 있습니다. Redshift 클러스터 또는 작업 그룹을 만들고 구성할 때 위젯을 사용하여 이 인스턴스를 만들 수 있습니다.

- 신뢰할 수 있는 토큰 발급자 구성 - 경우에 따라 신뢰 토큰을 발급하고 확인할 수 있는 기관인 신뢰할 수 있는 토큰 발급자를 사용해야 할 수도 있습니다. 이를 위해서는 AWS IAM Identity Center 통합을 구성하는 Redshift 관리자가 신뢰할 수 있는 토큰 발급자를 선택하고 필요한 속성을 추가하여 구성을 완료하기 전에 사전 단계를 거쳐야 합니다. 여기에는 신뢰할 수 있는 토큰 발급자 역할을 하도록 외부 ID 제공업체를 구성하고 AWS IAM Identity Center 콘솔에서 해당 속성을 추가하는 것이 포함될 수 있습니다. 이 단계를 완료하려면 [신뢰할 수 있는 토큰 발급자를 사용하여 애플리케이션 사용](#)을 참조하세요.

Note

모든 외부 연결에 신뢰할 수 있는 토큰 발급자를 설정할 필요는 없습니다. Amazon Redshift 쿼리 에디터 v2를 사용하여 Redshift 데이터베이스에 연결할 때는 신뢰할 수 있는 토큰 발급자 구성이 필요하지 않습니다. 하지만 대시보드와 같은 타사 애플리케이션 또는 ID 제공업체를 통해 인증하는 사용자 지정 애플리케이션에도 적용될 수 있습니다.

- 하나 이상의 IAM 역할 구성 - 다음 섹션에는 구성해야 하는 권한이 언급되어 있습니다. IAM 모범 사례에 따라 권한을 추가해야 합니다. 구체적인 권한은 다음 절차에 자세히 설명되어 있습니다.

자세한 내용은 [Getting Started with AWS IAM Identity Center](#)를 참조하세요.

AWS IAM Identity Center 작업을 위해 ID 제공업체 구성

사용자 및 그룹 ID 관리를 제어하는 첫 번째 단계는 AWS IAM Identity Center에 연결하고 ID 제공업체를 구성하는 것입니다. AWS IAM Identity Center 자체를 ID 제공업체로 사용하거나 Okta와 같은 타사 ID 저장소를 연결할 수 있습니다. ID 제공업체에 대한 연결 설정 및 구성에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [외부 ID 제공업체에 연결](#)을 참조하세요. 이 프로세스 말미에 테스트하기 위해 AWS IAM Identity Center에 소수의 사용자 및 그룹을 추가했는지 확인하세요.

관리 권한

Redshift/AWS IAM Identity Center 애플리케이션 수명 주기 관리에 필요한 권한

Redshift 관리자가 AWS IAM Identity Center와 함께 사용하도록 Redshift를 구성하는 데 사용하는 IAM ID를 만들어야 합니다. 대부분의 경우 권한이 있는 IAM 역할을 생성하고 필요에 따라 다른 ID에 할당합니다. 다음 작업을 수행할 수 있는 권한이 나열되어 있어야 합니다.

Redshift/AWS IAM Identity Center 애플리케이션 생성

- `sso:PutApplicationAssignmentConfiguration` - 보안에 사용됩니다.
- `sso:CreateApplication` - AWS IAM Identity Center 애플리케이션을 생성하는 데 사용됩니다.
- `sso:PutApplicationAuthenticationMethod` - Redshift 인증 액세스 권한을 부여합니다.
- `sso:PutApplicationGrant` - 신뢰할 수 있는 토큰 발급자 정보를 변경하는 데 사용됩니다.
- `sso:PutApplicationAccessScope` - Redshift AWS IAM Identity Center 애플리케이션을 설정하는 데 사용됩니다. 여기에는 AWS Lake Formation용 및 [Amazon S3 Access Grants](#)용이 포함됩니다.
- `redshift:CreateRedshiftIdcApplication` - Redshift AWS IAM Identity Center 애플리케이션을 생성하는 데 사용됩니다.

Redshift/AWS IAM Identity Center 애플리케이션 설명

- `sso:GetApplicationGrant` - 신뢰할 수 있는 토큰 발급자 정보를 나열하는 데 사용됩니다.
- `sso:ListApplicationAccessScopes` - Redshift AWS IAM Identity Center 애플리케이션 설정에서 다운스트림 통합(예: AWS Lake Formation용 및 S3 Access Grants용)을 나열하는 데 사용됩니다.
- `redshift:DescribeRedshiftIdcApplications` - 기존 AWS IAM Identity Center 애플리케이션을 설명하는 데 사용됩니다.

Redshift/AWS IAM Identity Center 애플리케이션 변경

- `redshift:ModifyRedshiftIdcApplication` - 기존 Redshift 애플리케이션을 변경하는 데 사용됩니다.
- `sso:UpdateApplication` - AWS IAM Identity Center 애플리케이션을 업데이트하는 데 사용됩니다.
- `sso:GetApplicationGrant` - 신뢰할 수 있는 토큰 발급자 정보를 가져옵니다.

- `sso:ListApplicationAccessScopes` - Redshift AWS IAM Identity Center 애플리케이션을 설정하는 데 사용됩니다.
- `sso:DeleteApplicationGrant` - 신뢰할 수 있는 토큰 발급자 정보를 삭제합니다.
- `sso:PutApplicationGrant` - 신뢰할 수 있는 토큰 발급자 정보를 변경하는 데 사용됩니다.
- `sso:PutApplicationAccessScope` - Redshift AWS IAM Identity Center 애플리케이션을 설정하는 데 사용됩니다. 여기에는 AWS Lake Formation용 및 [Amazon S3 Access Grants](#)용이 포함됩니다.
- `sso:DeleteApplicationAccessScope` - Redshift AWS IAM Identity Center 애플리케이션을 설정을 삭제하는 데 사용됩니다. 여기에는 AWS Lake Formation용 및 [Amazon S3 Access Grants](#)용이 포함됩니다.

Redshift/AWS IAM Identity Center 애플리케이션 삭제

- `sso:DeleteApplication` - AWS IAM Identity Center 애플리케이션을 삭제하는 데 사용됩니다.
- `redshift:DeleteRedshiftIdcApplication` - 기존 Redshift AWS IAM Identity Center 애플리케이션을 삭제할 수 있는 기능을 제공합니다.

Redshift/쿼리 에디터 v2 애플리케이션 수명 주기 관리에 필요한 권한

Redshift 관리자가 AWS IAM Identity Center와 함께 사용하도록 Redshift를 구성하는 데 사용하는 IAM ID를 만들어야 합니다. 대부분의 경우 권한이 있는 IAM 역할을 생성하고 필요에 따라 다른 ID에 할당합니다. 다음 작업을 수행할 수 있는 권한이 나열되어 있어야 합니다.

쿼리 에디터 v2 애플리케이션 생성

- `redshift:CreateQev2IdcApplication` - QEV2 애플리케이션을 생성하는 데 사용됩니다.
- `sso:CreateApplication` - AWS IAM Identity Center 애플리케이션을 생성하는 권한을 부여합니다.
- `sso:PutApplicationAuthenticationMethod` - Redshift 인증 액세스 권한을 부여합니다.
- `sso:PutApplicationGrant` - 신뢰할 수 있는 토큰 발급자 정보를 변경하는 데 사용됩니다.
- `sso:PutApplicationAccessScope` - Redshift AWS IAM Identity Center 애플리케이션을 설정하는 데 사용됩니다. 여기에는 쿼리 에디터 v2가 포함됩니다.
- `sso:PutApplicationAssignmentConfiguration` - 보안에 사용됩니다.

쿼리 에디터 v2 애플리케이션 설명

- `redshift:DescribeQev2IdcApplications` - AWS IAM Identity Center QEV2 애플리케이션을 설명하는 데 사용됩니다.

쿼리 에디터 v2 애플리케이션 변경

- `redshift:ModifyQev2IdcApplication` - AWS IAM Identity Center QEV2 애플리케이션을 변경하는 데 사용됩니다.
- `sso:UpdateApplication` - AWS IAM Identity Center QEV2 애플리케이션을 변경하는 데 사용됩니다.

쿼리 에디터 v2 애플리케이션 삭제

- `redshift>DeleteQev2IdcApplication` - QEV2 애플리케이션을 삭제하는 데 사용됩니다.
- `sso>DeleteApplication` - QEV2 애플리케이션을 삭제하는 데 사용됩니다.

Note

Amazon Redshift SDK에서는 다음과 같은 API를 사용할 수 없습니다.

- `CreateQev2IdcApplication`
- `DescribeQev2IdcApplications`
- `ModifyQev2IdcApplication`
- `DeleteQev2IdcApplication`

이러한 작업은 AWS 콘솔에서 AWS IAM Identity Center를 Redshift QEV2와 통합하는 경우에 만 사용할 수 있습니다. 자세한 내용은 [Actions defined by Amazon Redshift](#)를 참조하세요.

콘솔에서 새 리소스를 연결하기 위해 데이터베이스 관리자에게 필요한 권한

생성 프로세스 중에 새로 프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹을 연결하려면 이러한 권한이 필요합니다. 이러한 권한이 있는 경우 콘솔에 Redshift용 AWS IAM Identity Center 관리형 애플리케이션에 연결하도록 선택할 수 있는 선택 항목이 표시됩니다.

- `redshift:DescribeRedshiftIdcApplications`
- `sso:ListApplicationAccessScopes`

- sso:GetApplicationAccessScope
- sso:GetApplicationGrant

가장 좋은 방법은 권한 정책을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에 할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.

AWS IAM Identity Center와 연결된 AWS 관리형 애플리케이션으로 Redshift 설정

AWS IAM Identity Center에서 Amazon Redshift 프로비저닝 클러스터 또는 Amazon Redshift Serverless 작업 그룹의 ID를 관리하려면 먼저 Redshift 관리자가 Redshift를 AWS IAM Identity Center 관리형 애플리케이션으로 생성하기 위한 단계를 완료해야 합니다.

1. Amazon Redshift 또는 Amazon Redshift Serverless 콘솔 메뉴에서 AWS IAM Identity Center 통합을 선택한 다음 AWS IAM Identity Center에 연결을 선택합니다. 여기에서 일련의 선택 단계를 수행하여 AWS IAM Identity Center 통합을 위한 속성을 채웁니다.
2. Redshift의 AWS IAM Identity Center 관리형 애플리케이션에 대한 표시 이름과 고유한 이름을 선택합니다.
3. 조직의 네임스페이스를 지정합니다. 일반적으로 조직 이름의 축약된 버전입니다. Redshift 데이터베이스에서 AWS IAM Identity Center 관리형 사용자 및 역할의 접두사로 추가됩니다.
4. 사용할 IAM 역할을 선택합니다. 이 IAM 역할은 Redshift에 사용되는 다른 역할과 분리되어야 하며 다른 용도로는 사용하지 않는 것이 좋습니다. 구체적인 필수 정책 권한은 다음과 같습니다.
 - sso:DescribeApplication - 카탈로그에 ID 제공업체(idP) 항목을 생성하는 데 필요합니다.
 - sso:DescribeInstance - IdP 페더레이션 역할 또는 사용자를 수동으로 생성하는 데 사용됩니다.
5. 클라이언트 연결 및 신뢰할 수 있는 토큰 발급자를 구성합니다. 신뢰할 수 있는 토큰 발급자를 구성하면 외부 ID 제공업체와의 관계를 설정하여 신뢰할 수 있는 ID 전파가 용이해집니다. 예를 들어 ID 전파를 통해 사용자는 한 애플리케이션에 로그인하고 다른 애플리케이션의 특정 데이터에 액세스할 수 있습니다. 이를 통해 사용자는 서로 다른 위치에서 보다 원활하게 데이터를 수집할 수 있습니다. 이 단계에서는 콘솔에서 각 신뢰할 수 있는 토큰 발급자의 속성을 설정합니다. 속성에는 도구 또는 서비스의 구성 속성에서 가져와야 할 수 있는 이름 및 대상 클레임(aud 클레임이라고도 함)이 포함됩니다. 타사 도구의 JSON 웹 토큰(JWT)에서 가져온 애플리케이션 이름을 제공해야 할 수도 있습니다.

Note

각 타사 도구 또는 서비스에 필요한 `aud claim`은 토큰 유형(ID 제공업체가 발급한 액세스 토큰일 수 있음) 또는 ID 토큰 등의 다른 유형에 따라 달라질 수 있습니다. 벤더마다 다를 수 있습니다. 신뢰할 수 있는 ID 전파를 구현하고 Redshift와 통합할 때는 타사 도구가 AWS에 전송하는 토큰 유형에 대해 올바른 `aud` 값을 제공해야 합니다. 도구 또는 서비스 벤더의 권장 사항을 확인하세요.

신뢰할 수 있는 ID 전파에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [신뢰할 수 있는 ID 전파 개요](#)를 참조하세요.

Redshift 관리자가 단계를 완료하고 구성을 저장하면 Redshift 콘솔에 AWS IAM Identity Center 속성이 표시됩니다. 또한 시스템 뷰 [SVV_IDENTITY_PROVIDERS](#)를 쿼리하여 애플리케이션의 속성을 확인할 수 있습니다. 여기에는 애플리케이션 이름과 네임스페이스가 포함됩니다. 네임스페이스는 애플리케이션과 연결된 Redshift 데이터베이스 객체의 접두사로 사용됩니다. 이러한 작업을 완료하면 Redshift는 AWS IAM Identity Center 지원 애플리케이션이 됩니다. 콘솔의 속성에는 통합 상태가 포함됩니다. 통합이 완료되면 활성화됨이라고 표시됩니다. 이 프로세스가 끝나면 각각의 새 클러스터에서 AWS IAM Identity Center 통합을 활성화할 수 있습니다.

구성 후에는 사용자 또는 그룹 탭을 선택하고 할당을 선택하여 AWS IAM Identity Center에 있는 사용자 및 그룹을 Redshift에 포함할 수 있습니다.

새 Amazon Redshift 클러스터 또는 Amazon Redshift Serverless 작업 그룹에 AWS IAM Identity Center 통합 활성화

데이터베이스 관리자는 새로운 Redshift 리소스가 AWS IAM Identity Center와 연계되어 작동하도록 구성하여 로그인 및 데이터 액세스를 더 용이하게 만듭니다. 이는 프로비저닝된 클러스터 또는 서버리스 작업 그룹을 생성하는 단계의 일부로 수행됩니다. Redshift 리소스를 생성할 수 있는 권한이 있는 모든 사용자는 AWS IAM Identity Center 통합 태스크를 수행할 수 있습니다. 프로비저닝된 클러스터를 생성할 때 Amazon Redshift 콘솔에서 클러스터 생성을 선택하는 것부터 시작하면 됩니다. 다음 단계는 데이터베이스에 대한 AWS IAM Identity Center 관리를 활성화하는 방법을 보여줍니다. 클러스터를 생성하는 모든 단계가 포함되어 있지는 않습니다.

1. `create-cluster` 단계에서 IAM Identity Center 통합 섹션의 <클러스터 이름>에 대해 활성화를 선택합니다.

2. 통합을 활성화하는 프로세스에 한 단계가 있습니다. 콘솔에서 IAM Identity Center 통합 활성화를 선택하면 됩니다.
3. 새 클러스터 또는 작업 그룹의 경우 SQL 명령을 사용하여 Redshift에서 데이터베이스 역할을 생성합니다. 다음은 명령입니다.

```
CREATE ROLE <idcnamespace:rolename>;
```

네임스페이스와 역할 이름은 다음과 같습니다.

- IAM Identity Center 네임스페이스 접두사 - AWS IAM Identity Center와 Redshift 간의 연결을 설정할 때 정의한 네임스페이스입니다.
- 역할 이름 - 이 Redshift 데이터베이스 역할은 AWS IAM Identity Center의 그룹 이름과 일치해야 합니다.

Redshift는 AWS IAM Identity Center와 연결하여 데이터베이스 역할을 생성하고 AWS IAM Identity Center 그룹에 매핑하는 데 필요한 정보를 가져옵니다.

새 데이터 웨어하우스가 생성되면 AWS IAM Identity Center 통합을 위해 지정된 IAM 역할이 프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹에 자동으로 연결된다는 점에 유의하세요. 필요한 클러스터 메타데이터를 입력하고 리소스를 생성한 후에는 속성에서 AWS IAM Identity Center 통합 상태를 확인할 수 있습니다. AWS IAM Identity Center의 그룹 이름에 공백이 있는 경우 매칭되는 역할을 생성할 때 SQL에서 따옴표를 사용해야 합니다.

Redshift 데이터베이스를 활성화하고 역할을 생성했으면 Amazon Redshift 쿼리 에디터 v2 또는 Amazon Quick를 사용하여 데이터베이스에 연결할 준비가 된 것입니다. 자세한 내용은 다음 섹션에 자세히 설명되어 있습니다.

API를 사용하여 기본 **RedshiftIdcApplication** 설정

설정은 ID 관리자가 수행합니다. API를 사용하여 AWS IAM Identity Center 내의 Redshift 애플리케이션을 나타내는 RedshiftIdcApplication을 만들고 채웁니다.

1. 먼저 사용자를 생성하고 AWS IAM Identity Center의 그룹에 추가할 수 있습니다. AWS IAM Identity Center용 AWS 콘솔에서 이 작업을 수행하면 됩니다.
2. AWS IAM Identity Center 애플리케이션을 생성하고 Redshift와 호환해서 사용하도록 `create-redshift-idc-application`을 호출합니다. 필요한 값을 입력하여 애플리케이션을 생성합니

다. 표시 이름은 AWS IAM Identity Center 대시보드에 표시할 이름입니다. IAM 역할 ARN은 AWS IAM Identity Center에 대한 권한이 있는 ARN이며 Redshift에서도 이 역할을 맡을 수 있습니다.

```
aws redshift create-redshift-idc-application
--idc-instance-arn 'arn:aws:sso:::instance/ssoins-1234a01a1b12345d'
--identity-namespace 'MYCO'
--idc-display-name 'TEST-NEW-APPLICATION'
--iam-role-arn 'arn:aws:redshift:us-east-1:012345678901:role/TestRedshiftRole'
--redshift-idc-application-name 'myredshiftidcapplication'
```

다음 예는 create-redshift-idc-application에 대한 호출에서 반환된 샘플 RedshiftIdcApplication 응답을 보여줍니다.

```
"RedshiftIdcApplication": {
  "IdcInstanceArn": "arn:aws:sso:::instance/ssoins-1234a01a1b12345d",
  "RedshiftIdcApplicationName": "test-application-1",
  "RedshiftIdcApplicationArn": "arn:aws:redshift:us-east-1:012345678901:redshiftidcapplication:12aaa111-3ab2-3ab1-8e90-b2d72aea588b",
  "IdentityNamespace": "MYCO",
  "IdcDisplayName": "Redshift-Idc-Application",
  "IamRoleArn": "arn:aws:redshift:us-east-1:012345678901:role/TestRedshiftRole",
  "IdcManagedApplicationArn": "arn:aws:sso:::012345678901:application/ssoins-1234a01a1b12345d/apl-12345678910",
  "IdcOnboardStatus": "arn:aws:redshift:us-east-1:123461817589:redshiftidcapplication",
  "RedshiftIdcApplicationArn": "Completed",
  "AuthorizedTokenIssuerList": [
    "TrustedTokenIssuerArn": ...,
    "AuthorizedAudiencesList": [...]...
  ]
}
```

3. create-application-assignment를 사용하여 AWS IAM Identity Center의 관리형 애플리케이션에 특정 그룹이나 개별 사용자를 할당할 수 있습니다. 이렇게 하면 AWS IAM Identity Center를 통해 관리할 그룹을 지정할 수 있습니다. 데이터베이스 관리자가 Redshift에서 데이터베이스 역할을 생성하는 경우 AWS IAM Identity Center의 그룹 이름이 Redshift의 역할 이름에 매핑됩니다. 이 역할은 데이터베이스의 권한을 제어합니다. 자세한 내용은 [AWS IAM Identity Center 콘솔에서 애플리케이션에 사용자 액세스 권한 할당](#)을 참조하세요.

4. 애플리케이션을 활성화한 후에는 `create-cluster`를 호출하여 AWS IAM Identity Center에서 Redshift 관리형 애플리케이션 ARN을 포함합니다. 이렇게 하면 클러스터가 AWS IAM Identity Center의 관리형 애플리케이션과 연결됩니다.

AWS IAM Identity Center 애플리케이션을 기존 클러스터 또는 작업 그룹과 연결

AWS IAM Identity Center 통합을 활성화하려는 기존 클러스터 또는 작업 그룹이 있는 경우 SQL 명령을 실행하여 활성화할 수 있습니다. SQL 명령을 실행하여 통합 설정을 변경할 수도 있습니다. 자세한 내용은 [ALTER IDENTITY PROVIDER](#)를 참조하세요.

기존 ID 제공업체를 삭제할 수도 있습니다. 다음 예에서는 CASCADE가 ID 제공업체에 연결된 사용자 및 역할을 삭제하는 방법을 보여줍니다.

```
DROP IDENTITY PROVIDER
<provider_name> [ CASCADE ]
```

사용자 권한 설정

관리자는 ID 제공업체 내에서 또는 AWS IAM Identity Center 내에서 사용자의 ID 속성 및 그룹 멤버십을 기반으로 다양한 리소스에 대한 권한을 직접 구성합니다. 예를 들어, ID 제공업체 관리자는 맡은 역할에 적합한 그룹에 데이터베이스 엔지니어를 추가할 수 있습니다. 이 그룹 이름은 Redshift 데이터베이스 역할 이름에 매핑됩니다. 역할은 Redshift의 특정 테이블 또는 뷰에 대한 액세스를 제공하거나 제한합니다.

AWS IAM Identity Center에 대한 Amazon Redshift 역할 자동 생성

이 기능은 그룹 멤버십을 기반으로 Redshift에서 역할을 자동으로 생성할 수 있는 AWS IAM Identity Center와 통합됩니다.

역할을 자동으로 생성하면 몇 가지 이점이 있습니다. 역할을 자동 생성할 때 Redshift는 IdP에 그룹 멤버십이 있는 역할을 생성하므로 번거로운 수동 역할 생성 및 유지 관리를 피할 수 있습니다. 또한 포함 및 제외 패턴을 사용하여 Redshift 역할에 매핑되는 그룹을 필터링할 수 있습니다.

작동 방식

IdP 사용자로서 Redshift에 로그인하면 다음과 같은 이벤트 시퀀스가 발생합니다.

1. Redshift는 IdP에서 그룹 멤버십을 검색합니다.
2. Redshift는 역할 형식 `idp_namespace:rolename`을 사용하여 해당 그룹에 대한 역할 매핑을 자동으로 생성합니다.

3. Redshift는 매핑된 역할에 대한 권한을 부여합니다.

각 사용자 로그인 시 카탈로그에는 없지만 사용자가 속한 각 그룹이 자동으로 생성됩니다. 선택적으로 필터 포함 및 제외를 설정하여 Redshift 역할이 생성된 IdP 그룹을 제어할 수 있습니다.

역할 자동 생성 구성

CREATE IDENTITY PROVIDER 및 ALTER IDENTITY PROVIDER 명령을 사용하여 자동 역할 생성을 활성화하고 구성합니다.

```
-- Create a new IdP with auto role creation enabled
CREATE IDENTITY PROVIDER <idp_name> TYPE AWSIDC
  NAMESPACE '<namespace>'
  APPLICATION_ARN '<app_arn>'
  IAM_ROLE '<role_arn>'
  AUTO_CREATE_ROLES TRUE;

-- Enable on existing IdP
ALTER IDENTITY PROVIDER <idp_name>
  AUTO_CREATE_ROLES TRUE;

-- Disable
ALTER IDENTITY PROVIDER <idp_name>
  AUTO_CREATE_ROLES FALSE;
```

그룹 필터링

선택적으로 INCLUDE 및 EXCLUDE 패턴을 사용하여 Redshift 역할에 매핑되는 IdP 그룹을 필터링할 수 있습니다. 패턴이 충돌하면 EXCLUDE가 INCLUDE보다 우선합니다.

```
-- Only create roles for groups with 'dev'
CREATE IDENTITY PROVIDER <idp_name> TYPE AWSIDC
  ...
  AUTO_CREATE_ROLES TRUE
  INCLUDE GROUPS LIKE '%dev%';

-- Exclude 'test' groups
ALTER IDENTITY PROVIDER <idp_name>
  AUTO_CREATE_ROLES TRUE
  EXCLUDE GROUPS LIKE '%test%';
```

예제

다음 예제에서는 필터링 없이 역할 자동 생성을 활성화하는 방법을 보여줍니다.

```
CREATE IDENTITY PROVIDER prod_idc TYPE AWSIDC ...  
    AUTO_CREATE_ROLES TRUE;
```

다음 예제에는 개발 그룹이 포함되며 테스트 그룹은 제외됩니다.

```
ALTER IDENTITY PROVIDER prod_idc  
    AUTO_CREATE_ROLES TRUE  
    INCLUDE GROUPS LIKE '%dev%'  
    EXCLUDE GROUPS LIKE '%test%';
```

모범 사례

역할에 대해 자동 생성을 활성화할 때 다음 모범 사례를 고려하세요.

- INCLUDE 및 EXCLUDE 필터를 사용하여 역할을 가져오는 그룹을 제어합니다.
- 역할을 정기적으로 감사하고 사용하지 않는 역할을 정리합니다.
- Redshift 역할 계층 구조를 활용하여 권한 관리를 간소화합니다.

Amazon S3 Access Grants와 Amazon Redshift 통합

Amazon S3 Access Grants와의 통합을 사용하면 IAM Identity Center ID를 원활하게 전파하여 Amazon S3 데이터에 대한 액세스를 제어할 수 있습니다. 이 통합을 통해 IAM Identity Center 사용자 및 그룹을 기반으로 Amazon S3 데이터 액세스 권한을 부여할 수 있습니다.

Amazon S3 Access Grants에 대한 자세한 내용은 [S3 Access Grants를 통한 액세스 관리](#)를 참조하세요.

Amazon S3 Access Grants를 사용하면 애플리케이션에 다음과 같은 이점이 있습니다.

- IAM Identity Center ID를 기반으로 Amazon S3 데이터에 대한 세분화된 액세스 제어
- Amazon Redshift 및 Amazon S3에서 IAM Identity Center ID의 중앙 집중식 관리
- Amazon S3 액세스에 대한 별도의 IAM 권한 관리 불필요

작동 방식

애플리케이션을 Amazon S3 Access Grants와 통합하려면 다음을 수행합니다.

- 먼저 AWS Management Console 또는 AWS CLI를 사용하여 Amazon S3 Access Grants와 통합하도록 Amazon Redshift를 구성합니다.
- 다음으로 IdC 관리자 권한이 있는 사용자는 Amazon S3 Access Grants 서비스를 사용하여 특정 IdC 사용자/그룹에 대한 Amazon S3 버킷 또는 접두사 액세스 권한을 부여합니다. 자세한 내용은 [S3 Access Grants에서 권한 부여 사용](#)을 참조하세요.
- Redshift에 인증된 IdC 사용자가 S3에 액세스하는 쿼리(예: COPY, UNLOAD 또는 Spectrum 작업)를 실행하면 Amazon Redshift는 Amazon S3 Access Grants 서비스에서 해당 IdC ID로 범위가 지정된 임시 Amazon S3 액세스 자격 증명을 검색합니다.
- 그런 다음 Amazon Redshift는 검색된 임시 자격 증명을 사용하여 해당 쿼리에 대해 권한이 부여된 Amazon S3 위치에 액세스합니다.

Amazon S3 Access Grants와의 통합 설정

Amazon Redshift와 Amazon S3 Access Grants의 통합을 설정하려면 다음을 수행합니다.

주제

- [AWS Management Console을 사용하여 Amazon S3 Access Grants와의 통합 설정](#)
- [AWS CLI를 사용하여 Amazon S3 Access Grants와의 통합 활성화](#)

AWS Management Console을 사용하여 Amazon S3 Access Grants와의 통합 설정

1. Amazon Redshift 콘솔을 엽니다.
2. 클러스터 창에서 클러스터를 선택합니다.
3. 클러스터의 세부 정보 페이지의 ID 제공업체 통합 섹션에서 S3 Access Grants 서비스와의 통합을 활성화합니다.

Note

IAM Identity Center가 구성되어 있지 않으면 ID 제공업체 통합 섹션이 표시되지 않습니다. 자세한 내용은 [Enabling AWS IAM Identity Center](#)를 참조하세요.

AWS CLI를 사용하여 Amazon S3 Access Grants와의 통합 활성화

1. S3 통합이 활성화된 새 Amazon Redshift IdC 애플리케이션을 만들려면 다음을 수행합니다.

```
aws redshift create-redshift-idc-application <other parameters>
  --service-integrations '[ {"S3AccessGrants": [{"ReadWriteAccess":
{"Authorization": "Enabled"}]} ]'
```

2. S3 Access Grants 통합을 활성화하도록 기존 애플리케이션을 수정하려면 다음을 수행합니다.

```
aws redshift modify-redshift-idc-application <other parameters>
  --service-integrations '[ {"S3AccessGrants": [{"ReadWriteAccess":
{"Authorization": "Enabled"}]} ]'
```

3. S3 Access Grants 통합을 비활성화하도록 기존 애플리케이션을 수정하려면 다음을 수행합니다.

```
aws redshift modify-redshift-idc-application <other parameters>
  --service-integrations '[ {"S3AccessGrants": [{"ReadWriteAccess":
{"Authorization": "Disabled"}]} ]'
```

S3 Access Grants와의 통합 사용

S3 Access Grants 통합을 구성한 후 S3 데이터에 액세스하는 쿼리(예: COPY, UNLOAD 또는 Spectrum 쿼리)는 권한 부여를 위해 IdC ID를 사용합니다. IdC를 사용하여 인증되지 않은 사용자도 이러한 쿼리를 실행할 수 있지만 이러한 사용자 계정은 IdC가 제공하는 중앙 집중식 관리를 활용하지 않습니다.

다음 예시는 S3 Access Grants 통합으로 실행되는 쿼리를 보여줍니다.

```
COPY table FROM 's3://mybucket/data'; // -- Redshift uses IdC identity
UNLOAD ('SELECT * FROM table') TO 's3://mybucket/unloaded/' // -- Redshift uses IdC
identity
```

AWS Lake Formation을 통한 데이터 쿼리

AWS Lake Formation을 사용하면 데이터 레이크를 중앙에서 관리 및 보호하고 데이터 액세스를 제공하는 것이 더 쉬워집니다. AWS IAM Identity Center 및 Redshift를 통해 Lake Formation으로 ID를 전파하도록 구성하면 관리자가 조직의 ID 제공업체(IdP) 그룹을 기반으로 Amazon S3 데이터 레이크에 세밀하게 분류된 액세스를 허용할 수 있습니다. 이러한 그룹은 AWS IAM Identity Center를 통해 관리됩니다. 이 섹션에서는 Redshift와 함께 AWS IAM Identity Center를 활용하여 Lake Formation에서 관리하는 리소스에 연결하는 몇 가지 사용 사례를 구성하는 방법을 보여줍니다(예: 데이터 레이크에서 쿼리, 데이터 공유에서 쿼리).

AWS IAM Identity Center 및 Redshift 연결을 사용하여 데이터 레이크 쿼리

이 단계는 Redshift에 연결된 AWS IAM Identity Center를 사용하여 Lake Formation에서 관리하는 데이터 레이크를 쿼리하는 사용 사례를 다룹니다.

사전 조건:

이 절차에는 몇 가지 사전 조건과 단계가 있습니다.

1. Redshift를 통한 인증 및 ID 관리를 지원하려면 AWS IAM Identity Center를 설정해야 합니다. 콘솔에서 AWS IAM Identity Center를 활성화하고 ID 제공업체(IdP) 소스를 선택할 수 있습니다. 그런 다음 IdP 사용자 세트를 AWS IAM Identity Center와 동기화합니다. 또한 이 문서의 앞부분에 설명된 단계에 따라 AWS IAM Identity Center와 Redshift 간의 연결을 설정해야 합니다.
2. 새 Amazon Redshift 클러스터를 생성하고 구성 단계에서 AWS IAM Identity Center를 통한 ID 관리를 활성화합니다.
3. Lake Formation을 위한 관리형 AWS IAM Identity Center 애플리케이션을 생성하고 구성합니다. 이는 AWS IAM Identity Center와 Redshift 간의 연결 설정에 따른 것입니다. 단계는 다음과 같습니다.
 - a. AWS CLI에서 `modify-redshift-idc-application` 명령을 사용하여 Lake Formation 서비스를 Redshift용 AWS IAM Identity Center 관리 애플리케이션과 통합할 수 있도록 설정합니다. 이 호출에는 Lake Formation에 대한 권한 부여를 가능하게 하는 구성 문자열 값으로 설정된 `service-integrations` 파라미터가 포함됩니다.
 - b. `create-lake-formation-identity-center-configuration` 명령을 사용하여 Lake Formation을 구성합니다. 그러면 AWS IAM Identity Center 포털에서 볼 수 있는 Lake Formation용 AWS IAM Identity Center 애플리케이션이 생성됩니다. 관리자는 `--cli-input-json` 인수를 설정해야 하며, 이 인수의 값은 모든 AWS CLI API 직접 호출에 표준 형식을 사용하는 JSON 파일의 경로입니다. 다음에 대한 값을 포함해야 합니다.
 - `CatalogId` - Lake Formation 카탈로그 ID
 - `InstanceArn` - AWS IAM Identity Center 인스턴스 ARN 값

관리자가 사전 조건 구성을 완료한 후 데이터베이스 관리자는 데이터 레이크를 쿼리하기 위한 외부 스키마를 생성할 수 있습니다.

1. 관리자가 외부 스키마 생성 - Redshift 데이터베이스 관리자는 다음 SQL 문을 사용하여 데이터베이스에 연결하고 외부 스키마를 만듭니다.

```
CREATE EXTERNAL SCHEMA if not exists my_external_schema from DATA CATALOG database
'my_lf_integrated_db' catalog_id '12345678901234';
```

참고로, AWS IAM Identity Center를 통해 액세스를 관리하므로 이 경우에는 IAM 역할을 지정할 필요가 없습니다.

2. 관리자가 권한 부여 - 관리자가 AWS IAM Identity Center 그룹에 사용 권한을 부여하며, 이로 인해 Redshift 리소스에 대한 권한이 부여됩니다. 다음과 같은 SQL 문을 실행하면 됩니다.

```
GRANT USAGE ON SCHEMA "my_external_schema" to "MYC0:sales";
```

이후 관리자는 AWS CLI를 사용하여 조직의 요구 사항에 따라 객체에 대한 Lake Formation 권한을 부여합니다.

```
aws lakeformation grant-permissions ...
```

3. 사용자가 쿼리 실행 - 이제 예컨대 영업 그룹에 속한 AWS IAM Identity Center 사용자는 쿼리 에디터 v2를 통해 Redshift 데이터베이스에 로그인할 수 있습니다. 그런 다음, 아래 샘플과 같이 외부 스키마의 테이블에 액세스하는 쿼리를 실행할 수 있습니다.

```
SELECT * from my_external_schema.table1;
```

AWS IAM Identity Center와 Redshift 연결을 사용하여 데이터 공유에 연결

AWS IAM Identity Center를 통해 액세스를 관리하는 경우 다른 Redshift 데이터 웨어하우스에서 데이터 공유에 액세스할 수 있습니다. 이렇게 하려면 쿼리를 실행하여 외부 데이터베이스를 설정합니다. 이 단계를 완료하기 전에는 이전 절차에 설명된 대로 Redshift와 AWS IAM Identity Center 간에 연결을 설정하고 AWS Lake Formation 애플리케이션을 생성한 것으로 가정합니다.

1. 외부 데이터베이스 생성 - 관리자가 ARN을 통해 참조하여 데이터 공유를 위한 외부 데이터베이스를 만듭니다. 다음은 방법을 보여주는 샘플입니다.

```
CREATE DATABASE "redshift_external_db" FROM ARN 'arn:aws:glue:us-east-1:123456789012:database/redshift_external_db-iad' WITH NO DATA CATALOG SCHEMA;
```

ID 관리를 위해 Redshift와 함께 AWS IAM Identity Center를 사용하는 이 사용 사례에서는 IAM 역할이 포함되지 않습니다.

2. 관리자가 권한 설정 - 데이터베이스를 생성한 후 관리자는 AWS IAM Identity Center 그룹에 사용 권한을 부여합니다. 이렇게 하면 Redshift 리소스에 대한 권한이 부여됩니다.


```
GRANT USAGE ON DATABASE "my_external_db" to "MYC0:sales";
```

또한 관리자는 AWS CLI를 사용하여 객체에 대한 Lake Formation 권한을 부여합니다.

```
aws lakeformation grant-permissions ...
```

3. 사용자가 쿼리 실행 - 영업 그룹의 사용자는 할당된 권한에 따라 데이터베이스의 테이블을 쿼리할 수 있습니다.

```
select * from redshift_external_db.public.employees;
```

데이터 레이크에 대한 권한 부여 및 데이터 공유에 대한 권한 부여 관련 자세한 내용은 [Granting permissions to users and groups](#)를 참조하세요. 스키마 또는 데이터베이스에 사용 권한을 부여하는 방법에 대한 자세한 내용은 [GRANT](#)를 참조하세요.

신뢰할 수 있는 토큰 발급자를 사용하여 애플리케이션 또는 도구를 OAuth와 통합

AWS IAM Identity Center 연결을 통해 Redshift에 연결하기 위해 생성한 클라이언트 도구에 기능을 추가할 수 있습니다. AWS IAM Identity Center에 대한 Redshift 통합을 이미 구성한 경우 이 섹션에 자세히 설명된 속성을 사용하여 연결을 설정합니다.

AWS IAM Identity Center를 사용하여 Redshift에 연결하기 위한 인증 플러그인

다음 드라이버 플러그인을 사용하여 AWS IAM Identity Center를 사용하여 Amazon Redshift에 연결할 수 있습니다.

- **BrowserIdcAuthPlugin** - 이 플러그인은 AWS IAM Identity Center와의 원활한 통합 인증을 지원합니다. 사용자가 기업 ID 제공업체에 정의된 사용자 자격 증명을 사용하여 로그인할 수 있는 브라우저 창을 만듭니다.
- **IdpTokenAuthPlugin** - 이 플러그인은 Amazon Redshift 드라이버가 AWS IAM Identity Center 인증을 위해 브라우저 창을 열도록 하는 대신 자체적으로 인증 흐름을 관리하려는 애플리케이션에서 사용해야 합니다. Okta, PingOne 및 Microsoft Entra ID(Azure AD)와 같이 AWS IAM Identity Center와 연결된 모든 웹 ID 제공업체의 AWS IAM Identity Center 벤딩 액세스 토큰 또는 OIDC(OpenID Connect) JSON 웹 토큰(JWT)을 허용합니다. 클라이언트 애플리케이션은 이 필수 액세스 토큰/JWT를 생성할 책임이 있습니다.

BrowserIdcAuthPlugin을 사용하는 인증

Amazon Redshift 드라이버에 따라 BrowserIdcAuthPlugin을 사용하여 연결하려면 다음 플러그인 이름을 사용하세요.

드라이버	연결 옵션 키	값	설명
JDBC	plugin_name	com.amazon.redshift.plugin.BrowserIdcAuthPlugin	연결할 때 플러그인의 정규화된 클래스 이름을 입력해야 합니다.
ODBC	plugin_name	BrowserIdcAuthPlugin	
Python	credentials_provider	BrowserIdcAuthPlugin	Python 드라이버에 사용할 수 있는 plugin_name 옵션이 없습니다. 대신 credentials_provider를 사용합니다.

BrowserIdcAuthPlugin 플러그인에는 다음과 같은 추가 연결 옵션이 있습니다.

옵션 이름	필수?	설명	예제
idc_region	필수	AWS IAM Identity Center 인스턴스가 위치한 AWS 리전입니다.	us-east-1
issuer_url	필수	AWS IAM Identity Center 서버의 인스턴스 엔드포인트입니다. 이 값은 AWS IAM Identity Center 콘솔을 사용하여 찾을 수 있습니다.	https://identitycenter.amazonaws.com/ssoids-g5j2k70sn4yc5nsc

옵션 이름	필수?	설명	예제
listen_port	선택	Amazon Redshift 드라이버가 브라우저 리디렉션을 통해 AWS IAM Identity Center로부터 auth_code 응답을 수신하는 데 사용하는 포트입니다.	7890
idc_client_display_name	선택	AWS IAM Identity Center의 Single Sign-On 동의 팝업에서 AWS IAM Identity Center 클라이언트가 애플리케이션에 사용하는 이름입니다.	Amazon Redshift 드라이버
idp_response_timeout	선택	Redshift 드라이버가 인증 흐름이 완료될 때까지 기다리는 시간(초)입니다.	60

생성하고 연결하는 도구의 연결 속성에 이 값을 입력해야 합니다. 자세한 내용은 각각의 드라이버에 대한 연결 옵션 설명서를 참조하세요.

- [JDBC 드라이버 버전 2.x 구성 옵션](#)
- [ODBC 드라이버 옵션](#)
- [Amazon Redshift Python 커넥터의 구성 옵션](#)

IdpTokenAuthPlugin을 사용하는 인증

Amazon Redshift 드라이버에 따라 IdpTokenAuthPlugin을 사용하여 연결하려면 다음 플러그인 이름을 사용하세요.

드라이버	연결 옵션 키	값	설명
JDBC	plugin_name	com.amazon.redshift.plugin.IdpTokenAuthPlugin	연결할 때 플러그인의 정규화된 클래스 이름을 입력해야 합니다.
ODBC	plugin_name	IdpTokenAuthPlugin	
Python	credentials_provider	IdpTokenAuthPlugin	Python 드라이버에 사용할 수 있는 plugin_name 옵션이 없습니다. 대신 를 사용합니다credentials_provider

IdpTokenAuthPlugin 플러그인에는 다음과 같은 추가 연결 옵션이 있습니다.

옵션 이름	필수?	설명
토큰	필수	AWS IAM Identity Center에서 벤딩한 액세스 토큰 또는 AWS IAM Identity Center와 연결된 웹 ID 제공업체에서 제공하는 OIDC(OpenID Connect) JSON 웹 토큰(JWT)입니다. 애플리케이션은AWS IAM Identity Center 또는 AWS IAM Identity Center와 연결된 ID 제공업체를 통해 애플리케이션 사용자를 인증하여 이 토큰을 생성해야 합니다.
token_type	필수	IdpTokenAuthPlugin 에 사용되는 토큰 유형 가능한 값은 다음과 같습니다.

옵션 이름	필수?	설명
		• ACCESS_TOKEN - AWS IAM Identity Center에서 제공한 액세스 토큰을 사용하는 경우 이 값을 입력합니다. • EXT_JWT - AWS IAM Identity Center와 연결된 웹 기반 ID 제공업체가 제공하는 OpenID Connect(OIDC) JSON 웹 토큰(JWT)을 사용하는 경우 이 값을 입력합니다.

생성하고 연결하는 도구의 연결 속성에 이 값을 입력해야 합니다. 자세한 내용은 각각의 드라이버에 대한 연결 옵션 설명서를 참조하세요.

- [JDBC 드라이버 버전 2.x 구성 옵션](#)
- [ODBC 드라이버 옵션](#)
- [Amazon Redshift Python 커넥터의 구성 옵션](#)

Amazon Redshift 쿼리 에디터 v2의 연결 문제 해결

이 목록은 일반적으로 발생하는 오류를 자세히 설명하며 AWS IAM Identity Center 자격 증명을 사용하여 쿼리 에디터 v2와 Redshift 데이터베이스에 연결하는 데 도움이 될 수 있습니다.

- **오류: 연결 문제: Identity Center 세션 정보를 사용할 수 없습니다.** - 이 오류가 발생하면 브라우저의 보안 및 개인 정보 설정을 확인합니다. 이러한 브라우저 설정, 특히 Firefox의 Total Cookie Protection 기능과 같은 보안 쿠키 관련 설정을 사용하면 Amazon Redshift 쿼리 에디터 v2에서 Redshift 데이터베이스로의 연결 시도가 차단될 수 있습니다. 해당 브라우저에 대해 자세히 설명된 문제 해결 단계를 따르세요.
- **Firefox - 현재 서드 파티 쿠키는 기본적으로 차단되어 있습니다.** 브라우저 주소 표시줄의 방패 모양 아이콘을 클릭하고 토글을 전환하여 쿼리 에디터 v2의 강화 추적 방지 기능을 끕니다.
- **Chrome 시크릿 모드 - Chrome 시크릿 모드는 기본적으로 서드 파티 쿠키를 차단합니다.** 주소 표시줄의 눈 모양 아이콘을 클릭하여 쿼리 에디터 v2에 서드 파티 쿠키를 허용할 수 있습니다. 쿠키를 허용하도록 설정을 변경한 후에는 주소 표시줄에 눈 모양 아이콘이 보이지 않을 수 있습니다.

- Safari - Mac의 경우 Safari 앱을 엽니다. 설정을 선택한 후 고급을 선택합니다. 토글을 전환하여 모든 쿠키 차단을 끕니다.
- Edge - 설정을 선택하고 쿠키 및 사이트 권한을 선택합니다. 그런 다음 쿠키 및 사이트 데이터 관리 및 삭제 선택하고 타사 쿠키 차단을 끕니다.

설정을 변경한 후 연결을 시도해도 연결 문제: Identity Center 세션 정보를 사용할 수 없습니다 오류 메시지가 계속 표시되는 경우 AWS IAM Identity Center와의 연결을 새로 고치는 것이 좋습니다. 이렇게 하려면 Redshift 데이터베이스 인스턴스를 마우스 오른쪽 버튼으로 클릭하고 새로 고침을 선택합니다. 인증을 수행할 수 있는 새 창이 나타납니다.

- 오류: 연결 문제: Identity Ceneter 세션이 만료되었거나 유효하지 않습니다. - Redshift 프로비저닝된 클러스터 또는 Serverless 작업 그룹을 AWS IAM Identity Center와 통합한 후 사용자가 쿼리 에디터 v2에서 Redshift 데이터베이스에 연결하려고 하면 이 오류가 표시될 수 있습니다. 연결 시도가 성공한 후 이 문제가 발생할 수 있습니다. 이 경우 다시 인증하는 것이 좋습니다. 이렇게 하려면 Redshift 데이터베이스 인스턴스를 마우스 오른쪽 버튼으로 클릭하고 새로 고침을 선택합니다. 인증을 수행할 수 있는 새 창이 나타납니다.
- 오류: 범위가 잘못되었습니다. 사용자 자격 증명은 Redshift에 연결할 권한이 없습니다. - 자격 증명 관리를 위해 Redshift 프로비저닝된 클러스터 또는 Serverless 작업 그룹을 AWS IAM Identity Center와 통합한 후 사용자가 쿼리 에디터 v2에서 Redshift 데이터베이스에 연결하려고 하면 이 오류가 표시될 수 있습니다. 이 경우 쿼리 에디터 v2가 AWS IAM Identity Center를 통해 사용자를 성공적으로 연결하고 인증하여 올바른 리소스에 액세스하려면 관리자가 Redshift 콘솔을 통해 Redshift AWS IAM Identity Center 애플리케이션에 사용자를 할당해야 합니다. 이 작업은 IAM Identity Center 연결에서 완료됩니다. 이렇게 하면 사용자는 AWS IAM Identity Center 세션 캐시의 한도인 1시간 후에 성공적인 연결을 설정할 수 있습니다.
- 오류: 데이터베이스를 나열할 수 없습니다. FATAL: 클러스터가 자동으로 일시 중지될 때 쿼리가 실패했습니다. - Amazon Redshift Serverless 데이터베이스가 워크로드를 처리하지 않는 유휴 상태이면 AWS IAM Identity Center 자격 증명으로 연결할 때 일시 중지 상태를 유지할 수 있습니다. 이 문제를 해결하려면 다른 인증 방법으로 로그인하여 서버리스 작업 그룹을 다시 시작합니다. 그런 다음 AWS IAM Identity Center 자격 증명으로 데이터베이스에 연결합니다.
- 오류: AWS IAM Identity Center와 페더레이션하려고 시도하는 동안 오류가 발생했습니다. Amazon Redshift 관리자는 Redshift 콘솔을 사용하여 AWS IAM Identity Center QEV2 애플리케이션을 삭제하고 다시 생성해야 합니다. - 이 오류는 일반적으로 쿼리 에디터 v2와 연결된 AWS IAM Identity Center 애플리케이션 인스턴스가 삭제될 때 발생합니다. 이 문제를 해결하려면 Amazon Redshift 관리자가 AWS IAM Identity Center를 위해 Redshift 및 쿼리 에디터 v2 애플리케이션을 삭제하고 다시 생성해야 합니다. 이 작업은 Redshift 콘솔 또는 <https://docs.aws.amazon.com/cli/latest/reference/redshift/delete-redshift-idc-application.html> CLI 명령을 사용하여 수행할 수 있습니다.

Amazon Redshift에 대해 서비스 연결 역할 사용

Amazon Redshift는 AWS Identity and Access Management(IAM) [서비스 연결 역할](#)을 사용합니다. 서비스 연결 역할은 Amazon Redshift에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 Amazon Redshift에서 사전 정의하며, 해당 서비스가 Amazon Redshift 클러스터를 대신해 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 가지고 있습니다.

서비스 연결 역할을 사용하면 필요한 권한을 수동으로 추가할 필요가 없기 때문에 Amazon Redshift를 더 쉽게 설정할 수 있습니다. 서비스 연결 역할은 Amazon Redshift 사용 사례에 연결되고 사전 정의된 권한이 있습니다. Amazon Redshift만 역할을 수임할 수 있으며 서비스 연결 역할만 사전 정의된 권한 정책을 사용할 수 있습니다. Amazon Redshift는 클러스터 또는 Redshift 관리형 VPC 엔드포인트를 처음 생성할 때 계정에 서비스 연결 역할을 생성합니다. 계정에서 Amazon Redshift 클러스터 또는 Redshift 관리형 VPC 엔드포인트를 모두 삭제한 후에만 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 하면 리소스에 대한 액세스에 필요한 권한을 부주의하게 삭제할 수 없기 때문에 Amazon Redshift 리소스가 보호됩니다.

Amazon Redshift는 서비스가 제공되는 모든 리전에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 [AWS 리전 및 엔드포인트](#) 섹션을 참조하세요.

서비스 연결 역할을 지원하는 기타 서비스에 대한 자세한 내용은 [IAM으로 작업하는 AWS 서비스](#) 섹션을 참조하고 서비스 연결 역할 옆에 예가 표시된 서비스를 찾으세요. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

Amazon Redshift에 대한 서비스 연결 역할 권한

Amazon Redshift는 AWSServiceRoleForRedshift라는 서비스 연결 역할을 사용합니다. 이를 통해 Amazon Redshift가 사용자를 대신하여 AWS 서비스를 호출할 수 있습니다. 이 서비스 연결 역할은 관리형 정책 AmazonRedshiftServiceLinkedRolePolicy에 연결됩니다. 이 정책에 대한 업데이트는 [Amazon Redshift에 대한 AWS 관리형 \(사전 정의된\) 정책](#)을 참조하세요.

AWSServiceRoleForRedshift 서비스 연결 역할은 **redshift.amazonaws.com**만 신뢰하여 역할을 맡습니다.

AWSServiceRoleForRedshift 서비스 연결 역할 권한 정책은 Amazon Redshift가 관련된 모든 리소스에서 다음을 완료하도록 허용합니다.

- ec2:DescribeVpcs
- ec2:DescribeSubnets
- ec2:DescribeNetworkInterfaces

- `ec2:DescribeAddress`
- `ec2:AssociateAddress`
- `ec2:DisassociateAddress`
- `ec2:CreateNetworkInterface`
- `ec2>DeleteNetworkInterface`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:CreateVpcEndpoint`
- `ec2>DeleteVpcEndpoints`
- `ec2:DescribeVpcEndpoints`
- `ec2:ModifyVpcEndpoint`
- `ec2:DescribeVpcAttribute`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeInternetGateways`
- `ec2:DescribeSecurityGroupRules`
- `ec2:DescribeAvailabilityZones`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeRouteTables`
- `ec2:AssignIpv6Addresses`
- `ec2:UnassignIpv6Addresses`

네트워크 리소스에 대한 권한

다음 권한은 Amazon EC2에서 보안 그룹 규칙의 생성 및 관리 작업을 허용합니다. 이러한 보안 그룹 및 규칙은 특히 Amazon Redshift `aws:RequestTag/Redshift` 리소스 태그와 관련이 있습니다. 이렇게 하면 권한 범위가 특정 Amazon Redshift 리소스로 제한됩니다.

- `ec2:CreateSecurityGroup`
- `ec2:AuthorizeSecurityGroupEgress`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:RevokeSecurityGroupEgress`
- `ec2:RevokeSecurityGroupIngress`

- ec2:ModifySecurityGroupRules
- ec2>DeleteSecurityGroup

서비스 할당량에 대한 권한

다음 권한을 통해 호출자는 서비스 할당량을 받을 수 있습니다.

servicequotas:GetServiceQuota

다음 JSON 조각은 서비스 할당량의 작업 및 리소스 범위를 보여줍니다.

```
{
  "Sid": "ServiceQuotasToCheckCustomerLimits",
  "Effect": "Allow",
  "Action": [
    "servicequotas:GetServiceQuota"
  ],
  "Resource": [
    "arn:aws:servicequotas:*:*:ec2/L-0263D0A3",
    "arn:aws:servicequotas:*:*:vpc/L-29B6F2EB"
  ]
}
```

할당량 코드는 다음과 같습니다.

- L-0263D0A3 - EC2-VPC Elastic IP의 할당량 코드입니다.
- L-29B6F2EB - VPC당 인터페이스 VPC 엔드포인트의 할당량 코드입니다.

자세한 내용은 [AWS 서비스 할당량](#)을 참조하세요.

감사 로깅 작업

logs 접두사와 함께 나열된 작업은 감사 로깅 및 관련 기능과 관련이 있습니다. 특히, 로그 그룹 및 로그 스트림의 생성 및 관리와 관련이 있습니다.

- logs:CreateLogGroup
- logs:PutRetentionPolicy
- logs:CreateLogStream
- logs:PutLogEvents

- logs:DescribeLogStreams
- logs:GetLogEvents

다음 JSON은 감사 로깅을 위해 Amazon Redshift에 대한 작업 및 리소스 범위를 보여줍니다.

```
[
  {
    "Sid": "EnableCreationAndManagementOfRedshiftCloudwatchLogGroups",
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:PutRetentionPolicy"
    ],
    "Resource": [
      "arn:aws:logs:*:*:log-group:/aws/redshift/*"
    ]
  },
  {
    "Sid": "EnableCreationAndManagementOfRedshiftCloudwatchLogStreams",
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams",
      "logs:GetLogEvents"
    ],
    "Resource": [
      "arn:aws:logs:*:*:log-group:/aws/redshift/*:log-stream:*"
    ]
  }
]
```

서비스 연결 역할 및 AWS에서의 용도에 대한 자세한 내용은 [서비스 연결 역할 사용](#)을 참조하세요. Amazon Redshift의 특정 작업 및 기타 IAM 리소스에 대한 자세한 내용은 [Amazon Redshift에 사용되는 작업, 리소스 및 조건 키](#)를 참조하세요.

를 사용하여 관리자 보안 인증 정보를 관리하기 위한 작업AWS Secrets Manager

secretsmanager 접두사와 함께 나열된 작업은 Amazon Redshift를 사용하여 관리자 보안 인증 정보를 관리하는 것과 관련이 있습니다. 이러한 작업을 사용하면 Amazon Redshift는 AWS Secrets Manager를 사용하여 관리자 보안 인증 보안 암호를 생성하고 관리할 수 있습니다.

다음 JSON은 AWS Secrets Manager를 사용한 관리자 보안 인증 정보 관리와 관련하여 Amazon Redshift에 대한 작업 및 리소스 범위를 보여줍니다.

```
[
  {
    "Effect": "Allow",
    "Action": [
      "secretsmanager:DescribeSecret",
      "secretsmanager:DeleteSecret",
      "secretsmanager:PutSecretValue",
      "secretsmanager:UpdateSecret",
      "secretsmanager:UpdateSecretVersionStage",
      "secretsmanager:RotateSecret"
    ],
    "Resource": [
      "arn:aws:secretsmanager:*:*:secret:redshift!*"
    ],
    "Condition": {
      "StringEquals": {
        "secretsmanager:ResourceTag/aws:secretsmanager:owningService":
"redshift"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "secretsmanager:GetRandomPassword"
    ],
    "Resource": "*"
  }
]
```

AWS Glue Data Catalog에 클러스터 및 서버리스 네임스페이스를 등록하기 위한 작업

glue 접두사로 나열된 작업은 프로비저닝된 클러스터 또는 서버리스 네임스페이스를 등록하여 만든 AWS Glue Data Catalog의 카탈로그에 액세스하는 것과 관련이 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Amazon Redshift에 대한 Apache Iceberg 호환성](#)을 참조하세요.

다음 JSON은 AWS Glue Data Catalog의 카탈로그에 액세스하기 위한 Amazon Redshift에 대한 작업 및 리소스 범위를 보여줍니다.

```
[
  {
    "Sid": "DiscoverRedshiftCatalogs",
    "Effect": "Allow",
    "Action": [
      "glue:GetCatalogs",
      "glue:GetCatalog"
    ],
    "Resource": [
      "arn:aws:glue:*:*:catalog",
      "arn:aws:glue:*:*:catalog/*"
    ],
    "Condition": {
      {
        "Bool": {
          "glue:EnabledForRedshiftAutoDiscovery": "true"
        },
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "LakeFormationGetMetadataAccessForFederatedCatalogs",
      "Effect": "Allow",
      "Action": [
        "lakeformation:GetDataAccess"
      ],
      "Resource": [ "*" ],
      "Condition": {
        {
          "Bool": {
            "lakeformation:EnabledOnlyForMetaDataAccess": "true"
          },
          "StringEquals": {
            "aws:ResourceAccount": "${aws:PrincipalAccount}"
          },
          "ForAnyValue:StringEquals": {
            "aws:CalledVia": "glue.amazonaws.com"
          }
        }
      }
    }
  ]
}
```

```

    }
  }
}
]

```

glue:GetCatalog 및 glue:GetCatalogs 권한에는

glue:EnabledForRedshiftAutoDiscovery:true 조건이 있습니다. 즉, Amazon Redshift는 카탈로그를 자동으로 검색할 수 있는 IAM 액세스 권한을 부여합니다. 옵트아웃하려면 AWS Glue 계정 수준 리소스 정책을 추가하여 카탈로그에 대한 서비스 연결 역할 액세스를 선택적으로 거부합니다. 서비스 연결 역할에 이미 정책에 명시적 허용 작업이 있으므로 옵트아웃 정책은 해당 작업을 명시적으로 거부해야 합니다. 추가 정책이 Amazon Redshift에 대한 자동 검색을 거부하는 다음 예제를 생각해 보세요.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "glue:GetCatalog",
        "glue:GetCatalogs"
      ],
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/aws-service-role/redshift.amazonaws.com/AWSServiceRoleForRedshift"
      },
      "Resource": [
        "arn:aws:glue::*:catalog/<s3_table_catalog_name>",
        "arn:aws:glue::*:catalog/<s3_table_catalog_name>/*"
      ]
    }
  ]
}

```

IAM 엔터티가 AWSServiceRoleForRedshift 서비스 연결 역할을 생성하도록 허용

```

{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole"
  ]
}

```

```

    ],
    "Resource": "arn:aws:iam::<AWS-account-ID>:role/aws-service-role/
redshift.amazonaws.com/AWSServiceRoleForRedshift",
    "Condition": {"StringLike": {"iam:AWSServiceName": "redshift.amazonaws.com"}}
}

```

IAM 엔터티가 AWSServiceRoleForRedshift 서비스 연결 역할을 삭제하도록 허용

IAM 개체에 대한 권한에 다음 정책 설명을 추가합니다.

```

{
  "Effect": "Allow",
  "Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
  ],
  "Resource": "arn:aws:iam::<AWS-account-ID>:role/aws-service-role/
redshift.amazonaws.com/AWSServiceRoleForRedshift",
  "Condition": {"StringLike": {"iam:AWSServiceName": "redshift.amazonaws.com"}}
}

```

또는 AWS 관리형 정책을 사용하여 Amazon Redshift에 대한 [전체 액세스 권한을 제공](#)할 수 있습니다.

Amazon Redshift에 대한 서비스 연결 역할 생성

AWSServiceRoleForRedshift 서비스 연결 역할은 수동으로 생성할 필요가 없습니다. Amazon Redshift에서 서비스 연결 역할을 자동으로 생성합니다. 사용자 계정에서 AWSServiceRoleForRedshift 서비스 연결 역할이 삭제되면 새 Amazon Redshift 클러스터를 시작할 때 Amazon Redshift에서 해당 역할을 생성합니다.

Important

Amazon Redshift 서비스가 서비스 연결 역할을 지원하기 시작한 2017년 12월 18일 이전에 이 서비스를 사용했다면 Amazon Redshift가 사용자 계정에 AWSServiceRoleForRedshift 역할을 이미 생성했습니다. 자세한 내용은 [내 IAM 계정에 표시되는 새 역할](#)을 참조하세요.

Amazon Redshift에 대한 서비스 연결 역할 편집

Amazon Redshift는 AWSServiceRoleForRedshift 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 연결 역할을 생성한 후에는 다양한 엔터티가 역할을 참조할 수 있기 때문에 역할 이름을 변경할

수 없습니다. 그러나 IAM 콘솔, AWS Command Line Interface(AWS CLI) 또는 IAM API를 사용하여 역할에 대한 설명을 편집할 수 있습니다. 자세한 내용은 IAM User Guide의 [Modifying a role](#) 섹션을 참조하세요.

Amazon Redshift에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제할 것을 권합니다. 그렇게 하면 적극적으로 모니터링하거나 유지 관리하지 않은 미사용 엔터티가 없습니다.

계정에 대한 서비스 연결 역할을 삭제하려면 클러스터를 종료한 다음 계정에서 삭제해야 합니다. 자세한 내용은 [클러스터 종료 및 삭제](#) 섹션을 참조하세요.

IAM 콘솔, AWS CLI 또는 IAM API를 사용하여 서비스 연결 역할을 삭제할 수 있습니다. 자세한 내용은 [IAM 사용 설명서](#)의 서비스에 연결 역할 삭제 섹션을 참조하세요.

IAM 인증을 이용한 데이터베이스 사용자 자격 증명 생성

사용자의 Amazon Redshift 데이터베이스 액세스 권한을 관리하기 위해 AWS Identity and Access Management(IAM) 권한 정책을 통해 허용한 권한을 바탕으로 임시 데이터베이스 자격 증명을 생성할 수 있습니다.

일반적으로 Amazon Redshift 데이터베이스 사용자는 데이터베이스 사용자 이름과 암호를 제공하여 데이터베이스에 로그인합니다. 그러나 Amazon Redshift 데이터베이스에 사용자 이름과 암호를 유지할 필요는 없습니다. 대신 사용자 스스로 자격 증명을 새로 만들고 IAM 자격 증명을 바탕으로 데이터베이스에 로그인하도록 시스템 권한을 구성할 수 있습니다.

Amazon Redshift는 임시 데이터베이스 사용자 자격 증명을 생성하기 위해 [GetClusterCredentials](#) API 작업을 제공합니다. GetClusterCredentials 작업 호출 절차를 관리하는 Amazon Redshift JDBC 또는 ODBC 드라이버로 SQL 클라이언트를 구성할 수 있습니다. 이 과정은 데이터베이스 사용자 자격 증명을 검색하고 SQL 클라이언트와 Amazon Redshift 데이터베이스 사이를 연결함으로써 진행됩니다. 또한 데이터베이스 애플리케이션을 사용해 GetClusterCredentials 작업을 프로그램 방식으로 호출하고, 데이터베이스 사용자 자격 증명을 검색하고, 데이터베이스에 연결할 수 있습니다.

이미 AWS 밖에서 사용자 자격 증명을 관리하고 있는 경우, SAML(Security Assertion Markup Language) 2.0을 준수하는 자격 증명 공급자(IdP)를 통해 Amazon Redshift 리소스에 대한 액세스를 관리할 수 있습니다. 연합된 사용자가 IAM 역할에 액세스하도록 IdP를 구성합니다. 그 IAM 역할로 임시 데이터베이스 자격 증명을 생성하고 Amazon Redshift 데이터베이스에 로그인할 수 있습니다.

SQL 클라이언트가 사용자를 대신하여 GetClusterCredentials 작업을 호출할 수 있는 권한이 필요합니다. IAM 역할을 만들고 GetClusterCredentials 작업과 관련 작업의 액세스 권한을 허용하거나 제한하는 IAM 권한 정책을 연결하여 그러한 권한을 관리합니다. 가장 좋은 방법은 권한 정책

을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에 할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.

이 정책은 Amazon Redshift 클러스터, 데이터베이스, 데이터베이스 사용자 이름, 사용자 그룹 이름 등과 같은 리소스에 대한 액세스도 허용하거나 제한합니다.

Note

Amazon Redshift JDBC 또는 ODBC 드라이버를 이용해 `GetClusterCredentials` 작업 호출 및 데이터베이스 로그인 절차를 관리할 것을 권장합니다. 간단한 설명을 위해 이 주제에서는 JDBC 또는 ODBC 드라이버로 SQL 클라이언트를 사용한다고 가정하겠습니다.

`GetClusterCredentials` 작업 또는 병렬 `get-cluster-credentials` CLI 명령 사용의 구체적인 세부 내용과 예시는 [GetClusterCredentials](#) 및 [get-cluster-credentials](#)를 참조하십시오.

인증 및 권한 부여를 중앙에서 관리하기 위해 Amazon Redshift는 IAM을 통한 데이터베이스 인증을 지원하므로 엔터프라이즈 연동을 통해 사용자 인증을 사용할 수 있습니다. 사용자를 생성하는 대신 AWS Directory Service, 엔터프라이즈 사용자 디렉터리 또는 웹 자격 증명 공급자의 기존 자격 증명을 사용할 수 있습니다. 이 사용자를 페더레이션 사용자라고 합니다. AWS에서는 IdP를 통해 액세스가 요청되면 페더레이션 사용자에게 역할을 할당합니다.

조직의 사용자 또는 클라이언트 애플리케이션에 대한 연동 액세스를 제공하여 Amazon Redshift API 작업을 호출하기 위해 SAML 2.0 지원이 포함된 JDBC 또는 ODBC 드라이버를 사용하여 조직 IdP에서 인증을 요청할 수도 있습니다. 이 경우 조직의 사용자는 Amazon Redshift에 직접 액세스할 수 없습니다.

자세한 내용은 IAM User Guide의 [Identity Providers and Federation](#) 섹션을 참조하세요.

임시 IAM 보안 인증 정보 생성

이 단원에서는 IAM 기반의 임시 데이터베이스 사용자 자격 증명을 만들고 새 자격 증명을 이용해 데이터베이스에 로그인하도록 시스템을 구성하는 방법을 알아봅니다.

다음은 개괄적으로 살펴보는 프로세스의 흐름입니다

1. [1단계: IAM Single Sign-On 액세스에 대한 IAM 역할 생성](#)

(옵션) 사용자들이 IAM 인증과 서드 파티 자격 증명 공급자(IdP)를 통합하여 Amazon Redshift 데이터베이스에 액세스하도록 인증할 수 있습니다.

2. [2단계: IdP에 대한 SAML 어설션 구성](#)

(선택 사항) IdP를 이용해 IAM 인증을 사용하려면 조직의 사용자 내지 그룹을 IAM 역할에 매핑하는 IdP 애플리케이션에서 클레임 규칙을 정의해야 합니다. 혹은 속성 요소를 포함시켜 `GetClusterCredentials` 파라미터를 설정해도 됩니다.

3. [3단계: `GetClusterCredentialsWithIAM` 또는 `GetClusterCredentials` 직접 호출 권한이 있는 IAM 역할 생성](#)

SQL 클라이언트 애플리케이션은 `GetClusterCredentials` 작업을 호출할 때 사용자를 말합니다. IdP 액세스에 대해 IAM 역할을 생성한 경우, 그 역할에 필요한 권한을 추가할 수 있습니다.

4. [4단계: 데이터베이스 사용자 및 데이터베이스 그룹 생성](#)

(선택 사항) 기본적으로 사용자 이름이 없는 경우 `GetClusterCredentials`는 자격 증명을 반환하여 신규 사용자를 만듭니다. 로그인할 때 사용자가 조인하는 사용자 그룹을 지정하는 방안을 선택할 수도 있습니다. 기본적으로 데이터베이스 사용자는 PUBLIC 그룹에 조인합니다.

5. [5단계: IAM 자격 증명을 사용하도록 JDBC 또는 ODBC 연결 구성](#)

Amazon Redshift 데이터베이스에 연결하려면 SQL 클라이언트가 Amazon Redshift JDBC 또는 ODBC 드라이버를 사용하도록 구성합니다.

1단계: IAM Single Sign-On 액세스에 대한 IAM 역할 생성

SSO 액세스에 IdP를 사용하지 않으면 이 단계를 건너뛰어도 됩니다.

이미 AWS 외부에서 사용자 자격 증명을 관리하고 있다면 IAM 인증과 서드 파티 SAML-2.0 자격 증명 공급자(IdP)를 통합하여 사용자들이 Amazon Redshift 데이터베이스에 액세스하도록 인증할 수 있습니다.

자세한 내용은 IAM User Guide의 [Identity Providers and Federation](#) 섹션을 참조하세요.

Amazon Redshift IdP 인증을 사용하기 전에 AWS SAML 자격 증명 공급자를 생성합니다. IAM 콘솔에서 IdP를 생성하고 IdP와 해당 구성을 AWS에 통보합니다. 이렇게 하면 AWS 계정과 IdP 사이에 신뢰가 설정됩니다. 역할을 생성하는 단계는 IAM User Guide의 [Creating a Role for SAML 2.0 Federation \(Console\)](#)을 참조하세요.

2단계: IdP에 대한 SAML 어설션 구성

IAM 역할을 만든 다음에는 조직의 사용자 내지 그룹을 IAM 역할에 매핑하는 IdP 애플리케이션에서 클레임 규칙을 정의합니다. 자세한 내용은 IAM User Guide의 [Configuring SAML Assertions for the Authentication Response](#)를 참조하세요.

옵션으로 제공되는 GetClusterCredentials 파라미터인 DbUser, AutoCreate, DbGroups를 쓸 경우 두 가지 옵션이 있습니다. JDBC 또는 ODBC 연결로 파라미터의 값을 설정하거나 IdP에 SAML 속성 요소를 추가하여 값을 설정할 수 있습니다. DbUser, AutoCreate, DbGroups 파라미터에 대한 자세한 내용은 [5단계: IAM 자격 증명을 사용하도록 JDBC 또는 ODBC 연결 구성](#) 섹션을 참조하십시오.

Note

`${redshift:DbUser}`에서 설명한 대로 IAM 정책 변수 [GetClusterCredentials에 대한 리소스 정책](#)을 사용할 경우, DbUser의 값이, API 작업의 요청 컨텍스트에서 검색한 값으로 대체됩니다. Amazon Redshift 드라이버는 SAML 속성으로 제공된 값이 아니라 연결 URL에서 제공한 DbUser 변수의 값을 사용합니다.

이 구성의 보안을 위해 IAM 정책의 조건을 사용하여 DbUser으로 RoleSessionName 값을 확인할 것을 권장합니다. IAM 정책을 사용하여 조건을 설정하는 방법의 예는 [예제 8: GetClusterCredentials 사용을 위한 IAM 정책 예제](#)에서 볼 수 있습니다.

DbUser, AutoCreate, DbGroups 파라미터를 설정하도록 IdP를 구성하려면 다음 Attribute 요소를 포함시킵니다.

- Attribute 속성이 "https://redshift.amazon.com/SAML/Attributes/DbUser"로 설정된 Name 요소

AttributeValue 요소를 Amazon Redshift 데이터베이스에 연결할 사용자 이름으로 설정합니다.

AttributeValue 요소 값은 소문자여야 하며, 문자로 시작해서 알파벳 숫자, 밑줄(_), 더하기 기호(+), 마침표(.), 골뱅이(@), 하이픈(-)만 쓸 수 있고 길이는 128자 미만이어야 합니다. 일반적으로 사용자 이름은 사용자 ID(예: bobsmith) 또는 이메일 주소(예: bobsmith@example.com)입니다. 이 값에 공백이 들어가면 안 됩니다('Bob Smith'처럼 띄어 쓸 수 없음).

```
<Attribute Name="https://redshift.amazon.com/SAML/Attributes/DbUser">
  <AttributeValue>user-name</AttributeValue>
</Attribute>
```

- Name 속성이 "https://redshift.amazon.com/SAML/Attributes/AutoCreate"로 설정된 Attribute 요소

데이터베이스 사용자가 없으면 새로 하나 만들어서 AttributeValue 요소를 True로 설정하십시오. Amazon Redshift 데이터베이스에 데이터베이스 사용자가 반드시 존재해야 한다고 지정하려면 AttributeValue를 false로 설정합니다.

```
<Attribute Name="https://redshift.amazon.com/SAML/Attributes/AutoCreate">
```

```
<AttributeValue>true</AttributeValue>
</Attribute>
```

- Attribute 속성이 "https://redshift.amazon.com/SAML/Attributes/DbGroups"로 설정된 Name 요소

이 요소에는 하나 이상의 AttributeValue 요소가 포함되어 있습니다. Amazon Redshift 데이터베이스에 연결할 때 세션 기간 동안 DbUser가 조인하는 데이터베이스 그룹 이름으로 각 AttributeValue 요소를 설정합니다.

```
<Attribute Name="https://redshift.amazon.com/SAML/Attributes/DbGroups">
  <AttributeValue>group1</AttributeValue>
  <AttributeValue>group2</AttributeValue>
  <AttributeValue>group3</AttributeValue>
</Attribute>
```

3단계: GetClusterCredentialsWithIAM 또는 GetClusterCredentials 직접 호출 권한이 있는 IAM 역할 생성

SQL 클라이언트가 사용자를 대신하여 GetClusterCredentialsWithIAM 또는 GetClusterCredentials 작업을 직접 호출할 수 있는 권한이 필요합니다. 그 인증을 제공하려면 사용자 또는 역할을 새로 만들어 필요한 권한을 허용하는 정책을 연결합니다. 두 작업 모두 클러스터 자격 증명을 얻는 데 사용할 수 있지만 인증 방법은 다릅니다. GetClusterCredentialsWithIAM은 IAM 역할을 사용하여 역할에 매핑되는 데이터베이스 사용자를 자동으로 생성하므로 IAM 역할 수준에서 권한을 관리하는 데 유용한 반면 GetClusterCredentials는 데이터베이스의 지정된 사용자 이름에 대한 자격 증명을 제공합니다.

GetClusterCredentialsWithIAM 직접 호출 권한이 있는 IAM 역할 생성

1. IAM 서비스를 이용해 사용자 또는 역할을 새로 만듭니다. 기존의 사용자 또는 역할을 사용할 수 있습니다. 예를 들어 IdP 액세스에 대해 IAM 역할을 생성한 경우, 그 역할에 필요한 IAM 정책을 연결할 수 있습니다.
2. redshift:GetClusterCredentialsWithIAM 작업의 호출 권한이 있는 권한 정책을 연결합니다. 다음 정책 샘플은 특정 클러스터 및 데이터베이스, 클러스터의 모든 데이터베이스 및 클러스터의 모든 데이터베이스에 대한 작업을 허용하는 옵션을 보여줍니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

        "Sid": "SpecificClusterAndDBName",
        "Effect": "Allow",
        "Action": "redshift:GetClusterCredentialsWithIAM",
        "Resource": [
            "arn:aws:redshift:us-east-1:123456789012:dbname:testcluster/
testdatabase"
        ]
    },
    {
        "Sid": "SpecificClusterAndAnyDBName",
        "Effect": "Allow",
        "Action": "redshift:GetClusterCredentialsWithIAM",
        "Resource": "arn:aws:redshift:us-
east-1:123456789012:dbname:examplecluster/*",
    },
    {
        "Sid": "AnyClusterAnyDatabase",
        "Effect": "Allow",
        "Action": "redshift:GetClusterCredentialsWithIAM",
        "Resource": "*"
    }
]
}

```

GetClusterCredentials 호출 권한이 있는 IAM 역할 생성

1. IAM 서비스를 이용해 사용자 또는 역할을 새로 만듭니다. 기존의 사용자 또는 역할을 사용할 수 있습니다. 예를 들어 IdP 액세스에 대해 IAM 역할을 생성한 경우, 그 역할에 필요한 IAM 정책을 연결할 수 있습니다.
2. `redshift:GetClusterCredentials` 작업의 호출 권한이 있는 권한 정책을 연결합니다. 어떤 옵션 파라미터를 지정하느냐에 따라 정책에서 추가 작업 및 리소스를 허용하거나 제한할 수 있습니다.
 - SQL 클라이언트의 클러스터 ID, AWS 리전, 포트 검색을 허용하려면 Redshift 클러스터 리소스가 있는 `redshift:DescribeClusters` 작업의 호출 권한을 포함합니다.
 - `AutoCreate` 옵션을 사용하는 경우 `redshift:CreateClusterUser` 호출(`dbuser` 리소스가 있는) 권한을 포함하십시오. 다음 Amazon 리소스 이름(ARN)은 Amazon Redshift `dbuser`를 지정합니다. *region*, *account-id* 및 *cluster-name*을 AWS 리전, 계정 및 클러스터의 값으로 바꿉니다. *dbuser-name*에는 클러스터 데이터베이스에 로그인하는 데 사용할 사용자 이름을 지정합니다.

```
arn:aws:redshift:region:account-id:dbuser:cluster-name/dbuser-name
```

- (옵션) Amazon Redshift dbname 리소스를 지정하는 ARN을 다음 형식으로 추가합니다. *region*, *account-id* 및 *cluster-name*을 AWS 리전, 계정 및 클러스터의 값으로 바꿉니다. *database-name*에는 사용자가 로그인할 데이터베이스의 이름을 지정합니다.

```
arn:aws:redshift:region:account-id:dbname:cluster-name/database-name
```

- DbGroups 옵션을 사용하는 경우 redshift:JoinGroup 작업(Amazon Redshift dbgroup 리소스가 있는)의 호출 권한을 다음 형식대로 포함합니다. *region*, *account-id* 및 *cluster-name*을 AWS 리전, 계정 및 클러스터의 값으로 바꿉니다. *dbgroup-name*에는 사용자가 로그인할 때 조인하는 사용자 그룹의 이름을 지정합니다.

```
arn:aws:redshift:region:account-id:dbgroup:cluster-name/dbgroup-name
```

자세한 내용과 예제는 [GetClusterCredentials에 대한 리소스 정책](#) 섹션을 참조하세요.

4단계: 데이터베이스 사용자 및 데이터베이스 그룹 생성

선택에 따라 클러스터 데이터베이스에 로그인할 때 사용하는 데이터베이스 사용자를 새로 만들 수 있습니다. 기존 사용자에 임시 사용자 자격 증명을 만들 경우, 그 사용자의 암호를 비활성화하고 임시 암호로 로그인하도록 강제할 수 있습니다. 혹은 GetClusterCredentials Autocreate 옵션을 써서 새 데이터베이스 사용자를 자동으로 생성할 수도 있습니다.

IAM 데이터베이스 사용자가 로그인할 때 조인하게 하려는 권한을 가진 데이터베이스 사용자 그룹을 만들 수 있습니다. GetClusterCredentials 작업을 호출하면 새 사용자가 로그인 시 조인하는 사용자 그룹 이름의 목록을 지정할 수 있습니다. 이러한 그룹의 멤버십은 주어진 요청으로 생성한 자격 증명을 이용해 만든 세션에만 유효합니다.

데이터베이스 사용자 및 데이터베이스 그룹 생성

1. Amazon Redshift 데이터베이스에 로그인하여 [CREATE USER](#)를 이용해 데이터베이스 사용자를 생성하거나 [ALTER USER](#)를 이용해 기존 사용자를 변경합니다.
2. 선택적으로 그 사용자가 암호를 쓰지 못하도록 하려면 PASSWORD DISABLE 옵션을 지정합니다. 사용자 암호를 비활성화하면 그 사용자는 임시 보안 인증만 이용해 로그인할 수 있습니다. 사

용자 암호를 비활성화하지 않으면 원래 암호 혹은 임시 보안 인증 중 하나를 이용해 로그인할 수 있습니다. 슈퍼유저의 암호를 비활성화할 수는 없습니다.

사용자가 AWS Management Console 외부에서 AWS와 상호 작용하려면 프로그래밍 방식의 액세스 권한이 필요합니다. 프로그래밍 방식의 액세스 권한을 부여하는 방법은 AWS에 액세스하는 사용자 유형에 따라 다릅니다.

사용자에게 프로그래밍 방식의 액세스 권한을 부여하려면 다음 옵션 중 하나를 선택합니다.

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
IAM	(권장됨) 콘솔 자격 증명을 임시 자격 증명으로 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • AWS CLI의 경우 AWS Command Line Interface 사용 설명서의 AWS 로컬 개발을 위한 로그인을 참조하세요. • AWS SDK의 경우 AWS SDK 및 도구 참조 안내서의 AWS 로컬 개발을 위한 로그인을 참조하세요.

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
작업 인력 ID (IAM Identity Center에서 관리되는 사용자)	임시 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. - AWS CLI에 대해서는 AWS Command Line Interface 사용자 안내서에서 AWS IAM Identity Center을 사용하여 AWS CLI 구성을 참조하세요. - AWS SDK, 도구, AWS API에 대해서는 AWS SDK 및 도구 참조 가이드에서 IAM Identity Center 인증을 참조하세요.
IAM	임시 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	IAM 사용자 설명서의 AWS 리소스와 함께 임시 자격 증명 사용 에 나와 있는 지침을 따르세요.

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
IAM	(권장되지 않음) 장기 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • AWS CLI에 대해서는 AWS Command Line Interface 사용자 안내서에서 IAM 사용자 자격 증명을 사용한 인증을 참조하세요. • AWS SDK와 도구는 AWS SDK 및 도구 참조 가이드에서 장기 자격 증명을 사용한 인증을 참조하세요. • AWS API는 IAM 사용자 설명서에서 IAM 사용자의 액세스 키 관리를 참조하세요.

다음 예에서는 암호가 비활성화된 사용자를 만듭니다.

```
create user temp_creds_user password disable;
```

다음 예에서는 기존 사용자의 암호를 비활성화합니다.

```
alter user temp_creds_user password disable;
```

3. [CREATE GROUP](#)으로 데이터베이스 사용자 그룹을 새로 만듭니다.
4. [GRANT](#) 명령으로 그룹의 액세스 권한을 정의합니다.

5단계: IAM 자격 증명을 사용하도록 JDBC 또는 ODBC 연결 구성

Amazon Redshift JDBC 또는 ODBC 드라이버를 사용하여 SQL 클라이언트를 구성할 수 있습니다. 이 드라이버는 데이터베이스 사용자 자격 증명을 생성하고 SQL 클라이언트와 Amazon Redshift 데이터베이스 간의 연결을 설정하는 프로세스를 관리합니다.

인증에 자격 증명 공급자를 쓰는 경우, 자격 증명 공급자 플러그인의 이름을 지정합니다. Amazon Redshift JDBC 및 ODBC 드라이버에는 다음 SAML 기반의 자격 증명 공급자를 위한 플러그인이 포함됩니다.

- AD FS(Active Directory Federation Services)
- PingOne
- Okta
- Microsoft Azure AD

Microsoft Azure AD를 자격 증명 공급자로 설정하는 단계는 [JDBC 또는 ODBC Single Sign-On 인증 설정](#) 단원을 참조하십시오.

IAM 자격 증명을 사용하도록 JDBC 연결 구성

1. [Amazon Redshift용 JDBC 드라이버 버전 2.x 연결 구성](#) 페이지에서 최신 Amazon Redshift JDBC 드라이버를 다운로드합니다.
2. 다음 형식 중 한 가지로 IAM 자격 증명 옵션이 있는 JDBC URL을 새로 만듭니다. IAM 인증을 사용하려면 다음 예시에 보이는 것처럼 `jdbc:redshift:` 다음에 `iam:`을 Amazon Redshift JDBC URL에 넣습니다.

```
jdbc:redshift:iam://
```

`cluster-name`, `region` 및 `account-id`를 추가합니다. JDBC 드라이버는 IAM 계정 정보와 클러스터 이름을 이용해 클러스터 ID와 AWS 리전을 검색합니다. 이렇게 하려면 사용자 또는 역할에 지정된 클러스터로 `redshift:DescribeClusters` 작업을 호출할 권한이 있어야 합니다. 사용자 또는 역할에 `redshift:DescribeClusters` 작업 호출 권한이 없다면, 다음 예에 보이는 것처럼 클러스터 ID, AWS 리전, 포트를 포함합니다. 포트 번호는 선택에 따릅니다.

```
jdbc:redshift:iam://examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com:5439/dev
```

3. JDBC 옵션을 추가하여 IAM 자격 증명을 제공하십시오. 다른 JDBC 옵션 조합을 이용해 IAM 자격 증명을 제공할 수 있습니다. 자세한 내용은 [데이터베이스 사용자 자격 증명 생성을 위한 JDBC 및 ODBC 옵션](#)을 참조하세요.

다음 URL은 사용자에게 대한 `AccessKeyId`와 `SecretAccessKey`를 지정합니다.

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?
AccessKeyID=AKIAIOSFODNN7EXAMPLE&SecretAccessKey=wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY
```

다음 예에서는 IAM 자격 증명을 포함한 이름 있는 프로필을 지정합니다.

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?Profile=user2
```

4. JDBC 드라이버가 GetClusterCredentials API 작업을 호출하는 데 사용하는 JDBC 옵션을 추가합니다. GetClusterCredentials API 작업을 프로그래밍 방식으로 호출하는 경우 이러한 옵션을 포함하지 마십시오.

다음 예에는 JDBC GetClusterCredentials 옵션이 포함되어 있습니다.

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?
plugin_name=com.amazon.redshift.plugin.AzureCredentialsProvider&UID=user&PWD=password&idp_t
```

IAM 자격 증명을 사용하도록 ODBC 연결 구성

다음 절차에서는 IAM 인증 구성을 위한 단계만 안내합니다. 데이터베이스 사용자 이름과 암호를 이용한 표준 인증의 사용 단계는 [Amazon Redshift용 ODBC 드라이버 버전 2.x 연결 구성](#) 단원을 참조하십시오.

1. 운영 체제에 맞는 최신 Amazon Redshift ODBC 드라이버를 설치하고 구성합니다. 자세한 내용은 [Amazon Redshift용 ODBC 드라이버 버전 2.x 연결 구성](#) 페이지를 참조하십시오.

Important

Amazon Redshift ODBC 드라이버는 버전 1.3.6.1000 이상이어야 합니다.

2. 운영 체제에 맞는 단계를 따라 연결 설정을 구성하십시오.
3. Microsoft Windows 운영 체제에서는 [Amazon Redshift ODBC 드라이버 DSN 설정(Amazon Redshift ODBC Driver DSN Setup)] 창에 액세스합니다.
 - a. 연결 설정(Connection Settings)에서 다음 정보를 입력하세요.
 - Data Source Name(데이터 원본 이름)
 - 서버(선택 사항)

- 포트(선택 사항)
- 데이터베이스:

사용자 또는 역할에 `redshift:DescribeClusters` 작업을 호출할 수 있는 권한이 있는 경우 Data Source Name(데이터 소스 이름)과 Database(데이터베이스)만 필요합니다. Amazon Redshift는 [ClusterId]와 [리전(Region)]으로 `DescribeCluster` 작업을 호출하여 서버와 포트를 가져옵니다.

사용자 또는 역할에 `redshift:DescribeClusters` 작업을 호출할 권한이 없다면 서버 및 포트를 지정하십시오.

- b. 인증 아래에서 Auth Type(인증 유형) 값을 선택합니다.

각 인증 유형에 대해 다음과 같이 값을 입력합니다.

AWS 프로파일

다음 정보를 입력합니다.

- ClusterID
- 리전(Region)
- 프로파일 이름

ODBC 연결 옵션에 대한 값을 포함한 AWS config 파일에 프로파일 이름을 입력합니다. 자세한 내용은 [구성 프로파일 사용](#) 섹션을 참조하세요.

(선택 사항) ODBC 드라이버가 `GetClusterCredentials` API 작업을 호출하는 데 사용하는 옵션에 대한 세부 정보를 입력합니다.

- DbUser
- User AutoCreate
- DbGroups

자세한 내용은 [데이터베이스 사용자 자격 증명 생성을 위한 JDBC 및 ODBC 옵션](#) 섹션을 참조하세요.

IAM 자격 증명

다음 정보를 입력합니다.

- ClusterID
- 리전(Region)
- AccessKeyID 및 SecretAccessKey

IAM 데이터베이스 인증용으로 구성된 IAM 역할 또는 사용자에게 대한 액세스 키 ID와 비밀 액세스 키.

- SessionToken

SessionToken은 임시 자격 증명을 가진 IAM 역할에 필요합니다. 자세한 내용은 [임시 보안 자격 증명](#)을 참조하십시오.

ODBC 드라이버가 GetClusterCredentials API 작업을 호출하는 데 사용하는 옵션에 대한 세부 정보를 입력합니다.

- DbUser(필수)
- User AutoCreate(선택 사항)
- DbGroups(선택 사항)

자세한 내용은 [데이터베이스 사용자 자격 증명 생성을 위한 JDBC 및 ODBC 옵션](#) 섹션을 참조하세요.

자격 증명 공급자: AD FS

AD FS와의 Windows 통합 인증의 경우 User(사용자) 및 Password(암호)를 빈 칸으로 놔두십시오.

IdP 세부 정보를 제공합니다.

- IdP Host

기업 자격 증명 제공업체 호스트의 이름입니다. 이 이름에 슬래시(/)가 포함되면 안 됩니다.

- IdP Port(선택 사항)

자격 증명 공급자가 사용하는 포트. 기본값은 443입니다.

- Preferred Role

SAML 어설션에서 AttributeValue 속성에 대한 다중 값 Role 요소에서 IAM 역할의 Amazon 리소스 이름(ARN)입니다. 선호되는 역할에 적절한 값을 찾으려면 IdP 관리자

에게 문의하십시오. 자세한 내용은 [2단계: IdP에 대한 SAML 어설션 구성](#) 섹션을 참조하세요.

(선택 사항) ODBC 드라이버가 GetClusterCredentials API 작업을 호출하는 데 사용하는 옵션에 대한 세부 정보를 입력합니다.

- DbUser
- User AutoCreate
- DbGroups

자세한 내용은 [데이터베이스 사용자 자격 증명 생성을 위한 JDBC 및 ODBC 옵션](#) 섹션을 참조하세요.

자격 증명 공급자: PingFederate

사용자(User) 및 암호>Password)에 IdP 사용자 이름과 암호를 입력합니다.

IdP 세부 정보를 제공합니다.

- IdP Host

기업 자격 증명 제공업체 호스트의 이름입니다. 이 이름에 슬래시(/)가 포함되면 안 됩니다.

- IdP Port(선택 사항)

자격 증명 공급자가 사용하는 포트. 기본값은 443입니다.

- Preferred Role

SAML 어설션에서 AttributeValue 속성에 대한 다중 값 Role 요소에서 IAM 역할의 Amazon 리소스 이름(ARN)입니다. 선호되는 역할에 적절한 값을 찾으려면 IdP 관리자에게 문의하십시오. 자세한 내용은 [2단계: IdP에 대한 SAML 어설션 구성](#) 섹션을 참조하세요.

(선택 사항) ODBC 드라이버가 GetClusterCredentials API 작업을 호출하는 데 사용하는 옵션에 대한 세부 정보를 입력합니다.

- DbUser
- User AutoCreate
- DbGroups

자세한 내용은 [데이터베이스 사용자 자격 증명 생성을 위한 JDBC 및 ODBC 옵션](#) 섹션을 참조하세요.

자격 증명 공급자: Okta

User(사용자) 및 Password(암호)에 IdP 사용자 이름과 암호를 입력합니다.

IdP 세부 정보를 제공합니다.

- IdP Host

기업 자격 증명 제공업체 호스트의 이름입니다. 이 이름에 슬래시(/)가 포함되면 안 됩니다.

- IdP Port

이 값은 Okta에서 사용되지 않습니다.

- Preferred Role

SAML 어설션에서 AttributeValue 속성에 대한 Role 요소에서 IAM 역할의 Amazon 리소스 이름(ARN)입니다. 선호되는 역할에 적절한 값을 찾으려면 IdP 관리자에게 문의하십시오. 자세한 내용은 [2단계: IdP에 대한 SAML 어설션 구성](#) 섹션을 참조하세요.

- Okta App ID

Okta 애플리케이션을 위한 ID. Okta Application Embed Link에서 App ID의 값 뒤에 "amazon_aws"가 붙습니다. IdP 관리자에게 문의해 이 값을 얻으십시오.

(선택 사항) ODBC 드라이버가 GetClusterCredentials API 작업을 호출하는 데 사용하는 옵션에 대한 세부 정보를 입력합니다.

- DbUser

- User AutoCreate

- DbGroups

자세한 내용은 [데이터베이스 사용자 자격 증명 생성을 위한 JDBC 및 ODBC 옵션](#) 섹션을 참조하세요.

자격 증명 공급자: Azure AD

User(사용자) 및 Password(암호)에 IdP 사용자 이름과 암호를 입력합니다.

[클러스터 ID(Cluster ID)] 및 [리전(Region)]에 클러스터 ID와 Amazon Redshift 클러스터의 AWS 리전을 입력합니다.

[데이터베이스(Database)]에 Amazon Redshift 클러스터에서 생성한 데이터베이스를 입력합니다.

IdP 세부 정보를 제공합니다.

- IdP 테넌트

Azure AD에 사용되는 테넌트입니다.

- Azure 클라이언트 암호

Azure의 Amazon Redshift 엔터프라이즈 앱의 클라이언트 암호입니다.

- Azure 클라이언트 ID

Azure의 Amazon Redshift 엔터프라이즈 앱의 클라이언트 ID(애플리케이션 ID)입니다.

(선택 사항) ODBC 드라이버가 GetClusterCredentials API 작업을 호출하는 데 사용하는 옵션에 대한 세부 정보를 입력합니다.

- DbUser
- User AutoCreate
- DbGroups

자세한 내용은 [데이터베이스 사용자 자격 증명 생성을 위한 JDBC 및 ODBC 옵션](#) 섹션을 참조하세요.

IAM 자격 증명 제공 옵션

JDBC 또는 ODBC 연결에 IAM 자격 증명을 제공하려면 다음 옵션 중 한 가지를 선택합니다.

- AWS 프로필

JDBC 또는 ODBC 설정의 형태로 자격 증명 값을 제공하는 방식의 대안으로 이름 있는 프로필에 값을 입력할 수 있습니다. 자세한 내용은 [구성 프로필 사용](#) 섹션을 참조하세요.

- IAM 보안 인증

AccessKeyId, SecretAccessKey, 그리고 옵션으로 SessionToken에 대한 값을 JDBC 또는 ODBC 설정의 형태로 제공합니다. SessionToken은 임시 자격 증명을 가진 IAM 역할에만 필요합니다. 자세한 내용은 [IAM 자격 증명 제공을 위한 JDBC 및 ODBC 옵션](#) 섹션을 참조하세요.

- ID 제공업체 페더레이션

자격 증명 공급자 페더레이션을 사용하여 자격 증명 공급자의 사용자가 Amazon Redshift에 인증할 수 있도록 하는 경우 자격 증명 공급자 플러그인의 이름을 지정합니다. 자세한 내용은 [자격 증명 제공업체 플러그인](#) 섹션을 참조하세요.

Amazon Redshift JDBC 및 ODBC 드라이버에는 다음 SAML 기반의 자격 증명 페더레이션 자격 증명 공급자를 위한 플러그인이 포함됩니다.

- Microsoft Active Identity Federation Services(AD FS)
- PingOne
- Okta
- Microsoft Azure Active Directory (Azure AD)

JDBC 또는 ODBC 설정의 형태로 혹은 프로필을 이용해 플러그인 이름과 관련 값을 입력할 수 있습니다. 자세한 내용은 [JDBC 드라이버 버전 2.x 구성 옵션](#) 섹션을 참조하세요.

자세한 내용은 [5단계: IAM 자격 증명을 사용하도록 JDBC 또는 ODBC 연결 구성](#) 섹션을 참조하세요.

구성 프로필 사용

IAM 자격 증명 옵션과 GetClusterCredentials 옵션은 AWS 구성 파일에서 이름 있는 프로파일의 설정대로 제공할 수 있습니다. 프로필 이름을 제공하려면 Profile JDBC 옵션을 사용합니다. 구성은 홈 디렉터리의 .aws 폴더의 config 또는 credentials 파일에 저장되어 있습니다.

Amazon Redshift JDBC 또는 ODBC 드라이버에 포함된 SAML 기반 자격 증명 공급자 플러그인의 경우 [자격 증명 제공업체 플러그인](#)의 바로 앞에서 설명한 설정을 사용할 수 있습니다. plugin_name을 사용하지 않으면 다른 옵션은 무시됩니다.

다음 예에서는 2개의 프로파일이 있는 ~/.aws/credentials 파일을 보여줍니다.

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

[user2]
```



```
aws_access_key_id=AKIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
session_token=AQoDYXdzEPT/////////
wEXAMPLEtc764bNrC9SAPBSM22wD0k4x4HIZ8j4FZTwdQWLWsKWHGBuFqwAeMicRXmxfpSPfIeoIYRqTf1fKD8YUuwthAx7
qkPpKPi/kMcGd
QrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburEDy0KPkyQDYwT7WZ0wq5VSXDvp75YU
9HFv1Rd8Tx6q6fE8YQcHNVXAKiY9q6d+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
+scqKmlzm8FDrypNC9Yjc8fP0Ln9FX9KSYvKTr4rvx3iSi1tJabIQwj2ICCR/oLxBA==
```

user2 예시에 대한 자격 증명을 사용하려면 JDBC URL에서 Profile=user2를 지정하십시오.

프로파일 사용에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 [구성 및 자격 증명 파일 설정](#)을 참조하세요.

JDBC 드라이버에 대한 프로필 사용에 대한 자세한 내용은 [프로파일 지정](#)을 참조하세요.

ODBC 드라이버에 대한 프로필 사용에 대한 자세한 내용은 [인증 방법](#)을 참조하세요.

IAM 자격 증명 제공을 위한 JDBC 및 ODBC 옵션

다음 표는 IAM 자격 증명 제공을 위한 JDBC 및 ODBC 옵션을 목록으로 보여줍니다.

옵션	설명
Iam	ODBC 연결 문자열 전용. IAM 인증을 사용하려면 1로 설정합니다.
AccessKey ID	IAM 데이터베이스 인증용으로 구성한 IAM 역할 또는 사용자의 액세스 키 ID 및 비밀 액세스 키. SessionToken 은 임시 자격 증명에 필요합니다. SessionToken은 사용자에게 사용되지 않습니다. 자세한 내용은 임시 보안 자격 증명 을 참조하십시오.
SecretAccessKey	
SessionToken	
plugin_name	자격 증명 공급자를 실행하는 완전히 정규화된 클래스 이름. Amazon Redshift JDBC 드라이버에는 SAML 기반의 자격 증명 공급자 플러그인이 포함됩니다. plugin_name 을 제공하는 경우, 다른 관련 옵션도 제공할 수 있습니다. 자세한 내용은 자격 증명 제공업체 플러그인 섹션을 참조하세요.

옵션	설명
Profile	AWS 자격 증명의 프로파일 이름 또는 JDBC에 대한 값을 포함한 config 파일입니다. 자세한 내용은 구성 프로파일 사용 섹션을 참조하세요.

데이터베이스 사용자 자격 증명 생성을 위한 JDBC 및 ODBC 옵션

Amazon Redshift JDBC 또는 ODBC 드라이버로 데이터베이스 사용자 자격 증명을 생성하려면 데이터베이스 사용자 이름을 JDBC 또는 ODBC 옵션으로 제공합니다. 선택에 따라 데이터베이스 사용자가 없으면 드라이버가 새로 만들도록 할 수 있고, 사용자가 로그인 시 조인하는 데이터베이스 사용자 그룹 목록을 지정할 수 있습니다.

자격 증명 공급자(IdP)를 사용할 경우, IdP 관리자에게 문의하여 이들 옵션에 올바른 값을 정하십시오. IdP 관리자는 이러한 옵션 제공을 위해 사용자의 IdP를 구성할 수도 있으며, 이 경우 JDBC 또는 ODBC 옵션으로 제공하지 않아도 됩니다. 자세한 내용은 [2단계: IdP에 대한 SAML 어설션 구성](#) 섹션을 참조하세요.

Note

`${redshift:DbUser}`에서 설명한 대로 IAM 정책 변수 [GetClusterCredentials에 대한 리소스 정책](#)을 사용할 경우, DbUser의 값이, API 작업의 요청 컨텍스트에서 검색한 값으로 대체됩니다. Amazon Redshift 드라이버는 SAML 속성으로 제공된 값이 아니라 연결 URL에서 제공한 DbUser 변수의 값을 사용합니다.

이 구성의 보안을 위해 IAM 정책의 조건을 사용하여 DbUser로 RoleSessionName 값을 확인할 것을 권장합니다. IAM 정책을 사용하여 조건을 설정하는 방법의 예는 [예제 8: GetClusterCredentials 사용을 위한 IAM 정책 예제](#)에서 볼 수 있습니다.

다음 표는 데이터베이스 사용자의 자격 증명 생성 옵션을 목록으로 보여줍니다.

옵션	설명
DbUser	데이터베이스 사용자의 이름. 데이터베이스에 DbUser라는 사용자 이름이 있으면 임시 사용자 자격 증명이 기존 사용자와 동일한 권한을 갖습니다. 데이터베이스에 DbUser가 없고 AutoCreate가 True인 경우, 새 사용자 이름 DbUser를 만듭니다. 선택에 따라 기존 사용자의 암호를 비활성화하십시오. 자세한 내용은 ALTER_USER 를 참조하십시오.

옵션	설명
AutoCreate	데이터베이스 사용자가 없어서 DbUser에 지정된 이름으로 새로 만들려면 true를 지정합니다. 기본값은 false입니다.
DbGroups	데이터베이스 사용자가 현재 세션에서 조인하는 둘 이상의 기존 데이터베이스 그룹을 쉼표로 구분한 이름 목록. 기본적으로 PUBLIC에만 신규 사용자가 추가됩니다.

자격 증명 제공업체 플러그인

Amazon Redshift는 Single Sign-On 인증을 위해 자격 증명 공급자 플러그인을 사용합니다.

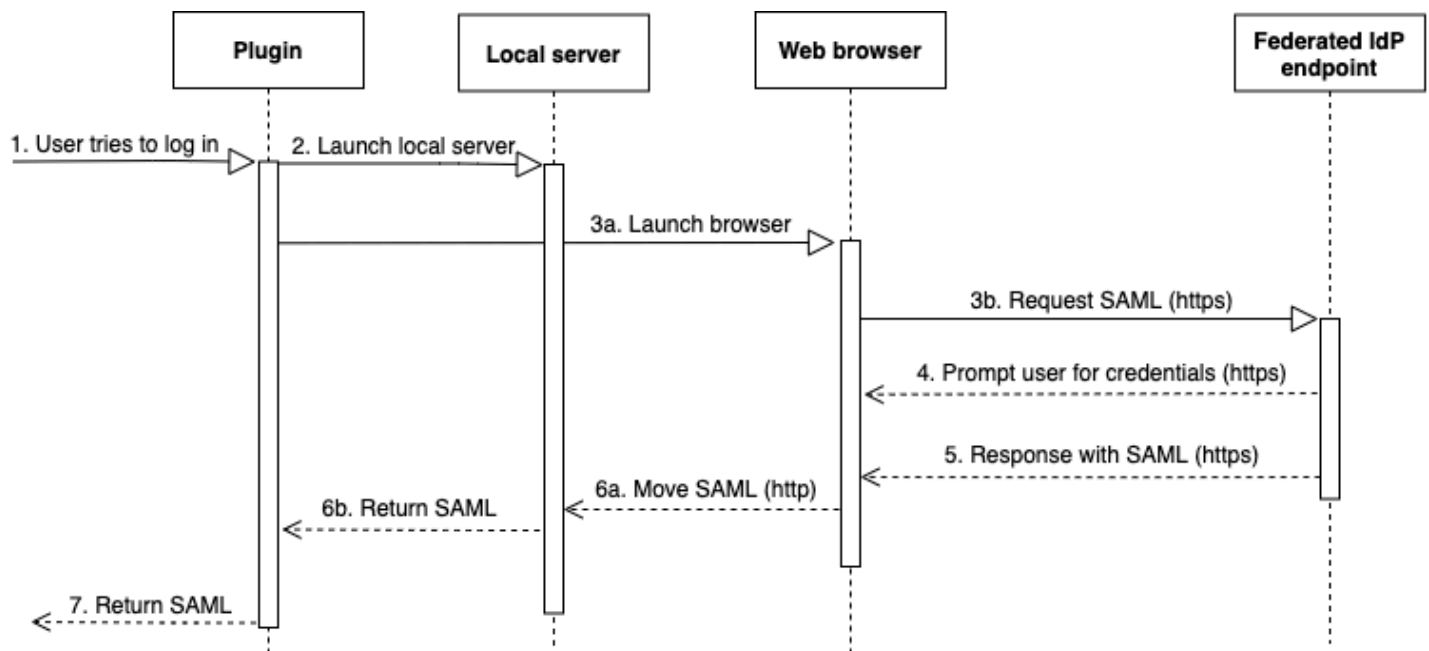
Single Sign-On 인증을 지원하기 위해 Amazon Redshift는 Microsoft Azure Active Directory용 Azure AD 플러그인을 제공합니다. 이 플러그인을 구성하는 방법에 대한 자세한 내용은 [JDBC 또는 ODBC Single Sign-On 인증 설정](#) 단원을 참조하십시오.

다중 인증

다중 인증

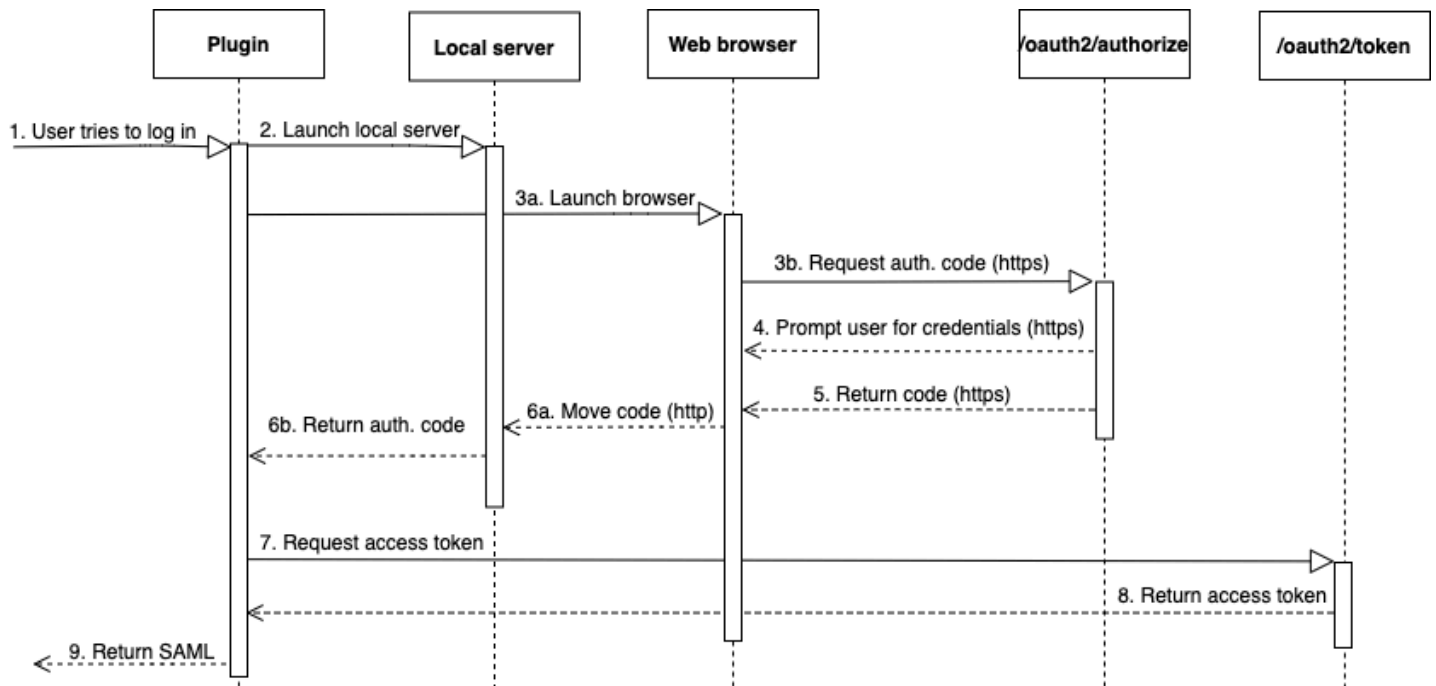
멀티 팩터 인증(MFA)을 지원하기 위해 Amazon Redshift는 브라우저 기반 플러그인을 제공합니다. Okta 및 PingOne용 브라우저 SAML 플러그인과 Microsoft Azure Active Directory용 브라우저 Azure AD 플러그인을 사용합니다.

브라우저 SAML 플러그인의 경우, OAuth 인증은 다음과 같이 흐릅니다.



1. 사용자가 로그인을 시도합니다.
2. 플러그인은 로컬 서버를 시작하여 로컬 호스트에서 들어오는 연결을 수신합니다.
3. 플러그인은 웹 브라우저를 시작하여 지정된 Single Sign-On 로그인 URL 페더레이션 자격 증명 공급자 엔드포인트에서 HTTPS를 통해 SAML 응답을 요청합니다.
4. 웹 브라우저는 링크를 따라 사용자에게 자격 증명을 입력하라는 메시지를 표시합니다.
5. 사용자가 인증하고 동의하면 연동 자격 증명 공급자 엔드포인트가 HTTPS를 통해 `redirect_uri`에 표시된 URI에 SAML 응답을 반환합니다.
6. 웹 브라우저는 SAML 응답과 함께 응답 메시지를 표시된 `redirect_uri`로 이동합니다.
7. 로컬 서버는 수신 연결을 수락하고 플러그인은 SAML 응답을 검색하여 Amazon Redshift에 전달합니다.

브라우저 Azure AD 플러그인의 경우, SAML 인증은 다음과 같이 흐릅니다.



1. 사용자가 로그인을 시도합니다.
2. 플러그인은 로컬 서버를 시작하여 로컬 호스트에서 들어오는 연결을 수신합니다.
3. 플러그인은 웹 브라우저를 시작하여 Azure AD `oauth2/authorize` 엔드포인트로부터 권한 부여 코드를 요청합니다.
4. 웹 브라우저는 HTTPS를 통해 생성된 링크를 따라 사용자에게 자격 증명을 입력하라는 메시지를 표시합니다. 링크는 테넌트 및 `client_id`와 같은 구성 속성을 사용하여 생성됩니다.

5. 사용자가 인증하고 동의하면 Azure AD oauth2/authorize 엔드포인트가 HTTPS를 통해 표시된 `redirect_uri`에 권한 부여 코드와 함께 응답을 반환하고 보냅니다.
6. 웹 브라우저는 SAML 응답과 함께 응답 메시지를 표시된 `redirect_uri`로 이동합니다.
7. 로컬 서버는 수신 연결 및 플러그인 요청을 수락하고 권한 부여 코드를 검색하고 Azure AD oauth2/token 엔드포인트에 POST 요청을 보냅니다.
8. Azure AD oauth2/token 엔드포인트는 표시된 `redirect_uri`에 액세스 토큰이 있는 응답을 반환합니다.
9. 플러그인은 SAML 응답을 검색하여 Amazon Redshift에 전달합니다.

다음 단원을 참조하십시오.

- AD FS(Active Directory Federation Services)

자세한 내용은 [JDBC 또는 ODBC Single Sign-On 인증 설정](#) 섹션을 참조하세요.

- PingOne(Ping)

Ping은 Forms 인증을 사용하여 사전에 지정한 PingOne IdP Adapter로만 지원됩니다.

자세한 내용은 [JDBC 또는 ODBC Single Sign-On 인증 설정](#) 섹션을 참조하세요.

- Okta

Okta는 AWS Management Console과 함께 사용되는 Okta 제공 애플리케이션에만 지원됩니다.

자세한 내용은 [JDBC 또는 ODBC Single Sign-On 인증 설정](#) 섹션을 참조하세요.

- Microsoft Azure Active Directory

자세한 내용은 [JDBC 또는 ODBC Single Sign-On 인증 설정](#) 섹션을 참조하세요.

플러그인 옵션

플러그인 옵션

SAML 기반의 자격 증명 공급자 플러그인을 사용하려면 JDBC 또는 ODBC 옵션을 사용하거나 명명된 프로파일에서 다음 옵션을 지정합니다. `plugin_name`을 지정하지 않으면 다른 옵션은 무시됩니다.

옵션	설명
plugin_name	JDBC의 경우 자격 증명 공급자를 실행하는 클래스 이름입니다. 다음 중 하나를 지정하세요. - Active Directory Federation Services의 경우 com.amazon.redshift.plugin.AdfsCredentialsProvider - Okta의 경우 com.amazon.redshift.plugin.OktaCredentialsProvider - PingFederate의 경우 com.amazon.redshift.plugin.PingCredentialsProvider - Microsoft Azure Active Directory의 경우 com.amazon.redshift.plugin.AzureCredentialsProvider - SAML MFA의 경우 com.amazon.redshift.plugin.BrowserSamlCredentialsProvider - MFA를 사용하는 Microsoft Azure Active Directory Single Sign-On의 경우 com.amazon.redshift.plugin.BrowserAzureCredentialsProvider ODBC의 경우 다음 중 하나를 지정하십시오. - Active Directory Federation Services의 경우: adfs - Okta의 경우: okta - PingFederate의 경우: ping - Microsoft Azure Active Directory의 경우: azure - SAML MFA의 경우: browser saml

옵션	설명
	MFA를 사용하는 Microsoft Azure Active Directory Single Sign-On의 경우: browser azure ad
idp_host	기업 자격 증명 제공업체 호스트의 이름입니다. 이 이름에 슬래시(/)가 포함되면 안 됩니다. Okta 자격 증명 공급자의 경우 idp_host의 값은 .okta.com 으로 끝나야 합니다.
idp_port	자격 증명 공급자가 사용하는 포트. 기본값은 443입니다. Okta의 경우 이 포트를 무시합니다.
preferred_role	SAML 어설션에서 AttributeValue 속성에 대한 Role 요소의 역할 Amazon 리소스 이름(ARN)입니다. 선호되는 역할에 적절한 값을 찾으려면 IdP 관리자에게 문의하십시오. 자세한 내용은 2단계: IdP에 대한 SAML 어설션 구성 섹션을 참조하세요.
user	해당되는 경우 도메인을 포함하는 기업 사용자 이름입니다. 예를 들어 Active Directory에서는 domain\username 형식의 도메인 이름이 필수입니다.
암호	기업 사용자의 암호. 이 옵션의 사용은 권장하지 않습니다. 대신 SQL 클라이언트를 사용해 암호를 공급하십시오.
app_id	Okta 애플리케이션을 위한 ID. Okta에만 사용됩니다. Okta application embed link에서 app_id 값 뒤에 amazon_aws 가 붙습니다. 이 값을 얻으려면 IdP 관리자에게 문의하십시오. 다음은 애플리케이션 내장 링크의 예입니다. https://example.okta.com/home/amazon_aws/0oa2hylwrpM8UGehd1t7/272
idp_tenant	Azure AD에 사용되는 테넌트입니다. Azure에서만 사용됩니다.
client_id	Azure AD의 Amazon Redshift 엔터프라이즈 애플리케이션의 클라이언트 ID입니다. Azure에서만 사용됩니다.

Amazon Redshift CLI 또는 API를 이용한 IAM ID에 대한 데이터베이스 자격 증명 생성

임시 데이터베이스 사용자 자격 증명을 프로그래밍 방식으로 생성하기 위해 Amazon Redshift는 [get-cluster-credentials](#) 명령을 AWS Command Line Interface(AWS CLI) 및 [GetClusterCredentials](#) API 작업에 제공합니다. 또는 GetClusterCredentials 작업을 호출하고, 데이터베이스 사용자 자격 증명을 검색하고, SQL 클라이언트와 Amazon Redshift 데이터베이스 사이를 연결하는 과정을 관리하는

Amazon Redshift JDBC 또는 ODBC 드라이버로 SQL 클라이언트를 구성할 수 있습니다. 자세한 내용은 [데이터베이스 사용자 자격 증명 생성을 위한 JDBC 및 ODBC 옵션](#) 섹션을 참조하세요.

Note

Amazon Redshift JDBC 또는 ODBC 드라이버를 사용해 데이터베이스 사용자 자격 증명을 만드는 방식을 권장합니다.

이 단원에서는 `GetClusterCredentials` 작업이나 `get-cluster-credentials` 명령을 프로그램 방식으로 호출하고, 데이터베이스 사용자 자격 증명을 검색하고, 데이터베이스에 연결하는 단계를 설명합니다.

임시 데이터베이스 자격 증명의 생성 및 사용

- 필수 권한으로 역할 또는 사용자를 새로 만들거나 수정하십시오. IAM 권한에 대한 자세한 내용은 [3단계: GetClusterCredentialsWithIAM 또는 GetClusterCredentials 직접 호출 권한이 있는 IAM 역할 생성](#) 단원을 참조하세요.
- 사용자 또는 역할로서 귀하는 이전 단계에서 `get-cluster-credentials` CLI 명령을 실행하거나 `GetClusterCredentials` API 작업을 호출하고 다음 값을 제공할 권한이 있습니다.
 - [클러스터 식별자(Cluster identifier)] – 데이터베이스가 포함된 클러스터의 이름.
 - [데이터베이스 사용자 이름(Database user name)] – 기존 또는 신규 데이터베이스 사용자의 이름.
 - 데이터베이스에 사용자가 없고 `AutoCreate`가 `True`인 경우, `PASSWORD` 비활성 상태로 새 사용자를 만듭니다.
 - 사용자가 없고 `AutoCreate`가 `False`인 경우 이 요청이 실패합니다.
 - 이 예에서 데이터베이스 사용자 이름은 `temp_creds_user`입니다.
 - [`Autocreate`] – (옵션) 데이터베이스 사용자 이름이 존재하지 않으면 새 사용자를 생성합니다.
 - [데이터베이스 이름(Database name)] – (옵션) 사용자가 로그인할 권한이 있는 데이터베이스의 이름. 데이터베이스 이름이 지정되지 않은 경우 사용자는 모든 클러스터 데이터베이스에 로그인할 수 있습니다.
 - [데이터베이스 그룹(Database groups)] – (옵션) 기존 데이터베이스 사용자 그룹의 목록. 성공적으로 로그인하면 지정된 사용자 그룹에 데이터베이스 사용자가 추가됩니다. 그룹을 지정하지 않으면 사용자는 `PUBLIC` 권한만 가집니다. 사용자 그룹 이름은 사용자 또는 역할에 연결된 IAM 정책에 지정된 `dbgroup` 리소스 ARN과 일치해야 합니다.

- [만료 시간(Expiration time)] – (옵션) 임시 자격 증명이 만료할 때까지 남은 시간(초). 900초(15분)~3600초(60분) 사이로 지정할 수 있습니다. 기본값은 900초입니다.
3. Amazon Redshift는 사용자가 지정된 리소스로 GetClusterCredentials 작업을 호출할 권한이 있는지 검증합니다.
 4. Amazon Redshift는 임시 암호와 데이터베이스 사용자 이름을 반환합니다.

다음 예에서는 Amazon Redshift CLI를 사용하여 temp_creds_user라는 기존 사용자에게 대한 임시 데이터베이스 자격 증명을 새로 만듭니다.

```
aws redshift get-cluster-credentials --cluster-identifier examplecluster --db-user temp_creds_user --db-name exampledb --duration-seconds 3600
```

쿼리 결과는 다음과 같습니다.

```
{
  "DbUser": "IAM:temp_creds_user",
  "Expiration": "2016-12-08T21:12:53Z",
  "DbPassword": "EXAMPLEjArE3hcnQj8zt4XQj9Xtma8oxYEM80yxpDHwXVPyJYBDm/gqX2Eeaq6P3DgTzgPg=="
}
```

다음 예에서는 Amazon Redshift CLI와 autocreate를 사용하여 신규 사용자에게 대한 임시 데이터베이스 자격 증명을 생성하고 이 사용자를 그룹 example_group에 추가합니다.

```
aws redshift get-cluster-credentials --cluster-identifier examplecluster --db-user temp_creds_user --auto-create --db-name exampledb --db-groups example_group --duration-seconds 3600
```

쿼리 결과는 다음과 같습니다.

```
{
  "DbUser": "IAMA:temp_creds_user:example_group",
  "Expiration": "2016-12-08T21:12:53Z",
  "DbPassword": "EXAMPLEjArE3hcnQj8zt4XQj9Xtma8oxYEM80yxpDHwXVPyJYBDm/gqX2Eeaq6P3DgTzgPg=="
}
```

5. Amazon Redshift 클러스터로 Secure Socket Layer(SSL) 인증을 연결하고 사용자 이름과 암호로 GetClusterCredentials 응답에서 로그인 요청을 보냅니다. 사용자 이름에 IAM: 또는

IAMA:의 예처럼 IAM:temp_creds_user 또는 IAMA:temp_creds_user 접두사를 넣으십시오.

Important

SQL 클라이언트를 구성하여 SSL을 요청합니다. 혹은 SQL 클라이언트가 자동으로 SSL과 연결 시도를 하면 어떤 식으로든 실패할 경우 SSL가 아닌 곳으로 대체할 수 있습니다. 이 경우 자격 증명이 만료되었거나 유효하지 않기 때문에 첫 번째 연결 시도가 실패할 수 있으며, 이후 SSL 연결이 아니기 때문에 두 번째 연결이 실패합니다. 이렇게 되면 첫 번째 오류 메시지가 누락될 수 있습니다. SSL을 이용한 클러스터 연결에 대한 자세한 내용은 [연결을 위한 보안 옵션 구성](#) 단원을 참조하십시오.

6. 연결이 SSL을 사용하지 않으면 연결 시도가 실패합니다.
7. 클러스터가 SQL 클라이언트로 authentication 요청을 보냅니다.
8. 그러면 SQL 클라이언트가 클러스터로 임시 암호를 보냅니다.
9. 암호가 유효하고 아직 만료되지 않았다면 클러스터가 연결을 완료합니다.

JDBC 또는 ODBC Single Sign-On 인증 설정

외부 ID 제공업체(idP)를 활용하여 사용자가 Amazon Redshift 클러스터에 액세스하도록 인증하고 권한을 부여하여 사용자 관리를 간소화하고 보안을 강화할 수 있습니다. 이를 통해 여러 서비스 전반에서 중앙 집중식 사용자 관리, 역할 기반 액세스 제어 및 감사 기능을 사용합니다. 일반적인 사용 사례에는 다양한 사용자 그룹에 대한 인증 간소화, 일관된 액세스 정책 적용, 규제 요구 사항 충족이 포함됩니다.

다음 페이지에서는 Redshift 클러스터와 idP 통합을 구성하는 방법을 안내합니다. IdP에 대한 서비스 공급자로 AWS 구성에 대한 자세한 내용은 IAM User Guide의 [Configuring Your SAML 2.0 IdP with Relying Party Trust and Adding Claims](#)를 참조하세요.

AD FS

이 자습서에서는 AD FS를 ID 제공업체(idP)로 사용하여 Amazon Redshift 클러스터에 액세스하는 방법을 보여 줍니다.

1단계: AD FS와 AWS 계정을 서로 신뢰하도록 설정

다음 절차에서는 신뢰 관계를 설정하는 방법을 설명합니다.

1. AD FS 사용자가 연결할 기존 Amazon Redshift 클러스터를 사용하거나 생성합니다. 연결을 구성하려면 이 클러스터의 특정 속성(예: 클러스터 식별자)이 필요합니다. 자세한 내용은 [클러스터 생성](#) 섹션을 참조하세요.
2. Microsoft Management Console에서 Amazon Redshift 액세스를 제어하도록 AD FS를 설정합니다.
 1. ADFS 2.0을 선택한 후 Add Relying Party Trust(신뢰 당사자 신뢰 추가)를 선택합니다. Add Relying Party Trust Wizard(신뢰 당사자 신뢰 추가 마법사) 페이지에서 시작을 선택합니다.
 2. Select Data Source(데이터 원본 선택) 페이지에서 Import data about the relying party published online or on a local network(온라인 또는 로컬 네트워크에 게시된 신뢰 당사자에 대한 데이터 가져오기)를 선택합니다.
 3. Federation metadata address (host name or URL)(연동 메타데이터 주소(호스트 이름 또는 URL))에 **https://signin.aws.amazon.com/saml-metadata.xml**을 입력합니다. 메타데이터 XML 파일은 AWS를 신뢰 당사자로 설명하는 표준 SAML 메타데이터 문서입니다.
 4. 표시 이름 지정 페이지에서 표시 이름에 값을 입력합니다.
 5. Choose Issuance Authorization Rules(발급 권한 부여 규칙 선택) 페이지에서 발급 권한 부여 규칙을 선택하여 모든 사용자가 이 신뢰 당사자에 액세스할 수 있도록 허용하거나 거부합니다.
 6. Ready to Add Trust(신뢰 추가 준비 완료) 페이지에서 설정을 검토합니다.
 7. Finish(완료) 페이지에서 Open the Edit Claim Rules dialog for this relying party trust when the wizard closes(마법사가 닫힐 때 이 신뢰 당사자 신뢰에 대한 클레임 규칙 편집 대화 상자 열기)를 선택합니다.
 8. 컨텍스트(오른쪽 클릭) 메뉴에서 Relying Party Trusts(신뢰 당사자 신뢰)를 선택합니다.
 9. 신뢰 당사자의 경우 컨텍스트(오른쪽 클릭) 메뉴를 열고 Edit Claim Rules(클레임 규칙 편집)를 선택합니다. 클레임 규칙 편집(Edit Claim Rules) 페이지에서 규칙 추가(Add Rule)를 선택합니다.
 10. Claim rule template(클레임 규칙 템플릿)에서 Transform an Incoming Claim(수신 클레임 변환)을 선택한 후 Edit Rule – NameId(규칙 편집 – NameId) 페이지에서 다음을 수행합니다.
 - 클레임 규칙 이름에 NameId를 입력합니다.
 - Incoming claim name(수신 클레임 이름)으로 Windows Account Name(Windows 계정 이름)을 선택합니다.
 - Outgoing claim name(발신 클레임 이름)으로 Name ID(이름 ID)를 선택합니다.
 - Outgoing name ID format(발신 이름 ID 형식)으로 Persistent Identifier(영구 식별자)를 선택합니다.
 - Pass through all claim values(모든 클레임 값 전달)를 선택합니다.

11 Edit Claim Rules(클레임 규칙 편집) 페이지에서 Add Rule(규칙 추가)를 선택합니다. Select Rule Template(규칙 템플릿 선택) 페이지에서 Claim rule template(클레임 규칙 템플릿)으로 Send LDAP Attributes as Claims(LDAP 속성을 클레임으로 전송)를 선택합니다.

12.규칙 구성 페이지에서 다음 작업을 수행합니다.

- 클레임 규칙 이름에 RoleSessionName을 입력합니다.
- 속성 저장의 경우 Active Directory를 선택합니다.
- LDAP Attribute(LDAP 속성)의 경우 이메일 주소를 선택합니다.
- [발신 클레임 유형(Outgoing Claim Type)]으로 <https://aws.amazon.com/SAML/Attributes/RoleSessionName>을 선택합니다.

13 Edit Claim Rules(클레임 규칙 편집) 페이지에서 Add Rule(규칙 추가)를 선택합니다. 규칙 템플릿 선택(Select Rule Template) 페이지에서 클레임 규칙 템플릿(Claim rule template)으로 사용자 지정 규칙을 사용하여 클레임 전송(Send Claims Using a Custom Rule)을 선택합니다.

14 Edit Rule – Get AD Groups(규칙 편집 – AD 그룹 가져오기) 페이지에서 클레임 규칙 이름에 Get AD Groups(AD 그룹 가져오기)를 입력합니다.

15 Custom rule(사용자 지정 규칙)에 다음을 입력합니다.

```
c:[Type ==
                                "http://schemas.microsoft.com/ws/2008/06/
identity/claims/windowsaccountname",
                                Issuer == "AD AUTHORITY"] => add(store =
    "Active Directory",
                                types = ("http://temp/variable"), query =
    ";tokenGroups;{0}",
                                param = c.Value);
```

16 Edit Claim Rules(클레임 규칙 편집) 페이지에서 Add Rule(규칙 추가)를 선택합니다. Select Rule Template(규칙 템플릿 선택) 페이지에서 Claim rule template(클레임 규칙 템플릿)으로 Send Claims Using a Custom Rule(사용자 지정 규칙을 사용하여 클레임 전송)을 선택합니다.

17 Edit Rule – Roles(규칙 편집 – 역할) 페이지에서 클레임 규칙 이름에 역할을 입력합니다.

18 Custom rule(사용자 지정 규칙)에 다음을 입력합니다.

```
c:[Type == "http://temp/variable", Value =~ "(?i)^AWS-"] =>
    issue(Type = "https://aws.amazon.com/SAML/Attributes/Role", Value =
    RegExReplace(c.Value, "AWS-", "arn:aws:iam::123456789012:saml-provider/
    ADFS,arn:aws:iam::123456789012:role/ADFS-"));
```

SAML 공급자 및 맡을 역할의 ARN을 적어 둡니다. 이 예에서
 arn:aws:iam:123456789012:saml-provider/ADFS는 SAML 공급자의 ARN이고
 arn:aws:iam:123456789012:role/ADFS-는 역할의 ARN입니다.

3. federationmetadata.xml 파일을 다운로드했는지 확인합니다. 문서 내용에 잘못된 문자가 없는지 확인합니다. 이는 AWS와의 신뢰 관계를 구성할 때 사용하는 메타데이터 파일입니다.
4. IAM 콘솔에서 IAM SAML 자격 증명 공급자를 만듭니다. 제공하는 메타데이터 문서는 Azure 엔터프라이즈 애플리케이션을 설정할 때 저장한 연동 메타데이터 XML 파일입니다. 자세한 단계는 IAM User Guide의 [Creating and Managing an IAM Identity Provider \(Console\)](#)를 참조하세요.
5. IAM 콘솔에서 SAML 2.0 연동을 위한 IAM 역할을 만듭니다. 자세한 단계는 IAM User Guide의 [Creating a Role for SAML](#)을 참조하세요.
6. IAM 콘솔에서 SAML 2.0 연동을 위해 생성한 IAM 역할에 연결할 수 있는 IAM 정책을 만듭니다. 자세한 단계는 IAM User Guide의 [Creating IAM Policies \(Console\)](#)를 참조하세요. Azure AD 예제는 [JDBC 또는 ODBC Single Sign-On 인증 설정](#) 단원을 참조하십시오.

2단계: AD FS에 대한 인증을 위해 JDBC 또는 ODBC 설정

JDBC

다음 절차에서는 AD FS에 대한 JDBC 관계를 설정하는 방법을 설명합니다.

- AD FS Single Sign-On을 사용하여 JDBC를 통해 클러스터에 연결하도록 데이터베이스 클라이언트를 구성합니다.

JDBC 드라이버를 사용하는 클라이언트를 사용하여 AD FS Single Sign-On으로 연결하거나 Java와 같은 언어를 사용하여 스크립트를 통해 연결할 수 있습니다. 설치 및 구성 정보는 [Amazon Redshift용 JDBC 드라이버 버전 2.x 연결 구성](#) 단원을 참조하십시오.

예를 들어 SQLWorkbench/J를 클라이언트로 사용할 수 있습니다. SQLWorkbench/j를 구성할 때 데이터베이스의 URL은 다음 형식을 사용합니다.

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

SQLWorkbench/J를 클라이언트로 사용하는 경우 다음 단계를 수행합니다.

- a. SQL Workbench/J를 시작합니다. [연결 프로파일 선택(Select Connection Profile)] 페이지에서 [프로파일 그룹(Profile Group)]을 추가합니다(예: **ADFS**).

- b. Connection Profile(연결 프로파일)에 연결 프로파일 이름을 입력합니다(예: **ADFS**).
- c. Manage Drivers(드라이버 관리)를 선택하고, Amazon Redshift를 선택합니다. [라이브러리(Library)] 옆에 있는 [폴더 열기(Open Folder)] 아이콘을 선택한 다음 적절한 JDBC .jar 파일을 선택합니다.
- d. Select Connection Profile(연결 프로파일 선택) 페이지에서 다음과 같이 연결 프로파일에 정보를 추가합니다.
 - 사용자에게 AD FS 사용자 이름을 입력합니다. 이는 Single Sign-On에 사용 중인 계정의 사용자 이름으로, 인증하려는 클러스터에 대한 권한이 있습니다.
 - 암호에 AD FS 암호를 입력합니다.
 - 드라이버(Drivers)에서 Amazon Redshift(`com.amazon.redshift.jdbc.Driver`)를 선택합니다.
 - URL에 **`jdbc:redshift:iam://your-cluster-identifier:your-cluster-region/your-database-name`**을 입력합니다.
- e. [확장 속성(Extended Properties)]을 선택합니다. `plugin_name`에 **`com.amazon.redshift.plugin.AdfsCredentialsProvider`**를 입력합니다. 이 값은 드라이버에 AD FS Single Sign-On을 인증 방법으로 사용하도록 지정합니다.

ODBC

AD FS에 대한 인증을 위해 ODBC를 설정하려면

- AD FS Single Sign-On을 사용하여 ODBC를 통해 클러스터에 연결하도록 데이터베이스 클라이언트를 구성합니다.

Amazon Redshift는 Linux, Windows 및 macOS 운영 체제용 ODBC 드라이버를 제공합니다. ODBC 드라이버를 설치하려면 먼저 SQL 클라이언트 도구가 32비트인지, 64비트인지 확인합니다. SQL 클라이언트 도구의 요구 사항에 맞는 ODBC 드라이버를 설치합니다.

Windows의 경우 Amazon Redshift ODBC 드라이버 DSN 설정(Amazon Redshift ODBC Driver DSN Setup) 페이지의 연결 설정(Connection Settings) 아래에 다음 정보를 입력합니다.

- 데이터 원본 이름(Data Source Name)에 **`your-DSN`**을 입력합니다. 이는 ODBC 프로파일 이름으로 사용되는 데이터 원본 이름을 지정합니다.

- [인증 유형(Auth type)]에서 [자격 증명 공급자: SAML(Identity Provider: SAML)]을 선택합니다. 이는 ODBC 드라이버가 AD FS Single Sign-On을 사용하여 인증할 때 사용하는 인증 방법입니다.
- 클러스터 ID(Cluster ID)에 ***your-cluster-identifier***를 입력합니다.
- 리전(Region)에 ***your-cluster-region***을 입력합니다.
- 데이터베이스(Database)에 ***your-database-name***을 입력합니다.
- 사용자(User)에 ***your-adfs-username***을 입력합니다. 이는 Single Sign-On에 사용 중인 AD FS 계정의 사용자 이름으로, 인증하려는 클러스터에 대한 권한이 있습니다. 인증 유형(Auth type)이 자격 증명 공급자: SAML(Identity Provider: SAML)인 경우에만 이를 사용합니다.
- Password(암호)에 ***your-adfs-password***를 입력합니다. Auth type(인증 유형)이 Identity Provider: SAML(자격 증명 공급자: SAML)인 경우에만 이를 사용합니다.

macOS 및 Linux에서는 다음과 같이 `odbc.ini` 파일을 편집합니다.

 Note

모든 항목은 대/소문자를 구분하지 않습니다.

- clusterid에 ***your-cluster-identifier***를 입력합니다. 이는 생성된 Amazon Redshift 클러스터의 이름입니다.
- 리전(region)에 ***your-cluster-region***을 입력합니다. 이는 생성된 Amazon Redshift 클러스터의 AWS 리전입니다.
- 데이터베이스(database)에 ***your-database-name***을 입력합니다. 이는 Amazon Redshift 클러스터에서 액세스하려는 데이터베이스의 이름입니다.
- 로캘(locale)에 ***en-us***를 입력합니다. 이는 오류 메시지가 표시되는 언어입니다.
- iam에 ***1***을 입력합니다. 이 값은 드라이버에 IAM 자격 증명을 사용하여 인증하도록 지정합니다.
- plugin_name의 경우 다음 중 하나를 수행합니다.
 - MFA를 사용하는 Azure AD Single Sign-On 구성에 ***BrowserSAML***을 입력합니다. 이는 ODBC 드라이버가 AD FS에 인증할 때 사용하는 인증 방법입니다.
 - Azure AD Single Sign-On 구성에 ***ADFS***를 입력합니다. 이는 ODBC 드라이버가 Azure AD Single Sign-On을 사용하여 인증할 때 사용하는 인증 방법입니다.

- uid에 ***your-adfs-username***을 입력합니다. 이는 Single Sign-On에 사용 중인 Microsoft Azure 계정의 사용자 이름으로, 인증하려는 클러스터에 대한 권한이 있습니다. plugin_name이 ADFS인 경우에만 이를 사용합니다.
- pwd에 ***your-adfs-password***를 입력합니다. plugin_name이 ADFS인 경우에만 이를 사용합니다.

macOS 및 Linux에서는 프로파일 설정을 편집하여 다음 내보내기도 추가합니다.

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

Azure

Microsoft Azure AD를 IdP(자격 증명 공급자)로 사용하여 Amazon Redshift 클러스터에 액세스할 수 있습니다. 이 자습서에서는 Azure를 ID 제공업체(idP)로 사용하여 Amazon Redshift 클러스터에 액세스하는 방법을 보여 줍니다.

[Federating Amazon Redshift access with Microsoft Azure AD single sign-on](#) 동영상을 시청하고 Amazon Redshift 액세스를 Microsoft Azure AD Single Sign-On과 페더레이션하는 방법에 대해 알아봅니다.

1단계: Azure와 AWS 계정을 서로 신뢰하도록 설정

다음 절차에서는 신뢰 관계를 설정하는 방법을 설명합니다.

Azure AD와 AWS 계정을 서로 신뢰하도록 설정하려면

1. Azure AD 사용자가 연결할 기존 Amazon Redshift 클러스터를 사용하거나 생성합니다. 연결을 구성하려면 이 클러스터의 특정 속성(예: 클러스터 식별자)이 필요합니다. 자세한 내용은 [클러스터 생성](#) 섹션을 참조하세요.
2. Microsoft Azure 포털에서 AWS에 사용되는 Azure Active Directory, 그룹, 사용자를 설정합니다.
3. Microsoft Azure 포털에서 Amazon Redshift를 AWS 콘솔에 대한 통합 인증 및 Amazon Redshift에 대한 페더레이션 로그인에 사용할 엔터프라이즈 애플리케이션으로 추가합니다. Enterprise application(엔터프라이즈 애플리케이션)을 선택합니다.
4. +New application(새 애플리케이션 추가)을 선택합니다. 애플리케이션 추가 페이지가 나타납니다.

5. 검색 필드에서 **AWS**를 검색합니다.
6. [Amazon Web Services(AWS)]와 [추가(Add)]를 차례로 선택합니다. 그러면 AWS 애플리케이션이 생성됩니다.
7. 관리에서 Single sign-on을 선택합니다.
8. SAML을 선택합니다. Amazon Web Services(AWS) | SAML -based Sign-on(SAML 기반 로그인) 페이지가 나타납니다.
9. 예를 선택하여 Set up Single Sign-On with SAML(SAML을 사용하여 Single Sign-On 설정) 페이지로 이동합니다. 이 페이지에는 미리 구성된 Single Sign-On 관련 속성 목록이 나와 있습니다.
10. Basic SAML Configuration(기본 SAML 구성)에서 편집 아이콘을 선택하고 저장을 선택합니다.
11. 둘 이상의 애플리케이션에 대해 구성하는 경우 식별자 값을 제공합니다. 예를 들면 **<https://signin.aws.amazon.com/saml#2>**를 입력합니다. 두 번째 애플리케이션부터 이 형식을 # 기호와 함께 사용하여 고유한 SPN 값을 지정합니다.
12. User Attributes and Claims(사용자 속성 및 클레임) 섹션에서 편집 아이콘을 선택합니다.

기본적으로 고유한 사용자 식별자(UID), 역할, RoleSessionName 및 SessionDuration 클레임은 미리 구성되어 있습니다.

13. + Add new claim(새 클레임 추가)을 선택하여 데이터베이스 사용자에게 대한 클레임을 추가합니다.

이름(Name)에 **DbUser**를 입력합니다.

네임스페이스(Namespace)에 **<https://redshift.amazon.com/SAML/Attributes>**를 입력합니다.

소스(Source)에서 속성(Attribute)를 선택합니다.

소스 속성에서 user.userprincipalname을 선택합니다. 그런 다음 저장(Save)을 선택합니다.

14. + Add new claim(새 클레임 추가)을 선택하여 AutoCreate에 대한 클레임을 추가합니다.

Name(이름)에 **AutoCreate**를 입력합니다.

네임스페이스(Namespace)에 **<https://redshift.amazon.com/SAML/Attributes>**를 입력합니다.

Source(소스)에서 Attribute(속성)를 선택합니다.

소스 속성에서 "true"를 선택합니다. 그런 다음 Save(저장)을 선택합니다.

여기에서 **123456789012**는 AWS 계정, **AzureSSO**는 생성한 IAM 역할, **AzureADProvider**는 IAM 공급자입니다.

클레임 이름	값
고유한 사용자 식별자(이름 ID)	user:userprincipalname
https://aws.amazon.com/SAML/Attributes/SessionDuration	"900"
https://aws.amazon.com/SAML/Attributes/Role	arn:aws:iam:: 123456789012 :role/ AzureSSO ,arn:aws:iam:: 123456789012 :saml-provider/ AzureADProvider
https://aws.amazon.com/SAML/Attributes/RoleSessionName	user:userprincipalname
https://redshift.amazon.com/SAML/Attributes/AutoCreate	"true"
https://redshift.amazon.com/SAML/Attributes/DbGroups	user:assignedroles
https://redshift.amazon.com/SAML/Attributes/DbUser	user:userprincipalname

15. 앱 등록 > **your-application-name** > 인증에서 모바일 및 데스크톱 애플리케이션을 추가합니다. URL을 http://localhost/redshift/로 지정합니다.
16. SAML Signing Certificate(SAML 서명 인증서) 섹션에서 다운로드를 선택하여 IAM SAML 자격 증명 공급자를 만들 때 사용할 연동 메타데이터 XML 파일을 다운로드하고 저장합니다. 이 파일은 Single Sign-On 페더레이션 자격 증명을 생성하는 데 사용됩니다.
17. IAM 콘솔에서 IAM SAML 자격 증명 공급자를 만듭니다. 제공하는 메타데이터 문서는 Azure 엔터프라이즈 애플리케이션을 설정할 때 저장한 연동 메타데이터 XML 파일입니다. 자세한 단계는 IAM User Guide의 [Creating and Managing an IAM Identity Provider \(Console\)](#)를 참조하세요.
18. IAM 콘솔에서 SAML 2.0 연동을 위한 IAM 역할을 만듭니다. 자세한 단계는 IAM User Guide의 [Creating a Role for SAML](#)을 참조하세요.

19. IAM 콘솔에서 SAML 2.0 연동을 위해 생성한 IAM 역할에 연결할 수 있는 IAM 정책을 만듭니다. 자세한 단계는 IAM User Guide의 [Creating IAM Policies \(Console\)](#)를 참조하세요.

환경에 맞게 다음 정책(JSON 형식)을 수정합니다.

- *us-west-1*을 해당 클러스터의 AWS 리전으로 대체합니다.
- *123456789012*를 AWS 계정으로 대체합니다.
- *cluster-identifier*를 클러스터 식별자(또는 모든 클러스터의 경우 *)로 대체합니다.
- *dev*를 데이터베이스(또는 모든 클러스터의 경우 *)로 대체합니다.
- 을 해당 IAM 역할의 고유 식별자로 대체합니다..*AROAJ2UCCR6DPCEXAMPLE*
- *example.com*은 해당 테넌트 또는 회사 이메일 도메인으로 대체합니다.
- 은 사용자를 할당할 데이터베이스 그룹으로 대체합니다..*my_dbgroup*

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "redshift:GetClusterCredentials",
      "Resource": [
        "arn:aws:redshift:us-west-1:123456789012:dbname:cluster-identifier/dev",
        "arn:aws:redshift:us-west-1:123456789012:dbuser:cluster-identifier/${redshift:DbUser}",
        "arn:aws:redshift:us-west-1:123456789012:cluster:cluster-identifier"
      ],
      "Condition": {
        "StringEquals": {
          "aws:userid": "AROAJ2UCCR6DPCEXAMPLE:${redshift:DbUser}@example.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "redshift:CreateClusterUser",
      "Resource": "arn:aws:redshift:us-west-1:123456789012:dbuser:cluster-identifier/${redshift:DbUser}"
    }
  ]
}
```

```

    {
      "Effect": "Allow",
      "Action": "redshift:JoinGroup",
      "Resource": "arn:aws:redshift:us-west-1:123456789012:dbgroup:cluster-
identifier/my_dbgroup"
    },
    {
      "Effect": "Allow",
      "Action": [
        "redshift:DescribeClusters",
        "iam:ListRoles"
      ],
      "Resource": "*"
    }
  ]
}

```

이 정책은 다음과 같이 권한을 부여합니다.

- 첫 번째 섹션에서는 지정된 클러스터에 대한 임시 자격 증명을 얻기 위해 GetClusterCredentials API 작업에 권한을 부여합니다. 이 예에서 리소스는 *cluster-identifier*로, 데이터베이스 *dev*가 있으며, 계정 *123456789012*와 AWS 리전 *us-west-1*에 있습니다. `${redshift:DbUser}` 절은 Azure AD에 지정된 DbUser 값과 일치하는 사용자만 연결하도록 허용합니다.
- 조건 절은 특정 사용자만 임시 자격 증명을 받도록 적용합니다. 이는 회사 이메일 도메인의 이메일 주소로 식별되는 IAM 계정에서 역할 고유 ID *AROAJ2UCCR6DPCEXAMPLE*로 지정된 역할을 가진 사용자입니다. 고유 ID에 대한 자세한 내용은 IAM User Guide의 [Unique IDs](#)를 참조하세요.

IdP(이 경우 Azure AD) 설정에 따라 조건 절의 작성 방법이 결정됩니다. 직원의 이메일이 johndoe@example.com인 경우, 먼저 `${redshift:DbUser}`를 해당 직원의 사용자 이름 johndoe와 일치하는 수퍼 필드로 설정합니다. 그런 다음 AWS SAML RoleSessionName 필드를 해당 직원의 이메일 johndoe@example.com과 일치하는 수퍼 필드로 설정하여 이 조건을 작동시킵니다. 이 접근 방식을 이용할 때는 다음 사항을 고려하십시오.

- `${redshift:DbUser}`를 직원의 이메일이 되도록 설정한 경우, @example.com과 일치하도록 JSON 예에서 RoleSessionName을 제거합니다.
- RoleSessionId를 직원의 사용자 이름이 되도록 설정한 경우, @example.com과 일치하도록 예에서 RoleSessionName을 제거합니다.

- JSON 예제에서 `${redshift:DbUser}` 및 `RoleSessionName` 모두 직원의 이메일로 설정됩니다. 이 JSON 예에서는 Amazon Redshift 데이터베이스 사용자 이름과 `@example.com`을 사용해 사용자를 로그인하여 클러스터에 액세스합니다.
- 두 번째 섹션에서는 지정된 클러스터에 `dbuser` 이름을 만들 수 있는 권한을 부여합니다. 이 JSON 예에서는 생성을 `${redshift:DbUser}`로 제한합니다.
- 세 번째 섹션에서는 사용자가 참여할 수 있는 `dbgroup`을 지정할 권한을 부여합니다. 이 JSON 예에서는 사용자가 지정된 클러스터의 `my_dbgroup` 그룹에 참여할 수 있습니다.
- 네 번째 섹션에서는 사용자가 모든 리소스에 대해 수행할 수 있는 작업에 대한 권한을 부여합니다. 이 JSON 예에서는 사용자가 `redshift:DescribeClusters`를 호출하여 클러스터 엔드포인트, AWS 리전 및 포트와 같은 클러스터 정보를 가져올 수 있도록 합니다. 또한 사용자가 `iam:ListRoles`를 호출하여 사용자가 맡을 수 있는 역할을 확인할 수 있도록 합니다.

2단계: Azure에 대한 인증을 위해 JDBC 또는 ODBC 설정

JDBC

Microsoft Azure AD에 대한 인증을 위해 JDBC를 설정하려면

- Azure AD Single Sign-On을 사용하여 JDBC를 통해 클러스터에 연결하도록 데이터베이스 클라이언트를 구성합니다.

JDBC 드라이버를 사용하는 클라이언트를 사용하여 Azure AD Single Sign-On으로 연결하거나 Java와 같은 언어를 사용하여 스크립트를 통해 연결할 수 있습니다. 설치 및 구성 정보는 [Amazon Redshift용 JDBC 드라이버 버전 2.x 연결 구성](#) 단원을 참조하십시오.

예를 들어 SQLWorkbench/J를 클라이언트로 사용할 수 있습니다. SQLWorkbench/J를 구성할 때 데이터베이스의 URL은 다음 형식을 사용합니다.

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

SQLWorkbench/J를 클라이언트로 사용하는 경우 다음 단계를 수행합니다.

- SQL Workbench/J를 시작합니다. [연결 프로파일 선택(Select Connection Profile)] 페이지에서 **AzureAuth**라는 [프로파일 그룹(Profile Group)]을 추가합니다.
- Connection Profile(연결 프로파일)에 **Azure**를 입력합니다.

- c. 드라이버 관리(Manage Drivers)를 선택하고, Amazon Redshift를 선택합니다. [라이브러리(Library)] 옆에 있는 [폴더 열기(Open Folder)] 아이콘을 선택한 다음 적절한 JDBC .jar 파일을 선택합니다.
- d. 연결 프로파일 선택(Select Connection Profile) 페이지에서 다음과 같이 연결 프로파일에 정보를 추가합니다.
 - 사용자에게 Microsoft Azure 사용자 이름을 입력합니다. 이는 Single Sign-On에 사용 중인 Microsoft Azure 계정의 사용자 이름으로, 인증하려는 클러스터에 대한 권한이 있습니다.
 - 암호에 Microsoft Azure 암호를 입력합니다.
 - 드라이버(Drivers)에서 Amazon Redshift(`com.amazon.redshift.jdbc.Driver`)를 선택합니다.
 - URL에 **`jdbc:redshift:iam://your-cluster-identifier:your-cluster-region/your-database-name`**을 입력합니다.
- e. Extended Properties(확장 속성)를 선택하여 다음 설명과 같이 연결 속성에 추가 정보를 추가합니다.

Azure AD Single Sign-On 구성의 경우 다음과 같이 추가 정보를 추가합니다.

- `plugin_name`에 **`com.amazon.redshift.plugin.AzureCredentialsProvider`**를 입력합니다. 이 값은 드라이버에 Azure AD Single Sign-On을 인증 방법으로 사용하도록 지정합니다.
- `idp_tenant`에 **`your-idp-tenant`**를 입력합니다. Microsoft Azure AD에만 사용됩니다. 이는 Azure AD에 구성된 회사의 테넌트 이름입니다. 이 값은 테넌트 이름이거나 하이픈이 있는 테넌트 고유 ID일 수 있습니다.
- `client_secret`에 **`your-azure-redshift-application-client-secret`**를 입력합니다. Microsoft Azure AD에만 사용됩니다. 이는 Azure Single Sign-On 구성을 설정할 때 생성한 Amazon Redshift 애플리케이션의 클라이언트 암호입니다. 이는 `com.amazon.redshift.plugin.AzureCredentialsProvider` 플러그인에만 적용됩니다.
- `client_id`에 **`your-azure-redshift-application-client-id`**를 입력합니다. Microsoft Azure AD에만 사용됩니다. 이는 Azure Single Sign-On 구성을 설정할 때 생성한 Amazon Redshift 애플리케이션의 클라이언트 ID(하이픈 포함)입니다.

MFA를 사용하는 Azure AD Single Sign-On 구성의 경우 다음과 같이 연결 속성에 추가 정보를 추가합니다.

- `plugin_name`에 **`com.amazon.redshift.plugin.BrowserAzureCredentialsProvider`**를 입력합니다. 이는 드라이버에 MFA를 사용하는 Azure AD Single Sign-On을 인증 방법으로 사용하도록 지정합니다.
- `idp_tenant`에 **`your-idp-tenant`**를 입력합니다. Microsoft Azure AD에만 사용됩니다. 이는 Azure AD에 구성된 회사의 테넌트 이름입니다. 이 값은 테넌트 이름이거나 하이픈이 있는 테넌트 고유 ID일 수 있습니다.
- `client_id`에 **`your-azure-redshift-application-client-id`**를 입력합니다. 이 옵션은 Microsoft Azure AD에만 사용됩니다. 이는 MFA를 사용하는 Azure AD Single Sign-On 구성을 설정할 때 생성한 Amazon Redshift 애플리케이션의 클라이언트 ID(하이픈 포함)입니다.
- `listen_port`에 **`your-listen-port`**를 입력합니다. 로컬 서버가 수신 대기하는 포트입니다. 기본값은 7890입니다.
- `idp_response_timeout`에 **`the-number-of-seconds`**를 입력합니다. 이는 IdP 서버가 응답을 다시 보낼 때 제한 시간이 지나기 전까지 대기하는 시간(초)입니다. 최소 시간(초)은 10이어야 합니다. 연결 구성이 이 제한 시간보다 오래 걸리면 연결이 중단됩니다.

ODBC

Microsoft Azure AD에 대한 인증을 위해 ODBC를 설정하려면

- Azure AD Single Sign-On을 사용하여 ODBC를 통해 클러스터에 연결하도록 데이터베이스 클라이언트를 구성합니다.

Amazon Redshift는 Linux, Windows 및 macOS 운영 체제용 ODBC 드라이버를 제공합니다. ODBC 드라이버를 설치하려면 먼저 SQL 클라이언트 도구가 32비트인지, 64비트인지 확인합니다. SQL 클라이언트 도구의 요구 사항에 맞는 ODBC 드라이버를 설치합니다.

Windows의 경우 Amazon Redshift ODBC 드라이버 DSN 설정(Amazon Redshift ODBC Driver DSN Setup) 페이지의 연결 설정(Connection Settings) 아래에 다음 정보를 입력합니다.

- 데이터 원본 이름(Data Source Name)에 **`your-DSN`**를 입력합니다. 이는 ODBC 프로파일 이름으로 사용되는 데이터 원본 이름을 지정합니다.
- Azure AD Single Sign-On 구성을 위한 Auth type(인증 유형)에서 **Identity Provider: Azure AD**를 선택합니다. 이는 ODBC 드라이버가 Azure Single Sign-On을 사용하여 인증할 때 사용하는 인증 방법입니다.

- MFA를 사용하는 Azure AD Single Sign-On 구성을 위한 Auth type(인증 유형)에서 **Identity Provider: Browser Azure AD**를 선택합니다. 이는 ODBC 드라이버가 MFA를 사용하는 Azure Single Sign-On을 사용하여 인증할 때 사용하는 인증 방법입니다.
- 클러스터 ID(Cluster ID)에 ***your-cluster-identifier***를 입력합니다.
- 리전(Region)에 ***your-cluster-region***을 입력합니다.
- 데이터베이스(Database)에 ***your-database-name***을 입력합니다.
- 사용자(User)에 ***your-azure-username***을 입력합니다. 이는 Single Sign-On에 사용 중인 Microsoft Azure 계정의 사용자 이름으로, 인증하려는 클러스터에 대한 권한이 있습니다. 인증 유형(Auth Type)이 자격 증명 공급자: Azure AD(Identity Provider: Azure AD)인 경우에만 이를 사용합니다.
- 암호>Password)에 ***your-azure-password***를 입력합니다. Auth Type(인증 유형)이 Identity Provider: Azure AD(자격 증명 공급자: Azure AD)인 경우에만 이를 사용합니다.
- IdP 테넌트에 ***your-idp-tenant***를 입력합니다. 이는 IdP(Azure)에 구성된 회사의 테넌트 이름입니다. 이 값은 테넌트 이름이거나 하이픈이 있는 테넌트 고유 ID일 수 있습니다.
- Azure 클라이언트 암호에 ***your-azure-redshift-application-client-secret***를 입력합니다. 이는 Azure Single Sign-On 구성을 설정할 때 생성한 Amazon Redshift 애플리케이션의 클라이언트 암호입니다.
- Azure 클라이언트 ID에 ***your-azure-redshift-application-client-id***를 입력합니다. 이는 Azure Single Sign-On 구성을 설정할 때 생성한 Amazon Redshift 애플리케이션의 클라이언트 ID(하이픈 포함)입니다.
- 수신 대기 포트(Listen Port)에 ***your-listen-port***를 입력합니다. 로컬 서버가 수신하는 기본 수신 대기 포트입니다. 기본값은 7890입니다. 이는 브라우저 Azure AD 플러그인에만 적용됩니다.
- 응답 제한 시간(Response Timeout)에 ***the-number-of-seconds***를 입력합니다. 이는 IdP 서버가 응답을 다시 보낼 때 제한 시간이 지나기 전까지 대기하는 시간(초)입니다. 최소 시간(초)은 10이어야 합니다. 연결 구성이 이 제한 시간보다 오래 걸리면 연결이 중단됩니다. 이 옵션은 브라우저 Azure AD 플러그인에만 적용됩니다.

macOS 및 Linux에서는 다음과 같이 `odbc.ini` 파일을 편집합니다.

 Note

모든 항목은 대/소문자를 구분하지 않습니다.

- clusterid에 **your-cluster-identifier**를 입력합니다. 이는 생성된 Amazon Redshift 클러스터의 이름입니다.
- 리전(region)에 **your-cluster-region**을 입력합니다. 이는 생성된 Amazon Redshift 클러스터의 AWS 리전입니다.
- 데이터베이스(database)에 **your-database-name**을 입력합니다. 이는 Amazon Redshift 클러스터에서 액세스하려는 데이터베이스의 이름입니다.
- 로캘(locale)에 **en-us**를 입력합니다. 이는 오류 메시지가 표시되는 언어입니다.
- iam에 **1**을 입력합니다. 이 값은 드라이버에 IAM 자격 증명을 사용하여 인증하도록 지정합니다.
- Azure AD Single Sign-On 구성을 위한 plugin_name에 **AzureAD**를 입력합니다. 이는 드라이버에 Azure Single Sign-On을 인증 방법으로 사용하도록 지정합니다.
- MFA를 사용하는 Azure AD Single Sign-On 구성을 위한 plugin_name에 **BrowserAzureAD**를 입력합니다. 이는 드라이버에 MFA를 사용하는 Azure Single Sign-On을 인증 방법으로 사용하도록 지정합니다.
- uid에 **your-azure-username**을 입력합니다. 이는 Single Sign-On에 사용 중인 Microsoft Azure 계정의 사용자 이름으로, 인증하려는 클러스터에 대한 권한이 있습니다. plugin_name이 AzureAD인 경우에만 이를 사용합니다.
- pwd에 **your-azure-password**를 입력합니다. plugin_name이 AzureAD인 경우에만 이를 사용합니다.
- idp_tenant에 **your-idp-tenant**를 입력합니다. 이는 IdP(Azure)에 구성된 회사의 테넌트 이름입니다. 이 값은 테넌트 이름이거나 하이픈이 있는 테넌트 고유 ID일 수 있습니다.
- client_secret에 **your-azure-redshift-application-client-secret**를 입력합니다. 이는 Azure Single Sign-On 구성을 설정할 때 생성한 Amazon Redshift 애플리케이션의 클라이언트 암호입니다.
- client_id에 **your-azure-redshift-application-client-id**를 입력합니다. 이는 Azure Single Sign-On 구성을 설정할 때 생성한 Amazon Redshift 애플리케이션의 클라이언트 ID(하이픈 포함)입니다.
- listen_port에 **your-listen-port**를 입력합니다. 로컬 서버가 수신 대기하는 포트입니다. 기본값은 7890입니다. 이는 브라우저 Azure AD 플러그인에 적용됩니다.
- idp_response_timeout에 **the-number-of-seconds**를 입력합니다. 이는 Azure로부터 응답을 기다리는 지정된 시간(초)입니다. 이 옵션은 브라우저 Azure AD 플러그인에 적용됩니다.

macOS 및 Linux에서는 프로파일 설정을 편집하여 다음 내보내기도 추가합니다.

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

문제 해결

브라우저 Azure AD 플러그인 문제를 해결하려면 다음을 고려합니다.

- 브라우저 Azure AD 플러그인을 사용하려면 요청에 지정된 회신 URL을 애플리케이션에 대해 구성된 회신 URL과 일치하도록 설정해야 합니다. Microsoft Azure 포털에서 SAML을 사용하여 SSO(Single Sign-On) 설정 페이지로 이동합니다. 그런 다음 회신 URL이 <http://localhost/redshift/>로 설정되어 있는지 확인합니다.
- IdP 테넌트 오류가 발생하면 IdP 테넌트 이름이 Microsoft Azure에서 Active Directory를 설정하는 데 처음 사용한 도메인 이름과 일치하는지 확인합니다.

Windows에서 [Amazon Redshift ODBC DSN 설정(Amazon Redshift ODBC DSN Setup)] 페이지의 [연결 설정(Connection Settings)] 섹션으로 이동합니다. 그런 다음 IdP(Azure)에 구성된 회사의 테넌트 이름이 Microsoft Azure에서 Active Directory를 설정하는 데 처음 사용한 도메인 이름과 일치하는지 확인합니다.

macOS 및 Linux에서 odbc.ini 파일을 찾습니다. 그런 다음 IdP(Azure)에 구성된 회사의 테넌트 이름이 Microsoft Azure에서 Active Directory를 설정하는 데 처음 사용한 도메인 이름과 일치하는지 확인합니다.

- 요청에 지정된 회신 URL이 애플리케이션에 대해 구성된 회신 URL과 일치하지 않는 오류가 발생하는 경우 리디렉션 URI가 회신 URL과 동일한지 확인합니다.

Microsoft Azure 포털에서 애플리케이션의 앱 등록 페이지로 이동합니다. 그런 다음 리디렉션 URI가 응답 URL과 일치하는지 확인합니다.

- 예기치 않은 응답: 무단 오류가 발생하는 경우 모바일 및 데스크톱 애플리케이션 구성을 완료했는지 확인합니다.

Microsoft Azure 포털에서 애플리케이션의 앱 등록 페이지로 이동합니다. 그런 다음 인증으로 이동하여 <http://localhost/redshift/>를 리디렉션 URI로 사용하도록 모바일 및 데스크톱 애플리케이션을 구성했는지 확인합니다.

Ping Identity

Ping Identity를 IdP(자격 증명 공급자)로 사용하여 Amazon Redshift 클러스터에 액세스할 수 있습니다. 이 자습서에서는 Ping Identity를 ID 제공업체(idP)로 사용하여 Amazon Redshift 클러스터에 액세스하는 방법을 보여 줍니다.

1단계: Ping Identity와 AWS 계정을 서로 신뢰하도록 설정

다음 절차에서는 PingOne 포털을 사용하여 신뢰 관계를 설정하는 방법을 설명합니다.

Ping Identity와 AWS 계정을 서로 신뢰하도록 설정하려면

1. Ping Identity 사용자가 연결할 기존 Amazon Redshift 클러스터를 사용하거나 생성합니다. 연결을 구성하려면 이 클러스터의 특정 속성(예: 클러스터 식별자)이 필요합니다. 자세한 내용은 [클러스터 생성](#) 섹션을 참조하세요.
2. Amazon Redshift를 PingOne 포털의 새 SAML 애플리케이션으로 추가합니다. 자세한 단계는 [Ping Identity 설명서](#)를 참조하십시오.
 1. 내 애플리케이션으로 이동합니다.
 2. 애플리케이션 추가에서 New SAML Application(새 SAML 애플리케이션)을 선택합니다.
 3. 애플리케이션 이름에 **Amazon Redshift**를 입력합니다.
 4. Protocol Version(프로토콜 버전)에서 SAML v2.0을 선택합니다.
 5. Category(범주)에서 **your-application-category**를 선택합니다.
 6. Assertion Consumer Service (ACS)에 **your-redshift-local-host-url**를 입력합니다. 이는 SAML 어설션이 리디렉션하는 로컬 호스트 및 포트입니다.
 7. Entity ID(개체 ID)에 urn:amazon:webservices를 입력합니다.
 8. 서명(Signing)에서 어설션 서명(Sign Assertion)을 선택합니다.
 9. SSO Attribute Mapping(SSO 속성 매핑) 섹션에서 다음 표와 같이 클레임을 만듭니다.

애플리케이션 속성	리터럴 값의 자격 증명 브리지 속성
https://aws.amazon.com/SAML/Attributes/Role	arn:aws:iam:: 123456789012 :role/ Ping ,arn:aws:iam:: 123456789012 :saml-provider/ PingProvider
https://aws.amazon.com/SAML/Attributes/RoleSessionName	이메일

애플리케이션 속성	리터럴 값의 자격 증명 브리지 속성
<code>https://redshift.amazonaws.com/SAML/Attributes/AutoCreate</code>	"true"
<code>https://redshift.amazonaws.com/SAML/Attributes/DbUser</code>	이메일
<code>https://redshift.amazonaws.com/SAML/Attributes/DbGroups</code>	"DbGroups" 속성의 그룹에는 @directory 접두사가 포함됩니다. 이를 제거하려면 [자격 증명 브리지(Identity Bridge)]에 [memberOf]를 입력합니다. [함수(Function)]에서 [ExtractByRegularExpression]을 선택합니다. [표현식(Expression)]에서 (.*)[@](?:.*)를 입력합니다.

- Group Access(그룹 액세스)에서 필요한 경우 다음 그룹 액세스를 설정합니다.
 - `https://aws.amazon.com/SAML/Attributes/Role`
 - `https://aws.amazon.com/SAML/Attributes/RoleSessionName`
 - `https://redshift.amazonaws.com/SAML/Attributes/AutoCreate`
 - `https://redshift.amazonaws.com/SAML/Attributes/DbUser`
- 필요한 경우 설정을 검토하고 변경합니다.
- Initiate Single Sign-On (SSO) URL(시작 Single Sign-On(SSO) URL)을 브라우저 SAML 플러그인의 로그인 URL로 사용합니다.
- IAM 콘솔에서 IAM SAML 자격 증명 공급자를 만듭니다. 제공하는 메타데이터 문서는 Ping Identity를 설정할 때 저장한 연동 메타데이터 XML 파일입니다. 자세한 단계는 IAM User Guide의 [Creating and Managing an IAM Identity Provider \(Console\)](#)를 참조하세요.
- IAM 콘솔에서 SAML 2.0 연동을 위한 IAM 역할을 만듭니다. 자세한 단계는 IAM User Guide의 [Creating a Role for SAML](#)을 참조하세요.
- IAM 콘솔에서 SAML 2.0 연동을 위해 생성한 IAM 역할에 연결할 수 있는 IAM 정책을 만듭니다. 자세한 단계는 IAM User Guide의 [Creating IAM Policies \(Console\)](#)를 참조하세요. Azure AD 예제는 [JDBC 또는 ODBC Single Sign-On 인증 설정](#) 단원을 참조하십시오.

2단계: Ping Identity에 대한 인증을 위해 JDBC 또는 ODBC 설정

JDBC

Ping Identity에 대한 인증을 위해 JDBC를 설정하려면

- Ping Identity Single Sign-On을 사용하여 JDBC를 통해 클러스터에 연결하도록 데이터베이스 클라이언트를 구성합니다.

JDBC 드라이버를 사용하는 클라이언트를 사용하여 Ping Identity Single Sign-On으로 연결하거나 Java와 같은 언어를 사용하여 스크립트를 통해 연결할 수 있습니다. 설치 및 구성 정보는 [Amazon Redshift용 JDBC 드라이버 버전 2.x 연결 구성](#) 단원을 참조하십시오.

예를 들어 SQLWorkbench/J를 클라이언트로 사용할 수 있습니다. SQLWorkbench/j를 구성할 때 데이터베이스의 URL은 다음 형식을 사용합니다.

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

SQLWorkbench/J를 클라이언트로 사용하는 경우 다음 단계를 수행합니다.

- a. SQL Workbench/J를 시작합니다. [연결 프로파일 선택(Select Connection Profile)] 페이지에서 [프로파일 그룹(Profile Group)]을 추가합니다(예: **Ping**).
- b. 연결 프로파일(Connection Profile)에 ***your-connection-profile-name***을 입력합니다(예: **Ping**).
- c. Manage Drivers(드라이버 관리)를 선택하고, Amazon Redshift를 선택합니다. [라이브러리(Library)] 옆에 있는 [폴더 열기(Open Folder)] 아이콘을 선택한 다음 적절한 JDBC .jar 파일을 선택합니다.
- d. Select Connection Profile(연결 프로파일 선택) 페이지에서 다음과 같이 연결 프로파일에 정보를 추가합니다.
 - User(사용자)에 PingOne 사용자 이름을 입력합니다. 이는 Single Sign-On에 사용 중인 PingOne 계정의 사용자 이름으로, 인증하려는 클러스터에 대한 권한이 있습니다.
 - Password(암호)에 PingOne 암호를 입력합니다.
 - 드라이버(Drivers)에서 Amazon Redshift(`com.amazon.redshift.jdbc.Driver`)를 선택합니다.
 - URL에 ***jdbc:redshift:iam://your-cluster-identifier:your-cluster-region/your-database-name***을 입력합니다.

- e. 확장 속성(Extended Properties)를 선택하고 다음 중 하나를 수행합니다.
- login_url에 **your-ping-ss0-login-url**을 입력합니다. 이 값은 URL에 Single Sign-On을 로그인하기 위한 인증으로 사용하도록 지정합니다.
 - Ping Identity의 경우 plugin_name에 **com.amazon.redshift.plugin.PingCredentialsProvider**를 입력합니다. 이 값은 드라이버에 Ping Identity Single Sign-On을 인증 방법으로 사용하도록 지정합니다.
 - Single Sign-On을 사용한 Ping Identity의 경우 plugin_name에 **com.amazon.redshift.plugin.BrowserSamlCredentialsProvider**를 입력합니다. 이 값은 드라이버에 Ping Identity PingOne과 함께 Single Sign-On을 인증 방법으로 사용하도록 지정합니다.

ODBC

Ping Identity에 대한 인증을 위해 ODBC를 설정하려면

- Ping Identity PingOne Single Sign-On을 사용하여 ODBC를 통해 클러스터에 연결하도록 데이터베이스 클라이언트를 구성합니다.

Amazon Redshift는 Linux, Windows 및 macOS 운영 체제용 ODBC 드라이버를 제공합니다. ODBC 드라이버를 설치하려면 먼저 SQL 클라이언트 도구가 32비트인지, 64비트인지 확인합니다. SQL 클라이언트 도구의 요구 사항에 맞는 ODBC 드라이버를 설치합니다.

Windows의 경우 Amazon Redshift ODBC 드라이버 DSN 설정(Amazon Redshift ODBC Driver DSN Setup) 페이지의 연결 설정(Connection Settings) 아래에 다음 정보를 입력합니다.

- 데이터 원본 이름(Data Source Name)에 **your-DSN**을 입력합니다. 이는 ODBC 프로파일 이름으로 사용되는 데이터 원본 이름을 지정합니다.
- 인증 유형(Auth type)에서 다음 중 하나를 수행합니다.
 - Ping Identity 구성의 경우 자격 증명 공급자: Ping Federate를 선택합니다. 이는 ODBC 드라이버가 Ping Identity Single Sign-On을 사용하여 인증할 때 사용하는 인증 방법입니다.
 - Single Sign-On 구성을 사용하는 Ping Identity의 경우 자격 증명 공급자: 브라우저 SAML을 선택합니다. 이는 ODBC 드라이버가 Single Sign-On이 포함된 Ping Identity를 사용하여 인증할 때의 인증 방법입니다.
- 클러스터 ID(Cluster ID)에 **your-cluster-identifier**를 입력합니다.
- 리전(Region)에 **your-cluster-region**을 입력합니다.

- 데이터베이스(Database)에 ***your-database-name***을 입력합니다.
- 사용자(User)에 ***your-ping-username***을 입력합니다. 이는 Single Sign-On에 사용 중인 PingOne 계정의 사용자 이름으로, 인증하려는 클러스터에 대한 권한이 있습니다. 인증 유형(Auth type)이 자격 증명 공급자: PingFederate(Identity Provider: PingFederate)인 경우에만 이를 사용합니다.
- 암호>Password)에 ***your-ping-password***를 입력합니다. 인증 유형(Auth type)이 자격 증명 공급자: PingFederate(Identity Provider: PingFederate)인 경우에만 이를 사용합니다.
- 수신 대기 포트(Listen Port)에 ***your-listen-port***를 입력합니다. 로컬 서버가 수신 대기하는 포트입니다. 기본값은 7890입니다. 이는 브라우저 SAML 플러그인에만 적용됩니다.
- 응답 제한 시간(Response Timeout)에 ***the-number-of-seconds***를 입력합니다. 이는 IdP 서버가 응답을 다시 보낼 때 제한 시간이 지나기 전까지 대기하는 시간(초)입니다. 최소 시간(초)은 10이어야 합니다. 연결 구성이 이 제한 시간보다 오래 걸리면 연결이 중단됩니다. 이는 브라우저 SAML 플러그인에만 적용됩니다.
- 로그인 URL에 ***your-login-url***을 입력합니다. 이는 브라우저 SAML 플러그인에만 적용됩니다.

macOS 및 Linux에서는 다음과 같이 `odbc.ini` 파일을 편집합니다.

 Note

모든 항목은 대/소문자를 구분하지 않습니다.

- clusterid에 ***your-cluster-identifier***를 입력합니다. 이는 생성된 Amazon Redshift 클러스터의 이름입니다.
- 리전(region)에 ***your-cluster-region***을 입력합니다. 이는 생성된 Amazon Redshift 클러스터의 AWS 리전입니다.
- 데이터베이스(database)에 ***your-database-name***을 입력합니다. 이는 Amazon Redshift 클러스터에서 액세스하려는 데이터베이스의 이름입니다.
- 로캘(locale)에 ***en-us***를 입력합니다. 이는 오류 메시지가 표시되는 언어입니다.
- iam에 ***1***을 입력합니다. 이 값은 드라이버에 IAM 자격 증명을 사용하여 인증하도록 지정합니다.
- plugin_name의 경우 다음 중 하나를 수행합니다.

- Ping Identity 구성의 경우 **BrowserSAML**을 입력합니다. 이는 ODBC 드라이버가 Ping Identity에 인증할 때 사용하는 인증 방법입니다.
- Single Sign-On을 사용하는 Ping Identity 구성에 **Ping**을 입력합니다. 이는 ODBC 드라이버가 Single Sign-On이 포함된 Ping Identity를 사용하여 인증할 때의 인증 방법입니다.
- uid에 **your-ping-username**을 입력합니다. 이는 Single Sign-On에 사용 중인 Microsoft Azure 계정의 사용자 이름으로, 인증하려는 클러스터에 대한 권한이 있습니다. plugin_name이 Ping인 경우에만 이를 사용합니다.
- pwd에 **your-ping-password**를 입력합니다. plugin_name이 Ping인 경우에만 이를 사용합니다.
- login_url에 **your-login-url**을 입력합니다. 이는 SAML 응답을 반환하는 시작 Single Sign-On URL입니다. 이는 브라우저 SAML 플러그인에만 적용됩니다.
- idp_response_timeout에 **the-number-of-seconds**를 입력합니다. 이는 PingOne Identity로부터 응답을 기다리는 지정된 시간(초)입니다. 이는 브라우저 SAML 플러그인에만 적용됩니다.
- listen_port에 **your-listen-port**를 입력합니다. 로컬 서버가 수신 대기하는 포트입니다. 기본값은 7890입니다. 이는 브라우저 SAML 플러그인에만 적용됩니다.

macOS 및 Linux에서는 프로파일 설정을 편집하여 다음 내보내기도 추가합니다.

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

Okta

Okta를 IdP(자격 증명 공급자)로 사용하여 Amazon Redshift 클러스터에 액세스할 수 있습니다. 이 자습서에서는 Okta를 ID 제공업체(idP)로 사용하여 Amazon Redshift 클러스터에 액세스하는 방법을 보여 줍니다.

1단계: Okta와 AWS 계정을 서로 신뢰하도록 설정

다음 절차에서는 신뢰 관계를 설정하는 방법을 설명합니다.

Okta와 AWS 계정을 서로 신뢰하도록 설정하려면

1. Okta 사용자가 연결할 기존 Amazon Redshift 클러스터를 사용하거나 생성합니다. 연결을 구성하려면 이 클러스터의 특정 속성(예: 클러스터 식별자)이 필요합니다. 자세한 내용은 [클러스터 생성](#) 섹션을 참조하세요.
2. Amazon Redshift를 Okta 포털의 새 애플리케이션으로 추가합니다. 자세한 단계는 [Okta documentation](#)을 참조하십시오.
 - 애플리케이션 추가를 선택합니다.
 - 애플리케이션 추가에서 Create New App(새 앱 생성)을 선택합니다.
 - Create a New Add Application Integration(새 애플리케이션 통합 추가 생성) 페이지의 플랫폼에서 웹을 선택합니다.
 - Sign on method(로그인 방법)에서 SAML v2.0을 선택합니다.
 - 일반 설정 페이지에서 앱 이름에 ***your-redshift-saml-sso-name***을 입력합니다. 이는 애플리케이션 이름입니다.
 - SAML 설정 페이지에서 SSO(Single Sign-On) URL에 ***your-redshift-local-host-url***을 입력합니다. 이는 SAML 어설션이 리디렉션하는 로컬 호스트 및 포트입니다(예: http://localhost:7890/redshift/).
3. [통합 인증 URL(Single sign on URL)] 값을 [수신자 URL(Recipient URL)] 및 [대상 URL(Destination URL)]로 사용합니다.
4. Signing(서명)에서 Sign Assertion(어설션 서명)을 선택합니다.
5. [대상 URI(SP 엔터티 ID)(Audience URI (SP Entity ID))]에 다음 표와 같이 클레임의 **urn:amazon:webservices**를 입력합니다.
6. 고급 설정 섹션에서 SAML 발급자 ID에 ***your-Identity-Provider-Issuer-ID***를 입력합니다. 이는 설정 지침 보기 섹션에서 찾을 수 있습니다.
7. Attribute Statements(속성 설명) 섹션에서 다음 표와 같이 클레임을 만듭니다.

클레임 이름	값
https://aws.amazon.com/SAML/Attributes/Role	arn:aws:iam:: <i>123456789012</i> :role/ <i>Okta</i> ,arn:aws:iam:: <i>123456789012</i> :saml-provider/ <i>Okta</i>
https://aws.amazon.com/SAML/Attributes/RoleSessionName	user.email

클레임 이름	값
<code>https://redshift.amazonaws.com/SAML/Attributes/AutoCreate</code>	<code>"true"</code>
<code>https://redshift.amazonaws.com/SAML/Attributes/DbUser</code>	<code>user.email</code>

- App Embed Link(앱 포함 링크) 섹션에서 브라우저 SAML 플러그인의 로그인 URL로 사용할 수 있는 URL을 찾습니다.
- IAM 콘솔에서 IAM SAML 자격 증명 공급자를 만듭니다. 제공하는 메타데이터 문서는 Okta를 설정할 때 저장한 연동 메타데이터 XML 파일입니다. 자세한 단계는 IAM User Guide의 [Creating and Managing an IAM Identity Provider \(Console\)](#)를 참조하세요.
- IAM 콘솔에서 SAML 2.0 연동을 위한 IAM 역할을 만듭니다. 자세한 단계는 IAM User Guide의 [Creating a Role for SAML](#)을 참조하세요.
- IAM 콘솔에서 SAML 2.0 연동을 위해 생성한 IAM 역할에 연결할 수 있는 IAM 정책을 만듭니다. 자세한 단계는 IAM User Guide의 [Creating IAM Policies \(Console\)](#)를 참조하세요. Azure AD 예제는 [JDBC 또는 ODBC Single Sign-On 인증 설정](#) 단원을 참조하십시오.

2단계: Okta에 대한 인증을 위해 JDBC 또는 ODBC 설정

JDBC

Okta에 대한 인증을 위해 JDBC를 설정하려면

- Okta Single Sign-On을 사용하여 JDBC를 통해 클러스터에 연결하도록 데이터베이스 클라이언트를 구성합니다.

JDBC 드라이버를 사용하는 클라이언트를 사용하여 Okta Single Sign-On으로 연결하거나 Java와 같은 언어를 사용하여 스크립트를 통해 연결할 수 있습니다. 설치 및 구성 정보는 [Amazon Redshift용 JDBC 드라이버 버전 2.x 연결 구성](#) 단원을 참조하십시오.

예를 들어 SQLWorkbench/J를 클라이언트로 사용할 수 있습니다. SQLWorkbench/J를 구성할 때 데이터베이스의 URL은 다음 형식을 사용합니다.

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

SQLWorkbench/J를 클라이언트로 사용하는 경우 다음 단계를 수행합니다.

- a. SQL Workbench/J를 시작합니다. [연결 프로파일 선택(Select Connection Profile)] 페이지에서 [프로파일 그룹(Profile Group)]을 추가합니다(예: **Okta**).
- b. 연결 프로파일(Connection Profile)에 ***your-connection-profile-name***을 입력합니다(예: **Okta**).
- c. Manage Drivers(드라이버 관리)를 선택하고, Amazon Redshift를 선택합니다. [라이브러리(Library)] 옆에 있는 [폴더 열기(Open Folder)] 아이콘을 선택한 다음 적절한 JDBC .jar 파일을 선택합니다.
- d. 연결 프로파일 선택(Select Connection Profile) 페이지에서 다음과 같이 연결 프로파일에 정보를 추가합니다.
 - 사용자에게 Okta 사용자 이름을 입력합니다. 이는 Single Sign-On에 사용 중인 Okta 계정의 사용자 이름으로, 인증하려는 클러스터에 대한 권한이 있습니다.
 - 암호에 Okta 암호를 입력합니다.
 - 드라이버(Drivers)에서 Amazon Redshift(`com.amazon.redshift.jdbc.Driver`)를 선택합니다.
 - URL에 ***jdbc:redshift:iam://your-cluster-identifier:your-cluster-region/your-database-name***을 입력합니다.
- e. 확장 속성(Extended Properties)를 선택하고 다음 중 하나를 수행합니다.
 - login_url에 ***your-okta-sso-login-url***을 입력합니다. 이 값은 URL에 Single Sign-On을 Okta에 로그인하기 위한 인증으로 사용하도록 지정합니다.
 - Okta Single Sign-On에서 plugin_name에 ***com.amazon.redshift.plugin.OktaCredentialsProvider***를 입력합니다. 이 값은 드라이버에 Okta Single Sign-On을 인증 방법으로 사용하도록 지정합니다.
 - MFA를 사용하는 Okta Single Sign-On에서 plugin_name에 ***com.amazon.redshift.plugin.BrowserSamlCredentialsProvider***를 입력합니다. 이는 드라이버에 MFA를 사용하는 Okta Single Sign-On을 인증 방법으로 사용하도록 지정합니다.

ODBC

Okta에 대한 인증을 위해 ODBC를 설정하려면

- Okta Single Sign-On을 사용하여 ODBC를 통해 클러스터에 연결하도록 데이터베이스 클라이언트를 구성합니다.

Amazon Redshift는 Linux, Windows 및 macOS 운영 체제용 ODBC 드라이버를 제공합니다. ODBC 드라이버를 설치하려면 먼저 SQL 클라이언트 도구가 32비트인지, 64비트인지 확인합니다. SQL 클라이언트 도구의 요구 사항에 맞는 ODBC 드라이버를 설치합니다.

Windows의 경우 Amazon Redshift ODBC 드라이버 DSN 설정(Amazon Redshift ODBC Driver DSN Setup) 페이지의 연결 설정(Connection Settings) 아래에 다음 정보를 입력합니다.

- 데이터 원본 이름(Data Source Name)에 ***your-DSN***을 입력합니다. 이는 ODBC 프로파일 이름으로 사용되는 데이터 원본 이름을 지정합니다.
- 인증 유형(Auth type)에서 다음 중 하나를 수행합니다.
 - Okta Single Sign-On 구성에서 **Identity Provider: Okta**를 선택합니다. 이는 ODBC 드라이버가 Okta Single Sign-On을 사용하여 인증할 때 사용하는 인증 방법입니다.
 - MFA를 사용하는 Okta Single Sign-On 구성에서 **Identity Provider: Browser SAML**을 선택합니다. 이는 ODBC 드라이버가 MFA를 사용하는 Okta Single Sign-On을 사용하여 인증할 때 사용하는 인증 방법입니다.
- 클러스터 ID(Cluster ID)에 ***your-cluster-identifier***를 입력합니다.
- 리전(Region)에 ***your-cluster-region***을 입력합니다.
- 데이터베이스(Database)에 ***your-database-name***을 입력합니다.
- 사용자(User)에 ***your-okta-username***을 입력합니다. 이는 Single Sign-On에 사용 중인 Okta 계정의 사용자 이름으로, 인증하려는 클러스터에 대한 권한이 있습니다. Auth type(인증 유형)이 Identity Provider: Okta(자격 증명 공급자: Okta)인 경우에만 이를 사용합니다.
- 암호>Password)에 ***your-okta-password***를 입력합니다. 인증 유형(Auth type)이 자격 증명 공급자: Okta(Identity Provider: Okta)인 경우에만 이를 사용합니다.

macOS 및 Linux에서는 다음과 같이 `odbc.ini` 파일을 편집합니다.

 Note

모든 항목은 대/소문자를 구분하지 않습니다.

- clusterid에 ***your-cluster-identifier***를 입력합니다. 이는 생성된 Amazon Redshift 클러스터의 이름입니다.
- 리전(region)에 ***your-cluster-region***을 입력합니다. 이는 생성된 Amazon Redshift 클러스터의 AWS 리전입니다.

- 데이터베이스(database)에 **your-database-name**를 입력합니다. 이는 Amazon Redshift 클러스터에서 액세스하려는 데이터베이스의 이름입니다.
- 로컬(locale)에 **en-us**를 입력합니다. 이는 오류 메시지가 표시되는 언어입니다.
- iam에 **1**을 입력합니다. 이 값은 드라이버에 IAM 자격 증명을 사용하여 인증하도록 지정합니다.
- plugin_name의 경우 다음 중 하나를 수행합니다.
 - MFA를 사용하는 Okta Single Sign-On 구성에 **BrowserSAML**을 입력합니다. 이는 ODBC 드라이버가 MFA가 포함된 Okta Single Sign-On을 사용하여 인증할 때의 인증 방법입니다.
 - Okta Single Sign-On 구성에 **Okta**를 입력합니다. 이는 ODBC 드라이버가 Okta Single Sign-On을 사용하여 인증할 때 사용하는 인증 방법입니다.
- uid에 **your-okta-username**를 입력합니다. 이는 Single Sign-On에 사용 중인 Okta 계정의 사용자 이름으로, 인증하려는 클러스터에 대한 권한이 있습니다. plugin_name이 Okta인 경우에만 이를 사용합니다.
- pwd에 **your-okta-password**를 입력합니다. plugin_name이 Okta인 경우에만 이를 사용합니다.
- login_url에 **your-login-url**을 입력합니다. 이는 SAML 응답을 반환하는 시작 Single Sign-On URL입니다. 이는 브라우저 SAML 플러그인에만 적용됩니다.
- idp_response_timeout에 **the-number-of-seconds**를 입력합니다. 이는 PingOne으로부터 응답을 기다리는 지정된 시간(초)입니다. 이는 브라우저 SAML 플러그인에만 적용됩니다.
- listen_port에 **your-listen-port**를 입력합니다. 로컬 서버가 수신 대기하는 포트입니다. 기본값은 7890입니다. 이는 브라우저 SAML 플러그인에만 적용됩니다.

macOS 및 Linux에서는 프로파일 설정을 편집하여 다음 내보내기도 추가합니다.

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

Amazon Redshift가 사용자를 대신하여 AWS 서비스에 액세스할 수 있도록 권한 부여

일부 Amazon Redshift 기능을 사용하려면 Amazon Redshift가 사용자 대신 다른 AWS 서비스에 액세스해야 합니다. 예를 들어 [COPY](#) 및 [UNLOAD](#) 명령은 Amazon S3 버킷을 사용하여 Amazon Redshift 클러스터로 데이터를 로드하거나 언로드할 수 있습니다. [CREATE EXTERNAL FUNCTION](#) 명령은 스칼라 Lambda 사용자 정의 함수(UDF)를 사용하여 AWS Lambda 함수를 호출할 수 있습니다. Amazon Redshift Spectrum은 Amazon Athena 또는 AWS Glue의 데이터 카탈로그를 사용할 수 있습니다. Amazon Redshift 클러스터가 사용자를 대신하여 작동하도록 하려면 클러스터에 보안 자격 증명을 제공해야 합니다. 보안 자격 증명을 제공하는 데 선호되는 방법은 AWS Identity and Access Management(IAM) 역할을 지정하는 것입니다. COPY 및 UNLOAD의 경우 임시 자격 증명을 제공할 수 있습니다.

사용자가 AWS Management Console 외부에서 AWS와 상호 작용하려면 프로그래밍 방식의 액세스 권한이 필요합니다. 프로그래밍 방식의 액세스 권한을 부여하는 방법은 AWS에 액세스하는 사용자 유형에 따라 다릅니다.

사용자에게 프로그래밍 방식의 액세스 권한을 부여하려면 다음 옵션 중 하나를 선택합니다.

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
IAM	(권장됨) 콘솔 자격 증명을 임시 자격 증명으로 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. - AWS CLI의 경우 AWS Command Line Interface 사용 설명서의 AWS 로컬 개발을 위한 로그인을 참조하세요. - AWS SDK의 경우 AWS SDK 및 도구 참조 안내서의 AWS 로컬 개발을 위한 로그인을 참조하세요.
작업 인력 ID	임시 자격 증명을 사용하여 AWS CLI, AWS SDK 또는	사용하고자 하는 인터페이스에 대한 지침을 따릅니다.

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
(IAM Identity Center에서 관리되는 사용자)	AWS API에 대한 프로그래밍 요청에 서명합니다.	- AWS CLI에 대해서는 AWS Command Line Interface 사용자 안내서에서 AWS IAM Identity Center을 사용하도록 AWS CLI 구성을 참조하세요. - AWS SDK, 도구, AWS API에 대해서는 AWS SDK 및 도구 참조 가이드에서 IAM Identity Center 인증을 참조하세요.
IAM	임시 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	IAM 사용자 설명서의 AWS 리소스와 함께 임시 자격 증명 사용 에 나와 있는 지침을 따르세요.
IAM	(권장되지 않음) 장기 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. - AWS CLI에 대해서는 AWS Command Line Interface 사용자 안내서에서 IAM 사용자 자격 증명을 사용한 인증을 참조하세요. - AWS SDK와 도구는 AWS SDK 및 도구 참조 가이드에서 장기 자격 증명을 사용한 인증을 참조하세요. - AWS API는 IAM 사용자 설명서에서 IAM 사용자의 액세스 키 관리를 참조하세요.

다음은 통해 다른 AWS 서비스에 액세스하는 데 필요한 권한과 함께 IAM 역할을 생성하는 방법을 알아봅니다. Amazon Redshift 명령을 실행할 때 역할을 클러스터와 연관시키고 역할의 Amazon 리소스 이름(ARN)을 지정해야 합니다. 자세한 내용은 [IAM 역할을 사용하여 COPY, UNLOAD, CREATE EXTERNAL FUNCTION 및 CREATE EXTERNAL SCHEMA 작업에 대한 권한 부여](#) 섹션을 참조하세요.

또한 슈퍼 사용자는 특정 사용자 및 그룹에 ASSUMEROLE 권한을 부여하여 COPY 및 UNLOAD 작업을 위한 역할에 대한 액세스 권한을 제공할 수 있습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [GRANT](#)를 참조하세요.

Amazon Redshift 클러스터가 AWS 서비스에 액세스 할 수 있도록 IAM 역할 생성

Amazon Redshift 클러스터가 AWS 서비스에 액세스 할 수 있도록 IAM 역할 생성

Amazon Redshift 클러스터에서 사용자를 대신하여 다른 AWS 서비스와 통신하도록 허용하는 IAM 역할을 생성하려면 다음 단계를 수행합니다. 이 섹션에 사용된 값은 예시이며 필요에 따라 값을 선택할 수 있습니다.

Amazon Redshift에서 AWS 서비스에 액세스하도록 허용하는 IAM 역할을 생성하려면

1. [IAM 콘솔\(IAM console\)](#)을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 역할 생성(Create role)을 선택합니다.
4. AWS 서비스를 선택한 다음 Redshift를 선택합니다.
5. 사용 사례 선택 아래에서, Redshift - 사용자 지정 가능(Redshift - Customizable)을 선택한 후 다음: 권한(Next: Permissions)을 선택합니다. 첨부 파일 권한 정책(Attach permissions policy) 페이지가 나타납니다.
6. 예를 들어 COPY를 사용하여 Amazon S3에 액세스하려면 **AmazonS3ReadOnlyAccess**를 사용하고 추가할 수 있습니다. COPY 또는 UNLOAD를 사용하여 Amazon S3에 액세스하려면 원하는 버킷 및 접두사에 대한 액세스를 제한하는 관리형 정책을 생성하는 것이 좋습니다. 읽기 및 쓰기 작업 모두에 대해 최소 권한을 적용하고 Amazon Redshift에 필요한 Amazon S3 버킷 및 키 접두사로만 제한하는 것이 좋습니다.

CREATE EXTERNAL FUNCTION 명령에 대한 Lambda 함수 호출에 액세스하려면 **AWSLambdaRole**을 추가합니다.

Redshift Spectrum의 경우 Amazon S3 액세스 외에 **AWSGlueConsoleFullAccess** 또는 **AmazonAthenaFullAccess**를 추가합니다.

다음: 태그를 선택합니다.

7. 태그 추가(Add tags) 페이지가 나타납니다. 선택적으로 태그를 추가할 수 있습니다. 다음: 검토를 선택합니다.
8. Role name(역할 이름)에는 역할 이름을 입력합니다(예: **RedshiftCopyUnload**). 역할 생성(Create role)을 선택합니다.
9. 새로운 역할은 해당 역할을 사용하는 클러스터의 사용자라면 모두에게 제공됩니다. 특정 클러스터의 일부 사용자로 액세스를 제한하거나, 혹은 특정 리전의 클러스터로 제한하려면 역할에 대한 신뢰 관계를 편집하세요. 자세한 내용은 [IAM 역할에 대한 액세스 제한](#) 섹션을 참조하세요.
10. 역할을 클러스터와 연결합니다. IAM 역할은 새롭게 생성한 클러스터와 연결할 수도 있고, 혹은 기존 클러스터에 추가할 수도 있습니다. 자세한 내용은 [IAM 역할을 클러스터와 연결](#) 섹션을 참조하세요.

Note

특정 데이터에 대한 액세스를 제한하려면 필요한 최소 권한을 부여하는 IAM 역할을 사용합니다.

IAM 역할에 대한 액세스 제한

기본적으로 Amazon Redshift 클러스터에서 사용할 수 있는 IAM 역할은 해당 클러스터의 모든 사용자가 사용할 수 있습니다. IAM 역할을 특정 클러스터 또는 특정 영역의 특정 Amazon Redshift 데이터베이스 사용자로 제한하도록 선택할 수 있습니다.

특정 데이터베이스 사용자에게 한해 IAM 역할을 사용할 수 있도록 하려면 다음 단계를 따르세요.

특정 데이터베이스 사용자로 IAM 역할에 대한 액세스 권한을 제한하려면

1. Amazon Redshift 클러스터의 데이터베이스 사용자에게 대한 Amazon 리소스 이름(ARN)을 확인합니다. 데이터베이스 사용자의 ARN 형식은 `arn:aws:redshift:region:account-id:dbuser:cluster-name/user-name`입니다.

Amazon Redshift Serverless의 경우 다음 ARN 형식을 사용합니다.

`arn:aws:redshift:region:account-id:dbuser:serverless-account-id-workgroup-id/user-name`

2. [IAM 콘솔\(IAM console\)](#)을 엽니다.
3. 탐색 창에서 역할을 선택합니다.

4. 특정 Amazon Redshift 데이터베이스 사용자로 제한하려는 IAM 역할을 선택합니다.
5. 신뢰 관계(Trust Relationships) 탭을 선택한 다음 신뢰 관계 편집(Edit Trust Relationship)을 선택합니다. Amazon Redshift가 귀하를 대신하여 다른 AWS 서비스에 액세스 할 수 있게 해주는 새로운 IAM 역할은 다음과 같이 신뢰 관계를 갖습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. 신뢰 관계의 sts:AssumeRole 작업 영역에 sts:ExternalId 필드를 지정하는 값으로 제한하는 조건을 추가합니다. 역할에 대한 액세스 권한을 부여할 각 데이터베이스 사용자의 ARN을 추가합니다. 외부 ID는 고유한 문자열일 수 있습니다.

예를 들어 다음 신뢰 관계는 user1 리전의 user2 클러스터에 속한 데이터베이스 사용자인 my-cluster와 us-west-2에게만 IAM 역할을 사용할 권한을 부여하고 있습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {

```

```

    "sts:ExternalId": [
      "arn:aws:redshift:us-west-2:123456789012:dbuser:my-cluster/user1",
      "arn:aws:redshift:us-west-2:123456789012:dbuser:my-cluster/user2"
    ]
  }
}
}]
}

```

7. 신뢰 정책 업데이트를 선택합니다.

IAM 역할을 한 AWS 리전으로 제한

특정 AWS 리전에만 액세스할 수 있도록 IAM 역할을 제한할 수 있습니다. 기본적으로 Amazon Redshift에 대한 IAM 역할은 단일 리전으로 제한되지 않습니다.

IAM 역할 사용을 리전별로 제한하려면 다음 단계를 수행합니다.

IAM 역할에 대해 허용되는 리전을 확인하려면

1. <https://console.aws.amazon.com/>에서 [IAM 콘솔](#)을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 특정 리전에 제한되도록 수정할 역할을 선택합니다.
4. 신뢰 관계(Trust Relationships) 탭을 선택한 다음 신뢰 관계 편집(Edit Trust Relationship)을 선택합니다. Amazon Redshift가 귀하를 대신하여 다른 AWS 서비스에 액세스 할 수 있게 해주는 새로운 IAM 역할은 다음과 같이 신뢰 관계를 갖습니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

}

5. Service의 Principal 목록을 역할 사용을 허용할 특정 리전의 목록으로 수정합니다. Service 목록에서 각 리전은 `redshift.region.amazonaws.com` 형식이어야 합니다.

예를 들어, 다음과 같이 편집된 신뢰 관계는 us-east-1 및 us-west-2 리전에서만 IAM 역할을 사용하도록 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "redshift.us-east-1.amazonaws.com",
          "redshift.us-west-2.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. 신뢰 정책 업데이트(Update Trust Policy)를 선택합니다.

Amazon Redshift에서 IAM 역할 연결

역할을 클러스터에 연결하면 클러스터가 Amazon S3, Amazon Athena, AWS Glue 및 AWS Lambda에 액세스할 수 있는 역할을 맡을 수 있습니다. 클러스터에 연결된 역할에 필요한 리소스에 대한 액세스 권한이 없는 경우 다른 계정에 속한 다른 역할을 함께 묶을 수 있습니다. 그러면 클러스터는 함께 묶은 역할을 일시적으로 수임하여 데이터에 액세스합니다. 역할을 함께 묶어 교차 계정 액세스를 부여할 수도 있습니다. 체인의 각 역할은 클러스터가 체인의 끝에 있는 역할을 수임할 때까지 체인의 다음 역할을 수임합니다. 연결할 수 있는 최대 IAM 역할 수는 할당량에 따라 달라집니다. 자세한 내용은 [Amazon Redshift 객체에 대한 할당량](#)에서 할당량 "Amazon Redshift Redshift가 다른 AWS 서비스에 액세스하기 위한 클러스터 IAM 역할" 섹션을 참조하세요.

Note

체인이 올바르게 작동하려면 IAM 역할을 지정해야 합니다.

예를 들어, 회사 A가 회사 B에 속한 Amazon S3 버킷의 데이터에 액세스하려고 한다고 가정합니다. 회사 A는 RoleA라는 Amazon Redshift에 대한 AWS 서비스 역할을 생성하고 해당 클러스터에 연결합니다. 회사 B는 RoleB라는 이름의 역할을 생성합니다. 이는 회사 B 버킷의 데이터에 액세스할 수 있는 권한을 받았습니다. 회사 B 버킷의 데이터에 액세스하려면 회사 A는 iam_role 파라미터를 사용하여 COPY 명령을 실행하고 RoleA와 RoleB를 묶습니다. COPY 작업 기간 동안 RoleA은 RoleB를 수임하여 Amazon S3 버킷에 액세스합니다.

역할을 함께 묶으려면 역할 간에 신뢰 관계를 구성해야 합니다. 또 다른 역할을 수임하는 역할(예: RoleA)에는 다음 묶인 역할(예: RoleB)을 수임하도록 허용하는 권한 정책이 있어야 합니다. 반대로 권한을 전달하는 역할(RoleB)에는 권한을 이전에 묶인 역할(RoleA)로 전달하도록 허용하는 신뢰 정책이 있어야 합니다. 자세한 내용은 IAM User Guide의 [Using IAM roles](#)를 참조하세요.

체인의 첫 번째 역할은 클러스터에 연결된 역할이어야 합니다. 첫 번째 역할과 체인의 다음 역할을 수임하는 각 후속 역할에는 특정 문이 포함된 정책이 있어야 합니다. 이 문은 Allow작업 및 sts:AssumeRole 요소에 있는 다음 역할의 Amazon 리소스 이름(ARN)에 Resource 효과를 줍니다. 예를 들어 RoleA에는 RoleB를 수임하도록 허용하는 권한 정책이 있고, 이는 AWS 계정 210987654321이 소유합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1487639602000",
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": "arn:aws:iam::111122223333:role/RoleB"
    }
  ]
}
```

다른 역할로 전달하는 역할은 역할을 수임하는 역할 또는 해당 역할을 소유한 AWS 계정과의 신뢰 관계를 설정해야 합니다. 예를 들어 RoleB에는 신뢰 정책이 있어 RoleA와의 신뢰 관계를 설정합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/RoleA"
      }
    }
  ]
}
```

다음 신뢰 정책은 RoleA의 소유자인 AWS 계정 123456789012와의 신뢰 관계를 설정합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      }
    }
  ]
}
```

Note

특정 사용자로 역할 연결 권한 부여를 제한하려면 조건을 정의합니다. 자세한 내용은 [IAM 역할에 대한 액세스 제한](#) 섹션을 참조하세요.

UNLOAD, COPY, CREATE EXTERNAL FUNCTION 또는 CREATE EXTERNAL SCHEMA 명령을 실행할 때 `iam_role` 파라미터에서 쉼표로 구분된 역할 ARN 목록을 포함해 역할을 함께 묶습니다. 다음은 `iam_role` 파라미터에서 역할을 묶기 위한 구문을 나타냅니다.

```
unload ('select * from venue limit 10')
to 's3://acmedata/redshift/venue_pipe_'
IAM_ROLE 'arn:aws:iam::<aws-account-id-1>:role/<role-name-1>[,arn:aws:iam::<aws-account-id-2>:role/<role-name-2>][,...]';
```

Note

전체 역할 체인은 작은따옴표로 묶여 있으며, 공백을 포함해서는 안 됩니다.

다음 예에서 RoleA는 AWS 계정 123456789012에 속하는 클러스터에 연결됩니다. 계정 210987654321에 속한 RoleB는 `s3://companyb/redshift/`라는 버킷에 액세스할 수 있는 권한을 가집니다. 다음 예에서는 RoleA와 RoleB를 묶어 `s3://companyb/redshift/` 버킷에 데이터 UNLOAD를 수행합니다.

```
unload ('select * from venue limit 10')
to 's3://companyb/redshift/venue_pipe_'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

다음 예에서는 COPY 명령을 사용하여 이전 예에서 언로드된 데이터를 로드합니다.

```
copy venue
from 's3://companyb/redshift/venue_pipe_'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

다음 예에서는 CREATE EXTERNAL SCHEMA가 묶인 역할을 사용하여 RoleB를 수입합니다.

```
create external schema spectrumexample from data catalog
```

```
database 'exempladb' region 'us-west-2'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

다음 예에서는 CREATE EXTERNAL FUNCTION이 묶인 역할을 사용하여 역할 RoleB를 수임합니다.

```
create external function lambda_example(varchar)
returns varchar
volatile
lambda 'exampleLambdaFunction'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

IAM 역할을 사용하여 COPY, UNLOAD, CREATE EXTERNAL FUNCTION 및 CREATE EXTERNAL SCHEMA 작업에 대한 권한 부여

[COPY](#) 명령을 사용하여 데이터를 Amazon Redshift에 로드(또는 가져오기)하고 [UNLOAD](#) 명령을 사용하여 Amazon Redshift에서 데이터를 언로드(또는 내보내기)할 수 있습니다. CREATE EXTERNAL FUNCTION 명령을 사용하여 AWS Lambda에서 함수를 호출하는 사용자 정의 함수를 생성할 수 있습니다.

Amazon Redshift Spectrum을 사용하면 [CREATE EXTERNAL SCHEMA](#) 명령을 사용하여 데이터가 들어있는 Amazon S3 버킷의 위치를 지정할 수 있습니다. COPY, UNLOAD 또는 CREATE EXTERNAL SCHEMA 명령을 실행할 때 보안 자격 증명을 제공합니다. 이러한 자격 증명을 통해 Amazon Redshift 클러스터는 Amazon S3 버킷과 같은 대상에서 데이터를 읽거나 쓸 수 있습니다.

CREATE EXTERNAL FUNCTION을 실행할 때 IAM 역할 파라미터를 사용하여 보안 자격 증명을 제공합니다. 이러한 자격 증명은 AWS Lambda에서 Lambda 함수를 호출하도록 Amazon Redshift 클러스터에 권한을 부여합니다. 보안 자격 증명을 제공하는 데 선호되는 방법은 AWS Identity and Access Management(IAM) 역할을 지정하는 것입니다. COPY 및 UNLOAD의 경우 임시 자격 증명을 제공할 수 있습니다. IAM 역할 생성에 대한 자세한 내용은 [Amazon Redshift가 사용자를 대신하여 AWS 서비스에 액세스할 수 있도록 권한 부여](#) 섹션을 참조하세요.

사용자가 AWS Management Console 외부에서 AWS와 상호 작용하려면 프로그래밍 방식의 액세스 권한이 필요합니다. 프로그래밍 방식의 액세스 권한을 부여하는 방법은 AWS에 액세스하는 사용자 유형에 따라 다릅니다.

사용자에게 프로그래밍 방식의 액세스 권한을 부여하려면 다음 옵션 중 하나를 선택합니다.

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
IAM	(권장됨) 콘솔 자격 증명을 임시 자격 증명으로 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • AWS CLI의 경우 AWS Command Line Interface 사용 설명서의 AWS 로컬 개발을 위한 로그인을 참조하세요. • AWS SDK의 경우 AWS SDK 및 도구 참조 안내서의 AWS 로컬 개발을 위한 로그인을 참조하세요.
작업 인력 ID (IAM Identity Center에서 관리되는 사용자)	임시 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • AWS CLI에 대해서는 AWS Command Line Interface 사용자 안내서에서 AWS IAM Identity Center을 사용하도록 AWS CLI 구성을 참조하세요. • AWS SDK, 도구, AWS API에 대해서는 AWS SDK 및 도구 참조 가이드에서 IAM Identity Center 인증을 참조하세요.
IAM	임시 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	IAM 사용자 설명서의 AWS 리소스와 함께 임시 자격 증명 사용 에 나와 있는 지침을 따르세요.

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
IAM	(권장되지 않음) 장기 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 AWS API에 대한 프로그래밍 요청에 서명합니다.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • AWS CLI에 대해서는 AWS Command Line Interface 사용자 안내서에서 IAM 사용자 자격 증명을 사용한 인증을 참조하세요. • AWS SDK와 도구는 AWS SDK 및 도구 참조 가이드에서 장기 자격 증명을 사용한 인증을 참조하세요. • AWS API는 IAM 사용자 설명서에서 IAM 사용자의 액세스 키 관리를 참조하세요.

IAM 역할 사용 단계는 다음과 같습니다.

- Amazon Redshift 클러스터에서 사용할 IAM 역할을 생성합니다.
- IAM 역할을 클러스터와 연결합니다.
- COPY, UNLOAD, CREATE EXTERNAL SCHEMA 또는 CREATE EXTERNAL FUNCTION 명령을 호출할 때 IAM 역할의 ARN을 포함합니다.

IAM 역할을 클러스터와 연결

Amazon Redshift가 다른 AWS 서비스에 액세스하도록 권한을 부여하는 IAM 역할을 만든 후에는 해당 역할을 Amazon Redshift 클러스터와 연결해야 합니다. 역할을 사용하여 데이터를 로드하거나 언로드하려면 먼저 이 작업을 수행해야 합니다.

IAM 역할을 클러스터와 연결하는 데 필요한 권한

IAM 역할을 클러스터와 연결하려면 사용자에게 해당 IAM 역할에 대한 iam:PassRole 권한이 필요합니다. 이 권한을 통해 관리자는 사용자가 Amazon Redshift 클러스터와 연결할 수 있는 IAM 역할을 제한할 수 있습니다. 가장 좋은 방법은 권한 정책을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에

할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.

다음은 이러한 작업이 가능하도록 사용자에게 추가할 수 있는 IAM 정책 예입니다.

- 해당 사용자의 계정이 소유한 모든 Amazon Redshift 클러스터에 대한 세부 정보를 가져옵니다.
- 3가지 IAM 역할 중 하나를 2개의 Amazon Redshift 클러스터 중 하나와 연결합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "redshift:DescribeClusters",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "redshift:ModifyClusterIamRoles",
        "redshift:CreateCluster"
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:123456789012:cluster:my-redshift-cluster",
        "arn:aws:redshift:us-east-1:123456789012:cluster:my-second-redshift-cluster"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": [
        "arn:aws:iam::123456789012:role/MyRedshiftRole",
        "arn:aws:iam::123456789012:role/SecondRedshiftRole",
        "arn:aws:iam::123456789012:role/ThirdRedshiftRole"
      ]
    }
  ]
}
```

```
}
```

사용자에게 적절한 권한이 부여되면 해당 사용자는 Amazon Redshift 클러스터와 IAM 역할을 연결할 수 있습니다. 그런 다음 IAM 역할은 COPY 또는 UNLOAD 명령이나 다른 Amazon Redshift 명령과 함께 사용할 수 있습니다.

IAM 정책에 대한 자세한 내용은 IAM User Guide의 [Overview of IAM policies](#)를 참조하세요.

클러스터와 IAM 역할 연결 관리

클러스터를 생성할 때는 IAM 역할을 Amazon Redshift 클러스터와 연결할 수 있습니다. 또는 기존 클러스터를 수정하고 하나 이상의 IAM 역할 연결을 추가하거나 제거할 수 있습니다.

다음에 유의하세요.

- 연결할 수 있는 최대 IAM 역할 수는 할당량에 따라 달라집니다.
- IAM 역할을 여러 Amazon Redshift 클러스터와 연결할 수 있습니다.
- IAM 역할과 클러스터가 동일한 AWS 계정에 의해 소유된 경우에만 IAM 역할을 Amazon Redshift 클러스터와 연결할 수 있습니다.

IAM 역할과 클러스터의 연결은 다음 단계에 따라 콘솔에서 관리할 수 있습니다.

IAM 역할 연결을 관리하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 업그레이드할 클러스터를 선택합니다.
3. 작업(Actions)에서 IAM 역할 관리(Manage IAM roles)를 선택하여 클러스터와 연결된 현재 목록 IAM 역할을 표시합니다.
4. IAM 역할 관리(Manage IAM roles) 페이지에서 추가하려는 사용 가능한 IAM 역할을 선택한 다음 Add IAM role(IAM 역할 추가)을 선택합니다.
5. 완료(Done)를 선택하여 변경 사항을 저장합니다.

다음과 같은 접근 방법을 사용하여 AWS CLI에서 클러스터에 대한 IAM 역할을 관리할 수 있습니다.

클러스터를 생성할 때 IAM 역할을 클러스터와 연결하려면 `--iam-role-arns` 명령의 `create-cluster` 파라미터에서 IAM 역할의 Amazon 리소스 이름(ARN)을 지정해야 합니다. `create-cluster` 명령을 호출할 때 추가할 수 있는 최대 IAM 역할 수에는 할당량이 적용됩니다.

Amazon Redshift 클러스터와 IAM 역할의 연결 및 연결 해제는 비동기 프로세스입니다. IAM 역할과 클러스터의 모든 연결 상태는 `describe-clusters` 명령을 호출하여 확인할 수 있습니다.

다음은 IAM 역할 2개를 새롭게 생성한 클러스터인 `my-redshift-cluster`와 연결하는 예입니다.

```
aws redshift create-cluster \  
  --cluster-identifier "my-redshift-cluster" \  
  --node-type "ra3.4xlarge" \  
  --number-of-nodes 16 \  
  --iam-role-arns "arn:aws:iam::123456789012:role/RedshiftCopyUnload" \  
                  "arn:aws:iam::123456789012:role/SecondRedshiftRole"
```

IAM 역할을 기존 Amazon Redshift 클러스터와 연결하려면 `modify-cluster-iam-roles` 명령의 `--add-iam-roles` 파라미터에 대한 IAM 역할의 Amazon 리소스 이름(ARN)을 지정합니다. `modify-cluster-iam-roles` 명령을 호출할 때 추가할 수 있는 최대 IAM 역할 수에는 할당량이 적용됩니다.

다음은 IAM 역할을 기존 클러스터인 `my-redshift-cluster`와 연결하는 예입니다.

```
aws redshift modify-cluster-iam-roles \  
  --cluster-identifier "my-redshift-cluster" \  
  --add-iam-roles "arn:aws:iam::123456789012:role/RedshiftCopyUnload"
```

클러스터에서 IAM 역할을 연결 해제하려면 `modify-cluster-iam-roles` 명령의 `--remove-iam-roles` 파라미터에서 IAM 역할의 ARN을 지정해야 합니다. `modify-cluster-iam-roles` 명령을 호출할 때 삭제할 수 있는 최대 IAM 역할 수에는 할당량이 적용됩니다.

다음은 `my-redshift-cluster`라는 클러스터에서 123456789012 AWS 계정에 속한 IAM 역할 연결을 제거하는 예입니다.

```
aws redshift modify-cluster-iam-roles \  
  --cluster-identifier "my-redshift-cluster" \  
  --remove-iam-roles "arn:aws:iam::123456789012:role/RedshiftCopyUnload"
```

AWS CLI를 사용하여 클러스터에 대한 IAM 역할 연결 나열

Amazon Redshift 클러스터와 관련된 모든 IAM 역할과 IAM 역할 연결의 상태를 나열하려면 `describe-clusters` 명령을 호출합니다. 그러면 다음 예 출력과 같이 클러스터와 연결되어 있는 IAM 역할 각각의 ARN이 `IamRoles` 목록으로 반환됩니다.

클러스터와 연결된 역할은 상태가 `in-sync`로 표시됩니다. 클러스터와 연결이 진행 중인 역할은 상태가 `adding`으로 표시됩니다. 클러스터와 연결 해제 중인 역할은 상태가 `removing`으로 표시됩니다.

```
{
  "Clusters": [
    {
      "ClusterIdentifier": "my-redshift-cluster",
      "NodeType": "ra3.4xlarge",
      "NumberOfNodes": 16,
      "IamRoles": [
        {
          "IamRoleArn": "arn:aws:iam::123456789012:role/MyRedshiftRole",
          "IamRoleApplyStatus": "in-sync"
        },
        {
          "IamRoleArn": "arn:aws:iam::123456789012:role/SecondRedshiftRole",
          "IamRoleApplyStatus": "in-sync"
        }
      ],
      ...
    },
    {
      "ClusterIdentifier": "my-second-redshift-cluster",
      "NodeType": "ra3.4xlarge",
      "NumberOfNodes": 10,
      "IamRoles": [
        {
          "IamRoleArn": "arn:aws:iam::123456789012:role/MyRedshiftRole",
          "IamRoleApplyStatus": "in-sync"
        },
        {
          "IamRoleArn": "arn:aws:iam::123456789012:role/SecondRedshiftRole",
          "IamRoleApplyStatus": "in-sync"
        },
        {
          "IamRoleArn": "arn:aws:iam::123456789012:role/ThirdRedshiftRole",

```

```

        "IamRoleApplyStatus": "in-sync"
    }
},
...
}
]
}

```

AWS CLI 사용에 대한 자세한 내용은 [AWS CLI User Guide](#)를 참조하세요.

Amazon Redshift의 기본값으로 IAM 역할 생성

Redshift 콘솔을 통해 IAM 역할을 생성하면 Amazon Redshift가 프로그래밍 방식으로 AWS 계정에 역할을 생성하고 기존 AWS 관리형 정책을 자동으로 해당 역할에 연결합니다. 이 접근 방식을 사용하면 Redshift 콘솔 내에 머물면서 역할 생성을 위해 IAM 콘솔로 전환할 필요가 없습니다. Amazon Redshift 콘솔에서 생성된 기존 IAM 역할에 대한 권한을 보다 세부적으로 제어하기 위해 IAM 역할에 사용자 지정된 관리형 정책을 연결할 수 있습니다.

콘솔에서 만들어진 IAM 역할

Amazon Redshift 콘솔을 사용하여 IAM 역할을 생성하면 Amazon Redshift는 콘솔을 통해 생성된 모든 IAM 역할을 추적합니다. Amazon Redshift는 모든 새 클러스터를 생성하고 스냅샷에서 클러스터를 복원하기 위한 가장 최근의 기본 IAM 역할을 미리 선택합니다.

SQL 명령을 실행할 수 있는 권한을 가진 정책이 있는 콘솔을 통해 IAM 역할을 생성할 수 있습니다. 이러한 명령에는 COPY, UNLOAD, CREATE EXTERNAL FUNCTION, CREATE EXTERNAL TABLE, CREATE EXTERNAL SCHEMA, CREATE MODEL 또는 CREATE LIBRARY가 포함됩니다. 필요에 따라 사용자 정의 정책을 생성하고 IAM 역할에 연결하여 AWS 리소스에 대한 사용자 액세스를 보다 세부적으로 제어할 수 있습니다.

IAM 역할을 생성하고 콘솔을 사용하여 클러스터의 기본값으로 설정한 경우 인증 및 권한 부여를 수행하기 위해 IAM 역할의 Amazon 리소스 이름(ARN)을 제공할 필요가 없습니다.

클러스터를 위해 콘솔을 통해 생성한 IAM 역할에는 AmazonRedshiftAllCommandsFullAccess 관리형 정책이 자동으로 연결됩니다. 이 IAM 역할을 통해 Amazon Redshift는 IAM 계정의 AWS 리소스에 대한 데이터를 복사, 언로드, 쿼리 및 분석할 수 있습니다. 관리형 정책은 [COPY](#), [UNLOAD](#), [CREATE EXTERNAL FUNCTION](#), [CREATE EXTERNAL SCHEMA](#), [CREATE MODEL](#) 및 [CREATE LIBRARY](#) 작업을 제공합니다. 이 정책은 Amazon S3, Amazon CloudWatch Logs, Amazon SageMaker AI 및 AWS Glue와 같은 관련 AWS 서비스에 대해 SELECT 문을 실행할 수 있는 권한도 부여합니다.

CREATE EXTERNAL FUNCTION, CREATE EXTERNAL SCHEMA, CREATE MODEL 및 CREATE LIBRARY 명령에는 default 키워드가 있습니다. 이러한 명령에 대한 이 키워드의 경우 Amazon Redshift는 기본값으로 설정되고 명령이 실행될 때 클러스터와 연결된 IAM 역할을 사용합니다.

[DEFAULT_IAM_ROLE](#) 명령을 실행하여 클러스터에 연결된 현재 기본 IAM 역할을 확인할 수 있습니다.

Redshift 클러스터에 대해 생성되고 기본값으로 설정된 IAM 역할의 액세스 권한을 제어하려면 ASSUMEROLE 권한을 사용합니다. 이 액세스 제어는 데이터베이스 사용자 및 그룹이 앞에 나열된 것과 같은 명령을 실행할 때 적용됩니다. IAM 역할에 대해 사용자 또는 그룹에 ASSUMEROLE 권한을 부여한 후 사용자 또는 그룹은 이러한 명령을 실행할 때 해당 역할을 수임할 수 있습니다. ASSUMEROLE 권한을 사용하여 필요에 따라 적절한 명령에 대한 액세스 권한을 부여할 수 있습니다.

Amazon Redshift 콘솔을 사용하여 다음 작업을 수행할 수 있습니다.

- [IAM 역할을 기본값으로 생성](#)
- [클러스터에서 IAM 역할 제거](#)
- [클러스터와 IAM 역할 연결](#)
- [IAM 역할을 기본값으로 설정](#)
- [클러스터의 기본값으로 IAM 역할 설정 취소](#)

AmazonRedshiftAllCommandsFullAccess 관리형 정책의 권한

다음 예에서는 클러스터에 대해 기본값으로 설정된 IAM 역할에 대한 특정 작업을 허용하는 AmazonRedshiftAllCommandsFullAccess 관리형 정책의 권한을 보여줍니다. 권한 정책이 연결된 IAM 역할은 사용자 또는 그룹이 할 수 있는 것과 할 수 없는 것을 승인합니다. 이러한 권한이 주어진다면 Amazon S3에서 COPY 명령을 실행하고 UNLOAD를 실행하고 CREATE MODEL 명령을 사용할 수 있습니다.

```
{
    "Effect": "Allow",
    "Action": [
        "s3:GetObject",
        "s3:GetBucketAcl",
        "s3:GetBucketCors",
        "s3:GetEncryptionConfiguration",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListAllMyBuckets",
```



```

        "s3:ListMultipartUploadParts",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject",
        "s3:PutBucketAcl",
        "s3:PutBucketCors",
        "s3:DeleteObject",
        "s3:AbortMultipartUpload",
        "s3:CreateBucket"
    ],
    "Resource": [
        "arn:aws:s3:::redshift-downloads",
        "arn:aws:s3:::redshift-downloads/*",
        "arn:aws:s3:::*redshift*",
        "arn:aws:s3:::*redshift/*"
    ]
}

```

다음 예에서는 클러스터에 대해 기본값으로 설정된 IAM 역할에 대한 특정 작업을 허용하는 AmazonRedshiftAllCommandsFullAccess 관리형 정책의 권한을 보여줍니다. 권한 정책이 연결된 IAM 역할은 사용자 또는 그룹이 할 수 있는 것과 할 수 없는 것을 승인합니다. 다음 권한이 주어지면 CREATE EXTERNAL FUNCTION 명령을 실행할 수 있습니다.

```

{
  "Action": [
    "lambda:InvokeFunction"
  ],
  "Resource": "arn:aws:lambda:*:*:function:*redshift*"
}

```

다음 예에서는 클러스터에 대해 기본값으로 설정된 IAM 역할에 대한 특정 작업을 허용하는 AmazonRedshiftAllCommandsFullAccess 관리형 정책의 권한을 보여줍니다. 권한 정책이 연결된 IAM 역할은 사용자 또는 그룹이 할 수 있는 것과 할 수 없는 것을 승인합니다. 다음 권한이 주어지면 Amazon Redshift Spectrum에 필요한 CREATE EXTERNAL SCHEMA 및 CREATE EXTERNAL TABLE 명령을 실행할 수 있습니다.

```

{
  "Effect": "Allow",
  "Action": [
    "glue:CreateDatabase",
    "glue:DeleteDatabase",
    "glue:GetDatabase",

```

```

        "glue:GetDatabases",
        "glue:UpdateDatabase",
        "glue:CreateTable",
        "glue>DeleteTable",
        "glue:BatchDeleteTable",
        "glue:UpdateTable",
        "glue:GetTable",
        "glue:GetTables",
        "glue:BatchCreatePartition",
        "glue:CreatePartition",
        "glue>DeletePartition",
        "glue:BatchDeletePartition",
        "glue:UpdatePartition",
        "glue:GetPartition",
        "glue:GetPartitions",
        "glue:BatchGetPartition"
    ],
    "Resource": [
        "arn:aws:glue::*:table/*redshift*/",
        "arn:aws:glue::*:catalog",
        "arn:aws:glue::*:database/*redshift*"
    ]
}

```

다음 예에서는 클러스터에 대해 기본값으로 설정된 IAM 역할에 대한 특정 작업을 허용하는 AmazonRedshiftAllCommandsFullAccess 관리형 정책의 권한을 보여줍니다. 권한 정책이 연결된 IAM 역할은 사용자 또는 그룹이 할 수 있는 것과 할 수 없는 것을 승인합니다. 다음 권한이 주어지면 연합 쿼리를 사용하여 CREATE EXTERNAL SCHEMA 명령을 실행할 수 있습니다.

```

{
    "Effect": "Allow",
    "Action": [
        "secretsmanager:GetResourcePolicy",
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:ListSecretVersionIds"
    ],
    "Resource": [
        "arn:aws:secretsmanager::*:secret:*Redshift*"
    ]
},
{
    "Effect": "Allow",

```

```

    "Action": [
        "secretsmanager:GetRandomPassword",
        "secretsmanager:ListSecrets"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "secretsmanager:ResourceTag/Redshift": "true"
        }
    }
},

```

콘솔을 사용하여 클러스터에 대해 생성된 IAM 역할 관리

Amazon Redshift 콘솔에서 생성된 IAM 역할을 생성, 수정 및 제거하려면 콘솔의 클러스터(Clusters) 섹션을 사용합니다.

IAM 역할을 기본값으로 생성

콘솔에서 AmazonRedshiftAllCommandsFullAccess 정책이 자동으로 연결된 클러스터에 대한 IAM 역할을 생성할 수 있습니다. 생성하는 새 IAM 역할을 통해 Amazon Redshift가 IAM 계정의 Amazon 리소스에서 데이터를 복사, 로드, 쿼리 및 분석할 수 있습니다.

클러스터의 기본값으로 설정된 IAM 역할은 하나만 있을 수 있습니다. 기존 IAM 역할이 현재 기본값으로 할당된 경우 다른 IAM 역할을 클러스터 기본값으로 생성하면 새 IAM 역할이 다른 역할을 기본값으로 대체합니다.

새 클러스터와 해당 클러스터의 기본값으로 설정된 IAM 역할을 생성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다.
3. 클러스터를 생성하려면 클러스터 생성(Create cluster)을 선택합니다.
4. 콘솔 페이지의 지침에 따라 클러스터 구성(Cluster configuration)에 속성을 입력합니다. 이 단계에 대한 자세한 내용은 [클러스터 생성](#) 섹션을 참조하세요.
5. (선택 사항) 샘플 데이터 로드(Load sample data)를 선택하여 Amazon Redshift 클러스터에 샘플 데이터 집합을 로드하여 쿼리 편집기로 데이터 쿼리를 시작합니다.

방화벽 뒤에 있는 경우 데이터베이스 포트는 인바운드 연결을 허용하는 개방형 포트여야 합니다.

6. 콘솔 페이지의 지침에 따라 데이터베이스 구성(Database configurations)에 대한 속성을 입력합니다.
7. 클러스터 권한(Cluster permissions)의 IAM 역할 관리(Manage IAM roles)에서 IAM 역할 생성(Create IAM role)을 선택합니다.
8. 다음 방법 중 하나를 선택하여 IAM 역할이 액세스할 Amazon S3 버킷을 지정합니다.
 - 특정 Amazon S3 버킷을 지정하지 않고 IAM 역할을 생성하려면 추가 Amazon S3 버킷 없음(No additional Amazon S3 bucket)을 선택합니다.
 - Amazon Redshift 클러스터에 대한 액세스 권한이 있는 사용자가 AWS 계정에 있는 Amazon S3 버킷 및 해당 콘텐츠에도 액세스할 수 있도록 하려면 모든 Amazon S3 버킷(Any Amazon S3 bucket)을 선택합니다.
 - 특정 Amazon S3 버킷(Specific Amazon S3 buckets)을 선택하여 생성 중인 IAM 역할에 액세스 권한이 있는 하나 이상의 Amazon S3 버킷을 지정합니다. 그런 다음 테이블에서 Amazon S3 버킷을 하나 이상 선택합니다.
9. 기본값으로 IAM 역할 생성(Create IAM role as default)을 선택합니다. Amazon Redshift는 IAM 역할을 자동으로 생성하고 클러스터의 기본값으로 설정합니다.
10. 클러스터를 생성하려면 클러스터 생성(Create cluster)을 선택합니다. 클러스터를 사용할 준비가 끝나려면 몇 분이 걸릴 수 있습니다.

클러스터에서 IAM 역할 제거

클러스터에서 IAM 역할을 하나 이상 제거할 수 있습니다.

클러스터에서 IAM 역할을 제거하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다.
3. IAM 역할을 제거할 클러스터를 선택합니다.
4. 클러스터 권한(Cluster permissions)에서 클러스터에서 제거할 IAM 역할을 하나 이상 선택합니다.
5. IAM 역할 관리(Manage IAM roles)에서 IAM 역할 제거(Remove IAM roles)를 선택합니다.

클러스터와 IAM 역할 연결

클러스터와 IAM 역할을 하나 이상 연결할 수 있습니다.

클러스터와 IAM 역할을 연결하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다.
3. IAM 역할을 연결할 클러스터를 선택합니다.
4. 클러스터 권한(Cluster permissions)에서 클러스터와 연결할 IAM 역할을 하나 이상 선택합니다.
5. IAM 역할 관리(Manage IAM roles)에서 IAM 역할 연결(Associate IAM roles)을 선택합니다.
6. 클러스터와 연결할 IAM 역할을 하나 이상 선택합니다.
7. IAM 역할 연결(Associate IAM role)을 선택합니다.

IAM 역할을 기본값으로 설정

IAM 역할을 클러스터의 기본값으로 설정할 수 있습니다.

IAM 역할을 클러스터의 기본값으로 설정하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다.
3. 기본 IAM 역할을 설정하려는 클러스터를 선택합니다.
4. 클러스터 권한(Cluster permissions)의 연결된 IAM 역할(Associated IAM roles)에서 클러스터의 기본값으로 설정할 IAM 역할을 선택합니다.
5. 기본값 설정(Set default)에서 기본값으로 설정(Make default)을 선택합니다.
6. 메시지가 나타나면 기본값 설정(Set default)을 선택하여 지정된 IAM 역할을 기본값으로 설정하도록 확인합니다.

클러스터의 기본값으로 IAM 역할 설정 취소

클러스터의 기본값으로 IAM 역할 설정을 취소할 수 있습니다.

클러스터의 기본값으로 IAM 역할 설정을 취소하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. 현재 AWS 리전의 계정에 대한 클러스터가 나열됩니다. 각 클러스터의 속성 하위 집합이 목록의 열에 표시됩니다.
3. IAM 역할을 연결할 클러스터를 선택합니다.
4. 클러스터 권한(Cluster permissions)의 연결된 IAM 역할(Associated IAM roles)에서 기본 IAM 역할을 선택합니다.
5. 기본값 설정(Set default)에서 기본값 지우기(Clear default)를 선택합니다.
6. 메시지가 나타나면 기본값 지우기(Clear default)를 선택하여 지정된 IAM 역할의 기본값 설정을 취소합니다.

AWS CLI를 사용하여 클러스터에 생성된 IAM 역할 관리

AWS CLI를 사용하여 클러스터에 생성된 IAM 역할을 관리할 수 있습니다.

IAM 역할이 기본값으로 설정된 Amazon Redshift 클러스터를 생성하려면

IAM 역할이 클러스터의 기본값으로 설정된 Amazon Redshift 클러스터를 생성하려면 `aws redshift create-cluster` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 Amazon Redshift 클러스터와 `myrole1`이라는 IAM 역할을 생성합니다. AWS CLI 명령은 또한 `myrole1`을 클러스터의 기본값으로 설정합니다.

```
aws redshift create-cluster \
  --node-type dc2.large \
  --number-of-nodes 2 \
  --master-username adminuser \
  --master-user-password TopSecret1 \
  --cluster-identifier mycluster \
  --iam-roles 'arn:aws:iam::012345678910:role/myrole1'
'arn:aws:iam::012345678910:role/myrole2' \
  --default-iam-role-arn 'arn:aws:iam::012345678910:role/myrole1'
```

다음은 코드 조각은 응답의 예입니다.

```
{
  "Cluster": {
```

```

    "ClusterIdentifier": "mycluster",
    "NodeType": "dc2.large",
    "MasterUsername": "adminuser",
    "DefaultIamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
    "IamRoles": [
      {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
        "ApplyStatus": "adding"
      },
      {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
        "ApplyStatus": "adding"
      }
    ]
    ...
  }
}

```

Amazon Redshift 클러스터에 IAM 역할을 하나 이상 추가하려면

클러스터에 연결된 IAM 역할을 하나 이상 추가하려면 `aws redshift modify-cluster-iam-roles` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 클러스터에 myrole3과 myrole4를 추가합니다.

```

aws redshift modify-cluster-iam-roles \
  --cluster-identifier mycluster \
  --add-iam-roles 'arn:aws:iam::012345678910:role/myrole3'
  'arn:aws:iam::012345678910:role/myrole4'

```

다음은 코드 조각은 응답의 예입니다.

```

{
  "Cluster": {
    "ClusterIdentifier": "mycluster",
    "NodeType": "dc2.large",
    "MasterUsername": "adminuser",
    "DefaultIamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
    "IamRoles": [
      {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
        "ApplyStatus": "in-sync"
      },

```

```

        {
            "IamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
            "ApplyStatus": "in-sync"
        },
        {
            "IamRoleArn": "arn:aws:iam::012345678910:role/myrole3",
            "ApplyStatus": "adding"
        },
        {
            "IamRoleArn": "arn:aws:iam::012345678910:role/myrole4",
            "ApplyStatus": "adding"
        }
    ],
    ...
}

```

Amazon Redshift 클러스터에서 IAM 역할을 하나 이상 제거하려면

클러스터에 연결된 IAM 역할을 하나 이상 제거하려면 `aws redshift modify-cluster-iam-roles` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 클러스터에서 `myrole3`과 `myrole4`를 제거합니다.

```

aws redshift modify-cluster-iam-roles \
  --cluster-identifier mycluster \
  --remove-iam-roles 'arn:aws:iam::012345678910:role/myrole3'
  'arn:aws:iam::012345678910:role/myrole4'

```

다음은 코드 조각은 응답의 예입니다.

```

{
  "Cluster": {
    "ClusterIdentifier": "mycluster",
    "NodeType": "dc2.large",
    "MasterUsername": "adminuser",
    "DefaultIamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
    "IamRoles": [
      {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
        "ApplyStatus": "in-sync"
      },
      {

```



```

        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
        "ApplyStatus": "in-sync"
    },
    {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole3",
        "ApplyStatus": "removing"
    },
    {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole4",
        "ApplyStatus": "removing"
    }
],
...
}
}

```

연결된 IAM 역할을 클러스터의 기본값으로 설정하려면

연결된 IAM 역할을 클러스터의 기본값으로 설정하려면 `aws redshift modify-cluster-iam-roles` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 `myrole2`를 클러스터의 기본값으로 설정합니다.

```

aws redshift modify-cluster-iam-roles \
  --cluster-identifier mycluster \
  --default-iam-role-arn 'arn:aws:iam::012345678910:role/myrole2'

```

다음은 코드 조각은 응답의 예입니다.

```

{
  "Cluster": {
    "ClusterIdentifier": "mycluster",
    "NodeType": "dc2.large",
    "MasterUsername": "adminuser",
    "DefaultIamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
    "IamRoles": [
      {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
        "ApplyStatus": "in-sync"
      },
      {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
        "ApplyStatus": "in-sync"
      }
    ]
  }
}

```

```

    }
  ],
  ...
}
}

```

연결되지 않은 IAM 역할을 클러스터의 기본값으로 설정하려면

연결되지 않은 IAM 역할을 클러스터의 기본값으로 설정하려면 `aws redshift modify-cluster-iam-roles` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 `myrole2`를 Amazon Redshift 클러스터에 추가하고 클러스터의 기본값으로 설정합니다.

```

aws redshift modify-cluster-iam-roles \
  --cluster-identifier mycluster \
  --add-iam-roles 'arn:aws:iam::012345678910:role/myrole3' \
  --default-iam-role-arn 'arn:aws:iam::012345678910:role/myrole3'

```

다음은 코드 조각은 응답의 예입니다.

```

{
  "Cluster": {
    "ClusterIdentifier": "mycluster",
    "NodeType": "dc2.large",
    "MasterUsername": "adminuser",
    "DefaultIamRoleArn": "arn:aws:iam::012345678910:role/myrole3",
    "IamRoles": [
      {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
        "ApplyStatus": "in-sync"
      },
      {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
        "ApplyStatus": "in-sync"
      },
      {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole3",
        "ApplyStatus": "adding"
      }
    ],
    ...
  }
}

```

```
}
}
```

스냅샷에서 클러스터를 복원하고 IAM 역할을 클러스터의 기본값으로 설정하려면

스냅샷에서 클러스터를 복원할 때 기존 IAM 역할을 연결하거나 새 IAM 역할을 생성하여 클러스터의 기본값으로 설정할 수 있습니다.

스냅샷에서 Amazon Redshift 클러스터를 복원하고 IAM 역할을 클러스터 기본값으로 설정하려면 `aws redshift restore-from-cluster-snapshot` AWS CLI 명령을 사용합니다.

다음 AWS CLI 명령은 스냅샷에서 클러스터를 복원하고 `myrole2`를 클러스터의 기본값으로 설정합니다.

```
aws redshift restore-from-cluster-snapshot \
  --cluster-identifier mycluster-clone \
  --snapshot-identifier my-snapshot-id
  --iam-roles 'arn:aws:iam::012345678910:role/myrole1'
  'arn:aws:iam::012345678910:role/myrole2' \
  --default-iam-role-arn 'arn:aws:iam::012345678910:role/myrole1'
```

다음은 코드 조각은 응답의 예입니다.

```
{
  "Cluster": {
    "ClusterIdentifier": "mycluster-clone",
    "NodeType": "dc2.large",
    "MasterUsername": "adminuser",
    "DefaultIamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
    "IamRoles": [
      {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
        "ApplyStatus": "adding"
      },
      {
        "IamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
        "ApplyStatus": "adding"
      }
    ],
    ...
  }
}
```

페더레이션형 ID를 사용하여 로컬 리소스 및 Amazon Redshift Spectrum 외부 테이블에 대한 Amazon Redshift 액세스 관리

GetDatabaseCredentials에서 제공한 보안 인증 정보와 함께 AWS에서 아이덴티티 페더레이션을 사용하면 로컬 데이터와 외부 데이터에 대한 권한 부여 및 액세스를 간소화할 수 있습니다. 이 자습서에서는 특정 IAM 역할을 사용하는 대신 AWS ID 페더레이션을 통해 리소스에 대한 액세스를 제공하는 방법을 보여 줍니다.

현재는 Amazon S3에 있는 외부 데이터에 대한 액세스 권한을 사용자에게 부여하려면 권한 정책에 정의된 권한으로 IAM 역할을 생성해야 합니다. 그러면 역할이 연결된 사용자가 외부 데이터에 액세스할 수 있습니다. 이 방법이 작동하지만 특정 사용자가 특정 열을 사용할 수 없도록 하는 것과 같은 세분화된 규칙을 제공하려는 경우 외부 스키마에서 추가 구성을 수행해야 할 수 있습니다.

GetDatabaseCredentials에서 제공된 자격 증명 인증 정보를 사용하는 ID 페더레이션은 지정 및 변경이 더 쉬운 세분화된 IAM 규칙을 통해 AWS Glue 및 Redshift Spectrum 리소스에 대한 액세스를 제공할 수 있습니다. 이렇게 하면 비즈니스 규칙을 준수하는 액세스를 쉽게 적용할 수 있습니다.

페더레이션형 보안 인증을 사용하면 다음과 같은 이점이 있습니다.

- Redshift Spectrum용으로 클러스터에 연결된 IAM 역할을 관리할 필요가 없습니다.
- 클러스터 관리자는 다양한 IAM 컨텍스트를 사용하는 소비자가 액세스할 수 있는 외부 스키마를 생성할 수 있습니다. 이는 예를 들어 여러 소비자가 한 테이블에서 동일한 외부 스키마를 쿼리하고 반환된 레코드에서 다양한 필드를 가져오는 열 필터링을 수행하는 데 유용합니다.
- 역할뿐만 아니라 IAM 권한이 있는 사용자를 사용하여 Amazon Redshift를 쿼리할 수 있습니다.

페더레이션형 ID로 로그인하기 위한 ID 준비

페더레이션형 ID로 로그인하려면 먼저 몇 가지 예비 단계를 수행해야 합니다. 이 지침에서는 Amazon S3 버킷에 저장된 데이터 파일을 참조하는 기존 Redshift Spectrum 외부 스키마가 있고 해당 버킷이 Amazon Redshift 클러스터 또는 Amazon Redshift Serverless 데이터 웨어하우스와 동일한 계정에 있다고 가정합니다.

1. IAM ID를 생성합니다. 이는 사용자 또는 IAM 역할일 수 있습니다. IAM에서 지원하는 이름을 사용합니다.
2. ID에 권한 정책을 연결합니다. 다음 중 하나를 지정합니다.
 - `redshift:GetClusterCredentialsWithIAM`(Amazon Redshift에서 프로비저닝된 클러스터의 경우)

- `redshift-serverless:GetCredentials`(Amazon Redshift Serverless의 경우)

IAM 콘솔을 사용하여 정책 편집기로 권한을 추가할 수 있습니다.

IAM ID에는 외부 데이터에 액세스할 수 있는 권한도 필요합니다. 다음 AWS 관리형 정책을 직접 추가하여 Amazon S3 대한 액세스 권한을 부여합니다.

- `AmazonS3ReadOnlyAccess`
- `AWSGlueConsoleFullAccess`

AWS Glue를 사용하여 외부 데이터를 준비하는 경우 마지막 관리형 정책이 필요합니다. Amazon Redshift Spectrum에 대한 액세스 권한을 부여하는 단계에 대한 자세한 내용은 Amazon Redshift 및 Redshift Spectrum 시작 안내서의 일부인 [Amazon Redshift에 대한 IAM 역할 생성](#) 섹션을 참조하세요. Redshift Spectrum에 액세스하기 위한 IAM 정책을 추가하는 단계를 보여 줍니다.

3. Amazon Redshift에 연결하도록 SQL 클라이언트를 설정합니다. Amazon Redshift JDBC 드라이버를 사용하고 사용자의 보안 인증 정보를 도구의 보안 인증 정보 속성에 추가합니다. SQL Workbench/J와 같은 클라이언트가 이 작업에 적합합니다. 다음과 같은 클라이언트 연결 확장 속성을 설정합니다.
 - `AccessKeyID` - 액세스 키 식별자입니다.
 - `SecretAccessKey` - 비밀 액세스 키입니다. (암호화를 사용하지 않는 경우 비밀 키를 전송할 때 발생할 수 있는 보안 위험에 유의하세요.)
 - `SessionToken` - IAM 역할용 임시 보안 인증 정보 집합입니다.
 - `groupFederation` - 프로비저닝된 클러스터의 페더레이션형 ID를 구성하는 경우 `true`로 설정합니다. Amazon Redshift Serverless를 사용하는 경우 이 파라미터를 설정하지 마세요.
 - `LogLevel` - 정수 로그 수준 값입니다. 이는 선택 사항입니다.
4. URL을 Amazon Redshift 또는 Amazon Redshift Serverless 콘솔에서 찾은 JDBC 엔드포인트로 설정합니다. URL 스키마를 `jdbc:redshift:iam:`으로 바꾸고 다음 형식을 사용합니다.
 - Amazon Redshift에서 프로비저닝된 클러스터의 형식: `jdbc:redshift:iam://<cluster_id>.<unique_suffix>.<region>.redshift.amazonaws.com:<port>/<database_name>`

예시: `jdbc:redshift:iam://test1.12345abcdefg.us-east-1.redshift.amazonaws.com:5439/dev`

- Amazon Redshift Serverless의 형식: `jdbc:redshift:iam://<workgroup-name>.<account-number>.<aws-region>.redshift-serverless.amazonaws.com:5439:<port>/<database_name>`

예시: `jdbc:redshift:iam://default.123456789012.us-east-1.redshift-serverless.amazonaws.com:5439/dev`

IAM ID를 사용하여 처음으로 데이터베이스에 연결한 후 Amazon Redshift는 접두사 IAM:(사용자의 경우) 또는 접두사 IAMR:(IAM 역할의 경우)가 붙은 동일한 이름의 Amazon Redshift ID를 자동으로 생성합니다. 이 주제의 나머지 단계에서는 사용자에 대한 예를 보여줍니다.

Redshift 사용자가 자동으로 생성되지 않는 경우 관리자 계정으로 IAM:<user name> 형식의 사용자 이름을 지정해 CREATE USER 문을 실행하여 사용자를 생성할 수 있습니다.

5. Amazon Redshift 클러스터 관리자로서 Redshift 사용자에게 외부 스키마에 액세스하는 데 필요한 권한을 부여합니다.

```
GRANT ALL ON SCHEMA my_schema to "IAM:my_user";
```

Redshift 사용자에게 외부 스키마에서 테이블을 생성할 수 있는 권한을 부여하려면 해당 사용자가 스키마 소유자여야 합니다. 예를 들면 다음과 같습니다.

```
ALTER SCHEMA my_schema owner to "IAM:my_user";
```

6. 구성을 확인하려면 권한이 부여된 후 SQL 클라이언트를 사용하여 사용자로 쿼리를 실행합니다. 이 쿼리 샘플은 외부 테이블에서 데이터를 검색합니다.

```
SELECT * FROM my_schema.my_table;
```

Redshift Spectrum에 ID 및 권한 부여 전파 시작하기

외부 테이블을 쿼리하기 위해 페더레이션형 ID를 전달하려면 SESSION을 CREATE EXTERNAL SCHEMA의 IAM_ROLE 쿼리 파라미터 값으로 설정합니다. 다음 단계에서는 SESSION을 설정하고 활용하여 외부 스키마에 대한 쿼리 권한을 부여하는 방법을 보여줍니다.

1. 로컬 테이블과 외부 테이블을 생성합니다. AWS Glue를 통해 카탈로그로 작성된 외부 테이블을 이 에 사용할 수 있습니다.

2. IAM ID로 Amazon Redshift에 연결합니다. 이전 섹션에서 설명한 것처럼 ID가 Amazon Redshift에 연결되면 Redshift 데이터베이스 사용자가 생성됩니다. 사용자가 이전에 존재하지 않은 경우 생성됩니다. 사용자가 새 사용자인 경우 관리자는 Amazon Redshift에서 쿼리 및 테이블 생성과 같은 작업을 수행할 수 있는 권한을 사용자에게 부여해야 합니다.
3. 관리자 계정으로 Redshift에 연결합니다. SESSION 값을 사용해 명령을 실행하여 외부 스키마를 생성합니다.

```
create external schema spectrum_schema from data catalog
database '<my_external_database>'
region '<my_region>'
iam_role 'SESSION'
catalog_id '<my_catalog_id>;'
```

이 경우 catalog_id가 설정됩니다. SESSION이 특정 역할을 대체하기 때문에 기능에 추가된 새로운 설정입니다.

이 예에서 쿼리의 값은 실제 값이 나타나는 방식을 모방한 것입니다.

```
create external schema spectrum_schema from data catalog
database 'spectrum_db'
region 'us-east-1'
iam_role 'SESSION'
catalog_id '123456789012'
```

이 경우의 catalog_id 값은 AWS 계정 ID입니다.

4. 2단계에서 연결한 IAM ID를 사용해 쿼리를 실행하여 외부 데이터에 액세스합니다. 예를 들면 다음과 같습니다.

```
select * from spectrum_schema.table1;
```

이 경우 table1은 예를 들어 Amazon S3 버킷에 있는 파일의 JSON 형식 데이터일 수 있습니다.

5. 외부 데이터베이스 또는 스키마를 가리키는 클러스터 연결 IAM 역할을 사용하는 외부 스키마가 이미 있는 경우 이 단계에 설명된 대로 기존 스키마를 교체하고 페더레이션형 ID를 사용하거나 새 스키마를 생성할 수 있습니다.

SESSION은 페더레이션형 ID 보안 인증 정보가 외부 스키마를 쿼리하는 데 사용됨을 나타냅니다. SESSION 쿼리 파라미터를 사용할 때는 catalog_id를 설정해야 합니다. 스키마에 사용되는 데이터

카탈로그를 가리키기 때문에 필요합니다. 이전에는 `iam_role`에 할당된 값에서 `catalog_id`가 검색되었습니다. 예를 들어 외부 스키마를 쿼리할 때 페더레이션형 보안 인증 정보를 사용하여 Redshift Spectrum에 이러한 방식으로 ID 및 권한 부여를 전파하도록 설정하면 IAM 역할을 통한 권한 부여가 필요하지 않습니다.

사용 노트

일반적인 연결 오류는 다음과 같습니다. 임시 자격 증명을 검색하는 중 IAM 오류: 제공된 언마셜러로 예외 응답을 언마셜링할 수 없음. 이 오류는 레거시 JDBC 드라이버가 있기 때문에 발생합니다. 페더레이션 ID에 필요한 최소 드라이버 버전은 2.1.0.9입니다. JDBC 드라이버는 [Amazon Redshift JDBC 드라이버, 버전 2.x 다운로드](#)에서 다운로드할 수 있습니다.

추가 리소스

이러한 링크는 외부 데이터에 대한 액세스를 관리하기 위한 추가 정보를 제공합니다.

- IAM 역할을 사용하여 Redshift Spectrum 데이터에 액세스할 수 있습니다. 자세한 내용은 [Amazon Redshift가 사용자를 대신하여 AWS 서비스에 액세스할 수 있도록 권한 부여](#) 섹션을 참조하세요.
- AWS Lake Formation으로 외부 테이블에 대한 액세스를 관리할 때 페더레이션 IAM 자격 증명과 함께 Redshift Spectrum을 사용하여 쿼리할 수 있습니다. 더 이상 AWS Lake Formation에 등록된 데이터를 쿼리하기 위해 Redshift Spectrum에 대한 클러스터 연결 IAM 역할을 관리할 필요가 없습니다. 자세한 내용은 [Amazon Redshift Spectrum과 함께 AWS Lake Formation 사용](#) 섹션을 참조하세요.

AWS Secrets Manager를 사용하여 Amazon Redshift 관리자 암호를 관리하는 방법

Amazon Redshift는 AWS Secrets Manager와 통합하여 암호화된 보안 암호 내에서 관리자 보안 인증 정보를 생성하고 관리할 수 있습니다. AWS Secrets Manager를 사용하면 관리자 암호를 API 호출로 대체하여 필요할 때 프로그래밍 방식으로 보안 암호를 검색할 수 있습니다. 하드 코딩된 보안 인증 정보 대신 암호를 사용하면 보안 인증 정보가 노출되거나 손상될 위험이 줄어듭니다. AWS Secrets Manager에 대한 자세한 내용은 [AWS Secrets Manager 사용 설명서](#)를 참조하세요.

다음 작업 중 하나를 수행할 때 Amazon Redshift가 AWS Secrets Manager를 사용하여 관리자 암호를 관리하도록 지정할 수 있습니다.

- 프로비저닝된 클러스터 또는 서버리스 네임스페이스 생성
- 프로비저닝된 클러스터 또는 서버리스 네임스페이스의 관리자 자격 증명 편집, 업데이트 또는 수정

- 스냅샷에서 클러스터 또는 서버리스 네임스페이스 복원

Amazon Redshift가 AWS Secrets Manager에서 관리자 암호를 관리하도록 지정하면 Amazon Redshift가 암호를 생성하여 Secrets Manager에 저장합니다. AWS Secrets Manager에서 직접 보안 암호에 액세스하여 관리자 사용자의 보안 인증 정보를 검색할 수 있습니다. 선택적으로, 다른 AWS 계정에서 보안 암호에 액세스해야 하는 경우 고객 관리형 키를 지정하여 보안 암호를 암호화할 수 있습니다. AWS Secrets Manager에서 제공하는 KMS 키를 사용할 수도 있습니다.

Amazon Redshift는 보안 암호 설정을 관리하고 기본적으로 30일마다 보안 암호를 교체합니다. 언제든지 보안 암호를 수동으로 교체할 수 있습니다. AWS Secrets Manager에서 보안 암호를 관리하는 프로비저닝된 클러스터 또는 서버리스 네임스페이스를 삭제하면 보안 암호와 관련 메타데이터도 삭제됩니다.

보안 암호 관리형 보안 인증 정보를 사용하여 클러스터 또는 서버리스 네임스페이스에 연결하려면 Secrets Manager 콘솔 또는 `GetSecretValue` Secrets Manager API 호출을 사용하여 AWS Secrets Manager에서 보안 암호를 검색하면 됩니다. 자세한 내용은 [AWS Secrets Manager에서 비밀 검색](#) 및 [AWS Secrets Manager 비밀에 있는 보안 인증 정보를 사용하여 SQL 데이터베이스에 연결](#)을 참조하세요.

AWS Secrets Manager 통합에 필요한 권한

사용자는 AWS Secrets Manager 통합과 관련된 작업을 수행하는 데 필요한 권한이 있어야 합니다. 사용자에게 필요한 지정된 리소스에서 특정 API 작업을 수행할 수 있는 권한을 부여하는 IAM 정책을 생성합니다. 그런 다음 해당 권한이 필요한 IAM 권한 세트 또는 역할에 이러한 정책을 연결합니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#) 섹션을 참조하세요.

Amazon Redshift가 AWS Secrets Manager에서 관리자 암호를 관리하도록 지정하는 사용자는 다음 작업을 수행할 수 있는 권한이 있어야 합니다.

- `secretsmanager:CreateSecret`
- `secretsmanager:RotateSecret`
- `secretsmanager:DescribeSecret`
- `secretsmanager:UpdateSecret`
- `secretsmanager>DeleteSecret`
- `secretsmanager:GetRandomPassword`
- `secretsmanager:TagResource`

프로비저닝된 클러스터의 MasterPasswordSecretKmsKeyId 파라미터 또는 서버리스 네임스페이스의 AdminPasswordSecretKmsKeyId 파라미터에 KMS 키를 전달하려는 사용자는 위에 나열된 권한 외에도 다음과 같은 권한이 필요합니다.

- kms:Decrypt
- kms:GenerateDataKey
- kms:CreateGrant
- kms:RetireGrant

관리자 보안 암호 교체

기본적으로 Amazon Redshift는 보안 인증 정보가 장기간 동일하게 유지되지 않도록 30일마다 보안 암호를 자동으로 교체합니다. Amazon Redshift가 관리자 암호 보안 암호를 교체하면 AWS Secrets Manager는 새 관리자 암호를 포함하도록 기존 보안 암호를 업데이트합니다. Amazon Redshift는 업데이트된 보안 암호의 암호와 일치하도록 클러스터의 관리자 암호를 변경합니다.

AWS Secrets Manager를 사용하여 예약된 교체 시점을 기다리지 않고 보안 암호를 즉시 교체할 수 있습니다. 보안 암호 교체에 대한 자세한 내용은 AWS Secrets Manager 사용 설명서에서 [AWS Secrets Manager 보안 암호 교체](#)를 참조하세요.

Amazon Redshift와 함께 AWS Secrets Manager 사용 시 고려 사항

AWS Secrets Manager를 사용하여 프로비저닝된 클러스터 또는 서버리스 네임스페이스의 관리자 보안 인증 정보를 관리할 때는 다음 사항을 고려하세요.

- 관리자 보안 인증 정보가 AWS Secrets Manager에서 관리되는 클러스터를 일시 중지해도 클러스터의 보안 암호는 삭제되지 않으며 보안 암호에 대한 요금이 계속 청구됩니다. 보안 암호는 클러스터를 삭제한 경우에만 삭제됩니다.
- Amazon Redshift가 연결된 보안 암호를 교체하려고 할 때 클러스터가 일시 중지되면 교체가 실패합니다. 이 경우 Amazon Redshift는 자동 교체를 중지하고 클러스터를 재개한 후에도 다시 교체를 시도하지 않습니다. secretsmanager:RotateSecret가 계속해서 보안 암호를 자동 교체하도록 하려면 AWS Secrets Manager API 호출을 사용하여 자동 교체 일정 예약을 다시 시작해야 합니다.
- Amazon Redshift가 연결된 보안 암호를 교체하려고 할 때 서버리스 네임스페이스에 연결된 작업 그룹이 없는 경우, 교체는 실패하고 작업 그룹을 연결한 후에도 다시 교체를 시도하지 않습니다. secretsmanager:RotateSecret가 계속해서 보안 암호를 자동 교체하도록 하려면 AWS Secrets Manager API 호출을 사용하여 자동 교체 일정 예약을 다시 시작해야 합니다.

Amazon Redshift에서 Amazon 리소스 이름(ARN) 보안 암호 검색

Amazon Redshift 콘솔을 사용하여 AWS Secrets Manager에서 관리되는 보안 암호의 Amazon 리소스 이름(ARN)을 볼 수 있습니다. 보안 암호의 ARN이 있으면 AWS Secrets Manager를 사용하여 보안 암호에 대한 세부 정보 및 보안 암호 내에 암호화된 데이터를 볼 수 있습니다. ARN을 사용하여 보안 암호를 검색하는 방법에 대한 자세한 내용은 AWS Secrets Manager 사용 설명서에서 [보안 암호 가져오기](#)를 참조하세요.

Amazon Redshift 프로비저닝 클러스터의 보안 암호에 대한 세부 정보 보기

Amazon Redshift 콘솔을 사용하여 다음 절차에 따라 클러스터에 대한 보안 암호의 Amazon 리소스 이름(ARN)을 확인합니다.

1. AWS Management Console에 로그인하고 Amazon Redshift 콘솔을 엽니다.
2. 클러스터 개요 창에서 보안 암호를 보려는 클러스터를 선택합니다.
3. 속성 탭을 선택합니다.
4. 관리자 보안 인증 정보 ARN에서 보안 암호의 ARN을 확인합니다. 이 ARN은 보안 암호의 식별자이며, AWS Secrets Manager에서 보안 암호의 세부 정보를 보는 데 사용할 수 있습니다.

Amazon Redshift Serverless 네임스페이스의 보안 암호에 대한 세부 정보 보기

Amazon Redshift 콘솔을 사용하여 다음 절차에 따라 서버리스 네임스페이스에 대한 보안 암호의 Amazon 리소스 이름(ARN)을 확인합니다.

1. AWS Management Console에 로그인하고 Amazon Redshift 콘솔을 엽니다.
2. 프로비저닝된 클러스터 대시보드에서 페이지 오른쪽 상단의 Serverless로 이동을 선택합니다.
3. Serverless 대시보드에서 네임스페이스 / 작업 그룹 창으로 스크롤하여 보안 암호를 보려는 네임스페이스를 선택합니다.
4. 일반 정보 창의 관리자 보안 인증 정보 ARN에서 보안 암호의 ARN을 확인합니다. 이 ARN은 보안 암호의 식별자이며, AWS Secrets Manager에서 보안 암호의 세부 정보를 보는 데 사용할 수 있습니다.

데이터베이스 연결 자격 증명을 위한 보안 암호 생성

Secrets Manager 보안 암호를 생성하여 Amazon Redshift 프로비저닝된 클러스터 또는 Redshift Serverless 네임스페이스 및 작업 그룹에 연결하는 데 사용되는 자격 증명을 저장할 수 있습니다. 이 보안 암호는 Amazon Redshift 쿼리 에디터 v2에서 쿼리를 예약할 때도 사용할 수 있습니다.

Secrets Manager 콘솔을 사용하여 Amazon Redshift 프로비저닝된 클러스터의 데이터베이스에 대한 보안 암호를 생성하려면

1. <https://console.aws.amazon.com/secretsmanager/>에서 Secrets Manager 콘솔을 엽니다.
2. 보안 암호 목록으로 이동한 다음 새 보안 암호 저장을 선택합니다.
3. Amazon Redshift 데이터 웨어하우스에 대한 자격 증명을 선택합니다. 보안 암호를 생성하는 단계에서 다음과 같이 정보를 입력합니다.
 - 사용자 이름 자격 증명에 데이터 웨어하우스의 관리 사용자 이름을 입력합니다.
 - 암호 자격 증명에 사용자 이름의 암호를 입력합니다.
 - 암호화 키에서 암호화 키를 선택합니다.
 - 데이터 웨어하우스에서 데이터가 포함된 Amazon Redshift 프로비저닝된 클러스터를 선택합니다.
 - 보안 암호 이름에 보안 암호 이름을 입력합니다.
 - 설명에 보안 암호의 설명을 입력합니다.
 - 태그에 **Redshift** 단어가 포함된 태그 키를 입력합니다. 이 태그 키는 Amazon Redshift 쿼리 에디터 v2를 사용하여 데이터 웨어하우스에 연결하려고 할 때 보안 암호를 나열하는 데 필요합니다. 보안 암호에 **Redshift** 문자열로 시작하는 태그 키가 있어야만 보안 암호가 관리 콘솔의 AWS Secrets Manager에 나열됩니다.
4. 검토 단계에서 변경 내용을 저장할 때까지 여러 화면을 통해 보안 암호에 대한 정보를 계속 입력합니다.

자격 증명, 엔진, 호스트, 포트 및 클러스터 식별자의 특정 값이 보안 암호에 저장됩니다. 또한 태그 키 Redshift를 사용하여 보안 암호에 태그가 지정됩니다.

Redshift Serverless 콘솔을 사용하여 Redshift Serverless 네임스페이스에서 데이터베이스의 보안 암호를 생성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. Redshift Serverless를 선택하고 네임스페이스 구성으로 이동합니다.
3. 보안 암호 자격 증명을 생성할 네임스페이스를 선택합니다.
4. 작업, 관리자 자격 증명 편집을 엽니다.
5. 관리자 암호에서 AWS Secrets Manager에서 관리자 자격 증명 관리를 선택합니다.
6. 변경 사항을 저장하려면 변경 사항 저장을 선택합니다.

암호가 성공적으로 변경되었다는 메시지가 나타나는지 확인합니다. Secrets Manager 콘솔에서도 보안 암호를 볼 수 있습니다. 이 보안 암호를 사용하면 AWS Secrets Manager 연결 방법을 사용하여 Redshift Serverless 콘솔 및 Amazon Redshift 쿼리 에디터 v2에 있는 작업 그룹의 데이터베이스에 연결할 수 있습니다. 보안 암호에 'Redshift' 문자열로 시작하는 태그 키가 있어야만 쿼리 에디터 v2 웹 애플리케이션에 나열됩니다. 보안 암호에 **Redshift** 문자열로 시작하는 태그 키가 있어야만 보안 암호가 관리 콘솔의 AWS Secrets Manager에 나열됩니다.

Secrets Manager 콘솔을 사용하여 Redshift Serverless 네임스페이스에서 데이터베이스의 보안 암호를 생성하려면

1. <https://console.aws.amazon.com/secretsmanager/>에서 Secrets Manager 콘솔을 엽니다.
2. 보안 암호 목록으로 이동한 다음 새 보안 암호 저장을 선택합니다.
3. Amazon Redshift 데이터 웨어하우스에 대한 자격 증명을 선택합니다. 보안 암호를 생성하는 단계에서 다음과 같이 정보를 입력합니다.
 - 사용자 이름 자격 증명에 데이터 웨어하우스의 관리 사용자 이름을 입력합니다.
 - 암호 자격 증명에 사용자 이름의 암호를 입력합니다.
 - 암호화 키에서 암호화 키를 선택합니다.
 - 데이터 웨어하우스에서 데이터가 포함된 Redshift Serverless 네임스페이스를 선택합니다.
 - 보안 암호 이름에 보안 암호 이름을 입력합니다.
 - 설명에 보안 암호의 설명을 입력합니다.
 - 태그에 **Redshift** 단어가 포함된 태그 키를 입력합니다. 이 태그 키는 Amazon Redshift 쿼리 에디터 v2를 사용하여 데이터 웨어하우스에 연결하려고 할 때 보안 암호를 나열하는 데 필요합니다. 보안 암호에 **Redshift** 문자열로 시작하는 태그 키가 있어야만 보안 암호가 관리 콘솔의 AWS Secrets Manager에 나열됩니다.
4. 검토 단계에서 변경 내용을 저장할 때까지 여러 화면을 통해 보안 암호에 대한 정보를 계속 입력합니다.

자격 증명, 데이터베이스 이름, 호스트, 포트, 네임스페이스 및 엔진의 특정 값이 보안 암호에 저장됩니다. 또한 태그 키 Redshift를 사용하여 보안 암호에 태그가 지정됩니다.

AWS CLI를 사용하여 Redshift Serverless 네임스페이스에서 데이터베이스의 보안 암호를 생성하려면

AWS CLI를 사용하여 보안 암호를 생성할 수 있습니다. 한 가지 방법은 AWS CloudShell을 사용하여 다음과 같이 Secrets Manager AWS CLI 명령을 실행하는 것입니다. 다음 절차에 표시된 AWS CLI 명령을 실행하려면 적절한 권한이 있어야 합니다.

1. AWS 콘솔에서 AWS CloudShell 명령 프롬프트를 엽니다. AWS CloudShell에 대한 자세한 내용은 AWS CloudShell 사용 설명서에서 [AWS CloudShell이란 무엇인가요](#)를 참조하세요.
2. 예를 들어 보안 암호가 MyTestSecret인 경우, 데이터베이스에 연결하거나 Amazon Redshift 쿼리 에디터 v2 쿼리를 예약하는 데 사용되는 보안 암호를 저장하는 Secrets Manager 명령을 입력합니다. 명령에서 다음 값을 해당 환경의 값으로 바꿉니다.
 - *admin*은 데이터 웨어하우스의 관리 사용자 이름입니다.
 - *passwd0rd*는 관리자의 암호입니다.
 - *dev*는 데이터 웨어하우스의 초기 데이터베이스 이름입니다.
 - *region*은 데이터 웨어하우스가 포함된 AWS 리전입니다. 예: us-east-1.
 - *123456789012*는 AWS 계정입니다.
 - *namespace-id*는 c3928f0e-c889-4d2b-97a5-5738324d5d3e와 비슷한 네임스페이스 식별자입니다. 이 식별자는 Amazon Redshift 콘솔의 서버리스 네임스페이스 세부 정보 페이지에서 찾을 수 있습니다.

```
aws secretsmanager create-secret \
--name MyTestSecret \
--description "My test secret created with the CLI." \
--secret-string "{\"username\":\"admin\",\"password\":\"passwd0rd\",\"dbname\":  
\"dev\",\"engine\":\"redshift\"}" \
--tags "[{\"Key\":\"redshift-serverless:namespaceArn\", \"Value\":  
\"arn:aws:redshift-serverless:region:123456789012:namespace/namespace-id\"}]"
```

Amazon Redshift의 로깅 및 모니터링

모니터링은 Amazon Redshift와 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. 다중 지점 실패가 발생할 경우 보다 쉽게 디버깅할 수 있도록 AWS 솔루션의 모든 부분으로부터 모니터링 데이터를 수집할 수 있습니다. AWS는 Amazon Redshift 리소스를 모니터링하고 잠재적 인시던트에 대응하기 위한 여러 도구를 제공합니다.

Amazon CloudWatch 경보

Amazon CloudWatch 경보를 사용하면 지정한 기간 동안 단일 지표를 감시합니다. 지표가 지정된 임계값을 초과하면 Amazon SNS 주제 또는 AWS Auto Scaling 정책으로 알림이 전송됩니다. CloudWatch 경보는 단순히 특정 상태에 있다고 해서 작업을 호출하지 않습니다. 대신, 상태가 변경

되어 지정된 기간 동안 유지되어야 합니다. 자세한 내용은 [경보 생성](#) 섹션을 참조하세요. 지표 목록은 [Amazon Redshift의 성능 데이터](#) 단원을 참조하십시오.

AWS CloudTrail 로그

CloudTrail은 Amazon Redshift에서 사용자, IAM 역할 또는 AWS 서비스가 수행한 API 작업에 대한 레코드를 제공합니다. CloudTrail에서 수집한 정보를 사용하여 Amazon Redshift에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다. 자세한 내용은 [CloudTrail을 사용하여 로깅](#) 섹션을 참조하세요.

데이터베이스 감사 로깅

Amazon Redshift는 연결 및 사용자 작업에 대한 정보를 데이터베이스에 기록합니다. 이렇게 기록되는 로그는 보안 및 문제 해결을 목적으로 데이터베이스를 모니터링하는 데 효과적입니다. 이러한 프로세스를 데이터베이스 감사라고 합니다. 로그는 다음 위치에 저장할 수 있습니다.

- Amazon S3 버킷 - 데이터베이스의 모니터링 작업을 담당하는 사용자를 위해 데이터 보안 기능으로 편리하게 액세스할 수 있습니다.
- Amazon CloudWatch - 시각화 기능 및 설정 작업과 같은 CloudWatch에 내장된 기능을 사용하여 감사 로깅 데이터를 볼 수 있습니다.

Note

[SYS_CONNECTION_LOG](#)는 Amazon Redshift Serverless에 대한 연결 로그 데이터를 수집합니다. Amazon Redshift Serverless에 대한 감사 로깅 데이터를 수집할 때는 로그 파일로 전송할 수 없으며 CloudWatch로만 전송할 수 있다는 점에 유의하세요.

주제

- [Amazon Redshift 로그](#)
- [감사 로그 및 Amazon CloudWatch](#)
- [감사 로깅 활성화](#)
- [보안 로깅](#)

Amazon Redshift 로그

Amazon Redshift는 다음 로그 파일에 정보를 로그합니다.

- 연결 로그(Connection log) - 인증 시도 횟수와 연결 및 연결 해제 정보를 로그합니다.
- 사용자 로그(User log) - 데이터베이스 사용자 정의 변경 사항에 대한 정보를 로그합니다.
- 사용자 작업 로그(User activity log) - 데이터베이스에서 실행하기 전에 각 쿼리를 로그합니다.

연결 및 사용자 로그는 주로 보안 목적으로 사용하기 좋습니다. 연결 로그를 사용하여 데이터베이스에 연결 중인 사용자에 대한 정보와 관련 연결 정보를 모니터링할 수 있습니다. 이 정보는 IP 주소, 요청한 시기, 사용한 인증 유형 등이 될 수 있습니다. 그리고 사용자 로그는 데이터베이스 사용자의 정의 변경 여부를 모니터링하는 데 사용됩니다.

사용자 작업 로그는 주로 문제 해결 목적으로 사용하기 좋습니다. 이 로그는 사용자와 시스템 모두 데이터베이스에서 실행하는 쿼리 유형에 대한 정보를 추적합니다.

연결 로그와 사용자 로그는 모두 데이터베이스의 시스템 테이블에 저장되는 정보와 일치합니다. 따라서 시스템 테이블을 사용해 동일한 정보를 얻을 수도 있지만 로그 파일이 가져오거나 살펴보는 데 더욱 간단한 메커니즘을 가지고 있습니다. 로그 파일은 데이터베이스 권한이 아닌 Amazon S3 권한을 이용하여 테이블에 대한 쿼리를 실행합니다. 또한 시스템 테이블에 대해 쿼리를 실행하기보다는 로그 파일의 정보를 확인하여 데이터베이스에 미치는 영향을 최소화합니다.

Note

로그 파일은 시스템 로그 테이블인 [STL_USERLOG](#) 및 [STL_CONNECTION_LOG](#)만큼 최신 상태가 아닙니다. 최신 레코드를 제외하고 이보다 오래된 레코드를 로그 파일에 복사하였습니다.

Note

Amazon Redshift Serverless의 경우 [SYS_CONNECTION_LOG](#)는 연결 로그 데이터를 수집합니다. Amazon Redshift Serverless에 대한 감사 로깅 데이터를 수집할 때는 로그 파일로 전송할 수 없으며 CloudWatch로만 전송할 수 있다는 점에 유의하세요.

연결 로그

인증 시도 횟수와 연결 및 차단 정보를 기록합니다. 다음 표는 연결 로그에 기록되는 정보를 설명한 것입니다. 이러한 필드에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [STL_CONNECTION_LOG](#)를 참조하세요. 수집된 Amazon Redshift Serverless의 연결 로그 데이터에 대한 자세한 내용은 [SYS_CONNECTION_LOG](#)를 참조하세요.

열 이름	설명
이벤트	연결 또는 인증 이벤트
recordtime	이벤트 발생 시간
remotehost	원격 호스트의 이름 또는 IP 주소
remoteport	원격 호스트의 포트 번호
pid	쿼리 문과 연결된 프로세스 ID
dbname	데이터베이스 이름.
사용자 이름	사용자 이름.
authmethod	인증 방법
기간	연결 지속 시간(마이크로초)
sslversion	SSL(Secure Sockets Layer) 버전
sslcipher	SSL 암호
mtu	integer
sslcompression	SSL 압축 유형
sslexpansion	SSL 확장 유형
iamauthguid	AWS CloudTrail 요청에 대한 AWS Identity and Access Management(IAM) 인증 ID입니다. 지정된 연결에 사용되는 보안 인증 정보를 생성하기 위한 GetClusterCredentials API 호출의 식별자입니다.
application_name	세션에서 애플리케이션의 초기 이름 또는 업데이트된 이름입니다.
os_version	Amazon Redshift 클러스터에 연결하는 클라이언트 시스템에 있는 운영 체제의 버전입니다.

열 이름	설명
driver_version	서드 파티 SQL 클라이언트 도구에서 Amazon Redshift 클러스터에 연결하는 ODBC 또는 JDBC 드라이버 버전입니다.
plugin_name	Amazon Redshift 클러스터에 연결하는 데 사용되는 플러그인의 이름입니다.
protocol_version	Amazon Redshift 드라이버가 서버와의 연결을 설정할 때 사용하는 내부 프로토콜 버전입니다.
sessionid	현재 세션에 대한 전역적으로 고유한 식별자입니다.
압축	연결에 사용 중인 압축 알고리즘입니다.

사용자 로그

다음과 같이 데이터베이스 사용자의 변경 사항에 대한 세부 정보를 기록합니다.

- 사용자 생성
- 사용자 삭제
- 사용자 변경(이름 변경)
- 사용자 변경(속성 변경)

열 이름	설명
userid	변경 사항이 적용되는 사용자 ID
사용자 이름	변경 사항이 적용되는 사용자의 이름
oldusername	이름 변경 작업의 경우 변경 전 사용자 이름. 그 외 다른 작업의 경우 이 필드는 비어 있습니다.
작업	실행 작업. 유효한 값: • Alter • 생성 • Drop

열 이름	설명
	이름 바꾸기
usecreatedb	true(1)인 경우 사용자에게 데이터베이스 생성 권한이 있다는 것을 의미합니다.
usesuper	true(1)인 경우 사용자가 슈퍼유저임을 의미합니다.
usecatupd	true(1)인 경우 사용자가 시스템 카탈로그를 업데이트할 수 있다는 것을 의미합니다.
valuntil	암호 만료 날짜
pid	프로세스 ID
xid	트랜잭션 ID
recordtime	쿼리 시작 시간(UTC)

사용자 변경 사항에 대한 추가 정보를 찾으려면 [SYS_USERLOG](#) 시스템 뷰를 쿼리합니다. 이 뷰에는 Amazon Redshift Serverless의 로그 데이터가 포함되어 있습니다.

사용자 작업 로그

데이터베이스에서 실행하기 전에 각 쿼리를 기록합니다.

열 이름	설명
recordtime	이벤트 발생 시간
db	데이터베이스 이름.
user	사용자 이름.
pid	쿼리 문과 연결된 프로세스 ID
userid	사용자 ID입니다.
xid	트랜잭션 ID
쿼리	LOG 접두사이며, 이어서 줄 바꿈을 포함한 쿼리 텍스트가 나옵니다.

감사 로그 및 Amazon CloudWatch

Amazon Redshift에서는 감사 로깅이 기본적으로 설정되어 있지 않습니다. 클러스터에서 로깅을 설정하면 Amazon Redshift가 로그를 생성한 후 Amazon S3으로 업로드합니다. 이 로그는 감사 로깅이 사용된 시점부터 현재까지 데이터를 로그에 수집합니다. 각 로깅 업데이트는 이전 로그의 연속입니다.

CloudWatch 또는 Amazon S3에 업로드되는 감사 로깅은 수동 프로세스(옵션)입니다. 시스템 테이블로 업로드되는 로깅은 옵션이 아니며 자동으로 실행됩니다. 시스템 테이블에 로깅에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [시스템 테이블 참조](#)를 참조하세요.

연결 로그, 사용자 로그 및 사용자 작업 로그는 모두 AWS Management Console, Amazon Redshift API Reference 또는 AWS Command Line Interface(AWS CLI)에서 사용됩니다. 사용자 작업 로그의 경우에는 `enable_user_activity_logging` 데이터베이스 파라미터도 활성화해야 합니다. 연결된 파라미터를 제외하고 감사 로깅 기능만 활성화하면 데이터베이스 감사 로그가 사용자 작업 로그를 제외한 연결 로그와 사용자 로그 정보만 기록합니다. `enable_user_activity_logging` 파라미터는 기본적으로 활성화되어 있지 않습니다(`false`). `true`로 설정하여 사용자 활동 로그를 활성화하도록 할 수 있습니다. 자세한 내용은 [Amazon Redshift 파라미터 그룹](#) 섹션을 참조하세요.

CloudWatch에 대한 로깅을 활성화하면 Amazon Redshift가 클러스터 연결, 사용자 및 사용자 활동 로그 데이터를 Amazon CloudWatch Logs 로그 그룹으로 내보냅니다. 스키마 측면에서 로그 데이터는 변경되지 않습니다. CloudWatch는 애플리케이션 모니터링을 위해 구축되었으며, 이를 사용하여 실시간 분석을 수행하거나 조치를 취하도록 설정할 수 있습니다. Amazon CloudWatch Logs를 사용하여 내구성이 우수한 스토리지에 로그 레코드를 저장할 수 있습니다.

Amazon S3 로그 파일을 저장하는 대신 CloudWatch를 사용하여 로그를 보는 것이 좋습니다. 구성이 많이 필요하지 않으며 모니터링 요구 사항에 맞을 수 있습니다. 특히 이미 다른 서비스 및 애플리케이션을 모니터링하는 데 사용하는 경우 더욱 그렇습니다.

Amazon CloudWatch Logs의 로그 그룹 및 로그 이벤트

내보낼 Amazon Redshift 로그를 선택한 후 Amazon CloudWatch Logs에서 로그 이벤트를 모니터링할 수 있습니다. 다음 접두사 아래에 Amazon Redshift Serverless에 대한 새 로그 그룹이 자동으로 생성됩니다. 여기서 `log_type`은 로그 유형을 나타냅니다.

```
/aws/redshift/cluster/<cluster_name>/<log_type>
```

예를 들어 연결 로그 내보내기를 선택하면 로그 데이터는 다음 로그 그룹에 저장됩니다.

```
/aws/redshift/cluster/cluster1/connectionlog
```

로그 이벤트는 로그 스트림을 사용하여 로그 그룹으로 보내집니다. 서버리스 엔드포인트에 대한 로그 이벤트 내에서 정보를 검색하려면 CloudWatch Logs 콘솔, AWS CLI 또는 Amazon CloudWatch Logs API를 사용합니다. 로그 데이터 검색 및 필터링에 대한 자세한 내용은 [필터를 사용하여 로그 이벤트에서 지표 생성](#)을 참조하세요.

CloudWatch에서는 세분성과 유연성을 제공하는 쿼리 구문을 사용하여 로그 데이터를 검색할 수 있습니다. 자세한 내용은 [CloudWatch Logs Insights 쿼리 구문](#)을 참조하세요.

Amazon CloudWatch 감사 로깅으로 마이그레이션

Amazon S3으로 로그를 전송하고 구성을 변경하는 경우(예: 로그를 CloudWatch로 전송하도록 변경) Amazon S3에 남아 있는 로그는 영향을 받지 않습니다. 로그 데이터가 있는 Amazon S3 버킷에서 계속 쿼리할 수 있습니다.

Amazon S3의 로그 파일

Amazon S3에서 Amazon Redshift 로그 파일의 수와 크기는 클러스터의 작업에 커다란 영향을 미칩니다. 다수의 로그를 생성하는 활성 클러스터가 하나 있다고 가정할 때 Amazon Redshift가 더욱 빈번하게 로그 파일을 생성할 수 있습니다. 같은 시간에 다수의 연결 로그가 만들어지는 등 동일한 유형의 작업에 대한 로그 파일이 연이어 생성되기도 합니다.

Amazon Redshift가 Amazon S3을 사용하여 로그를 저장하면 Amazon S3에서 사용하는 스토리지에 대한 요금이 발생합니다. Amazon S3에 대한 로깅을 구성하기 전에 로그 파일을 저장하는 데 걸리는 시간을 계획해야 합니다. 이 과정에서 감사 요구에 따라 로그 파일을 삭제하거나 보관할 수 있는 시기를 결정합니다. 저장 기간에 대한 계획은 규정 준수 또는 규제 요건에 관한 데이터처럼 저장하는 데이터 유형에 따라 크게 달라집니다. Amazon S3 요금에 대한 자세한 내용은 [Amazon Simple Storage Service\(S3\) 요금](#)을 참조하세요.

Amazon S3에 로깅을 활성화할 때의 제한

감사 로깅은 다음과 같은 제약이 있습니다.

- Amazon S3 관리형 키(SSE-S3) 암호화(AES-256)만 사용할 수 있습니다.
- Amazon S3 버킷에 S3 객체 잠금 기능이 해제되어 있어야 합니다.

Amazon Redshift 감사 로깅을 위한 버킷 권한

Amazon S3에 대한 로깅을 설정하면 Amazon Redshift가 로깅 정보를 수집한 후 Amazon S3에 저장된 로그 파일로 업로드합니다. 기존 버킷이나 새 버킷을 사용할 수 있습니다. Amazon Redshift에는 버킷에 대한 다음 IAM 권한이 필요합니다.

- `s3:GetBucketAcl` 이 서비스는 버킷 소유자 식별을 위해 Amazon S3 버킷에 대한 읽기 권한이 필요합니다.
- `s3:PutObject` 이 서비스는 로그 업로드를 위해 객체 업로드 권한이 필요합니다. 또한 로깅을 사용하는 사용자 또는 IAM 역할에 Amazon S3 버킷에 대한 `s3:PutObject` 권한이 있어야 합니다. 로그가 업로드될 때마다 서비스는 현재 버킷 소유자가 로깅 활성화 시점의 버킷 소유자와 일치하는지 확인합니다. 이러한 소유자가 일치하지 않으면 오류가 발생합니다.

감사 로깅을 사용하도록 설정할 때 새 버킷을 생성하는 옵션을 선택하면 버킷에 올바른 권한이 적용됩니다. 하지만 Amazon S3에서 고유한 버킷을 생성하거나 기존 버킷을 사용하는 경우에는 버킷 이름을 포함하는 버킷 정책을 추가해야 합니다. 로그는 서비스 보안 주체 자격 증명을 사용하여 전달됩니다. 대부분의 AWS 리전의 경우 Redshift 서비스 보안 주체 이름인 `redshift.amazonaws.com`을 추가합니다.

버킷 정책은 다음 형식을 사용합니다. `ServiceName`과 `BucketName`은 값의 자리 표시자입니다. 또한 버킷 정책에서 연결된 작업 및 리소스를 지정합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Put bucket policy needed for audit logging",
      "Effect": "Allow",
      "Principal": {
        "Service": "ServiceName"
      },
      "Action": [
        "s3:PutObject",
        "s3:GetBucketAcl"
      ],
      "Resource": [
        "arn:aws:s3:::BucketName",
        "arn:aws:s3:::BucketName/*"
      ]
    }
  ]
}
```

다음 예는 미국 동부(버지니아 북부) 리전과 AuditLogs라는 버킷에 대한 버킷 정책입니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Put bucket policy needed for audit logging",
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": [
        "s3:PutObject",
        "s3:GetBucketAcl"
      ],
      "Resource": [
        "arn:aws:s3:::AuditLogs",
        "arn:aws:s3:::AuditLogs/*"
      ]
    }
  ]
}
```

기본적으로 사용되지 않는 리전('옵트인' 리전이라고도 함)에는 리전별 서비스 보안 주체 이름이 필요합니다. 이러한 경우 서비스 보안 주체 이름에는 redshift.*region*.amazonaws.com 형식으로 리전이 포함됩니다. 아시아 태평양(홍콩) 리전의 *redshift.ap-east-1.amazonaws.com*을 예로 들 수 있습니다. 기본적으로 활성화되어 있지 않은 리전의 목록은 AWS 일반 참조의 [AWS 리전 관리](#)를 참조하세요.

Note

리전별 서비스 보안 주체 이름은 클러스터가 위치한 리전과 일치합니다.

로그 파일에 대한 모범 사례

Redshift가 Amazon S3에 로그 파일을 업로드할 때 대용량 파일을 부분적으로 업로드할 수 있습니다. 멀티파트 업로드에 실패하면 파일의 일부가 Amazon S3 버킷에 남아 있을 수 있습니다. 이로 인해 추

가 스토리지 비용이 발생할 수 있으므로 멀티파트 업로드가 실패할 때 어떤 일이 발생하는지 이해하는 것이 중요합니다. 감사 로그의 멀티파트 업로드에 대한 자세한 설명은 [멀티파트 업로드를 사용하여 객체 업로드 및 복사](#)와 [멀티파트 업로드 중단](#)을 참조하세요.

S3 버킷 생성 및 버킷 정책 추가에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [범용 버킷 생성](#) 및 [Amazon S3의 버킷 정책](#)을 참조하세요.

Amazon Redshift 감사 로깅을 위한 버킷 구조

기본적으로 Amazon Redshift는 다음 버킷 및 객체 구조를 사용하여 Amazon S3 버킷의 로그 파일을 구성합니다.

AWSLogs/AccountID/ServiceName/Region/Year/Month/Day/AccountID_ServiceName_Region

AWSLogs/123456789012/redshift/us-east-1/2013/10/29/123456789012_redshift_us-east-1_mycluster_userlog_2013-10-29T18:01.gz를 예로 들 수 있습니다.

Amazon S3 키 접두사를 제공하는 경우 접두사를 키 시작 부분에 넣습니다.

예를 들어 접두사로 myprefix를 지정한다면 다음과 같습니다. myprefix/AWSLogs/123456789012/redshift/us-east-1/2013/10/29/123456789012_redshift_us-east-1_mycluster_userlog_2013-10-29T18:01.gz

Amazon S3 키 접두사는 512자를 초과할 수 없습니다. 또한 공백(), 큰 따옴표("), 작은 따옴표('), 백슬래시(\)가 포함되어서도 안 됩니다. 허용되지 않는 특수 문자와 제어 문자도 다수 있습니다. 이러한 문자의 16진수 코드는 다음과 같습니다.

- x00~x20
- x22
- x27
- x5c
- x7f 이상

Amazon S3에서 감사 로깅 고려 사항

Amazon Redshift 감사 로깅은 다음과 같은 이유로 중단될 수 있습니다.

- Amazon Redshift에는 Amazon S3 버킷에 로그를 업로드할 권한이 없습니다. 버킷이 올바른 IAM 정책으로 구성되었는지 확인합니다. 자세한 내용은 [Amazon Redshift 감사 로깅을 위한 버킷 권한](#) 섹션을 참조하세요.
- 버킷 소유자가 변경된 경우. Amazon Redshift가 로그를 업로드할 때는 버킷 소유자가 로깅 사용 시점의 소유자와 동일한지 확인합니다. 버킷 소유자가 바뀐 경우에는 감사 로깅에 사용할 버킷이 다른 것으로 구성될 때까지 Amazon Redshift가 로그를 업로드하지 못합니다.
- 버킷을 찾을 수 없는 경우. Amazon S3에서 버킷이 삭제되면 Amazon Redshift가 로그를 업로드할 수 없습니다. 이때 버킷을 다시 생성하거나, Amazon Redshift를 구성하여 로그를 다른 버킷으로 업로드해야 합니다.

AWS CloudTrail을 사용한 API 직접 호출

Amazon Redshift는 Amazon Redshift에서 사용자, 역할 또는 AWS 서비스가 수행한 작업에 대한 레코드를 제공하는 서비스인 AWS CloudTrail과 통합됩니다. CloudTrail은 Amazon Redshift에 대한 모든 API 호출을 이벤트로 캡처합니다. Amazon Redshift와 AWS CloudTrail 통합에 대한 자세한 내용은 [CloudTrail을 사용한 로깅](#)을 참조하십시오.

CloudTrail은 Amazon Redshift 데이터베이스 감사 로깅과 별도로 또는 추가로 사용할 수 있습니다.

CloudTrail에 대한 자세한 내용은 [AWS CloudTrail 사용 설명서](#)를 참조하세요.

감사 로깅 활성화

감사 로그 데이터를 내보내도록 Amazon Redshift Redshift를 구성합니다. 로그는 CloudWatch로 내보내거나 Amazon S3 버킷에 파일로 내보낼 수 있습니다.

콘솔을 사용한 감사 로깅 활성화

콘솔 단계

클러스터에 대한 감사 로깅을 활성화하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 클러스터(Clusters)를 선택한 후 업그레이드할 클러스터를 선택합니다.
3. 속성 탭을 선택합니다. 데이터베이스 구성(Database configurations) 패널에서 편집(Edit)을 선택한 다음 감사 로깅 편집(Edit audit logging)을 선택합니다.

4. 감사 로깅 편집(Edit audit logging) 페이지에서 설정(Turn on)을 선택하고 S3 버킷(S3 bucket) 또는 CloudWatch를 선택합니다. 관리가 쉽고 데이터 시각화에 유용한 기능이 있으므로 CloudWatch를 사용하는 것이 좋습니다.
5. 내보낼 로그를 선택합니다.
6. 선택 사항을 저장하려면 변경 사항 저장(Save changes)을 선택합니다.

보안 로깅

Amazon Redshift가 하나 이상의 AWS Glue Data Catalog 뷰를 참조하는 쿼리를 로깅하는 경우, Amazon Redshift는 해당 쿼리에 대한 메타데이터를 로깅할 때 특정 시스템 테이블 및 뷰 열의 필드를 자동으로 마스킹합니다.

보안 로그 마스킹은 마스킹 조건에 맞는 쿼리를 실행하는 동안 Amazon Redshift가 생성하는 모든 시스템 테이블 및 뷰 항목에 적용됩니다. 다음 표에는 텍스트를 *****로, 숫자를 -1로 마스킹하여 보안 로깅이 적용된 시스템 뷰와 열이 나와 있습니다. 텍스트를 마스킹하는 데 사용되는 별표의 수는 원본 텍스트의 문자 수와 일치하며 최대 6자입니다. 6자를 초과하는 문자열은 6개의 별표로 표시됩니다.

시스템 테이블	중요한 열
SYS_EXTENSIONAL_QUERY_DETAIL	열: source_type, total_partitions, qualified_partitions, scanned_files, returned_rows, returned_bytes, file_format, file_location, external_query_text, warning_message
SYS_EXTENSIONAL_QUERY_ERROR	열: file_location, rowid, column_name, original_value, modified_value, trigger, action, action_value, error_code
SYS_QUERY_DETAIL	열: step_id, step_name, table_id, table_name, input_bytes, input_rows, output_bytes, output_rows, blocks_read, blocks_write, local_read_IO, remote_read_IO, spilled_block_local_disk, spilled_block_remote_disk, step_attribute
SYS_QUERY_HISTORY	열: returned_rows, returned_bytes
STL_AGGR	열: rows, bytes, tbl, type
STL_BCAST	열: rows, bytes, packets

시스템 테이블	중요한 열
STL_DDLTEXT	열: label, text
STL_DELETE	열: rows, tbl
STL_DIST	열: rows, bytes, packets
STL_ERROR	열: file, linenum, context, error
STL_EXPLAIN	열: plannode, info
STL_FILE_SCAN	열: name, line, bytes
STL_HASH	열: rows, bytes, tbl, est_rows
STL_HASHJOIN	열: rows, tbl, num_parts, join_type
STL_INSERT	열: rows, tbl
STL_LIMIT	열: rows
STL_MERGE	열: rows
STL_MERGEJOIN	열: rows, tbl
STL_NESTLOOP	열: rows, tbl
STL_PARSE	열: rows
STL_PLAN_INFO	열: startupcost, totalcost, rows, bytes
STL_PROJECT	열: rows, tbl
STL_QUERY	열: querytxt
STL_QUERY_METRICS	열: max_rows, rows, max_blocks_read, blocks_read, max_blocks_to_disk, blocks_to_disk, max_query_scan_size, query_scan_size
STL_QUERYTEXT	열: text
STL_RETURN	열: rows, bytes

시스템 테이블	중요한 열
STL_S3CLIENT	열: bucket, key, transfer_size, data_size
STL_S3CLIENT_ERROR	열: bucket, key, error, transfer_size
STL_SAVE	열: rows, bytes, tbl
STL_SCAN	열: rows, bytes, fetches, type, tbl, rows_pre_filter, rows_pre_user_filter, perm_table_name, scanned_mega_value
STL_SORT	열: rows, bytes, tbl
STL_SSHCLIENT_ERROR	열: ssh_username, endpoint, command, error
STL_TR_CONFLICT	열: table_id
STL_UNDONE	열: table_id
STL_UNIQUE	열: rows, type, bytes
STL_UTILITYTEXT	열: label, text
STL_WINDOW	열: rows
STV_BLOCKLIST	열: col, tbl, num_values, minvalue, maxvalue
STV_EXEC_STATE	열: rows, bytes, label
STV_INFLIGHT	열: label, text
STV_LOCKS	열: table_id
STV_QUERY_METRICS	열: rows, max_rows, blocks_read, max_blocks_read, max_blocks_to_disk, blocks_to_disk, max_query_scan_size, query_scan_size
STV_STARTUP_RECOVERY_STATE	열: table_id, table_name

시스템 테이블	중요한 열
STV_TBL_PERM	열: id, name, rows, sorted_rows, temp, block_count, query_scan_size
STV_TBL_TRANS	열: id, rows, size
SVCS_EXPLAIN	열: plannode, info
SVCS_PLAN_INFO	열: rows, bytes
SVCS_QUER Y_SUMMARY	열: step, rows, bytes, rate_row, rate_byte, label, rows_pre_filter
SVCS_S3LIST	열: bucket, prefix, retrieved_files, max_file_size, avg_file_size
SVCS_S3LOG	컬럼: message
SVCS_S3PARTITION_S UMMARY	열: total_partitions, qualified_partitions, min_assigned_partitions, max_assigned_partitions, avg_assigned_partitions
SVCS_S3QU ERY_SUMMARY	열: external_table_name, file_format, s3_scanned_rows, s3_scanned_bytes, s3query_returned_rows, s3query_returned_bytes
SVL_QUERY_METRICS	열: step_label, scan_row_count, join_row_count, nested_loop_join_row_count, return_row_count, spectrum_scan_row_count, spectrum_scan_size_mb
SVL_QUERY _METRICS_SUMMARY	열: step_label, scan_row_count, join_row_count, nested_loop_join_row_count, return_row_count, spectrum_scan_row_count, spectrum_scan_size_mb
SVL_QUERY_REPORT	열: rows, bytes, label, rows_pre_filter
SVL_QUERY _SUMMARY	열: rows, bytes, rows_pre_filter
SVL_S3LIST	열: bucket, prefix, retrieved_files, max_file_size, avg_file_size
SVL_S3LOG	컬럼: message
SVL_S3PARTITION	열: rows, bytes, label, rows_pre_filter

시스템 테이블	중요한 열
SVL_S3PARTITION_SUMMARY	열: total_partitions, qualified_partitions, min_assigned_partitions, max_assigned_partitions, avg_assigned_partitions
SVL_S3QUERY	열: external_table_name, file_format, s3_scanned_rows, s3_scanned_bytes, s3query_returned_rows, s3query_returned_bytes, files
SVL_S3QUERY_SUMMARY	열: external_table_name, file_format, s3_scanned_rows, s3_scanned_bytes, s3query_returned_rows, s3query_returned_bytes
SVL_S3RETRIES	열: file_size, location, message
SVL_SPECTRUM_SCAN_ERROR	열: location, rowid, colname, original_value, modified_value, trigger, action, action_value, error_code
SVL_STATEMENTTEXT	열: type, text
SVL_STORED_PROC_CALL	열: querytxt
SVL_STORED_PROC_MESSAGES	열: message, linenum, querytext
SVL_UDF_LOG	열: message, funcname
SVV_DISKUSAGE	열: name, col, tbl, blocknum, num_values, minvalue, maxvalue
SVV_QUERY_STATE	열: rows, bytes, label
SVV_TABLE_INFO	열: table_id, table
SVV_TRANSACTIONS	열: relation

시스템 테이블 및 뷰에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [시스템 테이블 및 뷰 참조](#)를 참조하세요. 쿼리 결과를 동적으로 마스킹하는 Amazon Redshift의 기능에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [동적 데이터 마스킹](#)을 참조하세요.

Amazon Redshift를 사용하여 AWS Glue Data Catalog에서 뷰를 생성하는 방법에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [AWS Glue Data Catalog 뷰](#)를 참조하세요.

CloudTrail을 사용하여 로깅

Amazon Redshift, 데이터 공유, Amazon Redshift Serverless, Amazon Redshift Data API 및 쿼리 에디터 v2는 모두 AWS CloudTrail에 통합됩니다. CloudTrail은 Amazon Redshift에서 사용자, 역할 또는 AWS 서비스가 수행한 작업의 레코드를 제공하는 서비스입니다. CloudTrail은 Amazon Redshift에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처된 호출에는 Redshift 콘솔의 호출과 Redshift 작업에 대한 코드 호출이 포함됩니다.

CloudTrail 추적을 생성하면 Redshift 이벤트를 비롯한 CloudTrail 이벤트를 Amazon S3 버킷에 지속적으로 전달할 수 있습니다. 추적을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 특정 사항을 확인할 수 있습니다. 여기에는 Redshift로 수행한 요청, 요청이 이루어진 IP 주소, 요청 주체, 요청 시점 및 추가 세부 정보가 포함됩니다.

CloudTrail은 Amazon Redshift 데이터베이스 감사 로깅과 별도로 또는 추가로 사용할 수 있습니다.

CloudTrail에 대한 자세한 설명은 [AWS CloudTrail 사용자 가이드](#)를 참조하십시오.

CloudTrail의 정보

CloudTrail은 계정 생성 시 AWS 계정에서 켜집니다. 활동이 발생하면 해당 활동이 Event history(이벤트 기록) 내의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서에서 [CloudTrail 이벤트 기록을 사용하여 이벤트 보기](#)를 참조하세요.

Redshift의 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려는 경우 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송합니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 지역에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 지역의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 작업하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 AWS CloudTrail User Guide의 다음 섹션을 참조하세요.

- [트레일 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에서 Amazon SNS 알림 구성](#)
- [여러 리전으로부터 CloudTrail 로그 파일 받기](#) 및 [여러 계정으로부터 CloudTrail 로그 파일 받기](#)

모든 Amazon Redshift, Amazon Redshift Serverless, Data API, 데이터 공유 및 쿼리 에디터 v2 작업은 CloudTrail이 로깅합니다. 예를 들어 AuthorizeDatashare, CreateNamespace, ExecuteStatement 및 CreateConnection 작업을 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 사용자 자격 증명으로 했는지 여부
- 역할 또는 페더레이션 사용자에 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지.

자세한 내용은 AWS CloudTrail User Guide의 [CloudTrail userIdentity Element](#)를 참조하세요.

로그 파일 항목

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있도록 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

Amazon Redshift 데이터 공유 예제

다음 예제는 AuthorizeDataShare 작업을 보여주는 CloudTrail 로그 항목입니다.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AKIAIOSFODNN7EXAMPLE:janedoe",
    "arn": "arn:aws:sts::111122223333:user/janedoe",
    "accountId": "111122223333",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAIOSFODNN7EXAMPLE:janedoe",
        "arn": "arn:aws:sts::111122223333:user/janedoe",
        "accountId": "111122223333",
        "userName": "janedoe"
      }
    }
  }
}
```



```

    },
    "attributes": {
      "creationDate": "2021-08-02T23:40:45Z",
      "mfaAuthenticated": "false"
    }
  },
  "eventTime": "2021-08-02T23:40:58Z",
  "eventSource": "redshift.amazonaws.com",
  "eventName": "AuthorizeDataShare",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "3.227.36.75",
  "userAgent": "aws-cli/1.18.118 Python/3.6.10
Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 boto3/1.17.41",
  "requestParameters": {
    "dataShareArn": "arn:aws:redshift:us-
east-1:111122223333:datashare:4c64c6ec-73d5-42be-869b-b7f7c43c7a53/testshare",
    "consumerIdentifier": "555555555555"
  },
  "responseElements": {
    "dataShareArn": "arn:aws:redshift:us-
east-1:111122223333:datashare:4c64c6ec-73d5-42be-869b-b7f7c43c7a53/testshare",
    "producerNamespaceArn": "arn:aws:redshift:us-
east-1:123456789012:namespace:4c64c6ec-73d5-42be-869b-b7f7c43c7a53",
    "producerArn": "arn:aws:redshift:us-
east-1:111122223333:namespace:4c64c6ec-73d5-42be-869b-b7f7c43c7a53",
    "allowPubliclyAccessibleConsumers": true,
    "dataShareAssociations": [
      {
        "consumerIdentifier": "555555555555",
        "status": "AUTHORIZED",
        "createdDate": "Aug 2, 2021 11:40:56 PM",
        "statusChangeDate": "Aug 2, 2021 11:40:57 PM"
      }
    ]
  },
  "requestID": "87ee1c99-9e41-42be-a5c4-00495f928422",
  "eventID": "03a3d818-37c8-46a6-aad5-0151803bdb09",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"

```

}

Amazon Redshift Serverless 예제

Amazon Redshift Serverless는 Amazon Redshift Serverless에서 수행한 작업의 기록을 제공하기 위해 AWS CloudTrail과 통합됩니다. CloudTrail은 Amazon Redshift Serverless에 대한 모든 API 호출을 이벤트로 캡처합니다. Amazon Redshift Serverless 기능에 대한 자세한 내용은 [Amazon Redshift Serverless 기능 개요](#)를 참조하십시오.

다음은 CreateNamespace 작업을 설명하는 CloudTrail 로그 항목을 보여 주는 예시입니다.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AAKEOFPINEXAMPLE",
    "arn": "arn:aws:sts::111111111111:assumed-role/admin/admin",
    "accountId": "111111111111",
    "accessKeyId": "AAKEOFPINEXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AAKEOFPINEXAMPLE",
        "arn": "arn:aws:iam::111111111111:role/admin",
        "accountId": "111111111111",
        "userName": "admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-03-21T20:51:58Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-03-21T23:15:40Z",
  "eventSource": "redshift-serverless.amazonaws.com",
  "eventName": "CreateNamespace",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "56.23.155.33",
  "userAgent": "aws-cli/2.4.14 Python/3.8.8 Linux/5.4.181-109.354.amzn2int.x86_64 exe/x86_64.amzn.2 prompt/off command/redshift-serverless.create-namespace",
  "requestParameters": {
    "adminUserPassword": "HIDDEN_DUE_TO_SECURITY_REASONS",
```

```

    "adminUsername": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "dbName": "dev",
    "namespaceName": "testnamespace"
  },
  "responseElements": {
    "namespace": {
      "adminUsername": "HIDDEN_DUE_TO_SECURITY_REASONS",
      "creationDate": "Mar 21, 2022 11:15:40 PM",
      "defaultIamRoleArn": "",
      "iamRoles": [],
      "logExports": [],
      "namespaceArn": "arn:aws:redshift-serverless:us-east-1:111111111111:namespace/befa5123-16c2-4449-afca-1d27cb40fc99",
      "namespaceId": "8b726a0c-16ca-4799-acca-1d27cb403599",
      "namespaceName": "testnamespace",
      "status": "AVAILABLE"
    }
  },
  "requestID": "ed4bb777-8127-4dae-aea3-bac009999163",
  "eventID": "1dbee944-f889-4beb-b228-7ad0f312464",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111111111111",
  "eventCategory": "Management",
}

```

Amazon Redshift Data API 예제

다음은 ExecuteStatement 작업을 설명하는 CloudTrail 로그 항목을 보여 주는 예시입니다.

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AKIAIOSFODNN7EXAMPLE:janedoe",
    "arn": "arn:aws:sts::123456789012:user/janedoe",
    "accountId": "123456789012",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "userName": "janedoe"
  },
  "eventTime": "2020-08-19T17:55:59Z",
  "eventSource": "redshift-data.amazonaws.com",

```

```

    "eventName": "ExecuteStatement",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-cli/1.18.118 Python/3.6.10
Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 boto3/1.17.41",
    "requestParameters": {
        "clusterIdentifier": "example-cluster-identifier",
        "database": "example-database-name",
        "dbUser": "example_db_user_name",
        "sql": "****OMITTED****"
    },
    "responseElements": {
        "clusterIdentifier": "example-cluster-identifier",
        "createdAt": "Aug 19, 2020 5:55:58 PM",
        "database": "example-database-name",
        "dbUser": "example_db_user_name",
        "id": "5c52b37b-9e07-40c1-98de-12ccd1419be7"
    },
    "requestID": "00c924d3-652e-4939-8a7a-cd0612eeb8ac",
    "eventID": "c1fb7076-102f-43e5-9ec9-40820bcc1175",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}

```

다음 예는 맥등성에 사용된 `clientToken`을 보여주는 `ExecuteStatement` 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.

```

{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE:janedoe",
        "arn": "arn:aws:sts::123456789012:user/janedoe",
        "accountId": "123456789012",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "janedoe"
    },
    "eventTime": "2020-08-19T17:55:59Z",
    "eventSource": "redshift-data.amazonaws.com",
    "eventName": "ExecuteStatement",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",

```

```

    "userAgent": "aws-cli/1.18.118 Python/3.6.10
Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 boto3/1.17.41",
    "requestParameters": {
        "clusterIdentifier": "example-cluster-identifier",
        "database": "example-database-name",
        "dbUser": "example_db_user_name",
        "sql": "****OMITTED****",
        "clientToken": "32db2e10-69ac-4534-b3fc-a191052616ce"
    },
    "responseElements": {
        "clusterIdentifier": "example-cluster-identifier",
        "createdAt": "Aug 19, 2020 5:55:58 PM",
        "database": "example-database-name",
        "dbUser": "example_db_user_name",
        "id": "5c52b37b-9e07-40c1-98de-12ccd1419be7"
    },
    "requestID": "00c924d3-652e-4939-8a7a-cd0612eeb8ac",
    "eventID": "c1fb7076-102f-43e5-9ec9-40820bcc1175",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}

```

Amazon Redshift 쿼리 에디터 v2 예제

다음은 CreateConnection 작업을 설명하는 CloudTrail 로그 항목을 보여 주는 예시입니다.

```

{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AAKE0FPINEXAMPLE:session",
        "arn": "arn:aws:sts::123456789012:assumed-role/MyRole/session",
        "accountId": "123456789012",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AAKE0FPINEXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/MyRole",
                "accountId": "123456789012",
                "userName": "MyRole"
            }
        }
    },

```

```

        "webIdFederationData": {},
        "attributes": {
            "creationDate": "2022-09-21T17:19:02Z",
            "mfaAuthenticated": "false"
        }
    },
    "eventTime": "2022-09-21T22:22:05Z",
    "eventSource": "sqlworkbench.amazonaws.com",
    "eventName": "CreateConnection",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "192.2.0.2",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:102.0)
    Gecko/20100101 Firefox/102.0",
    "requestParameters": {
        "password": "****",
        "databaseName": "****",
        "isServerless": false,
        "name": "****",
        "host": "redshift-cluster-2.c8robpbxvbf9.ca-central-1.redshift.amazonaws.com",
        "authenticationType": "****",
        "clusterId": "redshift-cluster-2",
        "username": "****",
        "tags": {
            "sqlworkbench-resource-owner": "AAKEOFPINEXAMPLE:session"
        }
    },
    "responseElements": {
        "result": true,
        "code": "",
        "data": {
            "id": "arn:aws:sqlworkbench:ca-central-1:123456789012:connection/ce56b1be-
            dd65-4bfb-8b17-12345123456",
            "name": "****",
            "authenticationType": "****",
            "databaseName": "****",
            "secretArn": "arn:aws:secretsmanager:ca-
            central-1:123456789012:secret:sqlworkbench!7da333b4-9a07-4917-b1dc-12345123456-qTCoFm",
            "clusterId": "redshift-cluster-2",
            "dbUser": "****",
            "userSettings": "****",
            "recordDate": "2022-09-21 22:22:05",
            "updatedAt": "2022-09-21 22:22:05",
            "accountId": "123456789012",

```

```

    "tags": {
      "sqlworkbench-resource-owner": "AAKE0FPINEXAMPLE:session"
    },
    "isServerless": false
  }
},
"requestID": "9b82f483-9c03-4cdd-bb49-a7009e7da714",
"eventID": "a7cdd442-e92f-46a2-bc82-2325588d41c3",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

AWS CloudTrail 로그의 Amazon Redshift 계정 ID

Amazon Redshift가 다른 AWS 서비스를 호출하면 Amazon Redshift에 속하는 계정 ID로 호출이 기록됩니다. 사용자의 계정 ID로 기록되지는 않습니다. 예를 들어 Amazon Redshift가 CreateGrant, Decrypt, Encrypt, RetireGrant 등의 AWS Key Management Service(AWS KMS) 작업을 호출하여 클러스터의 암호화를 관리한다고 가정합니다. 이 경우 호출은 AWS CloudTrail에 의해 Amazon Redshift 계정 ID를 사용하여 기록됩니다.

Amazon Redshift는 다른 AWS 서비스를 호출할 때 다음 표의 계정 ID를 사용합니다.

리전	리전	계정 ID
미국 동부(버지니아 북부) 리전	us-east-1	368064434614
US East (Ohio) Region	us-east-2	790247189693
US West (N. California) Region	us-west-1	703715109447
미국 서부(오레곤) 리전	us-west-2	473191095985
아프리카(케이프타운) 리전	af-south-1	420376844563
Asia Pacific (Hong Kong) Region	ap-east-1	651179539253
아시아 태평양(하이데라바드) 리전	ap-south-2	297058826802

리전	리전	계정 ID
Asia Pacific (Jakarta) Region	ap-southeast-3	623197973179
아시아 태평양(말레이시아) 리전	ap-southeast-5	590184011157
아시아 태평양(멜버른) 리전	ap-southeast-4	945512339897
Asia Pacific (Mumbai) Region	ap-south-1	408097707231
아시아 태평양(뉴질랜드)	ap-southeast-6	730335362089
아시아 태평양(오사카) 리전	ap-northeast-3	398671365691
Asia Pacific (Seoul) Region	ap-northeast-2	713597048934
아시아 태평양(싱가포르) 리전	ap-southeast-1	960118270566
아시아 태평양(시드니) 리전	ap-southeast-2	485979073181
아시아 태평양(타이베이)	ap-east-2	211125526206
아시아 태평양(태국) 리전	ap-southeast-7	767397930036
아시아 태평양(도쿄) 리전	ap-northeast-1	615915377779
캐나다(중부) 리전	ca-central-1	764870610256
캐나다 서부(캘거리) 리전	ca-west-1	830903446466
Europe (Frankfurt) Region	eu-central-1	434091160558
Europe (Ireland) Region	eu-west-1	246478207311
Europe (London) Region	eu-west-2	885798887673
유럽(밀라노) 리전	eu-south-1	041313461515
유럽(파리) 리전	eu-west-3	694668203235
유럽(스페인) 리전	eu-south-2	028811157404

리전	리전	계정 ID
유럽(스톡홀름) 리전	eu-north-1	553461782468
유럽(취리히) 리전	eu-central-2	668912161003
Israel (Tel Aviv) Region	il-central-1	901883065212
멕시코(중부) 리전	mx-central-1	058264411980
Middle East (Bahrain) Region	me-south-1	051362938876
중동(UAE) 리전	me-central-1	595013617770
남아메리카(상파울루) 리전	sa-east-1	392442076723

다음은 Amazon Redshift가 호출한 AWS KMS Decrypt 작업의 CloudTrail 로그 항목을 보여주는 예입니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AR0AI5QPCMKLTL4VHFCYY:i-0f53e22dbe5df8a89",
    "arn": "arn:aws:sts::790247189693:assumed-role/prod-23264-role-wp/i-0f53e22dbe5df8a89",
    "accountId": "790247189693",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2017-03-03T16:24:54Z"
      }
    },
    "sessionIssuer": {
      "type": "Role",
      "principalId": "AR0AI5QPCMKLTL4VHFCYY",
      "arn": "arn:aws:iam::790247189693:role/prod-23264-role-wp",
      "accountId": "790247189693",
      "userName": "prod-23264-role-wp"
    }
  }
}
```

```

    },
    "eventTime": "2017-03-03T17:16:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "52.14.143.61",
    "userAgent": "aws-internal/3",
    "requestParameters": {
      "encryptionContext": {
        "aws:redshift:createtime": "20170303T1710Z",
        "aws:redshift:arn": "arn:aws:redshift:us-east-2:123456789012:cluster:my-dw-
instance-2"
      }
    },
    "responseElements": null,
    "requestID": "30d2fe51-0035-11e7-ab67-17595a8411c8",
    "eventID": "619bad54-1764-4de4-a786-8898b0a7f40c",
    "readOnly": true,
    "resources": [
      {
        "ARN": "arn:aws:kms:us-east-2:123456789012:key/f8f4f94f-e588-4254-
b7e8-078b99270be7",
        "accountId": "123456789012",
        "type": "AWS::KMS::Key"
      }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012",
    "sharedEventID": "c1daefea-a5c2-4fab-b6f4-d8eaa1e522dc"
  }
}

```

Amazon Redshift의 규정 준수 확인

서드 파티 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 Amazon Redshift의 보안 및 규정 준수를 평가합니다. 여기에는 SOC, PCI, FedRAMP, HIPAA 등이 포함됩니다.

특정 규정 준수 프로그램 범위에 속하는 AWS 서비스의 목록은 [규정 준수 프로그램 제공 AWS 범위 내 서비스](#)를 참조하세요. 일반 정보는 [AWS 규정 준수 프로그램](#)을 참조하세요.

AWS Artifact를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 [AWS Artifact의 보고서 다운로드](#)를 참조하세요.

Amazon Redshift 사용 시 귀하의 규정 준수 책임은 데이터의 민감도, 조직의 규정 준수 목표, 관련 법률과 규정에 따라 결정됩니다. Amazon Redshift 사용 시 HIPAA, PCI 또는 FedRAMP와 같은 표준을 준수해야 하는 경우 다음과 같은 AWS의 도움말 리소스를 활용하세요.

- [보안 및 규정 준수 빠른 시작 안내서](#)에서는 아키텍처 고려 사항에 대해 설명하고 보안 및 규정 준수에 중점을 둔 기본 환경을 AWS에 배포하기 위한 단계를 안내합니다.
- [HIPAA 보안 및 규정 준수에 맞는 아키텍처 설계 백서](#)는 기업에서 AWS를 사용하여 HIPAA를 준수하는 애플리케이션을 생성하는 방법을 설명합니다.
- [AWS 규정 준수 리소스](#), 워크북 및 안내서는 특정 산업 및 위치에 적용할 수 있습니다.
- AWS 서비스인 [AWS Config](#)로 리소스 구성이 내부 관행, 업계 지침 및 규정을 준수하는 정도를 평가할 수 있습니다.
- AWS 서비스인 [AWS Security Hub CSPM](#)은 보안 업계 표준 및 모범 사례 준수 여부를 확인하는 데 도움이 되는 AWS 내 보안 상태에 대한 포괄적인 관점을 제공합니다. Security Hub CSPM은 보안 제어를 사용하여 리소스 구성 및 보안 표준을 평가하여 다양한 규정 준수 프레임워크를 준수할 수 있도록 지원합니다. Security Hub CSPM을 사용하여 Amazon Redshift 리소스를 평가하는 방법에 대한 자세한 내용은 AWS Security Hub 사용 설명서의 [Amazon Redshift 제어](#)를 참조하세요.

다음의 규정 준수 및 보안 문서는 Amazon Redshift를 다루고 있으며, 필요 시 AWS Artifact를 통해 다운로드할 수 있습니다. [자세한 내용은 AWS Artifact 단원을 참조하세요.](#)

- C5(Cloud Computing Compliance Controls Catalogue)
- ISO 27001:2013 적용성 보고서(SoA)
- ISO 27001:2013 인증서
- ISO 27017:2015 적용성 보고서(SoA)
- ISO 27017:2015 인증서
- ISO 27018:2015 적용성 보고서(SoA)
- ISO 27018:2014 인증서
- ISO 9001:2015 인증서
- PCI DSS 규정 준수 증명(AOC) 및 책임 요약
- SOC(Service Organization Controls) 1 보고서
- SOC(Service Organization Controls) 2 보고서
- 기밀성에 대한 SOC(Service Organization Controls) 2 보고서

Amazon Redshift의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역(AZ)을 중심으로 구축됩니다. AWS 리전에서는 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크에 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 가용 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성 및 확장성이 뛰어납니다.

거의 모든 AWS 리전은 여러 개의 가용 영역과 데이터 센터를 가지고 있습니다. 내결함성과 짧은 지연 시간을 위해 같은 리전에 있는 여러 가용 영역에 애플리케이션을 배포할 수 있습니다.

데이터 손실이나 애플리케이션 변경 없이 클러스터를 다른 가용 영역으로 이동하려면 클러스터에 대한 재배치를 구성할 수 있습니다. 재배치를 사용하면 클러스터에서 서비스 중단이 발생할 때 영향을 최소화하면서 작업을 계속할 수 있습니다. 클러스터 재배치가 커지면 Amazon Redshift가 일부 상황에서 클러스터를 재배치하도록 선택할 수 있습니다. Amazon Redshift의 재배치에 대한 자세한 내용은 [클러스터 재부팅](#) 섹션을 참조하세요.

가용 영역에서 예기치 않은 이벤트가 발생하는 장애 시나리오에서 여러 가용 영역(다중 AZ) 배포를 구성하여 Amazon Redshift 데이터 웨어하우스가 계속 작동하도록 할 수 있습니다. Amazon Redshift는 단일 엔드포인트를 통해 액세스할 수 있는 두 개의 가용 영역에 동일한 컴퓨팅 리소스를 배포합니다. 전체 가용 영역에 장애가 발생하는 경우 두 번째 가용 영역의 나머지 컴퓨팅 리소스를 계속해서 워크로드를 처리하는 데 사용할 수 있습니다. 다중 AZ 배포에 대한 자세한 내용은 [다중 AZ 배포](#) 단원을 참조하십시오.

AWS 리전 및 가용 영역에 대한 자세한 내용은 [AWS 글로벌 인프라](#)를 참조하십시오. 재해 복구에 Amazon Redshift를 사용하는 방법에 대한 자세한 내용은 [Implement disaster recovery with Amazon Redshift](#)를 참조하세요.

Amazon Redshift의 인프라 보안

관리형 서비스인 Amazon Redshift는 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스와 AWS의 인프라 보호 방법에 대한 자세한 내용은 [AWS 클라우드 보안](#)을 참조하세요. 인프라 보안에 대한 모범 사례를 사용하여 AWS 환경을 설계하려면 보안 원칙 AWS Well-Architected 프레임워크의 [인프라 보호](#)를 참조하세요.

AWS에서 게시한 API 호출을 사용하여 네트워크를 통해 Amazon Redshift에 액세스합니다. 클라이언트는 다음을 지원해야 합니다.

- Transport Layer Security(TLS). TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

네트워크 격리

Amazon VPC 서비스 기반의 Virtual Private Cloud(VPC)는 AWS 클라우드에서 개인적이고 논리적으로 격리된 네트워크입니다. 다음 단계를 수행하여 VPC 내에 Amazon Redshift 클러스터 또는 Redshift Serverless 작업 그룹을 배포할 수 있습니다.

- AWS 리전에서 VPC를 생성합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [Amazon VPC란 무엇입니까?](#)를 참조하세요.
- 프라이빗 VPC 서브넷을 두 개 이상 생성합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [VPC 및 서브넷](#)을 참조하세요.
- Amazon Redshift 클러스터 또는 Redshift Serverless 작업 그룹을 배포합니다. 자세한 내용은 [Redshift 리소스의 서브넷](#) 또는 [작업 그룹 및 네임스페이스](#) 섹션을 참조하세요.

Amazon Redshift 클러스터는 프로비저닝 시 기본적으로 잠금 상태입니다. Amazon Redshift 클라이언트로부터 들어오는 인바운드 네트워크 트래픽을 허용하려면 VPC 보안 그룹을 Amazon Redshift 클러스터에 연결합니다. 자세한 내용은 [Redshift 리소스의 서브넷](#) 섹션을 참조하세요.

특정 IP 주소 범위에서 들어 오고 나가는 트래픽만 허용하려면 VPC에서 보안 그룹을 업데이트합니다. 예를 들어 기업 네트워크에서 들어오고 나가는 트래픽만 허용할 수 있습니다.

Amazon Redshift 클러스터가 태깅된 서브넷과 관련한 네트워크 액세스 제어 목록을 구성하는 동안 각 AWS 리전의 S3 CIDR 범위를 수신 및 송신 규칙 모두의 허용 목록에 추가해야 합니다. 이렇게 하면 Redshift Spectrum, COPY 및 UNLOAD와 같은 S3 기반 작업을 중단 없이 실행할 수 있습니다.

다음 예제 명령은 us-east-1 리전의 Amazon S3에 사용된 모든 IPv4 주소에 대한 JSON 응답을 구문 분석합니다.

```
curl https://ip-ranges.amazonaws.com/ip-ranges.json | jq -r '.prefixes[] |
  select(.region=="us-east-1") | select(.service=="S3") | .ip_prefix'
```

54.231.0.0/17

52.92.16.0/20

52.216.0.0/15

특정 리전의 S3 IP 범위를 가져오는 방법에 대한 지침은 [AWS IP 주소 범위](#)를 참조하세요.

Amazon Redshift는 클러스터를 전용 테넌시 VPC에 배포하도록 지원합니다. 자세한 내용은 Amazon EC2 User Guide의 [Dedicated instances](#)를 참조하세요.

Amazon Redshift 보안 그룹

Amazon Redshift 클러스터를 프로비저닝할 때는 기본적으로 아무도 액세스하지 못하도록 클러스터가 잠깁니다. 이때 다른 사용자에게 Amazon Redshift 클러스터에 대한 인바운드 액세스 권한을 부여하려면 클러스터를 보안 그룹과 연결해야 합니다. EC2-VPC 플랫폼을 사용하는 기존 Amazon VPC 보안 그룹을 사용하거나, 혹은 새로운 보안 그룹을 정의한 후 클러스터와 연결할 수도 있습니다. EC2-VPC 플랫폼을 기반으로 클러스터를 관리하는 방법에 대한 자세한 내용은 [VPC의 Redshift 리소스](#) 단원을 참조하십시오.

인터페이스 VPC 엔드포인트

인터넷을 통해 연결하는 대신 Virtual Private Cloud(VPC)의 VPC 엔드포인트(AWS PrivateLink) 인터페이스를 사용하여 Amazon Redshift 및 Amazon Redshift Serverless API 서비스에 직접 연결할 수 있습니다. Amazon Redshift API 작업에 대한 자세한 내용은 Amazon Redshift API Reference의 [Actions](#)를 참조하세요. Redshift Serverless API 작업에 대한 자세한 내용은 Amazon Redshift Serverless API 참조의 [작업](#)을 참조하세요. AWS PrivateLink에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 VPC 엔드포인트\(AWS PrivateLink\)](#)를 참조하세요. 클러스터 또는 작업 그룹에 대한 JDBC/ODBC 연결은 Amazon Redshift API 서비스의 일부가 아닙니다.

인터페이스 VPC 엔드포인트를 사용하는 경우 VPC와 Amazon Redshift 또는 Redshift Serverless 사이의 통신은 모두 AWS 네트워크에서 수행되므로 더 안전할 수 있습니다. 각 VPC 엔드포인트는 하나 이상의 탄력적 네트워크 인터페이스 및 VPC 서브넷의 프라이빗 IP 주소로 표현됩니다. 탄력적 네트워크 인터페이스에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [탄력적 네트워크 인터페이스](#)를 참조하세요.

인터페이스 VPC 엔드포인트는 VPC를 Amazon Redshift에 직접 연결합니다. 인터넷 게이트웨이, NAT(Network Address Translation) 장치, 가상 사설 네트워크(VPN) 연결 또는 Direct Connect 연결을 사용하지 않습니다. VPC에 있는 인스턴스는 퍼블릭 IP 주소가 없어도 Amazon Redshift API와 통신할 수 있습니다.

2가지 옵션으로 VPC를 통해 Amazon Redshift 또는 Redshift Serverless를 사용할 수 있습니다. 하나는 VPC 내부에 있는 인스턴스에서 연결하는 것입니다. 다른 하나는 Site-to-Site VPN 옵션이나 Direct Connect를 사용하여 사설망을 VPC에 연결하는 것입니다. Site-to-Site VPN 옵션에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [VPN 연결](#)을 참조하세요. Direct Connect에 대한 자세한 내용은 Direct Connect 사용 설명서의 [연결 생성](#)을 참조하세요.

AWS Management Console 또는 AWS Command Line Interface(AWS CLI) 명령을 사용하여 Amazon Redshift에 연결할 인터페이스 VPC 엔드포인트를 생성할 수 있습니다. 자세한 내용은 [인터페이스 엔드포인트 생성](#)을 참조하세요.

인터페이스 VPC 엔드포인트를 생성한 후 엔드포인트에 대한 프라이빗 DNS 호스트 이름을 사용할 수 있습니다. 그렇게 하면 기본 엔드포인트는 다음과 같습니다.

- Amazon Redshift 프로비저닝: `https://redshift.Region.amazonaws.com`
- Amazon Redshift Serverless: `https://redshift-serverless.Region.amazonaws.com`

프라이빗 DNS 호스트 이름을 사용하지 않는 경우 Amazon VPC는 다음 형식으로 사용할 수 있는 DNS 엔드포인트 이름을 제공합니다.

- Amazon Redshift 프로비저닝: `VPC_endpoint_ID.redshift.Region.vpce.amazonaws.com`
- Amazon Redshift Serverless: `VPC_endpoint_ID.redshift-serverless.Region.vpce.amazonaws.com`

자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 VPC 엔드포인트\(AWS PrivateLink\)](#)를 참조하세요.

Amazon Redshift 및 Redshift Serverless는 VPC 내의 모든 [Amazon Redshift API 작업](#)과 [Redshift Serverless API 작업](#)에 대한 직접 호출을 지원합니다.

VPC 엔드포인트 정책을 VPC 엔드포인트에 연결하여 AWS Identity and Access Management(IAM) 보안 주체에 대한 액세스를 제어할 수 있습니다. 보안 그룹을 VPC 엔드포인트와 연결하여 네트워크 트래픽의 소스와 대상을 기반으로 인바운드 및 아웃바운드 액세스를 제어할 수도 있습니다. 예는 IP 주소의 범위입니다. 자세한 내용은 Amazon VPC 사용 설명서의 [VPC 엔드포인트를 통해 서비스에 대한 액세스 제어](#)를 참조하세요.

Amazon Redshift의 VPC 엔드포인트 정책

Amazon Redshift에 대한 VPC 엔드포인트 정책을 생성하여 다음을 지정할 수 있습니다.

- 작업을 수행할 수 있거나 수행할 수 없는 보안 주체
- 수행할 수 있는 작업
- 작업을 수행할 수 있는 리소스

자세한 내용은 Amazon VPC 사용 설명서의 [VPC 엔드포인트를 통해 서비스에 대한 액세스 제어](#)를 참조하세요.

다음으로 VPC 엔드포인트 정책의 예를 찾아볼 수 있습니다.

Amazon Redshift 프로비저닝된 엔드포인트 정책 예제

다음으로 Amazon Redshift 프로비저닝에 대한 VPC 엔드포인트 정책의 예제를 찾아볼 수 있습니다.

예: 지정된 AWS 계정의 모든 액세스를 거부하는 VPC 엔드포인트 정책

다음 VPC 엔드포인트 정책은 AWS 계정 **123456789012**가 이 엔드포인트를 사용하는 리소스에 대한 모든 액세스를 거부합니다.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      }
    }
  ]
}
```


예: 지정된 IAM 역할에 대해서만 VPC 액세스를 허용하는 VPC 엔드포인트 정책

다음 VPC 엔드포인트 정책은 AWS 계정 **123456789012**의 IAM 역할 **redshiftrole**에만 전체 액세스 권한을 허용합니다. 다른 모든 IAM 보안 주체는 엔드포인트를 사용하는 액세스가 거부됩니다.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/redshiftrole"
        ]
      }
    }
  ]
}
```

이것은 예시입니다. 대부분의 사용 사례에서는 특정 작업에 권한을 연결하여 권한 범위를 좁히는 것이 좋습니다.

예: 지정된 IAM 보안 주체(사용자)에 대해서만 VPC 액세스를 허용하는 VPC 엔드포인트 정책

다음 VPC 엔드포인트 정책은 AWS 계정 **123456789012**의 IAM 사용자 **redshiftadmin**에게만 전체 액세스 권한을 허용합니다. 다른 모든 IAM 보안 주체는 엔드포인트를 사용하는 액세스가 거부됩니다.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/redshiftadmin"
        ]
      }
    }
  ]
}
```

이것은 예시입니다. 대부분의 사용 사례에서는 사용자에게 할당하기 전에 역할에 권한을 연결하는 것이 좋습니다. 또한 특정 작업을 사용하여 권한 범위를 좁히는 것이 좋습니다.

예: 읽기 전용 Amazon Redshift 작업을 허용하는 VPC 엔드포인트 정책

다음 VPC 엔드포인트 정책은 AWS 계정 **123456789012**만 지정된 Amazon Redshift 작업을 수행할 수 있도록 허용합니다.

지정된 작업은 Amazon Redshift에 대한 읽기 전용 액세스 권한을 제공합니다. VPC의 다른 모든 작업은 지정된 계정에 대해 거부됩니다. 다른 모든 계정의 액세스도 거부됩니다. Amazon Redshift 작업 목록은 IAM User Guide의 [Actions, Resources, and Condition Keys for Amazon Redshift](#)를 참조하세요.

```
{
  "Statement": [
    {
      "Action": [
        "redshift:DescribeAccountAttributes",
        "redshift:DescribeClusterParameterGroups",
        "redshift:DescribeClusterParameters",
        "redshift:DescribeClusterSecurityGroups",
        "redshift:DescribeClusterSnapshots",
        "redshift:DescribeClusterSubnetGroups",
        "redshift:DescribeClusterVersions",
        "redshift:DescribeDefaultClusterParameters",
        "redshift:DescribeEventCategories",
        "redshift:DescribeEventSubscriptions",
        "redshift:DescribeHsmClientCertificates",
        "redshift:DescribeHsmConfigurations",
        "redshift:DescribeLoggingStatus",
        "redshift:DescribeOrderableClusterOptions",
        "redshift:DescribeQuery",
        "redshift:DescribeReservedNodeOfferings",
        "redshift:DescribeReservedNodes",
        "redshift:DescribeResize",
        "redshift:DescribeSavedQueries",
        "redshift:DescribeScheduledActions",
        "redshift:DescribeSnapshotCopyGrants",
        "redshift:DescribeSnapshotSchedules",
        "redshift:DescribeStorage",
        "redshift:DescribeTable",
        "redshift:DescribeTableRestoreStatus",
        "redshift:DescribeTags",
```

```

        "redshift:FetchResults",
        "redshift:GetReservedNodeExchangeOfferings"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Principal": {
        "AWS": [
            "123456789012"
        ]
    }
}
]
}

```

예: 지정된 클러스터에 대한 액세스를 거부하는 VPC 엔드포인트 정책

다음 VPC 엔드포인트 정책은 모든 계정 및 보안 주체에 대한 전체 액세스 권한을 허용합니다. 동시에 클러스터 ID가 *my-redshift-cluster*인 Amazon Redshift 클러스터에서 수행되는 작업에 대한 AWS 계정 *123456789012*의 액세스를 거부합니다. 클러스터에 대한 리소스 수준 권한을 지원하지 않는 다른 Amazon Redshift 작업은 여전히 허용됩니다. Amazon Redshift 작업 및 해당 리소스 유형 목록은 IAM User Guide의 [Actions, Resources, and Condition Keys for Amazon Redshift](#)를 참조하세요.

```

{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "arn:aws:redshift:us-east-1:123456789012:cluster:my-redshift-cluster",
      "Principal": {
        "AWS": [
            "123456789012"
        ]
      }
    }
  ]
}

```

```
]
}
```

Amazon Redshift Serverless 엔드포인트 정책 예제

다음으로 Redshift Serverless에 대한 VPC 엔드포인트 정책의 예제를 찾아볼 수 있습니다.

예: 읽기 전용 Redshift Serverless 작업을 허용하는 VPC 엔드포인트 정책

다음 VPC 엔드포인트 정책은 AWS 계정 **123456789012**만 지정된 Redshift Serverless 작업을 수행할 수 있도록 허용합니다.

지정된 작업은 Redshift Serverless에 대한 읽기 전용 액세스 권한을 제공합니다. VPC의 다른 모든 작업은 지정된 계정에 대해 거부됩니다. 다른 모든 계정의 액세스도 거부됩니다. Redshift Serverless 작업 목록은 IAM 사용 설명서에서 [Redshift Serverless의 작업, 리소스 및 조건 키](#)를 참조하세요.

```
{
  "Statement": [
    {
      "Action": [
        "redshift-serverless:DescribeOneTimeCredit",
        "redshift-serverless:GetCustomDomainAssociation",
        "redshift-serverless:GetEndpointAccess",
        "redshift-serverless:GetNamespace",
        "redshift-serverless:GetRecoveryPoint",
        "redshift-serverless:GetResourcePolicy",
        "redshift-serverless:GetScheduledAction",
        "redshift-serverless:GetSnapshot",
        "redshift-serverless:GetTableRestoreStatus",
        "redshift-serverless:GetUsageLimit",
        "redshift-serverless:GetWorkgroup"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      }
    }
  ]
}
```

```
]
}
```

예: 지정된 작업 그룹에 대한 액세스를 거부하는 VPC 엔드포인트 정책

다음 VPC 엔드포인트 정책은 모든 계정 및 보안 주체에 대한 전체 액세스 권한을 허용합니다. 동시에 작업 그룹 ID가 *my-redshift-workgroup*인 Amazon Redshift 작업 그룹에서 수행되는 작업에 대한 AWS 계정 *123456789012*의 액세스를 거부합니다. 작업 그룹에 대한 리소스 수준 권한을 지원하지 않는 다른 Amazon Redshift 작업은 여전히 허용됩니다. Redshift Serverless 작업 및 해당 리소스 유형 목록은 IAM 사용 설명서에서 [Redshift Serverless의 작업, 리소스 및 조건 키](#)를 참조하세요.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "arn:aws:redshift-serverless:us-east-1:123456789012:workgroup:my-redshift-workgroup",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      }
    }
  ]
}
```

Amazon Redshift의 구성 및 취약성 분석

AWS가 게스트 운영 체제(OS) 및 데이터베이스 패치 적용, 방화벽 구성, 재해 복구(DR) 등의 기본 보안 태스크를 처리합니다. 공인된 제3자가 이러한 절차를 검토했습니다. 자세한 내용은 [Amazon Redshift의 규정 준수 확인](#), [공동 책임 모델](#) 및 [보안, 자격 증명 및 규정 준수를 위한 모범 사례](#)를 참조하세요.

Amazon Redshift에서는 데이터 웨어하우스에 자동으로 업그레이드 및 패치를 적용하므로 애플리케이션 관리가 아니라 애플리케이션 자체에 더 집중할 수 있습니다. 구성 가능한 유지 관리 기간 동안에 패치 및 업그레이드가 적용됩니다. 자세한 내용은 [유지 관리 기간](#) 단원을 참조하십시오.

Amazon Redshift 쿼리 편집기 v2는 AWS 관리형 애플리케이션입니다. 모든 패치와 업데이트는 필요에 따라 AWS에 의해 적용됩니다.

자격 증명 강화 IAM 역할 세션을 사용하여 Redshift에 연결

IAM Identity Center를 사용하여 Amazon Redshift 클러스터 및 서버리스 작업 그룹에 대한 페더레이션 액세스를 제공할 수 있습니다. 이 접근 방식을 통해 사용자는 Identity Center 자격 증명을 사용하여 인증할 수 있습니다.

Amazon Redshift는 사용자 자격 증명 정보가 포함된 승인된 토큰을 생성하는 `GetIdentityCenterAuthToken` API 작업을 제공합니다. 이러한 API는 프로비저닝된 클러스터와 서버리스 작업 그룹 모두에 사용할 수 있습니다. 토큰을 사용하면 기존 Identity Center 설정을 사용하여 Amazon Redshift 데이터베이스에 원활하게 Single Sign-On 액세스할 수 있습니다.

사전 조건

Amazon Redshift에서 Identity Center 인증을 사용하기 전에 다음이 있는지 확인합니다.

- Identity Center 설정: 계정에 사용자 ID 및 적절한 애플리케이션 할당으로 구성된 IAM Identity Center가 있어야 합니다. 설정 지침은 [IAM Identity Center 설정](#)을 참조하세요.

Important

Redshift에 연결하려면 `redshift:connect` 범위를 사용해야 합니다.

- 자격 증명 강화 자격 증명: 애플리케이션에서는 임베디드 사용자 자격 증명 정보가 포함된 자격 증명 강화 자격 증명을 사용해야 합니다. 자세한 내용은 [자격 증명 강화 IAM 역할 세션 사용](#)을 참조하세요.
- IAM 권한: IAM 역할 또는 사용자에게 `GetIdentityCenterAuthToken` API를 호출하고 지정된 클러스터 또는 작업 그룹에 액세스할 수 있는 권한이 있어야 합니다. 필요한 권한:
 - 프로비저닝된 클러스터의 경우: 클러스터 ARN의 `redshift:GetIdentityCenterAuthToken`(형식: `arn:aws:redshift:region:account:cluster:cluster-name`)
 - 서버리스 작업 그룹의 경우: 작업 그룹 ARN의 `redshift-serverless:GetIdentityCenterAuthToken`(형식: `arn:aws:redshift-serverless:region:account:workgroup/workgroup-name`)
- 호환되는 드라이버: Identity Center 인증 토큰을 지원하는 Amazon Redshift JDBC 또는 ODBC 드라이버를 사용합니다.
- JDBC 드라이버: [Amazon Redshift JDBC 드라이버 버전 2.0 설치 및 구성](#)을 참조하세요.

- ODBC 드라이버: [Amazon Redshift ODBC 드라이버 버전 2.0 설치 및 구성](#)을 참조하세요.

Identity Center 인증 작동 방식

Amazon Redshift에 대한 Identity Center 인증은 다음 워크플로를 사용합니다.

1. 애플리케이션은 임베디드 사용자 자격 증명 정보가 포함된 자격 증명 강화 자격 증명을 사용하여 `GetIdentityCenterAuthToken` API를 직접적으로 호출합니다.
2. Amazon Redshift는 Identity Center 자격 증명을 검증하고 특정 클러스터 또는 작업 그룹으로 범위가 지정된 암호화된 권한 있는 토큰을 생성합니다. 예제 IAM 정책을 참조하세요.
3. 애플리케이션은 이 토큰을 사용하여 지정된 Amazon Redshift 클러스터 또는 작업 그룹에 연결합니다.
4. Amazon Redshift 데이터 플레인인 토큰을 검증하고 Identity Center 애플리케이션 내에서 Identity Center 사용자의 권한을 기반으로 액세스 권한을 부여합니다.

Important

이 API에는 자격 증명 강화 자격 증명が必要です. 자세한 내용은 [자격 증명 강화 IAM 역할 세션 사용](#)을 참조하세요.

자격 증명 강화 자격 증명 없이 API를 직접적으로 호출하면 `UnsupportedOperationFault` 오류가 발생합니다.

GetIdentityCenterAuthToken API 작업

Amazon Redshift는 프로비저닝된 클러스터용과 서버리스 작업 그룹용의 두 가지 개별 `GetIdentityCenterAuthToken` API 작업을 제공합니다. 두 작업의 이름은 동일하지만 대상 리소스 유형에 따라 다른 파라미터를 허용합니다.

프로비저닝된 클러스터에 대한 GetIdentityCenterAuthToken

프로비저닝된 Amazon Redshift 클러스터의 경우 Amazon Redshift 서비스의 `GetIdentityCenterAuthToken` API를 사용하여 승인된 토큰을 생성합니다.

요청 구문


```
{
  "ClusterIds": [ "string" ]
}
```

요청 파라미터

ClusterIds

토큰에 액세스할 수 있는 권한이 부여될 Amazon Redshift 클러스터 식별자 목록입니다. 토큰은 이 목록에 지정된 클러스터로만 인증하는 데 사용할 수 있습니다.

유형: 문자열 배열

길이 제한: 최소 항목은 1입니다. 최대 항목 수 20개.

필수 여부: 예

CLI 예

예: 단일 클러스터에 대해 승인된 토큰 가져오기

```
aws redshift get-identity-center-auth-token \
  --cluster-ids my-redshift-cluster
```

예: 여러 클러스터에 대해 승인된 토큰 가져오기

```
aws redshift get-identity-center-auth-token \
  --cluster-ids my-cluster-1 my-cluster-2
```

서버리스 작업 그룹을 위한 GetIdentityCenterAuthToken

Amazon Redshift Serverless 작업 그룹의 경우 Amazon Redshift Serverless 서비스의 GetIdentityCenterAuthToken API를 사용하여 승인된 토큰을 생성합니다.

요청 구문

```
{
  "WorkgroupNames": [ "string" ]
}
```

요청 파라미터

WorkgroupNames

토큰에 액세스할 수 있는 권한이 부여될 Amazon Redshift Serverless 작업 그룹 이름의 목록입니다. 토큰은 이 목록에 지정된 작업 그룹으로만 인증하는 데 사용할 수 있습니다.

유형: 문자열 배열

길이 제한: 최소 항목은 1입니다. 최대 항목 수 20개.

필수 여부: 예

CLI 예

예: 단일 작업 그룹에 대해 승인된 토큰 가져오기

```
aws redshift-serverless get-identity-center-auth-token \
  --workgroup-names my-workgroup
```

예: 여러 작업 그룹에 대해 승인된 토큰 가져오기

```
aws redshift-serverless get-identity-center-auth-token \
  --workgroup-names workgroup-1 workgroup-2
```

응답 구문

두 API 모두 동일한 응답 구조를 반환합니다.

```
{
  "AuthorizedToken": "string",
  "ExpirationTime": "timestamp"
}
```

응답 파라미터

AuthorizedToken

사용자 자격 증명 정보와 권한 있는 클러스터 또는 작업 그룹 목록이 포함된 암호화된 권한 있는 토큰입니다. 이 토큰은 민감한 데이터로 취급해야 합니다.

유형: 문자열

ExpirationTime

토큰이 만료되는 날짜 및 시간으로, UTC로 표시됩니다. 토큰은 생성 시점으로부터 1시간 동안 유효합니다.

유형: 타임스탬프

응답의 예

```
{
  "AuthorizedToken":
    "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJlJ1c2VyQG4YW1wbGUuY29tIiwiaWF0IjoxNjQwOTk1MjAw
    "ExpirationTime": "2024-01-01T12:00:00Z"
}
```

드라이버 통합

Amazon Redshift 드라이버는 직접 토큰 사용을 통해 Identity Center 인증을 지원합니다.

직접 토큰 사용

GetIdentityCenterAuthToken API를 직접적으로 호출하여 토큰을 얻은 후 SUBJECT_TOKEN 토큰 유형과 함께 IdpTokenAuthPlugin을 사용합니다.

연결 구성:

```
plugin_name = com.amazon.redshift.plugin.IdpTokenAuthPlugin
token_type = SUBJECT_TOKEN
```

```
token = {encrypted_token_from_api_response}
```

Identity Center 인증 플러그인 및 드라이버 구성에 대한 자세한 내용은 [Amazon Redshift 클러스터에 연결](#)을 참조하세요.

Java 코드 예제

Identity Center 인증을 사용하여 연결할 샘플 Java 코드:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

// Get token from GetIdentityCenterAuthToken API
String token = "your_encrypted_token_from_api_response";

// Configure connection properties
Properties props = new Properties();
props.setProperty("user", "your_username");
props.setProperty("plugin_name", "com.amazon.redshift.plugin.IdpTokenAuthPlugin");
props.setProperty("token_type", "SUBJECT_TOKEN");
props.setProperty("token", token);

// Connect to Redshift
String url = "jdbc:redshift://your-cluster.region.redshift.amazonaws.com:5439/your_database";
try (Connection conn = DriverManager.getConnection(url, props)) {
    // Use connection
    System.out.println("Connected successfully!");
} catch (SQLException e) {
    e.printStackTrace();
}
```

IAM 정책 요구 사항

Amazon Redshift에서 Identity Center 인증을 사용하려면 Amazon Redshift 클러스터 및 작업 그룹에 연결하는 데 필요한 표준 권한 외에 특정 IAM 권한이 필요합니다.

API 권한

프로비저닝된 클러스터의 경우 향상된 IAM 역할 세션에 다음이 있어야 합니다.

- 클러스터 ARN의 `redshift:GetIdentityCenterAuthToken`(형식: `arn:aws:redshift:region:account:cluster:cluster-name`)

서버리스 작업 그룹의 경우 향상된 IAM 역할 세션에 다음이 있어야 합니다.

- 작업 그룹 ARN의 `redshift-serverless:GetIdentityCenterAuthToken`(형식: `arn:aws:redshift-serverless:region:account:workgroup/workgroup-name`)

예제 IAM 정책

프로비저닝된 클러스터에 대한 정책 예제:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift:GetIdentityCenterAuthToken"
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:123456789012:cluster:my-cluster"
      ]
    }
  ]
}
```

서버리스 작업 그룹에 대한 정책 예제:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift-serverless:GetIdentityCenterAuthToken"
      ]
    }
  ]
}
```

```

    ],
    "Resource": [
        "arn:aws:redshift-serverless:us-east-1:123456789012:workgroup/my-
workgroup"
    ]
}
]
}

```

여러 리소스에 대한 정책 예제:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift:GetIdentityCenterAuthToken"
      ],
      "Resource": [
        "arn:aws:redshift:*:123456789012:cluster/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "redshift-serverless:GetIdentityCenterAuthToken"
      ],
      "Resource": [
        "arn:aws:redshift-serverless:*:123456789012:workgroup/*"
      ]
    }
  ]
}

```

리전별 가용성

Identity Center 인증은 다음 AWS 리전에서 사용할 수 있습니다.

- 상용 리전: 지원되는 모든 Amazon Redshift 리전
- AWS GovCloud: us-gov-east-1 및 us-gov-west-1에서 사용 가능

- 중국 리전: cn-north-1 및 cn-northwest-1에서 사용 가능

Note

초기 롤아웃 기간 동안 기능 사용 가능 여부는 달라질 수 있습니다.

Identity Center 인증 문제 해결

다음 섹션의 내용으로 Identity Center 인증 관련 문제를 해결할 수 있습니다.

주제

- [API 오류](#)
- [연결 오류](#)

API 오류

GetIdentityCenterAuthToken API는 다음과 같은 오류를 반환할 수 있습니다.

- `UnsupportedOperationFault` – 자격 증명 강화 자격 증명 없이 요청이 이루어졌거나 현재 구성에서 Identity Center 자격 증명 전파가 지원되지 않습니다.
- `ClusterNotFoundFault(redshift:GetIdentityCenterAuthToken)` - 지정된 클러스터 식별자 중 하나 이상이 존재하지 않거나 액세스 권한이 없습니다.
- `WorkgroupNotFoundFault(redshift-serverless:GetIdentityCenterAuthToken)` - 지정된 작업 그룹 이름 중 하나 이상이 없거나 액세스할 권한이 없습니다.
- `InvalidParameterValueException` - 클러스터 또는 작업 그룹의 빈 목록 또는 목록의 항목 20개 이상과 같이 요청 파라미터가 잘못되었습니다.

연결 오류

Identity Center 토큰을 사용하여 Amazon Redshift에 연결할 때 다음 오류가 발생할 수 있습니다.

- 토큰 자격 증명이 만료되었거나 유효하지 않음 - GetIdentityCenterAuthToken API를 사용하여 새 토큰을 생성하고 연결 구성을 업데이트합니다.
- 잘못된 토큰 - 토큰 형식을 확인하고 올바른 클러스터 또는 작업 그룹에 대해 생성되었는지 확인합니다.

- TYPE_INVALID_TOKEN - 연결 파라미터에서 token_type = SUBJECT_TOKEN을 사용하고 있는지 확인합니다.
- ID 제공업체 유형 "awsidc"는 지원되지 않음 - Amazon Redshift 클러스터 또는 작업 그룹이 Identity Center 인증을 지원하고 호환되는 버전으로 업데이트하는지 확인합니다.
- AWS IdC 자격 증명 공급자가 존재하지 않음 - AWS 콘솔을 통해 Amazon Redshift 리소스에 대한 Identity Center 통합을 구성합니다.
- 잘못된 범위. 사용자의 자격 증명이 Amazon Redshift에 연결할 권한이 없음 - 사용자에게 Identity Center에서 적절한 권한이 있고 올바른 리소스에 대해 토큰이 생성되었는지 확인합니다.

네트워킹 작업

Redshift 데이터베이스 연결 사용자 지정과 같은 네트워킹 작업을 수행할 수 있습니다. 보안 또는 기타 목적으로 트래픽을 제어하기 위해 이 작업을 수행할 수 있습니다. Redshift 리소스의 사용자 지정 도메인 이름 설정과 같은 DNS 관련 작업도 수행할 수 있습니다. Amazon Redshift 프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹이 있는 경우 이러한 구성 작업을 사용할 수 있습니다.

주제

- [클라이언트 연결의 사용자 지정 도메인 이름](#)
- [Redshift 관리형 VPC 엔드포인트](#)
- [VPC의 Redshift 리소스](#)
- [Redshift Enhanced VPC Routing으로 네트워크 트래픽 제어](#)

클라이언트 연결의 사용자 지정 도메인 이름

Amazon Redshift 클러스터 및 Amazon Redshift Serverless 작업 그룹에 대해 사용자 지정 URL이라고도 하는 사용자 지정 도메인 이름을 만들 수 있습니다. 이 도메인 이름은 SQL 클라이언트 연결을 엔드포인트로 라우팅하는 읽기 쉬운 DNS 레코드입니다. 언제든지 기존 클러스터 또는 작업 그룹에 대해 구성할 수 있습니다. 다음과 같은 몇 가지 이점이 있습니다.

- 사용자 지정 도메인 이름은 일반적으로 클러스터 이름 또는 작업 그룹 이름과 리전을 포함하는 기본 URL보다 더 단순한 문자열입니다. 기억하고 사용하기가 더 쉽습니다.
- 예를 들어 장애 조치 시 새 클러스터 또는 작업 그룹으로 트래픽을 신속하게 라우팅할 수 있습니다. 이렇게 하면 클라이언트가 다시 연결할 때 구성을 변경할 필요가 없습니다. 중단을 최소화하면서 중앙에서 연결을 다시 라우팅할 수 있습니다.
- 연결 URL에 서버 이름과 같은 개인 정보를 공유하는 것을 피할 수 있습니다. 사용자 지정 URL에서 숨길 수 있습니다.

CNAME을 사용하여 사용자 지정 도메인 이름을 설정하면 Amazon Redshift에서 추가 요금이 부과되지 않습니다. 새 도메인 이름을 만드는 경우 DNS 공급업체로부터 도메인 이름에 대한 요금이 청구될 수 있지만 일반적으로 이 비용은 적습니다.

도메인 이름 등록

사용자 지정 도메인 이름 설정은 몇 가지 작업으로 구성됩니다. 여기에는 도메인 이름을 DNS 공급자에 등록하고 인증서를 생성하는 작업이 포함됩니다. 이러한 작업을 수행한 후에는 Amazon Redshift 콘솔 또는 Amazon Redshift Serverless 콘솔에서 또는 AWS CLI 명령을 사용하여 사용자 지정 도메인 이름을 구성합니다.

Amazon Redshift에서 사용자 지정 도메인 이름을 구성하려면 등록된 인터넷 도메인 이름이 있어야 합니다. Route 53를 사용하거나 타사 도메인 등록 서비스 제공자를 사용하여 인터넷 도메인을 등록할 수 있습니다. 이러한 작업은 Amazon Redshift 콘솔 외부에서 완료합니다. 도메인 등록은 나머지 절차를 완료하여 사용자 지정 도메인을 만들기 위한 사전 조건입니다.

Note

프로비저닝된 클러스터를 사용하는 경우 사용자 지정 도메인 이름을 구성하는 단계를 수행하기 전에 클러스터를 재배포할 수 있도록 설정해야 합니다. 자세한 내용은 [클러스터 재배포](#) 섹션을 참조하세요. Amazon Redshift Serverless에서는 이 단계가 필요하지 않습니다.

사용자 지정 도메인 이름에는 일반적으로 루트 도메인과 하위 도메인 (예: mycluster.example.com)이 포함됩니다. 구성하려면 다음 단계를 수행합니다.

사용자 지정 도메인 이름에 대한 DNS CNAME 항목 생성

1. 루트 도메인을 등록합니다(예: example.com). 경우에 따라 기존 도메인을 사용할 수도 있습니다. 사용자 지정 이름은 특정 문자에 대한 제한이나 기타 이름 유효성 검사에 의해 제한될 수 있습니다. Route 53에 도메인을 등록하는 방법에 대한 자세한 내용은 [새 도메인 등록](#)을 참조하세요.
2. 사용자 지정 도메인 이름이 클러스터 또는 작업 그룹에 대한 Redshift 엔드포인트를 가리키도록 하는 DNS CNAME 레코드를 추가합니다. 엔드포인트는 Redshift 콘솔 또는 Amazon Redshift Serverless 콘솔의 클러스터 또는 작업 그룹 속성에서 찾을 수 있습니다. 클러스터 또는 작업 그룹 속성의 일반 정보에 있는 JDBC URL을 복사합니다. URL은 다음과 같습니다.

- Amazon Redshift 클러스터: redshift-cluster-sample.abc123456.us-east-1.redshift.amazonaws.com
- Amazon Redshift Serverless 작업 그룹: endpoint-name.012345678901.us-east-1-dev.redshift-serverless-dev.amazonaws.com

URL에 JDBC 접두사가 있는 경우 제거하세요.

 Note

각 이름은 고유해야 하며 조직 내에서 사용할 수 있어야 하므로 DNS 레코드는 사용 가능 여부에 따라 달라질 수 있습니다.

제한 사항:

사용자 지정 도메인의 CNAME 레코드 생성에는 몇 가지 제약이 있습니다.

- 동일한 프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹에 대해 사용자 지정 도메인 이름을 여러 개 생성하는 것은 지원되지 않습니다. 단 한 개의 CNAME 레코드만 연결할 수 있습니다.
- CNAME 레코드를 둘 이상의 클러스터 또는 작업 그룹과 연결하는 것은 지원되지 않습니다. 각 Redshift 리소스의 CNAME은 고유해야 합니다.

도메인을 등록하고 CNAME 레코드를 생성한 후 새 인증서 또는 기존 인증서를 선택합니다. 다음 단계는 AWS Certificate Manager를 사용하여 수행합니다.

AWS Certificate Manager에서 사용할 수 있는 관리형 갱신 자격을 충족하는 [DNS 검증이 완료된 인증서](#)를 생성하는 것이 좋습니다. 관리형 갱신이란 만료일이 다가오면 ACM이 인증서를 자동으로 갱신하거나 이메일 알림을 보내는 것입니다. 자세한 내용은 [ACM 인증서의 관리형 갱신](#)을 참조하세요.

도메인 이름에 대한 인증서 요청

Amazon Redshift 또는 Amazon Redshift Serverless는 통신을 안전하게 유지하고 도메인 이름의 소유권을 확인하기 위해 사용자 지정 엔드포인트에 대해 유효한 Secure Sockets Layer(SSL) 인증서를 요구합니다. 안전한 인증서 관리를 위해 AWS KMS key와 함께 AWS Certificate Manager 인증서 관리자 계정을 사용할 수 있습니다. 보안 유효성 검사에는 전체 호스트 이름 확인(sslmode=verify-full)이 포함됩니다.

인증서 갱신은 이메일 유효성 검사가 아닌 DNS 유효성 검사를 선택한 경우에만 Amazon Redshift에서 관리합니다. 이메일 유효성 검사를 사용하는 경우 인증서를 사용할 수 있지만 만료 전에 직접 갱신을 수행해야 합니다. 인증서에 대해 DNS 유효성 검사를 선택하는 것이 좋습니다. AWS Certificate Manager 인증서 관리자에서 가져온 인증서의 만료 날짜를 모니터링할 수 있습니다.

도메인 이름에 대해 ACM에 인증서 요청

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/acm/>에서 ACM 콘솔을 엽니다.
2. 인증서 요청을 선택합니다.
3. 도메인 이름 필드에 사용자 지정 도메인 이름을 입력합니다.

Note

여러 사용자 지정 도메인 레코드에 단일 인증서를 사용하기 위해 인증서 도메인 외에 여러 접두사를 지정할 수 있습니다. 예를 들어, 동일한 인증서로 `one.example.com` 및 `two.example.com`과 같은 추가 레코드 또는 `*.example.com`과 같은 와일드카드 DNS 레코드를 사용할 수 있습니다.

4. [Review and request]를 선택합니다.
5. [Confirm and request]를 선택합니다.
6. 요청이 유효하려면 ACM이 인증서를 발급하기 전에 인터넷 도메인의 등록 소유자가 요청에 동의해야 합니다. 단계를 완료하면 ACM 콘솔에서 상태가 발급됨으로 표시되는지 확인합니다.

사용자 지정 도메인 구성

Amazon Redshift 또는 Amazon Redshift Serverless 콘솔을 사용하여 사용자 지정 도메인 URL을 만들 수 있습니다. 구성하지 않은 경우 사용자 지정 도메인 이름 속성은 일반 정보 아래에 대시(-)로 표시됩니다. CNAME 레코드와 인증서를 만든 후에는 클러스터 또는 작업 그룹에 대한 사용자 지정 도메인 이름을 연결합니다.

사용자 지정 도메인 연결을 생성하려면 다음 IAM 권한이 필요합니다.

- `redshift:CreateCustomDomainAssociation` — ARN을 추가하여 특정 클러스터에 대한 권한을 제한할 수 있습니다.
- `redshiftServerless:CreateCustomDomainAssociation` - 작업 그룹의 ARN을 추가하여 해당 작업 그룹에 대한 권한을 제한할 수 있습니다.
- `acm:DescribeCertificate`

가장 좋은 방법은 권한 정책을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에 할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.

다음 단계를 수행하여 사용자 지정 도메인 이름을 할당합니다.

1. Redshift 콘솔에서 클러스터를 선택하거나 Amazon Redshift Serverless 콘솔에서 작업 그룹을 선택한 다음 작업 메뉴에서 사용자 지정 도메인 이름 생성을 선택합니다. 대화 상자가 나타납니다.
2. 사용자 지정 도메인 이름을 입력합니다.
3. AWS Certificate Manager에서 ACM 인증서에 대한 ARN을 선택합니다. 변경 내용을 확인합니다. 인증서 생성 단계의 지침에 따라 AWS Certificate Manager를 통해 관리형 갱신이 가능하고 DNS 검증이 완료된 인증서를 선택하는 것이 좋습니다.
4. 클러스터 속성에서 사용자 지정 도메인 이름 및 사용자 지정 도메인 인증서 ARN이 입력한 항목으로 채워졌는지 확인합니다. 사용자 지정 도메인 인증서 만료 날짜도 나열됩니다.

사용자 지정 도메인을 구성한 후에는 새 사용자 지정 도메인에 대해서만 `sslmode=verify-full`을 사용할 수 있습니다. 기본 엔드포인트에서는 작동하지 않습니다. 하지만 `sslmode=verify-ca`와 같은 다른 SSL 모드를 사용하여 기본 엔드포인트에 연결할 수 있습니다.

Note

[클러스터 재배포](#)는 추가 Redshift 네트워킹 기능을 구성하기 위한 사전 조건은 아니라는 점을 기억하시기 바랍니다. 다음 기능을 활성화하기 위해 이 기능을 켤 필요는 없습니다.

- 계정 간 또는 리전 간 VPC에서 Redshift로 연결 - 하나의 AWS Virtual Private Cloud(VPC)에서 Redshift 데이터베이스를 포함하는 다른 VPC로 연결할 수 있습니다. 따라서 데이터베이스에 연결하는 ID에 대한 로컬 VPC 액세스를 제공하지 않고도 서로 다른 계정이나 VPC에서의 클라이언트 액세스 등을 더 쉽게 관리할 수 있습니다. 자세한 내용은 [다른 계정 또는 리전의 Redshift VPC 엔드포인트에서 Amazon Redshift Serverless에 연결](#)을 참조하세요.
- 사용자 지정 도메인 이름 설정 - 이 주제에 설명된 대로 사용자 지정 도메인 이름을 생성하여 엔드포인트 이름을 보다 관련성 있고 간단하게 만들 수 있습니다.

Amazon Redshift 프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹에 연결

사용자 지정 도메인 이름에 연결하려면 프로비저닝된 클러스터

`redshift:DescribeCustomDomainAssociations`의 경우 다음 IAM 권한이 필요합니다. Amazon Redshift Serverless의 경우 권한을 추가할 필요가 없습니다.

가장 좋은 방법은 권한 정책을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에 할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.

콘솔에서 CNAME을 생성하고 클러스터 또는 작업 그룹에 할당하는 단계를 완료한 후에는 SQL 클라이언트의 연결 속성에서 사용자 지정 URL을 제공할 수 있습니다. CNAME 레코드가 생성된 직후에는 DNS 전파로 인해 지연이 있을 수 있습니다.

1. SQL 클라이언트를 엽니다. 예를 들어 SQL/Workbench J를 사용하여 연결 속성을 열고 연결 문자열에 대한 사용자 지정 도메인 이름을 추가할 수 있습니다. 예를 들어 `jdbc:redshift://mycluster.example.com:5439/dev?sslmode=verify-full`입니다. 이 예제에서는 dev가 기본 데이터베이스를 지정합니다.
2. 데이터베이스 사용자의 사용자 이름과 암호를 추가합니다.
3. 연결을 테스트합니다. 특정 테이블과 같은 데이터베이스 리소스를 쿼리하는 기능은 데이터베이스 사용자에게 부여된 권한 또는 할당된 Amazon Redshift 데이터베이스 역할에 부여된 권한에 따라 달라질 수 있습니다.

클러스터 또는 작업 그룹이 VPC에 있는 경우 클러스터 또는 작업 그룹에 연결하려면 공개적으로 액세스할 수 있도록 설정해야 할 수도 있습니다. 이 설정은 네트워크 속성에서 변경할 수 있습니다.

Note

사용자 지정 도메인 이름에 대한 연결은 JDBC, ODBC 및 Python 드라이버에서 지원됩니다.

사용자 지정 도메인이 할당된 클러스터 이름 변경

Note

이 일련의 단계는 Amazon Redshift Serverless 작업 그룹에는 적용되지 않습니다. 작업 그룹 이름은 변경할 수 없습니다.

사용자 지정 도메인 이름이 있는 클러스터의 이름을 바꾸려면 `acm:DescribeCertificate` IAM 권한이 필요합니다.

1. Amazon Redshift 콘솔로 이동하여 이름을 변경하려는 클러스터를 선택합니다. 편집을 선택하여 클러스터 속성을 편집합니다.

2. 클러스터 식별자를 편집합니다. 클러스터의 다른 속성도 변경할 수 있습니다. 변경 사항 저장 (Save changes)을 선택합니다.
3. 클러스터의 이름이 변경된 후에는 DNS 레코드를 업데이트하여 사용자 지정 도메인의 CNAME 항목이 업데이트된 Amazon Redshift 엔드포인트를 가리키도록 변경해야 합니다.

사용자 지정 도메인 연결 설명

이 섹션의 명령을 사용하여 프로비저닝된 특정 클러스터 또는 Amazon Redshift Serverless 작업 그룹과 연결된 사용자 지정 도메인 이름 목록을 가져올 수 있습니다.

다음 권한이 필요합니다.

- 프로비저닝된 클러스터: `redshift:DescribeCustomDomainAssociations`
- Amazon Redshift Serverless 작업 그룹: `redshiftServerless:ListCnameAssociations`

가장 좋은 방법은 권한 정책을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에 할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.

다음은 주어진 Amazon Redshift 클러스터에 대한 사용자 지정 도메인 이름을 나열하는 샘플 명령입니다.

```
aws redshift describe-custom-domain-associations --custom-domain-name customdomainname
```

사용자 지정 도메인 이름을 활성화한 경우 이 명령을 실행하여 클러스터와 연결된 사용자 지정 도메인 이름을 확인할 수 있습니다. 사용자 지정 도메인 연결을 설명하기 위한 CLI 명령에 대한 자세한 내용은 [describe-custom-domain-associations](#)를 참조하세요.

마찬가지로 다음은 주어진 Amazon Redshift Serverless 작업 그룹의 사용자 지정 도메인 이름을 나열하는 샘플 명령을 보여줍니다. 몇 가지 방법으로 수행할 수 있습니다. 사용자 지정 도메인 이름만 제공할 수 있습니다.

```
aws redshift-serverless list-custom-domain-associations --custom-domain-name customdomainname
```

인증서 ARN만 제공하여 연결을 가져올 수도 있습니다.

```
aws redshift-serverless list-custom-domain-associations --custom-domain-certificate-arn certificatearn
```

사용자 지정 도메인 이름을 활성화한 경우 이 명령을 실행하여 작업 그룹과 연결된 사용자 지정 도메인 이름을 확인할 수 있습니다. 명령을 실행하여 사용자 지정 도메인 연결의 속성을 가져올 수도 있습니다. 이렇게 하려면 사용자 지정 도메인 이름과 작업 그룹 이름을 파라미터로 제공해야 합니다. 인증서 ARN, 작업 그룹 이름, 사용자 지정 도메인의 인증서 만료 시간을 반환합니다.

```
aws redshift-serverless get-custom-domain-association --workgroup-name workgroupname --  
custom-domain-name customdomainname
```

Amazon Redshift Serverless에서 사용할 수 있는 CLI 참조 명령에 대한 자세한 내용은 [redshift-serverless](#)를 참조하세요.

사용자 지정 도메인을 다른 인증서와 연결

사용자 지정 도메인 이름에 대한 인증서 연결을 변경하려면 다음 IAM 권한이 필요합니다.

- redshift:ModifyCustomDomainAssociation
- acm:DescribeCertificate

가장 좋은 방법은 권한 정책을 IAM 역할에 연결한 다음 필요에 따라 사용자 및 그룹에 할당하는 것입니다. 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#)를 참조하세요.

다음 명령을 사용하여 사용자 지정 도메인을 다른 인증서와 연결합니다. --custom-domain-name 및 custom-domain-certificate-arn 인수는 필수입니다. 새 인증서의 ARN은 기존 ARN과 달라야 합니다.

```
aws redshift modify-custom-domain-association --cluster-id redshiftcluster --custom-  
domain-name customdomainname --custom-domain-certificate-arn certificatearn
```

다음 샘플은 사용자 지정 도메인을 Amazon Redshift Serverless 작업 그룹의 다른 인증서와 연결하는 방법을 보여줍니다.

```
aws redshift-serverless modify-custom-domain-association --workgroup-  
name redshiftworkgroup --custom-domain-name customdomainname --custom-domain-  
certificate-arn certificatearn
```

클러스터에 연결할 수 있기까지 최대 30초의 지연이 있습니다. 지연의 일부는 Amazon Redshift 클러스터가 속성을 업데이트할 때 발생하며, DNS가 업데이트될 때 약간의 추가 지연이 있습니다. API 및 각 속성 설정에 대한 자세한 내용은 [ModifyCustomDomainAssociation](#)을 참조하세요.

사용자 지정 도메인 삭제

사용자 지정 도메인 이름을 삭제하려면 사용자에게 다음 작업에 대한 권한이 있어야 합니다.

- 프로비저닝된 클러스터: `redshift:DeleteCustomDomainAssociation`
- Amazon Redshift Serverless 작업 그룹:
`redshiftserverless:DeleteCustomDomainAssociation`

콘솔에서

작업 버튼을 선택하고 사용자 지정 도메인 이름 삭제를 선택하여 사용자 지정 도메인 이름을 삭제할 수 있습니다. 이렇게 한 후에도 콘솔에 나열된 엔드포인트를 사용하도록 도구를 업데이트하여 서버에 연결할 수 있습니다.

CLI 명령 사용

다음 샘플은 사용자 지정 도메인 이름을 삭제하는 방법을 보여줍니다. 삭제 작업을 수행하려면 클러스터의 기존 사용자 지정 도메인 이름을 제공해야 합니다.

```
aws redshift delete-custom-domain-association --cluster-id redshiftcluster --custom-domain-name customdomainname
```

다음 샘플은 Amazon Redshift Serverless 작업 그룹의 사용자 지정 도메인 이름을 삭제하는 방법을 보여줍니다. 사용자 지정 도메인 이름은 필수 파라미터입니다.

```
aws redshift-serverless delete-custom-domain-association --workgroup-name workgroupname --custom-domain-name customdomainname
```

자세한 내용은 [DeleteCustomDomainAssociation](#)을 참조하세요.

Redshift 관리형 VPC 엔드포인트

기본적으로 Amazon Redshift 클러스터 또는 Amazon Redshift Serverless 작업 그룹은 Virtual Private Cloud(VPC)에서 프로비저닝됩니다. VPC는 퍼블릭 액세스를 허용하거나 인터넷 게이트웨이, NAT 디바이스 또는 AWS Direct Connect 연결을 설정하여 트래픽을 라우팅할 때 다른 VPC 또는 서브넷에서 액세스할 수 있습니다. 또한 Redshift 관리형 VPC 엔드포인트(AWS PrivateLink 기반)를 설정하여 클러스터 또는 작업에 액세스할 수 있습니다.

클러스터 또는 작업 그룹이 포함된 VPC와 클라이언트 도구를 실행하는 VPC 간의 프라이빗 연결로 Redshift 관리형 VPC 엔드포인트를 설정할 수 있습니다. 클러스터 또는 작업 그룹이 다른 계정에 있는 경우 계정 소유자(부여자)는 연결하는 계정(피부여자)에게 액세스 권한을 부여해야 합니다. 이 접근 방식을 사용하면 퍼블릭 IP 주소를 사용하거나 인터넷을 통해 트래픽을 라우팅하지 않고도 데이터 웨어하우스에 액세스할 수 있습니다.

Redshift 관리형 VPC 엔드포인트를 사용하여 액세스를 허용하는 일반적인 이유는 다음과 같습니다.

- AWS 계정 A는 AWS 계정 B의 VPC가 클러스터 또는 작업 그룹에 액세스할 수 있도록 허용하려고 합니다.
- AWS 계정 A는 AWS 계정 A에도 있는 VPC가 클러스터 또는 작업 그룹에 액세스할 수 있도록 허용하려고 합니다.
- AWS 계정 A는 AWS 계정 A 내의 VPC에 있는 다른 서브넷이 클러스터 또는 작업 그룹에 액세스할 수 있도록 허용하려고 합니다.

Redshift 관리형 VPC 엔드포인트를 설정하여 다른 계정의 클러스터 또는 작업 그룹에 액세스하는 워크플로는 다음과 같습니다.

1. 소유자 계정은 다른 계정에 액세스 권한을 부여하고 피부여자의 AWS 계정 ID와 VPC 식별자(또는 모든 VPC)를 지정합니다.
2. 피부여자 계정은 Redshift 관리형 VPC 엔드포인트를 생성할 권한이 있다는 알림을 받습니다.
3. 피부여자 계정은 Redshift 관리형 VPC 엔드포인트를 생성합니다.
4. 피부여자 계정은 Redshift 관리형 VPC 엔드포인트를 사용하여 소유자 계정의 클러스터 또는 작업 그룹에 액세스할 수 있습니다.

Amazon Redshift 콘솔, AWS CLI 또는 Amazon Redshift API를 사용하여 이를 수행할 수 있습니다.

Redshift 관리형 VPC 엔드포인트 사용 시 고려 사항

Note

RedShift 관리형 VPC 엔드포인트를 생성하거나 수정하려면 AWS 관리형 정책 AmazonRedshiftFullAccess에 지정된 기타 권한 외에도 IAM 정책 내에 `ec2:CreateVpcEndpoint` 또는 `ec2:ModifyVpcEndpoint` 권한이 필요합니다.

Redshift 관리형 VPC 엔드포인트를 사용할 때 다음 사항에 유의합니다.

- 프로비저닝된 클러스터를 사용하는 경우 노드 유형이 RA3이어야 합니다. Amazon Redshift Serverless 작업 그룹은 VPC 엔드포인트를 설정하는 데에도 사용할 수 있습니다.
- 프로비저닝된 클러스터의 경우 클러스터에 클러스터 재배포치 또는 다중 AZ가 활성화되어 있어야 합니다. 클러스터 재배포치를 설정하기 위한 요구 사항에 대한 자세한 내용은 [클러스터 재배포치](#) 섹션을 참조하세요. 다중 AZ 활성화에 대한 자세한 내용은 [새 클러스터 생성 시 다중 AZ 설정](#) 섹션을 참조하세요.
- 보안 그룹을 통해 액세스할 클러스터 또는 작업 그룹이 유효한 포트 범위 5431~5455 및 8191~8215 내에서 사용 가능한지 확인합니다. 기본값은 5439입니다.
- 기존 Redshift 관리형 VPC 엔드포인트와 연결된 VPC 보안 그룹을 수정할 수 있습니다. 다른 설정을 수정하려면 현재 Redshift 관리형 VPC 엔드포인트를 삭제하고 새 엔드포인트를 생성합니다.
- 생성할 수 있는 Redshift 관리형 VPC 엔드포인트 수는 VPC 엔드포인트 할당량으로 제한됩니다.
- Redshift 관리형 VPC 엔드포인트는 인터넷에서 액세스할 수 없습니다. Redshift 관리형 VPC 엔드포인트는 엔드포인트가 프로비저닝된 VPC 내에서 또는 라우팅 테이블 및 보안 그룹에서 허용하는 대로 엔드포인트가 프로비저닝된 VPC와 피어링된 VPC에서만 액세스할 수 있습니다.
- Amazon VPC 콘솔을 사용하여 Redshift 관리형 VPC 엔드포인트를 관리할 수 없습니다.
- 프로비저닝된 클러스터에 대해 Redshift 관리형 VPC 엔드포인트를 만들 때 선택하는 VPC에는 클러스터 서브넷 그룹이 있어야 합니다. 서브넷 그룹을 생성하려면 [클러스터 서브넷 그룹 생성](#) 섹션을 참조하세요.
- 가용 영역이 다운된 경우 Amazon Redshift는 다른 가용 영역에 새로운 탄력적 네트워크 인터페이스를 생성하지 않습니다. 이 경우 새 엔드포인트를 생성해야 할 수도 있습니다.

할당량 및 명명 제약 조건에 대한 자세한 내용은 [Amazon Redshift의 할당량 및 제한](#) 섹션을 참조하세요.

요금에 대한 자세한 정보는 [AWS PrivateLink 요금](#)을 참조하세요.

VPC에 대한 액세스 권한 부여

클러스터 또는 작업 그룹에 액세스하도록 하려는 VPC가 다른 AWS 계정에 있는 경우 소유자(부여자) 계정에서 권한을 부여해야 합니다.

다른 AWS 계정의 VPC가 클러스터 또는 작업 그룹에 액세스할 수 있도록 허용하는 방법

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.

2. 탐색 메뉴에서 클러스터(Clusters)를 선택합니다. Amazon Redshift Serverless의 경우 Serverless 대시보드를 선택합니다.
3. 액세스를 허용하려는 클러스터에서 클러스터 이름을 선택하여 세부 정보를 봅니다. 클러스터에 대한 [Properties(속성)] 탭을 선택합니다.

[부여된 계정(Granted accounts)] 섹션에는 클러스터에 액세스할 수 있는 계정과 해당 VPC가 표시됩니다. Amazon Redshift Serverless 작업 그룹의 경우 작업 그룹을 선택합니다. 부여된 계정은 데이터 액세스 탭에서 확인할 수 있습니다.

4. [액세스 권한 부여(Grant access)]를 선택하여 계정을 추가할 [피부여자 정보(Grantee information)]를 입력하는 양식을 표시합니다.
5. AWS 계정 ID에 액세스 권한을 부여할 계정의 ID를 입력합니다. 특정 VPC 또는 지정된 계정의 모든 VPC에 대한 액세스 권한을 부여할 수 있습니다.
6. [액세스 권한 부여(Grant access)]를 선택하여 액세스 권한을 부여합니다.

Redshift 관리형 VPC 엔드포인트 생성

클러스터 또는 작업 그룹을 소유하거나 관리 액세스 권한이 부여된 경우 이에 대한 Redshift 관리형 VPC 엔드포인트를 생성할 수 있습니다.

Redshift 관리형 VPC 엔드포인트를 생성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 창에서 구성(Configurations)을 선택합니다.

[구성(Configurations)] 페이지에 생성된 Redshift 관리형 VPC 엔드포인트가 표시됩니다. 엔드포인트에 대한 세부 정보를 보려면 해당 이름을 선택합니다. Amazon Redshift Serverless의 경우 작업 그룹을 선택하면 VPC 엔드포인트가 데이터 액세스 탭 아래에 있습니다.

3. [엔드포인트 생성(Create endpoint)]을 클릭하여 추가할 엔드포인트에 대한 정보를 입력할 양식을 표시합니다.
4. 엔드포인트 이름, 12자리 AWS 계정 ID, 엔드포인트가 위치한 Virtual Private Cloud(VPC), 서브넷 및 VPC 보안 그룹의 값을 입력합니다.

서브넷의 서브넷은 Amazon Redshift가 엔드포인트를 배포하는 서브넷과 IP 주소를 정의합니다. Amazon Redshift는 엔드포인트와 연결된 네트워크 인터페이스에 사용할 수 있는 IP 주소가 있는 서브넷을 선택합니다.

VPC 보안 그룹의 보안 그룹 규칙은 엔드포인트에 대해 권한을 부여하는 인바운드 트래픽의 포트, 프로토콜 및 소스를 정의합니다. 선택한 포트에 대해 워크로드가 실행되는 보안 그룹 또는 CIDR 범위를 통한 액세스를 허용합니다.

5. [엔드포인트 생성(Create endpoint)]을 선택하여 엔드포인트를 생성합니다.

엔드포인트가 생성되면 Redshift 관리형 VPC 엔드포인트에 대한 구성 설정의 엔드포인트 URL에 표시된 URL을 통해 클러스터 또는 작업 그룹에 액세스할 수 있습니다.

VPC의 Redshift 리소스

Amazon VPC 서비스를 기반으로 하는 EC2-VPC 플랫폼의 VPC에서 Amazon Redshift 클러스터 또는 Amazon Redshift Serverless 작업 그룹을 시작할 수 있습니다. 자세한 내용은 [EC2를 사용하여 클러스터 생성](#) 섹션을 참조하세요.

Note

클러스터 및 Serverless 작업 그룹을 전용 테넌시 VPC로 실행하는 것은 지원되지 않습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [전용 인스턴스](#)를 참조하세요.

VPC에서 리소스를 프로비저닝할 때는 다음을 수행해야 합니다.

- VPC 정보를 입력합니다.

VPC에서 프로비저닝된 클러스터를 만들 때는 클러스터 서브넷 그룹을 생성하여 VPC 정보를 제공해야 합니다. 이 정보에는 VPC ID 및 VPC의 서브넷 목록이 포함됩니다. 클러스터를 시작할 때 Redshift가 VPC의 서브넷 중 하나에서 프로비저닝할 수 있도록 서브넷 그룹을 제공합니다. Amazon Redshift Serverless를 사용하면 프로세스가 비슷합니다. Serverless 작업 그룹에 직접 서브넷을 할당합니다. 하지만 Serverless의 경우 서브넷 그룹을 만들지 않습니다. Amazon Redshift에서 서브넷 그룹 생성에 대한 자세한 내용은 [Redshift 리소스의 서브넷](#) 섹션을 참조하세요. 자세한 내용은 Amazon VPC 시작 안내서의 [Amazon VPC 시작하기](#) 섹션을 참조하세요.

- 선택적으로 접근성 옵션을 구성합니다.

Amazon Redshift에서 프로비저닝된 클러스터 및 서버리스 작업 그룹은 기본적으로 프라이빗입니다. 프로비저닝된 클러스터 또는 서버리스 작업 그룹을 퍼블릭 액세스가 가능하도록 구성하는 경우, Amazon Redshift는 외부 IP 주소로 탄력적 IP 주소를 사용합니다. 탄력적 IP 주소는 고정 IP 주소입니다. 이를 통해 클라이언트가 연결하는 데 사용하는 IP 주소에 영향을 주지 않고 기본 구성을 변경

할 수 있습니다. 이러한 접근 방식은 재해 복구 같은 상황에 유용할 수 있습니다. 탄력적 IP 주소 생성 여부는 가용 영역 재배포 설정에 따라 다릅니다. 두 가지 옵션이 있습니다.

1. 가용 영역 재배포가 켜져 있고 퍼블릭 액세스를 활성화하려는 경우 탄력적 IP 주소를 지정하지 않습니다. Amazon Redshift에서 관리하는 탄력적 IP 주소가 할당됩니다. AWS 계정과 연결되어 있습니다.
2. 가용 영역 재배포가 꺼져 있고 퍼블릭 액세스를 활성화하려는 경우 Amazon Redshift 클러스터 또는 작업 그룹을 시작하기 전에 Amazon EC2에서 VPC에 대한 탄력적 IP 주소를 생성하도록 선택할 수 있습니다. IP 주소를 생성하지 않는 경우 Amazon Redshift가 VPC에 사용할 구성된 탄력적 IP 주소를 제공합니다. 이러한 탄력적 IP 주소는 Amazon Redshift에서 관리하며 AWS 계정과 연결되어 있지 않습니다.

자세한 내용은 Amazon EC2 사용 설명서에서 [탄력적 IP 주소](#)를 참조하세요.

경우에 따라 VPC에 VPC 내에서 프라이빗 IP 주소를 사용하여 연결하려는 퍼블릭 액세스가 가능한 클러스터가 있을 수 있습니다. 이 경우 다음 VPC 파라미터를 true로 설정하세요.

- DNS resolution
- DNS hostnames

Amazon Redshift Serverless에서는 이러한 방식으로 연결할 수 없다는 점에 유의하세요.

VPC에 퍼블릭 액세스가 가능한 프로비저닝된 클러스터가 있지만 VPC에서 해당 파라미터를 true로 설정하지 않았다고 가정합니다. 이러한 경우 VPC 내에서 생성된 연결은 리소스의 프라이빗 IP 주소 대신 탄력적 IP 주소로 확인됩니다. VPC 내부에서 공개 액세스가 가능한 클러스터에 연결할 때는 위의 파라미터를 true로 설정하여 프라이빗 IP 주소를 사용하는 것이 바람직합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [VPC에서 DNS 사용](#)을 참조하세요.

Note

VPC에 기존 퍼블릭 액세스가 가능한 프로비저닝된 클러스터가 있는 경우 크기를 조정할 때까지 VPC 내의 연결은 계속해서 탄력적 IP 주소를 사용하여 연결합니다. 이는 위의 파라미터가 설정된 경우에도 발생합니다. 새로 생성된 클러스터는 동일한 VPC 내에서 퍼블릭 액세스가 가능한 클러스터에 연결할 때 프라이빗 IP 주소를 사용하는 새로운 동작을 따릅니다.

탄력적 IP 주소는 VPC 외부의 리소스에 액세스하기 위한 외부 IP 주소입니다. 프로비저닝된 클러스터의 경우, Amazon Redshift 콘솔의 노드 IP 주소 아래에 표시되는 퍼블릭 IP 주소 및 프라이빗 IP 주소와는 관련이 없습니다. 퍼블릭 및 프라이빗 클러스터 노드 IP 주소는 클러스터의 퍼블릭 액세스 여

부와 상관없이 표시됩니다. 특정 상황에서만 원격 호스트에서 수신 규칙을 구성하는 데 사용됩니다. 이러한 상황은 SSH(Secure Shell) 연결을 사용하여 Amazon EC2 인스턴스 또는 다른 원격 호스트에서 데이터를 로드할 때 발생합니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [1단계: 클러스터 퍼블릭 키 및 클러스터 노드 IP 주소 검색](#)을 참조하세요.

Note

노드 IP 주소는 Redshift Serverless 작업 그룹에 적용되지 않습니다.

프로비저닝된 클러스터를 탄력적 IP 주소와 연결하는 옵션은 클러스터를 생성하거나 클러스터를 스냅샷에서 복원할 때 사용할 수 있습니다. 경우에 따라 클러스터를 탄력적 IP 주소와 연결하거나 클러스터와 연관된 탄력적 IP 주소를 변경해야 할 때가 있습니다. 클러스터를 생성한 후 탄력적 IP 주소를 연결하려면 먼저 공개적으로 액세스할 수 없도록 클러스터를 업데이트한 다음 공개적으로 액세스할 수 있도록 만들고 동일한 작업에서 탄력적 IP 주소를 추가합니다.

프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹을 퍼블릭 액세스가 가능하도록 설정하고 탄력적 IP 주소를 할당하는 방법에 대한 자세한 내용은 [기본 또는 사용자 지정 보안 그룹 구성을 통한 퍼블릭 액세스 가능성](#) 섹션을 참조하세요.

- VPC 보안 그룹을 연결합니다.

VPC 보안 그룹을 사용하여 인바운드 액세스 권한을 부여합니다. 자세한 내용은 클라이언트와 프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹 간의 인바운드 및 아웃바운드 규칙 구성에 대한 지침을 제공하는 [Amazon Redshift 클러스터에 대한 보안 그룹 통신 설정 구성](#)을 참조하세요. 보안 그룹을 이해하는 데 도움이 되는 또 다른 리소스는 Amazon VPC 사용 설명서의 [VPC 보안](#)입니다.

VPC에서 프로비저닝된 클러스터 또는 Serverless 작업 그룹의 스냅샷 복원

VPC의 클러스터 또는 Serverless 작업 그룹의 스냅샷은 VPC 외부가 아닌 VPC 내부에서만 복원할 수 있습니다. 또한 계정 내 동일한 VPC는 물론이고 다른 VPC에서도 복원할 수 있습니다. 스냅샷 복사에 대한 자세한 내용은 [Amazon Redshift 스냅샷 및 백업](#) 섹션을 참조하세요.

VPC에서 Redshift 프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹 생성

다음은 Virtual Private Cloud(VPC)에서 클러스터 또는 작업 그룹을 배포할 수 있는 방법을 보여 주는 일반적인 단계입니다.

VPC에서 클러스터 또는 Serverless 작업 그룹을 만들려면 다음을 수행하세요.

1. VPC 구성 - 계정에 기본 VPC가 있는 경우 계정의 기본 VPC에서 또는 직접 만든 VPC에서 Redshift 리소스를 만들 수 있습니다. 자세한 내용은 [EC2를 사용하여 클러스터 생성](#) 섹션을 참조하세요. VPC를 만들려면 Amazon VPC 사용 설명서의 [VPC의 서브넷](#)을 참조하세요. 이때 VPC 식별자, 서브넷 및 서브넷의 가용 영역을 기록해 두세요. 클러스터 또는 작업 그룹을 시작할 때 이 정보가 필요합니다.

 Note

다음 단계에서 서브넷 그룹에 추가하려면 VPC에 서브넷이 하나 이상 정의되어 있어야 합니다. VPC에 서브넷 추가에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [VPC에 서브넷 추가](#)를 참조하세요.

2. Amazon Redshift 클러스터 서브넷 그룹을 생성하여 Amazon Redshift 클러스터가 VPC에서 사용할 수 있는 서브넷을 지정합니다. Redshift Serverless의 경우 서브넷 그룹을 생성하지 않고 작업 그룹을 생성할 때 서브넷 모음을 할당합니다. 작업 그룹을 만들 때 Serverless 대시보드에서 이 작업을 수행할 수 있습니다.

서브넷 그룹은 Amazon Redshift 콘솔을 사용하거나 프로그래밍 방식으로 생성할 수 있습니다. 자세한 내용은 [Redshift 리소스의 서브넷](#) 섹션을 참조하세요.

3. 클러스터 또는 작업 그룹과 연결할 VPC 보안 그룹에서 인바운드 연결을 위한 액세스 권한을 부여합니다. 퍼블릭 인터넷에서 VPC 외부의 클라이언트가 클러스터에 연결하도록 할 수 있습니다. 이렇게 하려면 클러스터를 인바운드 액세스를 허용하는 VPC 보안 그룹과 연결합니다. 자세한 내용은 [Amazon Redshift 클러스터에 대한 보안 그룹 통신 설정 구성 또는 Amazon Redshift Serverless 작업 그룹](#) 섹션을 참조하세요.
4. Redshift 프로비저닝된 콘솔 또는 작업 그룹 또는 Amazon Redshift Serverless 콘솔에서 클러스터를 생성하는 단계를 따르세요. 네트워크 및 보안에서 설정한 가상 프라이빗 클라우드(VPC), 클러스터 서브넷 그룹 및 VPC 보안 그룹을 지정합니다.

프로비저닝된 데이터 웨어하우스 클러스터를 생성하는 자세한 단계가 나와 있는 연습은 Amazon Redshift 시작 안내서에서 [Get started with Amazon Redshift provisioned data warehouses](#) 섹션을 참조하세요. Amazon Redshift Serverless 작업 그룹을 만드는 방법에 대한 자세한 내용은 Amazon Redshift 시작 안내서에서 [Amazon Redshift Serverless 데이터 웨어하우스 시작하기](#) 섹션을 참조하세요.

시작하기 단계에 따라 샘플 데이터를 업로드한 후 쿼리 예를 실행하면서 클러스터 또는 작업 그룹을 테스트할 수 있습니다. 자세한 내용은 Amazon Redshift 시작 안내서에서 [Amazon Redshift Serverless 데이터 웨어하우스 시작하기](#)를 참조하세요.

VPC 보안 그룹

Amazon Redshift 클러스터 또는 Amazon Redshift Serverless 작업 그룹을 프로비저닝할 때는 기본적으로 아무도 액세스할 수 없도록 액세스가 제한됩니다. 다른 사용자에게 인바운드 액세스 권한을 부여하려면 해당 사용자를 보안 그룹과 연결합니다. EC2-VPC 플랫폼을 사용하는 경우 기존 Amazon VPC 보안 그룹을 사용하거나 새로운 보안 그룹을 정의할 수 있습니다. 그런 다음 다음과 같이 클러스터 또는 작업 그룹과 연결합니다. EC2-Classik 플랫폼을 사용할 때는 보안 그룹을 정의한 후 클러스터 또는 작업 그룹과 연결합니다. EC2-Classik 플랫폼에서 보안 그룹을 사용하는 방법에 대한 자세한 내용은 [Amazon Redshift 보안 그룹](#) 섹션을 참조하세요.

VPC 보안 그룹은 클러스터 같은 VPC 인스턴스에 대한 액세스 권한을 제어하는 규칙 집합으로 구성됩니다. 각 규칙 집합은 IP 주소의 범위 또는 기타 VPC 보안 그룹을 기준으로 액세스를 설정합니다. VPC 보안 그룹을 클러스터 또는 작업 그룹과 연결하면 VPC 보안 그룹에 정의된 규칙에 따라 액세스가 제어됩니다.

EC2-VPC 플랫폼에 프로비저닝하는 클러스터는 각각 1개 이상의 Amazon VPC 보안 그룹이 연결되어 있습니다. Amazon VPC는 VPC를 생성할 때 자동으로 생성되는 기본값(default)이라는 VPC 보안 그룹을 제공합니다. Redshift 리소스를 사용할 때 다른 VPC 보안 그룹을 지정하지 않으면 VPC에서 실행하는 각 클러스터가 기본 VPC 보안 그룹에 자동으로 연결됩니다. VPC 보안 그룹은 클러스터 생성 시 클러스터와 연결하거나, 혹은 나중에 클러스터를 수정하면서 VPC 보안 그룹을 연결할 수도 있습니다.

다음 스크린샷은 기본 VPC 보안 그룹에 대한 기본 규칙을 보여줍니다.

Inbound			
Source	Protocol	Port Range	Comments
The security group ID (sg-xxxxxxx)	All	All	Allow inbound traffic from instances assigned to the same security group
Outbound			
Destination	Protocol	Port Range	Comments
0.0.0.0/0	All	All	Allow all outbound traffic

필요에 따라 기본 VPC 보안 그룹에 대한 규칙을 변경할 수 있습니다.

기본 VPC 보안 그룹으로 충분한 경우 더 생성할 필요는 없습니다. 그러나 선택적으로 추가 VPC 보안 그룹을 생성하여 인바운드 액세스를 더 잘 관리할 수 있습니다. 예를 들어 Amazon Redshift 클러스터 또는 Serverless 작업 그룹에서 서비스를 실행한다고 가정했을 때 고객에게 제공할 수 있는 서비스 수준이 몇 가지 있습니다. 모든 서비스 수준에서 동일한 액세스 권한을 부여하지 않으려는 경우 각 서비

스 수준에 대해 하나씩 개별 VPC 보안 그룹을 생성할 수 있습니다. 그런 다음 각 VPC 보안 그룹을 클러스터 또는 작업 그룹과 연결하면 됩니다.

VPC에 대해 최대 100개의 VPC 보안 그룹을 만들고 VPC 보안 그룹을 여러 클러스터 및 작업 그룹과 연결할 수 있습니다. 그러나 클러스터 또는 작업 그룹에 연결할 수 있는 VPC 보안 그룹의 수에는 제한이 있습니다.

Amazon Redshift는 변경 사항을 VPC 보안 그룹에 즉시 적용합니다. 따라서 VPC 보안 그룹을 클러스터와 연결하면 업데이트되는 VPC 보안 그룹의 인바운드 클러스터 액세스 규칙이 바로 적용됩니다.

<https://console.aws.amazon.com/vpc/>에서 VPC 보안 그룹을 생성하고 수정할 수 있습니다. 또한 AWS CLI, Amazon EC2 CLI, 및 AWS Tools for Windows PowerShell을 사용하여 VPC 보안 그룹을 프로그래밍 방식으로 관리할 수 있습니다. 보안 그룹 작업에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [VPC의 보안 그룹](#)을 참조하세요.

Amazon Redshift 클러스터에 대한 보안 그룹 통신 설정 구성 또는 Amazon Redshift Serverless 작업 그룹

이 주제는 네트워크 트래픽을 적절하게 라우팅하고 수신하도록 보안 그룹을 구성하는 데 도움이 됩니다. 다음은 몇 가지 일반적인 사용 사례입니다.

- Amazon Redshift 클러스터 또는 Amazon Redshift Serverless 작업 그룹에 대해 퍼블릭 액세스를 활성화했지만 트래픽이 수신되지 않는 경우. 이런 경우 인터넷에서 트래픽이 클러스터에 도달할 수 있도록 인바운드 규칙을 구성해야 합니다.
- 클러스터 또는 작업 그룹에 공개적으로 액세스할 수 없으며 인바운드 트래픽을 허용하기 위해 Redshift의 사전 구성된 기본 VPC 보안 그룹을 사용하는 경우. 하지만 기본값 이외의 보안 그룹을 사용해야 하는 요구 사항이 있으며 이 사용자 지정 보안 그룹은 인바운드 트래픽을 허용하지 않는 경우. 통신을 허용하도록 구성해야 합니다.

다음 섹션에서는 각 사용 사례에 대한 올바른 응답을 선택하고 요구 사항에 따라 네트워크 트래픽을 구성하는 방법을 보여 줍니다. 선택적으로 이 단계를 사용하여 다른 프라이빗 보안 그룹으로부터의 통신을 설정할 수 있습니다.

Note

대부분의 경우 네트워크 트래픽 설정은 Amazon Redshift에서 자동으로 구성되지 않습니다. 이는 트래픽 소스가 인터넷인지 프라이빗 보안 그룹인지에 따라 세부적인 수준에서 달라질 수 있고 보안 요구 사항이 다양하기 때문입니다.

기본 또는 사용자 지정 보안 그룹 구성을 통한 퍼블릭 액세스 가능성

클러스터 또는 작업 그룹을 만들고 있거나 이미 있는 경우 다음 구성 단계를 수행하여 퍼블릭 액세스를 허용하세요. 이는 기본 보안 그룹을 선택하는 경우와 사용자 지정 보안 그룹을 선택하는 경우 모두에 적용됩니다.

1. 네트워크 설정을 찾는 방법:

- 프로비저닝된 Amazon Redshift 클러스터의 경우 속성 탭을 선택한 다음 네트워크 및 보안 설정에서 클러스터의 VPC를 선택합니다.
- Amazon Redshift Serverless 작업 그룹의 경우 작업 그룹 구성을 선택합니다. 목록에서 작업 그룹을 선택합니다. 데이터 액세스의 네트워크 및 보안 패널에서 편집을 선택합니다.

2. VPC의 인터넷 게이트웨이 및 라우팅 테이블을 구성합니다. VPC를 이름으로 선택하여 구성을 시작합니다. 그러면 VPC 대시보드가 열립니다. 인터넷에서 퍼블릭 액세스가 가능한 클러스터 또는 작업 그룹에 연결하려면 인터넷 게이트웨이를 라우팅 테이블에 연결해야 합니다. VPC 대시보드에서 라우팅 테이블을 선택하여 이를 구성할 수 있습니다. 인터넷 게이트웨이의 대상이 소스 0.0.0.0/0 또는 퍼블릭 IP CIDR로 설정되어 있는지 확인합니다. 라우팅 테이블은 클러스터가 상주하는 VPC와 연결되어야 합니다. 여기에 설명된 것과 같이 VPC의 인터넷 액세스 설정에 대한 자세한 내용은 Amazon VPC 설명서에서 [인터넷 액세스 활성화](#)를 참조하세요. 라우팅 테이블 구성에 대한 자세한 내용은 [라우팅 테이블 구성](#)을 참조하세요.

3. 인터넷 게이트웨이 및 라우팅 테이블을 구성한 후 Redshift의 네트워크 설정으로 돌아가세요. 보안 그룹을 선택한 다음 인바운드 규칙을 선택하여 인바운드 액세스를 엽니다. 인바운드 규칙 편집을 선택합니다.
4. 요구 사항에 따라 인바운드 규칙의 프로토콜 및 포트를 선택하여 클라이언트의 트래픽을 허용합니다. RA3 클러스터의 경우 5431~5455 또는 8191~8215 범위 내의 포트를 선택하세요. 작업을 마쳤으면 각 규칙을 저장합니다.
5. 퍼블릭 액세스 가능 설정을 편집하여 활성화합니다. 클러스터 또는 작업 그룹의 작업 메뉴에서 이 작업을 수행할 수 있습니다.

퍼블릭 액세스 가능 설정을 켜면 Redshift가 탄력적 IP 주소를 생성합니다. 이는 AWS 계정과 연결되는 고정 IP 주소입니다. VPC 외부의 클라이언트는 이를 사용하여 연결할 수 있습니다.

보안 그룹 구성에 대한 자세한 내용은 [Amazon Redshift 보안 그룹](#)을 참조하세요.

클라이언트와 연결하여 규칙을 테스트할 수 있습니다. Amazon Redshift Serverless에 연결하는 경우 다음 작업을 수행하세요. 네트워크 구성을 완료한 후 [Amazon Redshift RSQL](#)과 같은 클라이언트 도구를 사용하여 연결합니다. Amazon Redshift Serverless 도메인을 호스트로 사용하여 다음을 입력합니다.

```
rsql -h workgroup-name.account-id.region.amazonaws.com -U admin -d dev -p 5439
```

기본 또는 사용자 지정 보안 그룹 구성을 통한 프라이빗 액세스 가능성

인터넷을 통해 클러스터 또는 작업 그룹과 통신하지 않는 경우 프라이빗 액세스가 가능하다고 합니다. 보안 그룹을 만들 때 기본 보안 그룹을 선택한 경우 보안 그룹에는 다음과 같은 기본 통신 규칙이 포함됩니다.

- 이 보안 그룹에 할당된 모든 리소스로부터의 인바운드 트래픽을 허용하는 인바운드 규칙
- 모든 아웃바운드 트래픽을 허용하는 아웃바운드 규칙. 이 규칙의 대상은 0.0.0.0/0입니다. Classless Inter-Domain Routing(CIDR) 표기법에서는 가능한 모든 IP 주소를 나타냅니다.

클러스터나 작업 그룹의 보안 그룹을 선택하여 콘솔에서 규칙을 볼 수 있습니다.

클러스터나 작업 그룹 및 클라이언트가 모두 기본 보안 그룹을 사용하는 경우 네트워크 트래픽을 허용하는 데 추가 구성이 필요하지 않습니다. 그러나 Redshift 또는 클라이언트의 기본 보안 그룹에서 규칙을 삭제하거나 변경하면 추가 구성이 필요해집니다. 이 경우 인바운드 및 아웃바운드 통신을 허용하도록 규칙을 구성해야 합니다. 다음은 일반적인 보안 그룹 구성입니다.

- 클라이언트 Amazon EC2 인스턴스의 경우:
 - 클라이언트의 IP 주소를 허용하는 인바운드 규칙
 - Redshift 사용을 위해 제공된 모든 서브넷의 IP 주소 범위(CIDR 블록)를 허용하는 아웃바운드 규칙. 아니면 모든 IP 주소 범위인 0.0.0.0/0을 지정해도 됩니다.
- Redshift 클러스터 또는 작업 그룹의 경우:
 - 클라이언트 보안 그룹을 허용하는 인바운드 규칙
 - 0.0.0.0/0으로의 트래픽을 허용하는 아웃바운드 규칙. 일반적으로 이 아웃바운드 규칙은 모든 아웃바운드 트래픽을 허용합니다. 선택적으로 아웃바운드 규칙을 추가하여 클라이언트 보안 그룹에

대한 트래픽을 허용할 수 있습니다. 이 경우에는 각 요청에 대한 응답 트래픽이 인스턴스에 도달하도록 허용되므로 아웃바운드 규칙이 항상 필요한 것은 아닙니다. 요청 및 응답 동작에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [보안 그룹](#)을 참조하세요.

Redshift 사용을 위해 지정된 서브넷 또는 보안 그룹의 구성을 변경하는 경우 통신을 계속 유지하기 위해 트래픽 규칙을 적절히 변경해야 할 수 있습니다. 인바운드 및 아웃바운드 규칙 생성에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [VPC CIDR 블록](#)을 참조하세요. 클라이언트에서 Amazon Redshift에 연결하는 방법에 대한 자세한 내용은 [Amazon Redshift에서 연결 구성](#)을 참조하세요.

Redshift 리소스의 서브넷

Virtual Private Cloud(VPC)에서 프로비저닝된 클러스터를 만드는 경우 서브넷 그룹을 만듭니다. 각 VPC에는 보안 및 운영 요구 사항에 따라 리소스를 그룹화할 수 있는 VPC 내의 IP 주소 하위 집합인 서브넷이 하나 이상 있을 수 있습니다. 프로비저닝된 클러스터를 만들 때 서브넷 그룹을 생성하여 VPC에 서브넷 세트를 지정합니다. 프로비저닝된 클러스터 대시보드의 구성에서 클러스터 서브넷 그룹을 찾아 편집할 수 있습니다. 프로비저닝된 클러스터에 대한 초기 구성 중에 서브넷 그룹을 지정하면 Amazon Redshift가 해당 서브넷 중 하나에 클러스터를 생성합니다. VPC 서비스에 대한 자세한 내용은 [Amazon VPC](#) 제품 세부 정보 페이지를 참조하세요.

Amazon Redshift Serverless 작업 그룹의 서브넷 구성은 프로비저닝된 클러스터와 유사하지만 단계가 약간 다릅니다. Serverless 작업 그룹을 만들고 설정할 때 작업 그룹의 서브넷을 지정하면 해당 서브넷이 목록에 추가됩니다. Serverless 대시보드에서 작업 그룹 속성을 선택하면 기존 작업 그룹의 서브넷을 볼 수 있습니다. 네트워크 및 보안 속성에서 사용할 수 있습니다. 자세한 내용은 [네임스페이스를 사용하여 작업 그룹 생성](#)을 참조하세요.

VPC 생성에 대한 자세한 내용은 [Amazon VPC 사용 설명서](#)를 참조하세요.

프로비저닝된 클러스터의 서브넷 그룹을 만들거나 Serverless 작업 그룹의 서브넷을 선택한 후에는 이전에 추가한 서브넷을 제거하거나 더 추가할 수 있습니다. 이러한 변경 작업은 콘솔을 사용하거나 API 작업을 사용하여 수행할 수 있습니다. 프로비저닝된 클러스터의 API 작업에 대한 자세한 내용은 [ModifyClusterSubnetGroup](#)을 참조하세요. Serverless 작업 그룹에 대한 API 작업은 [UpdateWorkgroup](#)을 참조하세요.

서브넷 그룹의 서브넷 중 하나에 클러스터를 프로비저닝할 수 있습니다. 클러스터 서브넷 그룹을 사용하면 Virtual Private Cloud(VPC)에서 서브넷 세트를 지정할 수 있습니다.

⚠ Warning

클래식 크기 조정, 일시 중지 및 다시 시작, 다중 AZ 장애 조치 또는 기타 이벤트와 같은 클러스터 유지 관리 작업 중에 프로비저닝된 컴퓨팅 노드가 Amazon Redshift 클러스터 서브넷 그룹 내의 다른 서브넷으로 이동될 수 있습니다. 서브넷 그룹의 모든 서브넷에는 동일한 네트워크 ACL 인바운드 및 아웃바운드 규칙과 동일한 라우팅 테이블 경로가 있어야 합니다. 이렇게 하면 이러한 유지 관리 이벤트 후에도 Amazon Redshift 컴퓨팅 리소스가 최적으로 통신하고 작동할 수 있도록 연결이 보장됩니다. 네트워크 ACL 또는 라우팅 테이블 구성이 다른 서브넷을 동일한 Amazon Redshift 클러스터 서브넷 그룹에 추가하지 마세요.

서브넷 구성에 대한 자세한 내용은 Amazon VPC 사용 설명서에서 [VPC의 서브넷](#)을 참조하세요. Redshift 다중 AZ 배포에 대한 자세한 내용은 Redshift 관리 가이드에서 [다중 AZ 배포](#)을 참조하세요. [클러스터 크기 조정](#)도 Redshift 관리 가이드에서 다루고 있습니다.

클러스터 서브넷 그룹 생성

다음 절차에서는 프로비저닝된 클러스터를 위한 서브넷 그룹을 생성하는 방법을 안내합니다. VPC에서 클러스터를 프로비저닝하려면 적어도 하나의 클러스터 서브넷 그룹이 정의되어야 합니다.

프로비저닝된 클러스터에 대한 클러스터 서브넷 그룹을 생성하려면 다음을 수행하세요.

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 구성(Configurations)을 선택한 다음 서브넷 그룹(Subnet groups)을 선택합니다. 서브넷 그룹 목록이 표시됩니다.
3. 클러스터 서브넷 그룹 생성을 선택하여 생성 페이지를 표시합니다.
4. 추가할 서브넷을 포함하여 서브넷 그룹에 대한 정보를 입력합니다.
5. 클러스터 서브넷 그룹 생성을 선택하여 선택한 서브넷이 포함된 그룹을 생성합니다.

i Note

서브넷 컬렉션으로 Amazon Redshift Serverless 작업 그룹을 만드는 방법에 대한 자세한 내용은 Amazon VPC 사용 설명서에서 [네임스페이스가 있는 작업 그룹 생성](#) 또는 [서브넷 생성](#)을 참조하세요.

클러스터 서브넷 그룹 수정

서브넷 그룹을 생성한 후 Amazon Redshift 콘솔에서 해당 정보를 수정할 수 있습니다. 다음 절차는 프로비저닝된 클러스터의 서브넷 그룹을 수정하는 방법을 안내합니다.

프로비저닝된 클러스터에 대한 클러스터 서브넷 그룹을 수정하려면 다음을 수행하세요.

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 구성(Configurations)을 선택한 다음 서브넷 그룹(Subnet groups)을 선택합니다. 서브넷 그룹 목록이 표시됩니다.
3. 수정할 서브넷 그룹을 선택합니다.
4. 작업에서 수정을 선택하여 서브넷 그룹의 세부 정보를 표시합니다.
5. 서브넷 그룹에 대한 정보를 업데이트합니다.
6. 저장을 선택하여 그룹을 수정합니다.

경우에 따라 서브넷을 변경하거나 제거하려면 추가 단계가 필요합니다. 예를 들어, 이 AWS 지식 센터 문서인 [프로비저닝된 Amazon Redshift 클러스터를 다른 서브넷으로 이동하려면 어떻게 해야 하나요?](#)에서는 클러스터 이동을 다루는 사용 사례를 설명합니다.

프로비저닝된 클러스터에 대한 클러스터 서브넷 그룹을 삭제하려면 다음을 수행하세요.

클러스터 서브넷 그룹 사용을 마치면 그룹을 삭제하여 정리해야 합니다. 다음 절차에서는 프로비저닝된 클러스터의 서브넷 그룹을 삭제하는 단계를 안내합니다.

클러스터 서브넷 그룹을 삭제하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 구성(Configurations)을 선택한 다음 서브넷 그룹(Subnet groups)을 선택합니다. 서브넷 그룹 목록이 표시됩니다.
3. 삭제할 서브넷 그룹을 선택하고 삭제를 선택합니다.

Note

클러스터가 사용 중인 클러스터 서브넷 그룹은 삭제할 수 없습니다.

VPC 및 서브넷에 대한 퍼블릭 액세스 차단

VPC 퍼블릭 액세스 차단(BPA)은 중앙 집중식 보안 기능으로, 인터넷 게이트웨이 및 이그레스 전용 인터넷 게이트웨이를 통한 AWS 리전 소유 VPC 및 서브넷의 리소스의 인터넷 송신 또는 수신을 차단할 수 있습니다. AWS 계정에서 이 기능을 켜면 기본적으로 Amazon Redshift가 사용하는 VPC 또는 서브넷에 영향을 미칩니다. 즉, Amazon Redshift는 퍼블릭에 대한 모든 작업을 차단합니다.

VPC BPA가 켜져 있고 퍼블릭 인터넷을 통해 Amazon Redshift API를 사용하려는 경우 VPC 또는 서브넷에 Amazon EC2 API를 사용하기 위한 제외를 추가해야 합니다. 제외 항목에는 다음 모드 중 하나를 사용할 수 있습니다.

- 양방향: 제외된 VPC 및 서브넷을 오가는 모든 인터넷 트래픽이 허용됩니다.
- 송신 전용: 제외된 VPC 및 서브넷의 아웃바운드 인터넷 트래픽만 허용됩니다. 제외된 VPC 및 서브넷으로의 인바운드 인터넷 트래픽은 차단됩니다. 이는 BPA가 양방향으로 설정된 경우에만 적용됩니다.

VPC BPA 제외는 전체 VPC 또는 VPC 내의 특정 서브넷을 퍼블릭 액세스 가능으로 지정합니다. 해당 경계 내의 네트워크 인터페이스는 해당 인터페이스에 퍼블릭 인터넷에 대한 라우팅 및 액세스 권한이 있는지와 관련하여 보안 그룹, 라우팅 테이블 및 네트워크 ACL과 같은 일반 VPC 네트워킹 제어를 준수합니다. 제외 추가에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [Create and delete exclusions](#) 섹션을 참조하세요.

프로비저닝된 클러스터

서브넷 그룹은 동일한 VPC의 서브넷 조합입니다. 프로비저닝된 클러스터의 서브넷 그룹이 VPC BPA가 켜져 있는 계정에 있는 경우 다음 기능이 차단됩니다.

- 퍼블릭 클러스터 만들기
- 퍼블릭 클러스터 복원
- 프라이빗 클러스터를 퍼블릭으로 수정
- 그룹 내에 퍼블릭 클러스터가 하나 이상 있는 경우 VPC BPA가 켜져 있는 서브넷을 서브넷 그룹에 추가

Serverless 클러스터

Redshift Serverless는 서브넷 그룹을 사용하지 않습니다. 대신 각 클러스터에는 자체 서브넷 세트가 있습니다. 작업 그룹이 VPC BPA가 켜져 있는 계정에 있는 경우 다음 기능이 차단됩니다.

- 퍼블릭 액세스 작업 그룹 만들기
- 프라이빗 작업 그룹을 퍼블릭으로 수정
- 작업 그룹이 퍼블릭일 때 VPC BPA가 켜져 있는 서브넷을 작업 그룹에 추가

Redshift Enhanced VPC Routing으로 네트워크 트래픽 제어

Amazon Redshift Enhanced VPC Routing을 사용하면 Amazon Redshift는 클러스터와 데이터 리포지토리 사이의 [COPY](#) 및 [UNLOAD](#) 트래픽이 모두 Amazon VPC 서비스를 기반으로 하는 Virtual Private Cloud(VPC)를 통과하도록 강제합니다. Enhanced VPC Routing을 사용하면 Amazon VPC 사용 설명서에 설명된 대로 [VPC 보안 그룹](#), [네트워크 액세스 제어 목록\(ACL\)](#), [VPC 엔드포인트](#), [VPC 엔드포인트 정책](#), [인터넷 게이트웨이](#), [도메인 이름 시스템\(DNS\)](#) 서버 등의 표준 VPC 기능을 사용할 수 있습니다. 이러한 기능을 사용하여 Amazon Redshift 클러스터와 다른 리소스 간의 데이터 흐름을 제어할 수 있습니다. Enhanced VPC Routing 기능을 사용하여 트래픽이 VPC를 통과하도록 라우팅하면 [VPC 흐름 로그](#)를 사용하여 COPY 및 UNLOAD 트래픽을 모니터링할 수도 있습니다.

Amazon Redshift 클러스터 및 Amazon Redshift Serverless 작업 그룹은 모두 향상된 VPC 라우팅을 지원합니다. Redshift Spectrum에서 향상된 VPC 라우팅을 사용할 수 없습니다. 자세한 내용은 [Redshift Spectrum으로 Amazon S3 버킷에 액세스](#) 섹션을 참조하세요.

향상된 VPC 라우팅 기능을 켜지 않은 경우 Amazon Redshift는 AWS 네트워크 내의 다른 서비스로 전송되는 트래픽을 포함하여 인터넷을 통해 트래픽을 라우팅합니다.

Important

Enhanced VPC Routing은 Amazon Redshift가 다른 리소스에 액세스하는 방식에 영향을 미치기 때문에, VPC를 올바르게 구성하지 않으면 COPY 및 UNLOAD 명령이 실패할 수 있습니다. 따라서 다음 설명과 같이 클러스터의 VPC와 사용자의 데이터 리소스 사이의 네트워크 경로를 명확하게 생성해야 합니다.

향상된 VPC 라우팅 기능을 켜 클러스터에서 COPY 또는 UNLOAD 명령을 실행하면 VPC는 가능한 가장 엄격한 또는 가장 특정한 네트워크 경로를 사용하여 트래픽을 지정된 리소스로 라우팅합니다.

예를 들어 VPC에서 다음과 같은 경로를 구성할 수 있습니다.

- VPC 엔드포인트 - 트래픽이 클러스터 또는 작업 그룹과 동일한 AWS 리전에 속한 Amazon S3 버킷으로 전송되는 경우에는 VPC 엔드포인트를 생성하여 트래픽을 버킷으로 직접 보낼 수 있습니다. VPC 엔드포인트를 사용하면 엔드포인트 정책을 연결하여 Amazon S3에 대한 액세스를 관리할

수 있습니다. Redshift에서 엔드포인트 사용에 대한 자세한 내용은 [VPC 엔드포인트를 사용하여 데이터베이스 트래픽 제어](#) 섹션을 참조하세요. Lake Formation 를 사용하는 경우, VPC와 AWS Lake Formation 간에 프라이빗 연결을 설정하는 방법에 대한 자세한 정보를 [AWS Lake Formation 및 인터페이스 VPC 엔드포인트\(AWS PrivateLink\)](#)에서 확인할 수 있습니다.

Note

Amazon S3 VPC 게이트웨이 엔드포인트와 함께 Redshift VPC 엔드포인트를 사용하는 경우 Redshift에서 향상된 VPC 라우팅을 활성화해야 합니다. 자세한 내용은 [Amazon S3용 게이트웨이 엔드포인트](#)를 참조하세요.

- NAT 게이트웨이 - 다른 AWS 리전의 Amazon S3 버킷에 연결하고 AWS 네트워크 내 다른 서비스에 연결할 수 있습니다. 또한 AWS 네트워크 외부에 있는 호스트 인스턴스에도 액세스할 수 있습니다. 이렇게 하려면 Amazon VPC 사용 설명서의 설명에 따라 [네트워크 주소 변환\(NAT\) 게이트웨이](#)를 구성합니다.
- 인터넷 게이트웨이 - VPC 외부의 AWS 서비스에 연결하기 위해 Amazon VPC 사용 설명서의 설명에 따라 [인터넷 게이트웨이](#)를 VPC 서브넷에 연결할 수 있습니다. 인터넷 게이트웨이를 사용하려면 클러스터 또는 작업 그룹이 다른 서비스에서 통신할 수 있도록 공개적으로 액세스할 수 있어야 합니다.

자세한 내용은 Amazon VPC 사용 설명서의 [VPC 엔드포인트](#) 섹션을 참조하세요.

Enhanced VPC Routing 사용에 대한 추가 요금은 없습니다. 특정 작업에 대해 데이터 전송 요금이 추가로 발생할 수 있습니다. 여기에는 다른 AWS 리전의 Amazon S3로 UNLOAD 등의 작업이 포함됩니다. 퍼블릭 IP 주소를 사용하여 Amazon EMR 또는 SSH(Secure Shell)에서 COPY합니다. 요금에 대한 자세한 정보는 [Amazon EC2 요금](#)을 참조하세요.

주제

- [VPC 엔드포인트를 사용하여 데이터베이스 트래픽 제어](#)
- [향상된 VPC 라우팅 켜기](#)
- [Redshift Spectrum으로 Amazon S3 버킷에 액세스](#)

VPC 엔드포인트를 사용하여 데이터베이스 트래픽 제어

VPC 엔드포인트를 사용하여 VPC의 Amazon Redshift 클러스터 또는 Serverless 작업 그룹과 Amazon Simple Storage Service(Amazon S3) 간에 관리형 연결을 생성할 수 있습니다. 이때는 Amazon S3의

데이터베이스와 데이터 간 COPY 및 UNLOAD 트래픽이 Amazon VPC를 벗어나지 않습니다. 따라서 엔드포인트 정책을 엔드포인트에 연결하여 데이터에 대한 액세스를 더욱 밀접하게 관리할 수 있습니다. 예를 들어 계정 내에서 특정 Amazon S3 버킷에만 데이터를 언로드하도록 허용하는 정책을 VPC 엔드포인트에 추가할 수 있습니다.

VPC 엔드포인트를 사용하려면 데이터 웨어하우스가 있는 VPC에 대한 VPC 엔드포인트를 생성한 다음 향상된 VPC 라우팅 기능을 켭니다. 클러스터 또는 작업 그룹을 생성할 때 향상된 VPC 라우팅 기능을 켜거나 향상된 VPC 라우팅을 사용하도록 VPC의 클러스터 또는 작업 그룹을 수정할 수 있습니다.

VPC 엔드포인트는 라우팅 테이블을 사용하여 VPC의 클러스터 또는 작업 그룹과 Amazon S3 사이의 트래픽 라우팅을 제어합니다. 지정된 라우팅 테이블과 연결된 서브넷의 클러스터 및 작업 그룹은 모두 자동으로 해당 엔드포인트를 사용하여 서비스에 액세스합니다.

VPC는 트래픽과 가장 명확하게, 즉 가장 한정적으로 일치하는 라우팅을 사용하여 트래픽의 라우팅 방식을 결정합니다. 예를 들어 라우팅 테이블에 인터넷 게이트웨이 및 Amazon S3 엔드포인트를 가리키는 모든 인터넷 트래픽(0.0.0.0/0)에 대한 라우팅이 있다고 가정해 보겠습니다. 이러한 경우 엔드포인트 라우팅은 Amazon S3로 전달되는 모든 트래픽에 우선합니다. 이는 Amazon S3 서비스에 대한 IP 주소 범위가 0.0.0.0/0보다 더 구체적이기 때문입니다. 이러한 예에서는 다른 AWS 리전의 Amazon S3 버킷으로 향하는 트래픽을 포함하여 그 밖의 다른 인터넷 트래픽 모두 인터넷 게이트웨이로 이동합니다.

엔드포인트 생성에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [VPC 엔드포인트 생성](#)을 참조하세요.

엔드포인트 정책은 클러스터 또는 작업 그룹에서 데이터 파일이 저장되어 있는 Amazon S3 버킷에 액세스하는 것을 제어할 때 사용됩니다. 더욱 명확하게 제어하려면 옵션으로 사용자 지정 엔드포인트 정책을 연결할 수 있습니다. 자세한 내용은 AWS PrivateLink 안내서의 [엔드포인트 정책을 사용하여 서비스에 대한 액세스 제어](#)를 참조하세요.

Note

AWS Database Migration Service(AWS DMS)는 관계형 데이터베이스, 데이터 웨어하우스 및 기타 유형의 데이터 스토어를 마이그레이션할 수 있는 클라우드 서비스입니다. VPC가 활성화된 Amazon Redshift 데이터베이스를 비롯한 모든 AWS 소스 또는 대상 데이터베이스에 연결할 수 있지만 일부 구성 제한이 있습니다. Amazon VPC 엔드포인트를 지원하므로 AWS DMS에서 복제 태스크에 대한 엔드 투 엔드 네트워크 보안을 더 쉽게 유지할 수 있습니다. AWS DMS와 함께 Redshift를 사용하는 방법에 대한 자세한 내용은 AWS Database Migration Service 사용 설명서의 [VPC 엔드포인트를 AWS DMS 소스 및 대상 엔드포인트로 구성](#)을 참조하세요.

엔드포인트 사용에 따르는 추가 요금은 없습니다. 데이터 전송 및 리소스 사용량에 대한 표준 요금이 그대로 적용됩니다. 요금에 대한 자세한 정보는 [Amazon EC2 요금](#)을 참조하세요.

향상된 VPC 라우팅 켜기

클러스터를 생성 또는 수정할 때, 그리고 Amazon Redshift Serverless 작업 그룹을 생성 또는 수정할 때 향상된 VPC 라우팅 기능을 켤 수 있습니다.

향상된 VPC 라우팅을 사용하려면 클러스터 또는 Serverless 작업 그룹이 다음 요구 사항 및 제약 조건을 충족해야 합니다.

- 클러스터가 VPC에 속해야 합니다.

Amazon S3 VPC 엔드포인트를 연결할 경우 VPC 엔드포인트는 동일한 AWS 리전에 있는 Amazon S3 버킷에 액세스하는 데만 사용됩니다. VPC 엔드포인트를 사용하지 않고 다른 AWS 리전에 속한 버킷에 액세스하거나, 혹은 다른 AWS 서비스에 액세스하려면 클러스터 또는 Serverless 작업 그룹을 공개 액세스로 변경하거나 [네트워크 주소 변환\(NAT\) 게이트웨이](#)를 사용합니다. 자세한 내용은 [VPC에서 Redshift 프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹 생성](#) 섹션을 참조하세요.

- VPC에서 DNS(Domain Name Service) 확인을 활성화해야 합니다. 또한 자체 DNS 서버를 사용하는 경우 Amazon S3에 대한 DNS 요청이 AWS에서 유지 관리하는 IP 주소로 올바르게 확인되어야 합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [VPC에서 DNS 사용](#)을 참조하세요.
- DNS 호스트 이름이 VPC에서 활성화되어야 합니다. 기본적으로 DNS 호스트 이름을 사용하도록 되어 있습니다.
- VPC 엔드포인트 정책이 Amazon Redshift에서 COPY, UNLOAD 또는 CREATE LIBRARY 호출 시 사용되는 모든 Amazon S3 버킷에 대한 액세스를 허용해야 합니다. 여기에는 관련된 모든 매니페스트 파일에 대한 액세스도 포함됩니다. 원격 호스트에서 호출하는 COPY의 경우에는 엔드포인트 정책이 각 호스트 컴퓨터에 대한 액세스를 허용해야 합니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [COPY, UNLOAD 및 CREATE LIBRARY 작업을 위한 IAM 권한](#) 섹션을 참조하세요.

프로비저닝된 클러스터에 대해 Enhanced VPC Routing을 설정하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.

2. 탐색 메뉴에서 Provisioned clusters dashboard(프로비저닝된 클러스터 대시보드)를 선택한 다음 Create cluster(클러스터 생성)를 선택하고 Cluster details(클러스터 세부 정보) 속성을 입력합니다.
3. 추가 구성 섹션을 표시하려면 기본값 사용을 끕니다.
4. Network and security(네트워크 및 보안) 섹션으로 이동합니다.
5. 향상된 VPC 라우팅 기능을 켜려면 Turn on(켜기)을 선택하여 클러스터 트래픽이 VPC를 통과하도록 합니다.
6. 클러스터를 생성하려면 클러스터 생성(Create cluster)을 선택합니다. 클러스터를 사용할 준비가 끝나려면 몇 분이 걸릴 수 있습니다.

Amazon Redshift Serverless에 대해 Enhanced VPC Routing을 설정하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 Serverless dashboard(Serverless 대시보드)를 선택한 다음 Create workgroup(작업 그룹 생성)을 선택하고 작업 그룹의 속성을 입력합니다.
3. Network and security(네트워크 및 보안) 섹션으로 이동합니다.
4. Turn on enhanced VPC routing(향상된 VPC 라우팅 켜기)을 선택하여 VPC를 통해 네트워크 트래픽을 라우팅합니다.
5. Next(다음)를 선택하고 작업 그룹을 생성할 때까지 작업 그룹 속성 입력을 마칩니다.

Redshift Spectrum으로 Amazon S3 버킷에 액세스

일반적으로 Amazon Redshift Spectrum은 프로비저닝된 클러스터에서 향상된 VPC 라우팅을 지원하지 않지만, 향상된 VPC 라우팅이 활성화된 경우 프로비저닝된 클러스터가 Amazon S3에서 외부 테이블을 쿼리할 수 있습니다.

Amazon Redshift의 향상된 VPC 라우팅은 VPC를 통해 특정 트래픽을 전송합니다. 즉, 클러스터와 Amazon S3 버킷 간의 모든 트래픽이 Amazon VPC를 통과하게 됩니다. Redshift Spectrum은 Amazon Redshift가 소유하지만 VPC 외부에 있는 AWS 관리형 리소스에서 실행되므로 Redshift Spectrum은 향상된 VPC 라우팅을 사용하지 않습니다.

Redshift Spectrum과 Amazon S3 간의 트래픽은 VPC 외부에서 AWS 사설 네트워크를 통해 안전하게 라우팅됩니다. 이동 중인 트래픽은 Amazon Signature Version 4 프로토콜(SIGv4)로 서명되고 HTTPS를 사용하여 암호화됩니다. 이 트래픽에는 Amazon Redshift 클러스터에 연결된 IAM 역할을 기반으

로 권한이 부여됩니다. Redshift Spectrum 트래픽을 추가로 관리하기 위해 클러스터의 IAM 역할과 Amazon S3 버킷에 연결된 정책을 수정할 수 있습니다. 또한 아래의 설명처럼 클러스터가 AWS Glue 또는 Athena에 액세스할 수 있도록 VPC를 구성해야 할 수도 있습니다.

Enhanced VPC Routing은 Amazon Redshift가 다른 리소스에 액세스하는 방식에 영향을 미치기 때문에, VPC를 올바르게 구성하지 않으면 쿼리가 실패할 수 있다는 점에 유의하세요. 자세한 내용은 Amazon S3 버킷으로 트래픽을 전송하기 위해 VPC 엔드포인트, NAT 게이트웨이 및 기타 네트워킹 리소스를 생성하는 방법에 대해 상세히 설명하는 [Redshift Enhanced VPC Routing으로 네트워크 트래픽 제어](#) 섹션을 참조하세요.

Note

Amazon Redshift Serverless는 Amazon S3의 외부 테이블에 대한 쿼리를 위한 향상된 VPC 라우팅을 지원합니다. 구성에 대한 자세한 내용은 Amazon Redshift Serverless 시작 안내서에서 [Amazon S3에서 데이터 로드](#)를 참조하세요.

Amazon Redshift Spectrum 사용 시 권한 정책 구성

Redshift Spectrum을 사용할 때는 다음을 고려하세요.

- [Amazon S3 버킷 액세스 정책 및 IAM 역할](#)
- [IAM 역할을 맡기 위한 권한](#)
- [Amazon S3 액세스 로깅 및 감사](#)
- [AWS Glue 또는 Amazon Athena에 액세스](#)

Amazon S3 버킷 액세스 정책 및 IAM 역할

버킷에 연결된 버킷 정책을 사용하고, 프로비저닝된 클러스터에 연결된 IAM 역할을 사용하여 Amazon S3 버킷에서 데이터에 대한 액세스를 제어할 수 있습니다.

프로비저닝된 클러스터의 Redshift Spectrum은 지정된 VPC 엔드포인트에 대한 액세스만 제한하는 버킷 정책을 사용하는 Amazon S3 버킷에 저장된 데이터에 액세스할 수 없습니다. 대신 특정 보안 주체(예: 특정 AWS 계정 또는 특정 사용자)에 대한 액세스를 제한하는 버킷 정책을 사용합니다.

버킷에 대한 액세스 권한이 부여된 IAM 역할의 경우 Amazon Redshift 서비스 보안 주체만 이 역할을 맡도록 허용하는 신뢰 관계를 사용합니다. 클러스터에 연결할 때 이 역할은 Amazon Redshift의 컨텍

스트 내에서만 사용할 수 있으며 클러스터 외부에서는 공유할 수 없습니다. 자세한 내용은 [IAM 역할에 대한 액세스 제한](#) 섹션을 참조하세요. 서비스 제어 정책(SCP)을 사용하여 역할을 추가로 제한할 수도 있습니다. AWS Organizations 사용 설명서의 [IAM 사용자 및 역할이 지정된 변경을 수행하지 못하도록 방지\(지정된 관리자 역할은 제외\)](#)를 참조하세요.

Note

Redshift Spectrum을 사용하기 위해 Amazon S3의 미리 서명된 URL 사용을 차단하는 IAM 정책을 적용할 수 없습니다. Amazon Redshift Spectrum에서 생성한 미리 서명된 URL은 1시간 동안 유효하므로 Amazon Redshift가 충분한 시간을 갖고 Amazon S3 버킷에서 모든 파일을 로드할 수 있습니다. Redshift Spectrum으로 스캔한 각 파일에 대해 미리 서명된 고유한 URL이 생성됩니다. s3:signatureAge 작업이 포함된 버킷 정책의 경우 값을 최소 3,600,000밀리초로 설정해야 합니다.

다음 예시 버킷 정책은 AWS 계정 123456789012가 소유한 지정된 버킷에 액세스하도록 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BucketPolicyForSpectrum",
      "Effect": "Allow",
      "Principal": {
        "AWS": ["arn:aws:iam::123456789012:role/redshift"]
      },
      "Action": [
        "s3:GetObject",
        "s3:ListBucketVersions",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```


IAM 역할을 맡기 위한 권한

클러스터에 연결된 역할에는 다음과 같이 Amazon Redshift 서비스에서만 이 역할을 맡을 수 있도록 허용하는 신뢰 관계가 있어야 합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Redshift Spectrum에 대한 IAM 정책](#) 단원을 참조하세요.

Amazon S3 액세스 로깅 및 감사

Amazon Redshift Enhanced VPC Routing 사용의 한 가지 이점은 모든 COPY 및 UNLOAD 트래픽이 VPC 흐름 로그에 로깅된다는 것입니다. Redshift Spectrum에서 시작해 Amazon S3로 전송되는 트래픽이 VPC를 통과하면 안 되기 때문에 이 트래픽은 VPC 흐름 로그에 로깅되지 않습니다. Redshift Spectrum이 Amazon S3의 데이터에 액세스하는 경우 AWS 계정 및 각 역할 권한의 맥락에서 이러한 작업을 수행합니다. AWS CloudTrail 및 Amazon S3에서 서버 액세스 로깅을 사용하여 Amazon S3 액세스를 기록하고 감사할 수 있습니다.

S3 IP 범위가 허용 목록에 추가되었는지 확인합니다. 필요한 S3 IP 범위에 대한 자세한 내용은 [네트워크 격리](#)를 참조하세요.

AWS CloudTrail 로그

Redshift Spectrum 액세스를 비롯해 Amazon S3의 객체에 대한 모든 액세스를 추적하려면 Amazon S3 객체에 대한 CloudTrail 로깅을 사용합니다.

CloudTrail을 사용하여 AWS 인프라 전반에서 계정 활동을 확인, 검색, 다운로드, 보관 및 응답할 수 있습니다. 자세한 내용은 [CloudTrail 시작하기](#)를 참조하십시오.

기본적으로 CloudTrail은 버킷 수준의 작업만 추적합니다. 객체 수준 작업(예: GetObject)을 추적하려면 로깅된 각 버킷에 대해 데이터 및 관리 이벤트를 활성화하십시오.

Amazon S3 서버 액세스 로깅

서버 액세스 로깅은 버킷에 대해 이루어진 요청에 따른 상세 레코드를 제공합니다. 액세스 로그 정보는 보안 및 액세스 감사에 유용할 수 있습니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [서버 액세스 로깅 활성화 방법](#)을 참조하세요.

자세한 내용은 AWS Security Blog 게시물 [How to Use Bucket Policies and Apply Defense-in-Depth to Help Secure Your Amazon S3 Data](#)를 참조하세요.

AWS Glue 또는 Amazon Athena에 액세스

Redshift Spectrum은 AWS Glue 또는 Athena에서 데이터 카탈로그에 액세스합니다. 또 다른 옵션은 데이터 카탈로그에 전용 Hive 메타스토어를 사용하는 것입니다.

AWS Glue 또는 Athena에 액세스하도록 하려면 인터넷 게이트웨이 또는 NAT 게이트웨이를 사용해 VPC를 구성합니다. AWS Glue 및 Athena의 퍼블릭 엔드포인트에 대한 아웃바운드 트래픽을 허용하도록 VPC 보안 그룹을 구성합니다. 또는 AWS Glue에 액세스하도록 AWS Glue Data Catalog의 인터페이스 VPC 엔드포인트를 구성할 수 있습니다. VPC 인터페이스 엔드포인트를 사용하는 경우 VPC와 AWS Glue 간의 통신은 AWS 네트워크에서 수행됩니다. 자세한 내용은 [인터페이스 엔드포인트 생성](#)을 참조하세요.

VPC에서 다음과 같은 경로를 구성할 수 있습니다.

- 인터넷 게이트웨이 – VPC 외부의 AWS 서비스에 연결하기 위해 Amazon VPC 사용 설명서의 설명에 따라 [인터넷 게이트웨이](#)를 VPC 서브넷에 연결할 수 있습니다. 인터넷 게이트웨이를 사용하려면 프로비저닝된 클러스터에 다른 서비스에서 통신할 수 있는 퍼블릭 IP 주소가 있어야 합니다.
- NAT 게이트웨이 – 다른 AWS 리전의 Amazon S3 버킷이나 AWS 네트워크 내의 다른 서비스에 연결하려면 Amazon VPC 사용 설명서에 설명된 대로 [네트워크 주소 변환\(NAT\)](#) 게이트웨이를 구성합니다. 이 구성은 AWS 네트워크 외부에 있는 호스트 인스턴스에 액세스하는 데에도 사용됩니다.

자세한 내용은 [Redshift Enhanced VPC Routing으로 네트워크 트래픽 제어](#) 섹션을 참조하세요.

Amazon Redshift 이벤트

Amazon Redshift는 클러스터 이벤트를 추적하여 이벤트 관련 정보를 AWS 계정에 몇 주간 저장합니다. 또한 이벤트가 있을 때마다 Amazon Redshift는 이벤트 발생 날짜, 설명, 이벤트 소스(클러스터, 파라미터 그룹 또는 스냅샷), 소스 ID 등의 정보를 보고합니다.

Amazon Redshift에서는 일부 이벤트에 대해 사전에 알림을 제공합니다. 이러한 이벤트의 이벤트 범주는 pending입니다. 예를 들어, 클러스터의 노드 중 하나에 대해 하드웨어 업데이트가 필요한 경우 사전 알림을 보냅니다. 다른 Amazon Redshift 이벤트와 마찬가지로 보류 중인 이벤트를 구독할 수 있습니다. 자세한 내용은 [Amazon Redshift 클러스터 이벤트 알림 구독](#) 섹션을 참조하세요.

Amazon Redshift 관리 콘솔, Amazon Redshift API 또는 AWS SDK를 사용하여 이벤트 정보를 얻을 수 있습니다. 이벤트 전체 목록을 가져오거나, 이벤트 지속 시간 또는 시작 및 종료 날짜 같은 필터를 적용하여 특정 기간 동안의 이벤트 정보를 가져올 수 있습니다.

그 밖에 클러스터 이벤트나 파라미터 그룹 이벤트 같이 특정 소스 유형에서 발생한 이벤트도 가져올 수 있습니다. [소스(Source)] 옆에는 지정된 작업을 트리거하는 리소스 이름과 리소스 유형이 표시됩니다.

이벤트 필터 집합을 지정하여 Amazon Redshift 이벤트 알림 구독을 생성할 수 있습니다. 이후 필터 기준과 일치하는 이벤트가 발생하면 Amazon Redshift가 Amazon Simple Notification Service를 사용하여 이벤트 발생 사실을 능동적으로 알립니다.

소스 유형 및 범주별 Amazon Redshift 이벤트 목록은 [the section called “프로비저닝된 클러스터 이벤트 알림”](#) 섹션을 참조하세요.

Amazon Redshift 클러스터 이벤트 알림 구독

Amazon Redshift는 Amazon Simple Notification Service(Amazon SNS)를 사용하여 Amazon Redshift 이벤트 알림을 전달합니다. 알림은 Amazon Redshift 이벤트 구독을 생성하여 사용할 수 있습니다. 주어진 클러스터, 스냅샷, 보안 그룹 또는 파라미터 그룹에 대해 이벤트가 발생하면 알림을 받을 수 있습니다. 가장 간단한 구독 생성 방법은 Amazon SNS 콘솔을 이용하는 것입니다. Amazon SNS 주제 생성 및 구독에 대한 자세한 내용은 [Amazon SNS 시작하기](#)를 참조하세요.

Amazon Redshift 구독에서는 Amazon Redshift 이벤트 및 Amazon SNS 주제에 대한 필터 집합을 지정합니다. 필터 기준과 일치하는 이벤트가 발생할 때마다 Amazon Redshift는 Amazon SNS 주제에 알림 메시지를 게시합니다.

그런 다음 주제에 대한 Amazon SNS 구독이 있는 모든 Amazon SNS 소비자에게 메시지를 전송합니다. Amazon SNS 사용자에게 전송되는 메시지는 이메일, 텍스트 메시지, HTTP 엔드포인트 호출 등과

같이 AWS 리전에서 Amazon SNS가 지원하는 모든 형식일 수 있습니다. 예를 들어 모든 리전이 이메일 알림을 지원하지만 SMS 알림은 미국 동부(버지니아 북부) 리전에서만 생성됩니다.

Note

현재는 Amazon SNS 표준 주제(Amazon SNS FIFO 주제 제외)에 대한 이벤트 구독만 생성할 수 있습니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 [Amazon SAN 이벤트 소스](#)를 참조하세요.

이벤트 알림 구독을 생성할 때 이벤트 필터를 하나 이상 지정합니다. Amazon Redshift는 모든 필터 기준과 일치하는 이벤트가 발생할 때마다 구독을 통해 알림을 보냅니다. 필터 기준으로는 소스 유형(클러스터, 스냅샷 등), 소스 ID(클러스터 또는 스냅샷 이름 등), 이벤트 카테고리(모니터링, 보안 등), 이벤트 심각도(INFO, ERROR 등)가 있습니다.

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.

Amazon Redshift CLI 또는 API를 사용하여 AWS Management Console에서 [사용(Enabled)] 라디오 버튼을 No로 설정하거나 Enabled 파라미터를 false로 설정하여 구독을 삭제하지 않고 알림을 쉽게 해제할 수 있습니다.

Amazon Redshift 이벤트 구독에서는 다음과 같은 이벤트 기준을 지정할 수 있습니다.

- 소스 유형 - 값은 클러스터, 스냅샷, 파라미터 그룹 및 보안 그룹입니다.
- 리소스의 소스 ID(my-cluster-1 또는 my-snapshot-20130823) - ID는 이벤트 구독과 동일한 AWS 리전에 속한 리소스 ID가 되어야 합니다.
- 이벤트 카테고리의 값은 Configuration(구성), Management(관리), Monitoring(모니터링), Security(보안), Pending(보류 중)입니다.
- 이벤트 심각도 - 값은 INFO 또는 ERROR입니다.

이벤트 기준은 따로 지정할 수 있습니다. 단, 콘솔에서 소스 ID를 지정하기 전에 소스 유형을 먼저 지정해야 하는 경우는 제외입니다. 예를 들어 소스 유형, 소스 ID 또는 심각도를 지정하지 않고도 이벤트 카테고리를 지정할 수 있습니다. 소스 유형에서 지정한 유형이 아닌 리소스의 소스 ID도 지정할 수는 있지만 이러한 리소스에서 발생하는 이벤트에 대해서는 알림이 전송되지 않습니다. 예를 들어 소스 유형

을 클러스터로, 그리고 ID를 보안 그룹으로 지정하였다고 가정할 때 보안 그룹에서 발생하는 이벤트가 소스 유형 필터 기준과 일치하지 않으면 해당 이벤트에 대해 어떠한 알림도 전송되지 않습니다.

Amazon Redshift는 구독에서 지정한 모든 기준과 일치하는 이벤트에 대해 알림을 전송합니다. 반환되는 이벤트 집합에 대한 몇 가지 예를 소개하면 다음과 같습니다.

- 구독에서 소스 유형을 클러스터로, 소스 ID를 my-cluster-1로, 카테고리를 모니터링으로, 심각도를 ERROR로 지정합니다. 그러면 my-cluster-1에서 심각도가 ERROR인 모니터링 이벤트에 대해서만 알림이 전송됩니다.
- 구독에서 소스 유형을 클러스터로, 카테고리를 구성으로, 심각도를 INFO로 지정합니다. 그러면 AWS 계정에 속한 모든 Amazon Redshift 클러스터에서 심각도가 INFO인 구성 이벤트에 대해 알림이 전송됩니다.
- 구독에서 카테고리를 구성으로, 심각도를 INFO로 지정합니다. 그러면 AWS 계정에 속한 모든 Amazon Redshift 리소스에서 심각도가 INFO인 구성 이벤트에 대해 알림이 전송됩니다.
- 구독에서 심각도를 ERROR로 지정합니다. 그러면 AWS 계정에 속한 모든 Amazon Redshift 리소스에서 심각도가 ERROR인 모든 이벤트에 대해 알림이 전송됩니다.

기존 구독에서 이름이 소스 ID로 참조되는 객체를 삭제하거나, 혹은 객체 이름을 변경하더라도 해당 구독은 활성 상태를 유지하지만 해당 객체에서 발생하는 이벤트 알림은 더 이상 전송되지 않습니다. 나중에 구독 소스 ID에서 참조하는 이름과 동일한 이름으로 새로운 객체를 생성하면 새로운 객체에서 발생하는 이벤트 알림에 대한 전송을 시작합니다.

Amazon Redshift는 이벤트 알림을 Amazon SNS 주제에 게시하며, 이 주제는 Amazon 리소스 이름(ARN)으로 구분할 수 있습니다. Amazon Redshift 콘솔에서 이벤트 구독을 생성할 때는 기존 Amazon SNS 주제를 지정하거나, 혹은 구독 생성 시 콘솔에서 주제를 생성하도록 요청할 수도 있습니다.

이렇게 Amazon SNS 주제로 전송된 모든 Amazon Redshift 이벤트 알림은 이어서 해당 주제를 구독하는 Amazon SNS 고객 모두에게 전송됩니다. 주제에 대한 고객 구독을 추가하거나 제거하는 등 Amazon SNS 주제를 변경할 때는 Amazon SNS 콘솔을 사용하면 됩니다.

다음 섹션은 수신되는 모든 카테고리 및 이벤트를 나열합니다. 또한 Amazon Redshift 이벤트 구독 및 구독 작업에 대한 정보도 제공합니다.

이벤트 알림 구독 생성

임의의 Amazon Redshift 클러스터, 스냅샷, 보안 그룹 또는 파라미터 그룹에 대한 이벤트 발생 시 알림을 전송할 수 있도록 Amazon Simple Notification Service(Amazon SNS) 이벤트 알림 구독을 생성할 수

있습니다. 이벤트에 대한 알림은 SNS 주제에 게시된 후 이어서 해당 주제를 구독하는 SNS 고객 모두에게 전송됩니다.

고객에게 전송되는 SNS 메시지는 이메일, 텍스트 메시지, HTTP 엔드포인트 호출 등과 같이 AWS 리전에서 Amazon SNS가 지원하는 모든 알림 형식일 수 있습니다. 예를 들어 모든 리전이 이메일 알림을 지원하지만 SMS 알림은 미국 동부(버지니아 북부) 리전에서만 생성됩니다. 자세한 내용은 [Amazon Redshift 프로비저닝 클러스터 이벤트 알림](#) 섹션을 참조하세요.

이벤트 구독을 생성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 이벤트(Events)를 선택합니다.
3. 이벤트 구독 탭을 선택하고 이벤트 구독 생성을 선택합니다.
4. 이름, 소스 유형, 범주 및 심각도와 같은 이벤트 구독의 속성을 입력합니다. 이벤트 알림을 받도록 Amazon SNS 주제를 사용할 수도 있습니다.
5. 구독을 생성하려면 Create event subscriptions(이벤트 구독 생성)를 선택합니다.

Amazon Redshift 프로비저닝 클러스터 이벤트 알림

이 페이지에서는 각 Amazon Redshift 소스 유형에 따른 이벤트 ID와 범주에 대해 살펴봅니다.

클러스터 소스 유형의 범주 및 이벤트

다음 표에는 클러스터가 소스 유형일 때 이벤트 카테고리 및 이벤트 목록이 나와 있습니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
구성	REDSHIFT-EVENT-1000	INFO	파라미터 그룹 [파라미터 그룹 이름]이 [시간]에 업데이트되었습니다. 동적 파라미터만 변경한 경우에는 지금 연관된 클러스터가 수정됩니다. 정적 파라미터를 변경한 경우에는 연관된 클러스터를 재부팅할 때 동적 파라미터를 포함한 모든 업데이트 사항이 적용됩니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
구성	REDSHIFT-EVENT-1001	INFO	[클러스터 이름] Amazon Redshift 클러스터가 파라미터 그룹 [파라미터 그룹 이름]을 사용하도록 [시간]에 수정되었습니다.
구성	REDSHIFT-EVENT-1500	오류	[VPC 이름] Amazon VPC가 존재하지 않습니다. 클러스터 [클러스터 이름]에 대한 구성 변경이 적용되지 않았습니다. AWS Management Console에서 문제를 해결하세요.
구성	REDSHIFT-EVENT-1501	오류	[VPC 이름] Amazon VPC에 지정한 고객 서브넷 [서브넷 이름]이 존재하지 않거나 잘못되었습니다. 클러스터 [클러스터 이름]에 대한 구성 변경이 적용되지 않았습니다. AWS Management Console에서 문제를 해결하세요.
구성	REDSHIFT-EVENT-1502	오류	클러스터 서브넷 그룹의 서브넷 [서브넷 그룹 이름]에는 사용 가능한 IP 주소가 없습니다. [클러스터 이름] 클러스터를 생성할 수 없습니다.
구성	REDSHIFT-EVENT-1503	오류	[VPC 이름] Amazon VPC에 인터넷 게이트웨이가 연결되어 있지 않습니다. 클러스터 [클러스터 이름]에 대한 구성 변경이 적용되지 않았습니다. AWS Management Console에서 문제를 해결하세요.
구성	REDSHIFT-EVENT-1504	오류	클러스터 [클러스터 이름] HSM에 연결할 수 없습니다.
구성	REDSHIFT-EVENT-1505	오류	클러스터 [클러스터 이름] HSM을 등록할 수 없습니다. 다른 구성을 시도하세요.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
구성	REDSHIFT-EVENT-1506	오류	Amazon Redshift가 계정의 탄력적 네트워크 인터페이스 제한을 초과했습니다. 탄력적 네트워크 인터페이스를 [최대 탄력적 네트워크 인터페이스 수]개까지 삭제하거나 EC2를 사용하여 AWS 리전당 네트워크 인터페이스 수의 한도 증가를 요청합니다.
구성	REDSHIFT-EVENT-1509	오류	계정의 VPC 엔드포인트 한도에 도달했기 때문에 [클러스터 이름] Amazon Redshift 클러스터를 생성할 수 없습니다. 사용하지 않는 VPC 엔드포인트를 삭제하거나 VPC 엔드포인트 한도 증가를 요청하세요. 자세한 내용은 Amazon VPC 사용 설명서의 VPC 엔드포인트 를 참조하세요.
구성	REDSHIFT-EVENT-1510	오류	Amazon Redshift 클러스터 [클러스터 이름]에 샘플 데이터를 로드하려는 시도에 성공하지 못했습니다. 샘플 데이터를 로드하려면 먼저 Amazon S3 버킷에 액세스할 수 있도록 VPC 구성한 다음 새 클러스터를 생성하고 샘플 데이터를 로드합니다. 자세한 내용은 Amazon Redshift 관리 가이드의 Enhanced VPC Routing 활성화 섹션을 참조하세요.
구성	REDSHIFT-EVENT-1511	오류	계정의 탄력적 IP 주소 한도를 초과하여 Amazon Redshift 클러스터 [클러스터 이름]을 (를) 생성할 수 없습니다. Amazon EC2에서 사용하지 않는 탄력적 IP 주소를 삭제하거나 한도 증가를 요청하세요.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
관리	REDSHIFT-EVENT-2000	INFO	[클러스터 이름] Amazon Redshift 클러스터가 생성되어 사용할 준비가 되었습니다.
관리	REDSHIFT-EVENT-2001	INFO	[클러스터 이름] Amazon Redshift 클러스터가 [시간]에 삭제되었습니다. 최종 스냅샷은 [저장되었습니다 / 저장되지 않았습니다.]
관리	REDSHIFT-EVENT-2002	INFO	[클러스터 이름] 클러스터에 대한 VPC 보안 그룹이 [UTC 기준 시간]에 업데이트되었습니다.
관리	REDSHIFT-EVENT-2003	INFO	[클러스터 이름] 클러스터에 대한 유지 관리가 [UTC 기준 시간]에 시작되었습니다.
관리	REDSHIFT-EVENT-2004	INFO	[클러스터 이름] 클러스터에 대한 유지 관리가 [UTC 기준 시간]에 완료되었습니다.
관리	REDSHIFT-EVENT-2006	INFO	[클러스터 이름] 클러스터 크기 조정이 [UTC 기준 시간]에 시작되었습니다. 클러스터는 읽기 전용 모드입니다.
관리	REDSHIFT-EVENT-2007	INFO	[클러스터 이름] 클러스터에 대한 크기 조정 요청이 승인되었습니다.
관리	REDSHIFT-EVENT-2008	INFO	새로운 Amazon Redshift 클러스터 [클러스터 이름] 스냅샷 [스냅샷 이름]을 생성하는 복원 작업이 [시간]에 시작되었습니다. 복원 프로세스를 모니터링하려면 AWS Management Console로 이동하세요.
관리	REDSHIFT-EVENT-2013	INFO	Amazon Redshift 클러스터 [클러스터 이름]의 이름이 [시간]에 변경되었습니다.
관리	REDSHIFT-EVENT-2014	INFO	Amazon Redshift 클러스터 [클러스터 이름]에 대한 테이블 복원 요청이 수신되었습니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
관리	REDSHIFT-EVENT-2015	INFO	Amazon Redshift 클러스터 [클러스터 이름]에 대한 테이블 복원이 [시간]에 취소되었습니다.
관리	REDSHIFT-EVENT-2016	INFO	Amazon Redshift 클러스터 [클러스터 이름] 교체가 [시간]에 시작되었습니다.
관리	REDSHIFT-EVENT-2017	INFO	Amazon Redshift 클러스터 [클러스터 이름]에 대해 고객이 실행한 유지 관리가 [시간]에 시작되었습니다. 유지 관리 중에는 클러스터를 사용하지 못할 수도 있습니다.
관리	REDSHIFT-EVENT-2018	INFO	Amazon Redshift 클러스터 [클러스터 이름]에 대해 고객이 실행한 유지 관리가 [시간]에 완료되었습니다.
관리	REDSHIFT-EVENT-2019	오류	Amazon Redshift 클러스터 [클러스터 이름]에 대해 고객이 실행한 유지 관리가 [시간]에 실패했습니다. 클러스터를 원래 상태로 되돌립니다.
관리	REDSHIFT-EVENT-2020	INFO	Amazon Redshift 클러스터 [클러스터 이름]의 트랙이 [원본 트랙]에서 [대상 트랙]으로 수정되었습니다.
관리	REDSHIFT-EVENT-2021	오류	용량 풀에서 용량을 확보하는 동안 Amazon Redshift 클러스터 [클러스터 이름]의 [작업]에 성공하지 못했습니다. 용량을 확보하기 위해 노력하고 있지만 현재 귀하의 요청이 취소되었습니다. 이 클러스터를 삭제하고 나중에 다시 시도하세요.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
관리	REDSHIFT-EVENT-2022	오류	용량 풀에서 용량을 확보하는 동안 Amazon Redshift 클러스터 [클러스터 이름]의 [작업]에 성공하지 못했습니다. 용량을 확보하기 위해 노력하고 있지만 현재 귀하의 요청이 취소되었습니다. 용량은 [대체 가용 영역]에서 사용할 수 있습니다. 이 클러스터를 삭제하고 대체 가용 영역에서 재시도하세요.
관리	REDSHIFT-EVENT-2023	오류	단일 노드 Amazon Redshift 클러스터 [클러스터 이름]에서 하드웨어 오류가 감지되었으므로 쿼리가 중단되거나 클러스터의 간헐적인 사용 가능성이 발생할 수 있습니다. 용량 풀에서 용량을 확보하는 동안 클러스터를 교체하지 못했습니다. 스냅샷에서 새 클러스터를 복원해야 합니다. 이 클러스터를 삭제하고 사용할 수 있는 최신 스냅샷을 선택한 다음 해당 스냅샷에서 새 클러스터를 복원합니다. 이렇게 하면 정상적인 하드웨어가 자동으로 제공됩니다.
관리	REDSHIFT-EVENT-2024	오류	단일 노드 Amazon Redshift 클러스터 [클러스터 이름]에서 하드웨어 오류가 감지되었으므로 쿼리가 중단되거나 클러스터의 간헐적인 사용 가능성이 발생할 수 있습니다. 용량 풀에서 용량을 확보하는 동안 클러스터를 교체하지 못했습니다. 용량은 [대체 가용 영역] 가용 영역에서 사용할 수 있습니다. 이 클러스터를 삭제하고 사용할 수 있는 최신 스냅샷을 선택한 다음 해당 스냅샷에서 새 클러스터를 복원합니다. 이렇게 하면 정상적인 하드웨어가 자동으로 제공됩니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
관리	REDSHIFT-EVENT-3011	INFO	Amazon Redshift 클러스터 '[클러스터]'에 대한 탄력적 크기 조정이 [시간]에 시작되었습니다. 크기 조정 중에는 데이터베이스 연결을 보류합니다. 이 작업 중 일부 쿼리 및 연결이 종료되거나 시간이 초과될 수 있습니다.
관리	REDSHIFT-EVENT-3012	INFO	[시간]에 시작된 '[클러스터 이름]' 클러스터에 대한 탄력적 크기 조정 요청을 받았습니다. 크기 조정이 시작되면 이벤트 알림을 제공해 드립니다.
보류 중	REDSHIFT-EVENT-2025	INFO	[클러스터 이름] 클러스터에 대한 데이터베이스가 [시작 시간] ~ [종료 시간] 사이에 업데이트됩니다. 클러스터에 액세스할 수 없습니다. 이에 따라 적절한 계획을 수립해야 합니다.
보류 중	REDSHIFT-EVENT-2026	INFO	[클러스터 이름] 클러스터가 [시작 시간] ~ [종료 시간] 사이에 업데이트됩니다. 클러스터에 액세스할 수 없습니다. 이에 따라 적절한 계획을 수립해야 합니다.
모니터링	REDSHIFT-EVENT-2050	INFO	Amazon Redshift 클러스터 [클러스터 이름]에서 하드웨어 문제가 발견되었습니다. 교체 요청이 [시간]에 시작되었습니다.
모니터링	REDSHIFT-EVENT-3000	INFO	[클러스터 이름] Amazon Redshift 클러스터가 [시간]에 재부팅되었습니다.
모니터링	REDSHIFT-EVENT-3001	INFO	Amazon Redshift 클러스터 [클러스터 이름]의 노드가 [시간]에 자동 교체되어 현재 정상적으로 작동하고 있습니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
모니터링	REDSHIFT-EVENT-3002	INFO	Amazon Redshift 클러스터 [클러스터 이름]의 크기 조정이 완료되어 이제 클러스터에서 읽기 및 쓰기 작업이 가능합니다. 크기 조정 작업은 [시간]에 시작되어 완료하는 데 [시간]시간이 걸렸습니다.
모니터링	REDSHIFT-EVENT-3003	INFO	[클러스터 이름] Amazon Redshift 클러스터가 스냅샷 [스냅샷 이름]에서 성공적으로 생성되어 현재 사용할 수 있습니다.
모니터링	REDSHIFT-EVENT-3007	INFO	Amazon Redshift 스냅샷 [스냅샷 이름]이 성공적으로 [소스 AWS 리전]에서 [대상 AWS 리전]으로 [시간]에 복사되었습니다.
모니터링	REDSHIFT-EVENT-3008	INFO	Amazon Redshift 클러스터 [클러스터 이름]에 대한 테이블 복원이 [시간]에 시작되었습니다.
모니터링	REDSHIFT-EVENT-3009	INFO	Amazon Redshift 클러스터 [클러스터 이름]에 대한 테이블 복원이 [시간]에 성공적으로 완료되었습니다.
모니터링	REDSHIFT-EVENT-3010	오류	Amazon Redshift 클러스터 [클러스터 이름]에 대한 테이블 복원이 [시간]에 실패했습니다.
모니터링	REDSHIFT-EVENT-3013	오류	Amazon Redshift 클러스터 [클러스터 이름]에 대해 요청된 탄력적 크기 조정 작업이 [이유]로 인해 [시간]에 실패했습니다.
모니터링	REDSHIFT-EVENT-3014	INFO	Amazon Redshift가 [클러스터 이름] 클러스터를 [시간]에 재부팅했습니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
보류 중	REDSHIFT-EVENT-3015	INFO	[시작 시간(UTC)]과 [종료 시간(UTC)] 사이에 유지 관리를 위해 [클러스터 이름] 클러스터의 재부팅을 예약했습니다. 이 시간 동안에는 클러스터에 액세스할 수 없습니다. 차질이 없도록 중단에 대한 계획을 세우는 것이 좋습니다.
모니터링	REDSHIFT-EVENT-3016	INFO	클러스터가 재부팅됩니다.
모니터링	REDSHIFT-EVENT-3500	오류	Amazon Redshift 클러스터 [클러스터 이름]에 대한 크기 조정이 실패했습니다. 크기 조정이 몇 분 후 자동으로 재시도됩니다.
모니터링	REDSHIFT-EVENT-3501	오류	스냅샷 [스냅샷 이름]에서 [클러스터 이름] Amazon Redshift 클러스터를 생성하는 복원 작업이 [시간]에 실패했습니다. 작업을 다시 시도하세요.
모니터링	REDSHIFT-EVENT-3504	오류	[클러스터 이름] 클러스터를 로깅하는 데 Amazon S3 버킷 [버킷 이름]이 잘못되었습니다.
모니터링	REDSHIFT-EVENT-3505	오류	Amazon S3 버킷 [버킷 이름]에 [클러스터 이름] 클러스터를 위한 올바른 IAM 정책이 없습니다.
모니터링	REDSHIFT-EVENT-3506	오류	Amazon S3 버킷 [버킷 이름]이 존재하지 않습니다. 클러스터 [클러스터 이름] 로깅을 계속할 수 없습니다.
모니터링	REDSHIFT-EVENT-3507	오류	EIP [IP 주소]를 사용하여 [클러스터 이름] Amazon Redshift 클러스터를 생성할 수 없습니다. 이 EIP는 이미 사용 중입니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
모니터링	REDSHIFT-EVENT-3508	오류	EIP [IP 주소]를 사용하여 [클러스터 이름] Amazon Redshift 클러스터를 생성할 수 없습니다. EIP를 찾을 수 없습니다.
모니터링	REDSHIFT-EVENT-3509	오류	클러스터 [클러스터 이름]에서 리전 간 스냅샷 복사가 활성화되어 있지 않습니다.
모니터링	REDSHIFT-EVENT-3510	오류	Amazon Redshift 클러스터 [클러스터 이름]에 대한 테이블 복원이 [시간]에 시작되지 않았습니다. 이유: [이유].
모니터링	REDSHIFT-EVENT-3511	오류	Amazon Redshift 클러스터 [클러스터 이름]에 대한 테이블 복원이 [시간]에 실패했습니다.
모니터링	REDSHIFT-EVENT-3512	오류	하드웨어 문제로 인해 [클러스터 이름] Amazon Redshift 클러스터가 실패했습니다. 클러스터는 [시간]에 생성된 최근 스냅샷 [스냅샷 이름]에서 자동 복원됩니다.
모니터링	REDSHIFT-EVENT-3513	오류	하드웨어 문제로 인해 [클러스터 이름] Amazon Redshift 클러스터가 실패했습니다. 클러스터는 [시간]에 생성된 최근 스냅샷 [스냅샷 이름]에서 자동 복원됩니다. 이 시간 이후 데이터베이스의 변경 내용은 다시 제출해야 합니다.
모니터링	REDSHIFT-EVENT-3514	오류	하드웨어 문제로 인해 [클러스터 이름] Amazon Redshift 클러스터가 실패했습니다. 클러스터가 하드웨어 장애 상태입니다. 클러스터를 삭제하고 [시간]에 생성된 최근 스냅샷 [스냅샷 이름]에서 복원하세요.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
모니터링	REDSHIFT-EVENT-3515	오류	하드웨어 문제로 인해 [클러스터 이름] Amazon Redshift 클러스터가 실패했습니다. 클러스터가 하드웨어 장애 상태입니다. 클러스터를 삭제하고 [시간]에 생성된 최근 스냅샷 [스냅샷 이름]에서 복원하세요. 이 시간 이후 데이터베이스의 변경 내용은 다시 제출해야 합니다.
모니터링	REDSHIFT-EVENT-3516	오류	하드웨어 문제로 인해 [클러스터 이름] Amazon Redshift 클러스터가 실패했지만 클러스터 백업이 없습니다. 클러스터가 하드웨어 장애 상태로 전환되며, 삭제할 수 있습니다.
모니터링	REDSHIFT-EVENT-3519	INFO	클러스터 [클러스터 이름]의 재시작이 [시간]에 시작되었습니다.
모니터링	REDSHIFT-EVENT-3520	INFO	클러스터 [클러스터 이름]의 재시작이 [시간]에 완료되었습니다.
모니터링	REDSHIFT-EVENT-3521	INFO	클러스터 [클러스터 이름]에서 연결 문제가 발견되어 자동 진단 검사가 [시간]에 시작되었습니다.
모니터링	REDSHIFT-EVENT-3522	INFO	클러스터 [클러스터 이름] 클러스터의 복원 작업이 [시간]에 중단되었습니다. Amazon Redshift 팀이 해결 방법을 찾고 있습니다.
모니터링	REDSHIFT-EVENT-3533	오류	'[클러스터 이름]'에 대한 클러스터 크기 조정이 [시간]에 취소되었습니다. [이유] 때문에 작업이 취소되었습니다. [필요한 조치].

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
모니터링	REDSHIFT-EVENT-3534	INFO	Amazon Redshift 클러스터 '[클러스터 이름]'에 대한 탄력적 크기 조정이 [시간]에 완료되었습니다. 이제 데이터를 전송하는 동안 읽기 및 쓰기 작업에 클러스터를 사용할 수 있습니다. 데이터 전송이 완료될 때까지 일부 쿼리를 완료하는데 시간이 더 걸릴 수 있습니다.
모니터링	REDSHIFT-EVENT-3537	INFO	클러스터 '[클러스터 이름]' 데이터 전송이 [시간(UTC)]에 완료되었습니다.
모니터링	REDSHIFT-EVENT-3600	INFO	Amazon Redshift 클러스터 '[클러스터 이름]'에 대해 요청된 크기 조정 작업이 과거에 취소되었습니다. 롤백이 [시간]에 완료되었습니다.
보류 중	REDSHIFT-EVENT-3601	INFO	[클러스터 이름] 클러스터의 노드가 [시작 시간] ~ [종료 시간] 사이에 교체됩니다. 이 유지 관리는 연기할 수 없습니다. 이에 따라 적절한 계획을 수립해야 합니다.
보류 중	REDSHIFT-EVENT-3602	INFO	[클러스터 이름] 클러스터의 노드가 [시작 시간] ~ [종료 시간] 사이에 교체되도록 예약되어 있습니다. 클러스터에 액세스할 수 없습니다. 이에 따라 적절한 계획을 수립해야 합니다.
관리	REDSHIFT-EVENT-3603	INFO	내부 오류로 인해 스냅샷 [스냅샷 이름]에서 클러스터[클러스터 이름]를 생성하는 복원 작업이 중단되었습니다. 클러스터가 호환되지 않는 복원 상태로 전환되며 삭제될 수 있습니다. 스냅샷을 다른 구성의 클러스터로 복원해 보세요.
관리	REDSHIFT-EVENT-3614	INFO	예약된 작업 [예약된 작업 이름]이 [시간(UTC)]에 생성되었습니다. 첫 번째 호출은 [시간(UTC)]에 예약됩니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
관리	REDSHIFT-EVENT-3615	INFO	예약된 작업 [예약된 작업 이름]이 [시간(UTC)]에 예약됩니다.
모니터링	REDSHIFT-EVENT-3616	INFO	[시간(UTC)]에 예약된 작업 [예약된 작업 이름]이 'SUCCEEDED' 상태로 완료되었습니다.
모니터링	REDSHIFT-EVENT-3617	오류	지연으로 인해 [시간(UTC)]에 예약된 작업 [예약된 작업 이름]을 건너뛰었습니다.
모니터링	REDSHIFT-EVENT-3618	INFO	클러스터 [클러스터 이름] 일시 중지 작업이 [시간(UTC)]에 시작되었습니다. 일시 중지 시작됨
모니터링	REDSHIFT-EVENT-3619	INFO	[클러스터 이름] Amazon Redshift 클러스터가 [UTC 시간]에 일시 중지되었습니다.
관리	REDSHIFT-EVENT-3626	INFO	예약된 작업 [예약된 작업 이름]이 [시간(UTC)]에 수정되었습니다. 첫 번째 호출은 [시간(UTC)]에 예약됩니다.
관리	REDSHIFT-EVENT-3627	INFO	예약된 작업 [예약된 작업 이름]이 [시간(UTC)]에 삭제되었습니다.
모니터링	REDSHIFT-EVENT-3628	오류	[시간(UTC)]에 예약된 작업 [예약된 작업 이름]이 'FAILED' 상태로 완료되었습니다.
모니터링	REDSHIFT-EVENT-3629	INFO	클러스터 [클러스터 이름]에 대한 내부 장애 조치 요청이 시작되었습니다.
모니터링	REDSHIFT-EVENT-3630	INFO	복구를 위해 Amazon Redshift 클러스터 [클러스터 이름]을 [가용 영역]에서 [가용 영역]으로 성공적으로 재배치했습니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
관리	REDSHIFT-EVENT-3631	INFO	Amazon Redshift [클러스터 이름]에서 재배포치 요청을 받았습니다. 가용 영역 재배포치가 완료 되면 Amazon Redshift가 이벤트 알림을 보냅니다.
모니터링	REDSHIFT-EVENT-3632	INFO	Amazon Redshift 클러스터 [클러스터 이름]이 [가용 영역]에서 [가용 영역]으로 성공적으로 재배포되었습니다. 이제 클러스터를 사용할 수 있습니다.
모니터링	REDSHIFT-EVENT-3658	오류	Redshift 클러스터 [클러스터 ID]에 대해 EC2-Classic에서 EC2-VPC로 마이그레이션에 실패했습니다.
모니터링	REDSHIFT-EVENT-3659	INFO	Redshift 클러스터 [클러스터 ID]에 대해 EC2-Classic에서 EC2-VPC로 마이그레이션이 성공했습니다.
모니터링	REDSHIFT-EVENT-3660	INFO	클러스터가 하드웨어 장애 상태입니다. EC2-Classic 클러스터를 삭제하고 [시간(UTC)]에 생성된 최신 스냅샷 [스냅샷 이름]에서 EC2-VPC 클러스터로 복원하세요.
관리	REDSHIFT-EVENT-3666	INFO	Amazon Redshift 다중 AZ 클러스터([클러스터 이름])가 [시간(UTC)]에 장애를 감지하고 자동 복구를 트리거했습니다.
관리	REDSHIFT-EVENT-3667	INFO	Amazon Redshift 다중 AZ 클러스터([클러스터 이름])가 [시간(UTC)]에 성공적으로 복구되었으며 [첫 번째 가용 영역]에서 사용할 수 있습니다. 다른 AZ의 보조 컴퓨팅도 곧 사용할 수 있게 될 예정입니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
모니터링	REDSHIFT-EVENT-3668	오류	Amazon Redshift 다중 AZ 클러스터([클러스터 이름])가 [시간(UTC)]에 복구되지 못했습니다.
관리	REDSHIFT-EVENT-3669	INFO	Amazon Redshift 다중 AZ 클러스터([클러스터 이름])가 [시간(UTC)]에 성공적으로 복구되었으며 [첫 번째 가용 영역] 및 [두 번째 가용 영역]의 컴퓨팅 리소스와 함께 사용할 수 있습니다.
관리	REDSHIFT-EVENT-3670	INFO	Amazon Redshift 클러스터([클러스터 이름])에 대한 유지 관리가 [시간(UTC)]에 완료되어 [첫 번째 가용 영역]의 컴퓨팅 리소스와 함께 사용할 수 있습니다. 다른 AZ의 보조 컴퓨팅도 곧 사용할 수 있게 될 예정입니다.
관리	REDSHIFT-EVENT-3671	INFO	Amazon Redshift 클러스터([클러스터 이름])에 대한 크기 조정이 [시간(UTC)]에 완료되어 [첫 번째 가용 영역]에서 사용할 수 있습니다. 다른 AZ의 보조 컴퓨팅도 곧 사용할 수 있게 될 예정입니다.
관리	REDSHIFT-EVENT-3672	INFO	Amazon Redshift 다중 AZ 클러스터([클러스터 이름])가 [시간(UTC)]에 [두 번째 가용 영역]에서 장애를 감지하고 자동 복구를 트리거했습니다.
관리	REDSHIFT-EVENT-3673	INFO	Amazon Redshift 클러스터([클러스터 이름])에 대한 다중 AZ를 활성화하는 작업이 [시간(UTC)]에 시작되었습니다.
관리	REDSHIFT-EVENT-3674	INFO	Amazon Redshift 클러스터([클러스터 이름])에 대한 다중 AZ를 활성화하는 작업이 [시간(UTC)]에 성공적으로 완료되었습니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
모니터링	REDSHIFT-EVENT-3675	오류	Amazon Redshift 클러스터([클러스터 이름])에 대한 다중 AZ를 활성화하는 작업이 [시간(UTC)]에 실패했습니다.
관리	REDSHIFT-EVENT-3676	INFO	Amazon Redshift 다중 AZ 클러스터([클러스터 이름])에 대한 다중 AZ를 비활성화하는 작업이 [시간(UTC)]에 시작되었습니다.
관리	REDSHIFT-EVENT-3677	INFO	Amazon Redshift 클러스터([클러스터 이름])에 대한 다중 AZ를 비활성화하는 작업이 [시간(UTC)]에 성공적으로 완료되었습니다.
모니터링	REDSHIFT-EVENT-3678	오류	Amazon Redshift 클러스터([클러스터 이름])에 대한 다중 AZ를 비활성화하는 작업이 [시간(UTC)]에 실패했습니다.
구성	REDSHIFT-EVENT-3679	INFO	Amazon Redshift 클러스터([클러스터 이름])의 포트가 성공적으로 수정되었습니다.
구성	REDSHIFT-EVENT-3680	오류	Amazon Redshift는 이 작업에 필요한 SLR(서비스 연결 역할)에 액세스할 수 없기 때문에 [클러스터 이름] 클러스터를 만들지 못했습니다. Amazon Redshift 콘솔에서 다시 만들어 보세요. Amazon Redshift는 SLR을 자동으로 생성합니다.
모니터링	REDSHIFT-EVENT-3684	오류	Amazon S3의 [버킷 이름] 버킷이 알 수 없거나 액세스할 수 없는 AWS KMS 키로 암호화되었습니다. Amazon S3 버킷의 암호화를 수정하세요.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
관리	REDSHIFT-EVENT-3685	오류	사용 가능한 디스크 공간이 충분하지 않아 [클러스터 이름] 클러스터의 복원 작업이 실패했습니다. 작업이 롤백되고 있습니다. 다른 구성의 클러스터로 복원해 보세요.
관리	REDSHIFT-EVENT-3686	오류	사용 가능한 디스크 공간이 충분하지 않아 [클러스터 이름] 클러스터의 크기 조정 작업이 실패했습니다. 작업이 롤백되고 있습니다. 다른 구성의 클러스터로 크기를 조정해 보세요.
관리	REDSHIFT-EVENT-3687	INFO	Amazon Redshift 클러스터 [클러스터 이름]에 대해 요청된 크기 조정 작업이 [UTC 시간]에 완료되었습니다.
보안	REDSHIFT-EVENT-3688	오류	Amazon Redshift 클러스터 [클러스터 이름]에 대한 암호화 키 교체를 완료할 수 없었습니다.
데이터 공유	REDSHIFT-EVENT-3689	INFO	Amazon Redshift 네임스페이스(<클러스터 이름>)를 Glue Data Catalog 계정(<계정 ID>)에 등록하는 작업이 <UTC 기준 시간>에 시작되었습니다.
데이터 공유	REDSHIFT-EVENT-3690	INFO	Amazon Redshift 네임스페이스(<클러스터 이름>)를 Glue Data Catalog 계정(<계정 ID>)에 등록하는 작업이 <UTC 기준 시간>에 성공적으로 완료되었습니다.
데이터 공유	REDSHIFT-EVENT-3691	INFO	Amazon Redshift 네임스페이스(<클러스터 이름>)를 Glue Data Catalog 계정(<계정 ID>)에 등록하는 작업이 <UTC 기준 시간>에 실패했습니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
데이터 공유	REDSHIFT-EVENT-3692	INFO	Amazon Redshift 네임스페이스(<클러스터 이름>)를 Glue Data Catalog 계정(<계정 ID>)에서 등록 취소하는 작업이 <UTC 기준 시간>에 시작되었습니다.
데이터 공유	REDSHIFT-EVENT-3693	INFO	Amazon Redshift 네임스페이스(<클러스터 이름>)를 Glue Data Catalog 계정(<계정 ID>)에서 등록 취소하는 작업이 <UTC 기준 시간>에 성공적으로 완료되었습니다.
데이터 공유	REDSHIFT-EVENT-3694	INFO	Amazon Redshift 네임스페이스(<클러스터 이름>)를 Glue Data Catalog 계정(<계정 ID>)에서 등록 취소하는 작업이 <UTC 기준 시간>에 실패했습니다.
보안	REDSHIFT-EVENT-4000	INFO	Amazon Redshift 클러스터: [클러스터 이름]의 관리자 자격 증명이 [시간]에 업데이트되었습니다.
보안	REDSHIFT-EVENT-4001	INFO	보안 그룹 [보안 그룹 이름]이 [시간]에 수정되었습니다. 변경 내용은 연결된 모든 클러스터에 자동으로 적용됩니다.
보안	REDSHIFT-EVENT-4005	오류	[클러스터 이름] 클러스터에 잘못된 AWS KMS key 권한이 있습니다. 클러스터가 "INACCESSIBLE_KMS_KEY" 상태가 되며 더 이상 액세스할 수 없습니다.
보안	REDSHIFT-EVENT-4006	INFO	AWS KMS key 권한이 복원되어 이제 [클러스터 이름] 클러스터가 이전 상태로 복원되었습니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
보안	REDSHIFT-EVENT-4500	오류	Amazon Redshift 클러스터([클러스터 이름])에 대해 제공한 보안 그룹 중 하나 이상을 찾을 수 없습니다. 유효한 보안 그룹이 있었다면 클러스터에 적용되었습니다. 무효한 보안 그룹은 적용되지 않았습니다. 기존 클러스터를 수정 중이었고 유효한 보안 그룹이 없었다면 클러스터는 원래 보안 그룹을 유지했습니다. 새 클러스터를 만드는 중이었고 유효한 보안 그룹이 없었다면 클러스터 만들기에 실패했습니다. 어느 경우든 유효한 보안 그룹으로 요청을 다시 시도하세요. Amazon Redshift에서 보안 그룹을 관리하는 방법에 대한 자세한 내용은 Amazon Redshift 관리 안내서의 VPC 보안 그룹 섹션을 참조하세요.
보안	REDSHIFT-EVENT-4501	오류	클러스터 보안 그룹 [클러스터 보안 그룹 이름]에서 지정한 보안 그룹 [보안 그룹 이름]을 찾을 수 없습니다. 권한 부여를 완료할 수 없습니다.
보안	REDSHIFT-EVENT-4502	오류	Amazon Redshift 클러스터 클러스터 이름의 관리자 자격 증명이 동시 활동으로 인해 시간에 업데이트하지 못했습니다. 현재 워크로드에서 활성 워크로드를 완료하거나 줄일 수 있게 허용한 다음 작업을 다시 시도하세요.
보안	REDSHIFT-EVENT-4503	오류	Amazon Redshift가 [클러스터 이름] 클러스터의 보안 암호에 액세스할 수 없습니다.
보안	REDSHIFT-EVENT-4504	오류	Amazon Redshift가 [클러스터 이름] 클러스터의 관리자 보안 인증 보안 암호를 암호화하는데 사용된 [KMS 키] KMS 키에 액세스할 수 없습니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
보안	REDSHIFT-EVENT-4505	오류	Amazon Redshift가 [클러스터 이름] 클러스터에 대한 보안 암호를 교체할 수 없습니다. 클러스터에서 진행 중인 작업이 있기 때문입니다.
보안	REDSHIFT-EVENT-4506	오류	Amazon Redshift [클러스터 이름] 클러스터가 일시 중지되었습니다. Amazon Redshift는 일시 중지된 클러스터의 보안 암호를 교체할 수 없습니다.

파라미터 그룹 소스 유형의 범주 및 이벤트

다음 표는 파라미터 그룹이 소스 유형일 때 이벤트 카테고리 및 이벤트 목록을 나타냅니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
구성	REDSHIFT-EVENT-1002	INFO	파라미터 [파라미터 이름]가 [시간]에 [값]에서 [값](으)로 업데이트되었습니다.
구성	REDSHIFT-EVENT-1003	INFO	클러스터 파라미터 그룹 [그룹 이름]이 생성되었습니다.
구성	REDSHIFT-EVENT-1004	INFO	클러스터 파라미터 그룹 [그룹 이름]이 삭제되었습니다.
구성	REDSHIFT-EVENT-1005	INFO	클러스터 파라미터 그룹 [이름]이 [시간]에 업데이트되었습니다. 동적 파라미터만 변경한 경우에는 지금 연관된 클러스터가 수정됩니다. 정적 파라미터를 변경한 경우에는 연관된 클러스터를 재부팅할 때 동적 파라미터를 포함한 모든 업데이트 사항이 적용됩니다.

보안 그룹 소스 유형의 범주 및 이벤트

다음 표는 보안 그룹이 소스 유형일 때 이벤트 카테고리 및 이벤트 목록을 나타냅니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
보안	REDSHIFT-EVENT-4002	INFO	클러스터 보안 그룹 [그룹 이름]이 생성되었습니다.
보안	REDSHIFT-EVENT-4003	INFO	클러스터 보안 그룹 [그룹 이름]이 삭제되었습니다.
보안	REDSHIFT-EVENT-4004	INFO	클러스터 보안 그룹 [그룹 이름]이 [시간]에 생성되었습니다. 변경 내용은 연결된 모든 클러스터에 자동으로 적용됩니다.

스냅샷 소스 유형의 범주 및 이벤트

다음 표는 스냅샷이 소스 유형일 때 이벤트 카테고리 및 이벤트 목록을 나타냅니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
관리	REDSHIFT-EVENT-2009	INFO	Amazon Redshift 클러스터 [클러스터 이름]의 사용자 스냅샷 [스냅샷 이름]이 [시간]에 시작되었습니다. 스냅샷 프로세스를 모니터링하려면 AWS Management Console로 이동하세요.
관리	REDSHIFT-EVENT-2010	INFO	Amazon Redshift 클러스터 [클러스터 이름]의 사용자 스냅샷 [스냅샷 이름]이 [시간]에 취소되었습니다.
관리	REDSHIFT-EVENT-2011	INFO	Amazon Redshift 클러스터 [클러스터 이름]의 사용자 스냅샷 [스냅샷 이름]이 [시간]에 삭제되었습니다.

Amazon Redshift 범주	이벤트 ID	이벤트 심각도	설명
관리	REDSHIFT-EVENT-2012	INFO	Amazon Redshift 클러스터 [클러스터 이름]의 최종 스냅샷 [스냅샷 이름]이 [시간]에 시작되었습니다.
모니터링	REDSHIFT-EVENT-3004	INFO	Amazon Redshift 클러스터 [클러스터 이름]의 사용자 스냅샷 [스냅샷 이름]이 [시간]에 성공적으로 완료되었습니다.
모니터링	REDSHIFT-EVENT-3005	INFO	Amazon Redshift 클러스터 [이름]의 최종 스냅샷 [이름]이 [시간]에 성공적으로 완료되었습니다.
모니터링	REDSHIFT-EVENT-3006	INFO	Amazon Redshift 클러스터 [클러스터 이름]의 최종 스냅샷 [스냅샷 이름]이 [시간]에 취소되었습니다.
모니터링	REDSHIFT-EVENT-3502	오류	Amazon Redshift 클러스터 [클러스터 이름]의 최종 스냅샷 [스냅샷 이름]이 [시간]에 실패했습니다. 담당 팀이 현재 문제를 조사 중입니다. AWS Management Console에서 작업을 다시 시도하세요.
모니터링	REDSHIFT-EVENT-3503	오류	Amazon Redshift 클러스터 [클러스터 이름]의 사용자 스냅샷 [스냅샷 이름]이 [시간]에 실패했습니다. 담당 팀이 현재 문제를 조사 중입니다. AWS Management Console에서 작업을 다시 시도하세요.

Amazon EventBridge로 Amazon Redshift Serverless 이벤트 알림

Amazon Redshift Serverless는 Amazon EventBridge를 통해 이벤트 알림을 관리하여 데이터 웨어하우스의 변경 사항에 대한 최신 정보를 제공합니다. Amazon EventBridge는 애플리케이션을 다양한 소스의 데이터와 연결하는 데 사용할 수 있는 서버리스 이벤트 버스 서비스입니다. 이 경우 이벤트 소스는

Amazon Redshift입니다. 환경에서 모니터링되는 변경 사항인 이벤트는 Amazon Redshift 데이터 웨어하우스에서 EventBridge로 자동으로 전송됩니다. 이벤트는 거의 실시간으로 전송됩니다.

EventBridge는 특정 이벤트에 대해 수행할 작업을 지정할 수 있는 이벤트 규칙을 작성할 수 있는 환경을 제공합니다. 또한 EventBridge가 이벤트를 보낼 수 있는 리소스인 대상을 설정할 수도 있습니다. 대상에는 API 대상, Amazon CloudWatch 로그 그룹 등이 포함될 수 있습니다. 규칙에 대한 자세한 내용은 [Amazon EventBridge 규칙](#)을 참조하세요. 대상에 대한 자세한 내용은 [Amazon EventBridge 대상](#)을 참조하세요.

이벤트는 심각도 및 범주로 분류될 수 있습니다. 다음과 같은 필터를 사용할 수 있습니다.

- 리소스 필터링 - 이벤트가 연결된 리소스에 따라 메시지를 수신합니다. 리소스에는 작업 그룹, 스냅샷 등이 있습니다.
- 기간 필터링 - 이벤트의 범위를 특정 기간으로 제한합니다.
- 범주 필터링 - 지정된 범주의 모든 이벤트에 대한 이벤트 알림을 받습니다.

다음 표에는 추가 메타데이터와 함께 Amazon Redshift Serverless 이벤트가 나와 있습니다.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
RateChange	REDSHIFT-SERVERLESS-EVENT-1001	INFO	작업 그룹 기본 RPU 변경이 <time in UTC>에 완료되었습니다.
RateChange	REDSHIFT-SERVERLESS-EVENT-1002	오류	작업 그룹 기본 RPU 변경이 <time in UTC>에 실패했습니다.
모니터링	REDSHIFT-SERVERLESS-EVENT-1003	INFO	<time in UTC>에 Amazon Redshift 데이터 웨어하우스 <endpoint name>에

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
			서 소프트웨어가 업데이트되었습니다.
구성	REDSHIFT-SERVERLESS-EVENT-1011	오류	Amazon Redshift Serverless는 이 작업에 필요한 SLR(서비스 연결 역할)에 액세스할 수 없기 때문에 [작업 그룹 이름] 작업 그룹을 만들지 못했습니다. Amazon Redshift 콘솔에서 다시 만들어 보세요. Amazon Redshift는 SLR을 자동으로 생성합니다.
모니터링	REDSHIFT-SERVERLESS-EVENT-1029	오류	사용 가능한 디스크 공간이 부족하여 작업 그룹 기본 RPU 변경을 [시간(UTC)]에 완료하지 못했습니다. 다른 구성으로 다시 시도해 보세요.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
모니터링	REDSHIFT-SERVERLESS-EVENT-1500	오류	계정의 탄력적 IP 주소 한도를 초과했기 때문에 <workgroup name> 작업 그룹을 생성하거나 업데이트할 수 없습니다. Amazon EC2에서 사용하지 않는 탄력적 IP 주소를 삭제하거나 한도 증가를 요청하세요.
모니터링	REDSHIFT-SERVERLESS-EVENT-1501	오류	<subnet id> 서브넷에 사용 가능한 IP 주소가 없습니다. 이로 인해 <workgroup name> 작업 그룹에서 EMR, 통합 쿼리, Amazon EC2 COPY/UNLOAD와 같은 쿼리 유형이 정상적으로 실행되지 않습니다. 이 문제를 해결하려면 ENI를 삭제하여 서브넷에서 IP를 확보하세요.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
모니터링	REDSHIFT-SERVERLESS-EVENT-1502	오류	<subnet id> 서브넷에 사용 가능한 IP 주소가 없습니다. 이로 인해 <workgroup name> 작업 그룹에서 Amazon EMR, Redshift 연합 쿼리, Redshift COPY/UNLOAD, Redshift ML 쿼리 유형이 정상적으로 실행되지 않습니다. 이 문제를 해결하려면 사용되지 않는 탄력적 네트워크 인터페이스(ENI)를 삭제하여 서브넷에서 IP를 확보하세요.
관리	REDSHIFT-SERVERLESS-EVENT-1008	INFO	Amazon Redshift 작업 그룹 <workgroup name>이(가) 생성되어 사용할 준비가 되었습니다.
관리	REDSHIFT-SERVERLESS-EVENT-1009	INFO	Amazon Redshift 작업 그룹 <workgroup name>이(가) <time in UTC>에 삭제되었습니다.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
모니터링	REDSHIFT-SERVERLESS-EVENT-1000	INFO	<time in UTC>에 <snapshot name> 스냅샷 생성이 완료되었습니다.
관리	REDSHIFT-SERVERLESS-EVENT-1004	INFO	<namespace name> 네임스페이스의 스냅샷에서 복원하는 작업이 <time in UTC>에 완료되었습니다.
관리	REDSHIFT-SERVERLESS-EVENT-1005	오류	<namespace name> 네임스페이스의 스냅샷에서 복원하는 작업이 <time in UTC>에 실패했습니다.
관리	REDSHIFT-SERVERLESS-EVENT-1006	INFO	<namespace name> 네임스페이스의 복구 시점에서 복원하는 작업이 <time in UTC>에 완료되었습니다.
관리	REDSHIFT-SERVERLESS-EVENT-1007	INFO	<namespace name> 네임스페이스의 복구 시점에서 복원하는 작업이 <time in UTC>에 실패했습니다.
보안	REDSHIFT-SERVERLESS-EVENT-1012	오류	Amazon Redshift가 <네임스페이스 이름> 네임스페이스의 보안 암호에 액세스할 수 없습니다.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
보안	REDSHIFT-SERVERLESS-EVENT-1013	오류	Amazon Redshift가 <네임스페이스 이름> 네임스페이스의 관리자 보안 인증 보안 암호를 암호화하는 데 사용된 KMS 키에 액세스할 수 없습니다.
보안	REDSHIFT-SERVERLESS-EVENT-1014	오류	작업 그룹에 진행 중인 작업이 있기 때문에 Amazon Redshift가 <네임스페이스 이름> 네임스페이스의 보안 암호를 교체할 수 없습니다.
보안	REDSHIFT-SERVERLESS-EVENT-1015	오류	<네임스페이스 이름> 네임스페이스에 연결된 작업 그룹이 없습니다. Amazon Redshift는 작업 그룹이 연결된 네임스페이스의 보안 암호만 교체할 수 있습니다.
보안	REDSHIFT-SERVERLESS-EVENT-1016	INFO	<네임스페이스 이름> 네임스페이스의 관리자 보안 인증 정보가 <시간(UTC)>에 업데이트되었습니다.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
보안	REDSHIFT-SERVERLESS-EVENT-1030	INFO	Amazon Redshift 네임스페이스(<네임스페이스 이름>)를 Glue Data Catalog 계정(<계정 ID>)에 등록하는 작업이 <UTC 기준 시간>에 시작되었습니다.
보안	REDSHIFT-SERVERLESS-EVENT-1031	INFO	Amazon Redshift 네임스페이스(<네임스페이스 이름>)를 Glue Data Catalog 계정(<계정 ID>)에 등록하는 작업이 <UTC 기준 시간>에 성공적으로 완료되었습니다.
보안	REDSHIFT-SERVERLESS-EVENT-1032	INFO	Amazon Redshift 네임스페이스(<네임스페이스 이름>)를 Glue Data Catalog 계정(<계정 ID>)에 등록하는 작업이 <UTC 기준 시간>에 실패했습니다.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
보안	REDSHIFT-SERVERLESS-EVENT-1033	INFO	Amazon Redshift 네임스페이스(<네임스페이스 이름>)를 Glue Data Catalog 계정(<계정 ID>)에서 등록 취소하는 작업이 <UTC 기준 시간>에 시작되었습니다.
보안	REDSHIFT-SERVERLESS-EVENT-1034	INFO	Amazon Redshift 네임스페이스(<네임스페이스 이름>)를 Glue Data Catalog 계정(<계정 ID>)에서 등록 취소하는 작업이 <UTC 기준 시간>에 성공적으로 완료되었습니다.
보안	REDSHIFT-SERVERLESS-EVENT-1035	INFO	Amazon Redshift 네임스페이스(<네임스페이스 이름>)를 Glue Data Catalog 계정(<계정 ID>)에서 등록 취소하는 작업이 <UTC 기준 시간>에 실패했습니다.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
관리	REDSHIFT-SERVERLESS-EVENT-1036	오류	Redshift Serverless 작업 그룹 <작업 그룹 이름>에서 고객이 시작한 트랙 업데이트에 실패했습니다. 작업 그룹을 원래 트랙으로 되돌립니다.
관리	REDSHIFT-SERVERLESS-EVENT-1037	오류	Redshift Serverless 작업 그룹 <작업 그룹 이름>에서 트랙 업데이트가 실패했습니다. 당시 작업 그룹이 사용 중이어서 Amazon Redshift Serverless 작업 그룹에 대한 트랙 업데이트가 실패했습니다. 따라서 작업 그룹이 원래 트랙으로 반환되었습니다. 트랙 업데이트를 다시 시도하려면 작업 그룹에서 활동이 줄어들 때까지 기다린 다음 트랙을 다시 업데이트해 보세요.
관리	REDSHIFT-SERVERLESS-EVENT-1038	INFO	Amazon Redshift 작업 그룹 <작업 그룹 이름> 트랙이 수정되었습니다. 트랙 변경이 완료되었습니다.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
네임스페이스	REDSHIFT-SERVERLESS-EVENT-1039	오류	<네임스페이스 이름> 네임스페이스에 잘못된 AWS KMS key 권한이 있습니다. 네임스페이스가 "INACCESSIBLE_KMS_KEY" 상태가 되며 더 이상 액세스할 수 없습니다.
네임스페이스	REDSHIFT-SERVERLESS-EVENT-1040	INFO	AWS KMS key 권한이 복원되어 이제 <네임스페이스 이름> 네임스페이스가 이전 상태로 복원되었습니다.
네임스페이스	REDSHIFT-SERVERLESS-EVENT-1045	INFO	서버리스 복원으로 인해 Redshift Serverless 네임스페이스 <네임스페이스 이름>의 <데이터 공유 이름> 데이터 공유가 삭제되었습니다.
네임스페이스	REDSHIFT-SERVERLESS-EVENT-1046	INFO	Redshift Serverless 네임스페이스 <네임스페이스 이름>의 데이터 공유 <데이터 공유 이름>에 대한 <소비자 위탁자 ID> 소비자가 서버리스 복원으로 인해 취소되었습니다.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
네임스페이스	REDSHIFT-SERVERLESS-EVENT-1047	INFO	서버리스 복원으로 인해 Redshift Serverless 네임스페이스 <네임스페이스 이름>의 데이터 공유 <데이터 공유 이름>에 대한 퍼블릭 액세스 가능성이 변경되었습니다.

Amazon EventBridge를 사용하여 제로 ETL 통합 이벤트 알림

제로 ETL 통합이 Amazon EventBridge를 통해 이벤트 알림을 관리하여 통합의 변경 사항에 대한 최신 정보를 제공합니다. Amazon EventBridge는 애플리케이션을 다양한 소스의 데이터와 연결하는 데 사용할 수 있는 서버리스 이벤트 버스 서비스입니다. 이 경우 이벤트 소스는 Amazon Redshift입니다. 환경에서 모니터링되는 변경 사항인 이벤트는 Amazon Redshift 데이터 웨어하우스에서 EventBridge로 자동으로 전송됩니다. 이벤트는 거의 실시간으로 전송됩니다.

EventBridge는 특정 이벤트에 대해 수행할 작업을 지정하는 이벤트 규칙을 작성할 수 있는 환경을 제공합니다. 또한 EventBridge가 이벤트를 보낼 수 있는 리소스인 대상을 설정할 수도 있습니다. 대상에는 API 대상, Amazon CloudWatch 로그 그룹 등이 포함될 수 있습니다. 규칙에 대한 자세한 내용은 [Amazon EventBridge 규칙](#)을 참조하세요. 대상에 대한 자세한 내용은 [Amazon EventBridge 대상](#)을 참조하세요.

이벤트는 심각도 및 범주로 분류될 수 있습니다. 다음과 같은 필터를 사용할 수 있습니다.

- 리소스 필터링 - 이벤트가 연결된 리소스에 따라 메시지를 수신합니다. 리소스에는 작업 그룹 또는 스냅샷이 있습니다.
- 기간 필터링 - 이벤트의 범위를 특정 기간으로 제한합니다.
- 범주 필터링 - 지정된 범주의 모든 이벤트에 대한 이벤트 알림을 받습니다.

다음 테이블에는 추가 메타데이터와 함께 제로 ETL 통합 이벤트가 나와 있습니다.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
모니터링	REDSHIFT-INTEGRATION-EVENT-0000	INFO	<통합 이름> 제로 ETL 통합이 생성되어 <시간(UTC)>에 활성화 상태가 되었습니다.
모니터링	REDSHIFT-INTEGRATION-EVENT-0001	INFO	<통합 이름> 제로 ETL 통합이 <UTC(시간)>에 삭제되었습니다.
모니터링	REDSHIFT-INTEGRATION-EVENT-0002	INFO	<통합 이름> 제로 ETL 통합 삭제를 <시간(UTC)>에 시작했습니다.
모니터링	REDSHIFT-INTEGRATION-EVENT-0003	INFO	<통합 이름> 제로 ETL 통합이 트랜잭션 데이터를 Amazon Redshift 데이터 웨어하우스에 동기화하고 있습니다.
모니터링	REDSHIFT-INTEGRATION-EVENT-0004	WARNING	테이블에 프라이머리 키가 없기 때문에 Amazon Redshift는 테이블을 복제할 수 없습니다. 소스 테이블에 프라이머리 키를 추가하면 테이블이 자동으로 다시 동기화됩니다.
모니터링	REDSHIFT-INTEGRATION-EVENT-0005	WARNING	하나 이상의 열이 지원되지 않는 데이터

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
			형식을 사용하기 때문에 Amazon Redshift가 테이블을 복제할 수 없습니다. 통합을 수정하여 필터에서 이 테이블을 제외하거나 소스 테이블에서 열을 삭제하고 'ALTER DATABASE <Redshift database name> INTEGRATION REFRESH TABLE <schema name>.<table name>'을 실행하여 이 테이블을 동기화하세요.
모니터링	REDSHIFT-INTEGRATION-EVENT-0006	오류	통합을 만들 수 없습니다. 통합을 삭제하고 다시 만드세요. 오류가 지속될 경우 AWS Support에 문의하세요.
모니터링	REDSHIFT-INTEGRATION-EVENT-0007	오류	내부 오류로 인해 데이터를 로드할 수 없습니다. 통합을 삭제하고 다시 만드세요. 오류가 지속될 경우 AWS Support에 문의하세요.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
모니터링	REDSHIFT-INTEGRATION-EVENT-0008	오류	소스 DB 클러스터에서 권한이 취소되어 권한 부여에 실패했습니다. 고객 관리형 키 (CMK)를 사용하여 통합을 암호화하는 경우 KMS 키 권한을 확인하세요. 그런 다음 통합을 삭제하고 다시 만드세요.
모니터링	REDSHIFT-INTEGRATION-EVENT-0009	오류	테이블 및 스키마 수가 Amazon Redshift 한도를 초과하므로 Amazon Redshift로 데이터를 전송할 수 없습니다. 통합을 삭제하고 다시 만드세요.
모니터링	REDSHIFT-INTEGRATION-EVENT-0012	오류	대상 서버리스 네임스페이스에서 복구 시점에서의 복원을 호출했습니다. 통합을 삭제하고 다시 만드세요.
모니터링	REDSHIFT-INTEGRATION-EVENT-0013	INFO	<통합 이름> 제로 ETL 통합이 현재 활성 상태입니다.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
모니터링	REDSHIFT-INTEGRATION-EVENT-0014	오류	<통합 이름> 통합은 내부 오류로 인해 수정할 수 없으므로 실패 상태입니다. 통합을 삭제하고 다시 만드세요. 오류가 지속될 경우 AWS Support에 문의하세요.
연산	REDSHIFT-INTEGRATION-EVENT-0016	INFO	<통합 이름> 제로 ETL 통합이 수정 요청을 처리하고 있습니다.
연산	REDSHIFT-INTEGRATION-EVENT-0017	INFO	<통합 이름> 제로 ETL 통합에 대한 수정이 적용되었습니다.
연산	REDSHIFT-INTEGRATION-EVENT-0018	WARNING	대상 Amazon Redshift 클러스터를 일시 중지하는 중입니다. 클러스터가 일시 중지될 때까지 기다린 다음 클러스터를 다시 시작하여 데이터 스트리밍을 계속하세요.
연산	REDSHIFT-INTEGRATION-EVENT-0019	WARNING	대상 Amazon Redshift 클러스터가 일시 중지되었습니다. 데이터를 계속 스트리밍하려면 클러스터를 재개해야 합니다.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
연산	REDSHIFT-INTEGRATION-EVENT-0020	WARNING	대상 Amazon Redshift 클러스터를 재개하는 중입니다. 데이터 스트리밍을 계속하려면 클러스터가 활성화될 때까지 기다리세요.
구성	REDSHIFT-INTEGRATION-EVENT-1000	오류	소스 Aurora DB 클러스터의 파라미터 하나 이상이 잘못 구성되었습니다. 파라미터 그룹을 수정하고 클러스터를 재부팅하여 변경 사항을 적용한 다음 통합을 다시 만드세요.
구성	REDSHIFT-INTEGRATION-EVENT-1001	오류	enable_case_sensitive_identifier 파라미터의 값이 잘못되어 통합에 실패했습니다. 소스 Aurora DB 클러스터의 값을 true로 설정한 다음 통합을 삭제하고 다시 만드세요.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
구성	REDSHIFT-INTEGRATION-EVENT-1002	오류	cdc_insert_enabled 파라미터의 값이 잘못되어 통합에 실패했습니다. 소스 Aurora DB 클러스터의 값을 true로 설정한 다음 통합을 삭제하고 다시 만드세요.
구성	REDSHIFT-INTEGRATION-EVENT-1003	오류	소스 DB 클러스터 파라미터 그룹의 binlog_format 파라미터는 ROW로 설정해야 합니다. 파라미터 그룹을 수정하고 클러스터를 재부팅하여 변경 사항을 적용한 다음 통합을 다시 만드세요.
구성	REDSHIFT-INTEGRATION-EVENT-1004	오류	binlog_transaction_compression 클러스터 파라미터가 활성화되었으므로 데이터를 로드할 수 없습니다. 파라미터 값을 OFF로 설정하고 라이터 인스턴스를 재부팅하여 변경 사항을 적용한 다음 통합을 다시 만드세요.

Amazon Redshift 범주	외부 이벤트 ID	이벤트 심각도	메시지 설명
구성	REDSHIFT-INTEGRATION-EVENT-1005	오류	binlog_row_value_options 클러스터 파라미터가 지원되지 않는 PARTIAL_JSON으로 설정되어 있기 때문에 데이터를 로드할 수 없습니다. 파라미터 그룹을 수정하고 라이터 인스턴스를 재부팅하여 변경 사항을 적용한 다음 통합을 다시 만드세요.
구성	REDSHIFT-INTEGRATION-EVENT-1006	WARNING	통합 필터를 구문 분석할 수 없습니다. 필터 구문을 수정하세요.

Amazon Redshift의 할당량 및 제한

Amazon Redshift에는 AWS 리전별로 AWS 계정의 여러 리소스 사용을 제한하는 할당량이 있습니다. 각 할당량에 대한 기본값이 있으며 일부 할당량은 조정할 수 있습니다.

Amazon Redshift 객체에 대한 할당량

Amazon Redshift에는 여러 객체 유형 사용을 제한하는 할당량이 있습니다. 각각에 대한 기본값이 있습니다.

할당량 이름	AWS 기본 값	조정 가능	설명
AWS스냅샷당 스냅샷을 복원하도록 권한을 부여할 수 있는 계정	20	아니요	스냅샷당 스냅샷을 복원하도록 권한을 부여할 수 있는 최대 AWS 계정 수입니다.
AWS별로 스냅샷을 복원하도록 권한을 부여할 수 있는 계정 AWS KMS key	100	아니요	KMS 키당 스냅샷을 복원하도록 권한을 부여할 수 있는 최대 AWS 계정 수입니다. 다시 말해서 단일 KMS 키로 암호화되어 있는 스냅샷이 10개라고 가정하면 AWS 계정 10개에 각 스냅샷을 복원할 수 있는 권한을 부여하거나, 혹은 최대 100개까지 계정을 추가하지만 각 스냅샷마다 계정 20개를 초과하지 않는 기타 조합에 복원 권한을 부여할 수 있습니다.
Amazon Redshift가 다른 AWS 서비스에 액세스하기 위한 클러	50 ¹	아니요	클러스터 및 IAM 역할을 소유한 사용자의 다른 AWS 서비스에 액세스할 수 있도록 Amazon Redshift에 권한을 부여하기 위해 클러스터와 연결할 수 있는 최대 IAM 역할 수입니다. ¹ us-iso-east-1, us-iso-west-1, us-isob-east-1 AWS 리전에서는 할당량이 10입니다.

할당량 이름	AWS 기본 값	조정 가능	설명
스터 IAM 역할			
모든 사용자 정의 수동 WLM 대기열에 대한 동시성 수준(쿼리 슬롯)	50	아니요	수동 워크로드 관리로 정의된 모든 사용자 정의 대기열에 대한 최대 쿼리 슬롯입니다.
동시성 확장 클러스터	10	예	동시성 확장 클러스터의 최대 수입니다.
클러스터의 DC2 노드	128	예	클러스터에 할당할 수 있는 최대 DC2 노드 수입니다. 노드 유형에 따른 노드 제한에 대한 자세한 내용은 Amazon Redshift의 클러스터 및 노드 섹션을 참조하세요.
이벤트 구독	20	예	현재 AWS 리전에 있는 이 계정에 대한 최대 이벤트 구독 수입니다.
노드	200	예	현재 AWS 리전에 있는 이 계정에 대한 모든 데이터베이스 인스턴스의 최대 노드 수입니다.
파라미터 그룹	20	아니요	현재 AWS 리전에 있는 이 계정에 대한 최대 파라미터 그룹 수입니다.
클러스터의 RA3 노드	128	예	클러스터에 할당할 수 있는 최대 RA3 노드 수입니다. 노드 유형에 따른 노드 제한에 대한 자세한 내용은 Amazon Redshift의 클러스터 및 노드 섹션을 참조하세요.

할당량 이름	AWS 기본 값	조정 가능	설명
클러스터에 연결된 Redshift 관리형 VPC 엔드포인트	30	예	클러스터에 연결할 수 있는 최대 Redshift 관리형 VPC 엔드포인트 수입니다. Redshift 관리형 VPC 엔드포인트에 대한 자세한 내용은 Redshift 관리형 VPC 엔드포인트 섹션을 참조하세요.
Redshift 관리형 VPC 엔드포인트를 통해 액세스되는 클러스터에 대한 피부여자	10	예	클러스터 소유자가 클러스터에 대한 Redshift 관리형 VPC 엔드포인트 생성 권한을 부여할 수 있는 최대 피부여자 수입니다. Redshift 관리형 VPC 엔드포인트에 대한 자세한 내용은 Redshift 관리형 VPC 엔드포인트 섹션을 참조하세요.
권한 부여당 Redshift 관리형 VPC 엔드포인트	10	예	권한 부여별로 생성할 수 있는 최대 Redshift 관리형 VPC 엔드포인트 수입니다. Redshift 관리형 VPC 엔드포인트에 대한 자세한 내용은 Redshift 관리형 VPC 엔드포인트 섹션을 참조하세요.
예약 노드	200	예	현재 AWS 리전에 있는 이 계정에 대한 최대 예약 노드 수입니다.
클러스터당 각 데이터베이스의 스키마	9,900	아니요	각 데이터베이스에서 클러스터당 생성할 수 있는 최대 스키마 수입니다. 그러나 pg_temp_* 스키마는 이 할당량에 포함되지 않습니다.
보안 그룹	20	예	현재 AWS 리전에 있는 이 계정에 대한 최대 보안 그룹 수입니다.
COPY로 로드할 때 단일 행 크기	4	아니요	COPY 명령을 사용하여 로드할 때 단일 행의 최대 크기 (MB)입니다.

할당량 이름	AWS 기본 값	조정 가능	설명
스냅샷	700	예	현재 AWS 리전에 있는 이 계정에 대한 최대 사용자 스냅샷 수입입니다.
서브넷 그룹	50	예	현재 AWS 리전에 있는 이 계정에 대한 최대 서브넷 그룹 수입입니다.
서브넷 그룹의 서브넷	20	예	서브넷 그룹의 최대 서브넷 수입입니다.
large 클러스터 노드 유형에 대한 테이블	9,900	아니요	대형 클러스터 노드 유형의 최대 테이블 수입입니다. 이 제한에는 영구 테이블, 임시 테이블, datashare 테이블 및 구체화된 보기가 포함됩니다. 외부 테이블은 임시 테이블로 계산됩니다. 임시 테이블에는 사용자 정의 임시 테이블을 비롯해 쿼리 처리 또는 시스템 유지 관리 시 Amazon Redshift에서 생성된 임시 테이블이 포함됩니다. 보기 및 시스템 테이블은 이러한 제한에 포함되지 않습니다.
xlarge 클러스터 노드 유형에 대한 테이블	9,900	아니요	xlarge 클러스터 노드 유형의 최대 테이블 수입입니다. 이 제한에는 영구 테이블, 임시 테이블, datashare 테이블 및 구체화된 보기가 포함됩니다. 외부 테이블은 임시 테이블로 계산됩니다. 임시 테이블에는 사용자 정의 임시 테이블을 비롯해 쿼리 처리 또는 시스템 유지 관리 시 Amazon Redshift에서 생성된 임시 테이블이 포함됩니다. 보기 및 시스템 테이블은 이러한 제한에 포함되지 않습니다.

할당량 이름	AWS 기본 값	조정 가능	설명
테이블 x1plus 싱글 노드 클러스터 를 사용하 는 클러스 터 노드 유 형입니다.	9,900	아니요	싱글 노드 클러스터가 있는 x1plus 클러스터 노드 유형의 최대 테이블 수입니다. 이 제한에는 영구 테이블, 임시 테이블, datashare 테이블 및 구체화된 보기가 포함됩니다. 외부 테이블은 임시 테이블로 계산됩니다. 임시 테이블에는 사용자 정의 임시 테이블을 비롯해 쿼리 처리 또는 시스템 유지 관리 시 Amazon Redshift에서 생성된 임시 테이블이 포함됩니다. 보기 및 시스템 테이블은 이러한 제한에 포함되지 않습니다.
테이블 x1plus 다중 노드 클러스터 가 있는 클 러스터 노 드 유형입 니다.	20,000건	아니요	다중 노드 클러스터가 있는 x1plus 클러스터 노드 유형의 최대 테이블 수입니다. 이 제한에는 영구 테이블, 임시 테이블, datashare 테이블 및 구체화된 보기가 포함됩니다. 외부 테이블은 임시 테이블로 계산됩니다. 임시 테이블에는 사용자 정의 임시 테이블을 비롯해 쿼리 처리 또는 시스템 유지 관리 시 Amazon Redshift에서 생성된 임시 테이블이 포함됩니다. 보기 및 시스템 테이블은 이러한 제한에 포함되지 않습니다.
4xlarge 클러스터 노드 유형 에 대한 테 이블	200,000	아니요	4xlarge 클러스터 노드 유형의 최대 테이블 수입니다. 이 제한에는 영구 테이블, 임시 테이블, datashare 테이블 및 구체화된 보기가 포함됩니다. 외부 테이블은 임시 테이블로 계산됩니다. 임시 테이블에는 사용자 정의 임시 테이블을 비롯해 쿼리 처리 또는 시스템 유지 관리 시 Amazon Redshift에서 생성된 임시 테이블이 포함됩니다. 보기 및 시스템 테이블은 이러한 제한에 포함되지 않습니다.

할당량 이름	AWS 기본 값	조정 가능	설명
8xlarge 클러스터 노드 유형에 대한 테이블	200,000	아니요	8xlarge 클러스터 노드 유형의 최대 테이블 수입니다. 이 제한에는 영구 테이블, 임시 테이블, datashare 테이블 및 구체화된 보기가 포함됩니다. 외부 테이블은 임시 테이블로 계산됩니다. 임시 테이블에는 사용자 정의 임시 테이블을 비롯해 쿼리 처리 또는 시스템 유지 관리 시 Amazon Redshift에서 생성된 임시 테이블이 포함됩니다. 보기 및 시스템 테이블은 이러한 제한에 포함되지 않습니다.
16xlarge 클러스터 노드 유형에 대한 테이블	200,000	아니요	16xlarge 클러스터 노드 유형의 최대 테이블 수입니다. 이 제한에는 영구 테이블, 임시 테이블, datashare 테이블 및 구체화된 보기가 포함됩니다. 외부 테이블은 임시 테이블로 계산됩니다. 임시 테이블에는 사용자 정의 임시 테이블을 비롯해 쿼리 처리 또는 시스템 유지 관리 시 Amazon Redshift에서 생성된 임시 테이블이 포함됩니다. 보기 및 시스템 테이블은 이러한 제한에 포함되지 않습니다.
데이터베이스 수	60	아니요	Amazon Redshift 클러스터에서 허용되는 최대 데이터베이스 수입니다. 여기에는 데이터 공유에서 생성된 데이터베이스는 제외됩니다.
유휴 또는 비활성 세션에 대한 시간 제한	4시간	아니요	이 설정은 클러스터에 적용됩니다. 사용자에게 대한 유휴 세션 시간 제한 값을 설정하는 방법에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 ALTER USER 섹션을 참조하세요. 사용자 설정이 클러스터 설정보다 우선합니다.
유휴 트랜잭션에 대한 시간 제한	6시간	아니요	Amazon Redshift가 트랜잭션과 연결된 세션을 종료하기 전에 진행 중인 트랜잭션의 최대 비활성 기간입니다. 이 설정은 사용자가 정의한 유휴 시간 제한 설정보다 우선 적용됩니다. 클러스터에 적용됩니다.

할당량 이름	AWS 기본 값	조정 가능	설명
데이터베이스의 저장 프로시저	10,000개	아니요	최대 저장 프로시저 수입니다. 추가 제한은 저장 프로시저 지원에 대한 제한 및 차이점 섹션을 참조하세요.
RA3 노드에 대한 최대 연결 수	2,000	아니요	RA3 클러스터에 대한 최대 연결 수입니다. 허용되는 최대 연결은 노드 유형에 따라 다릅니다.
DC2 노드에 대한 최대 연결 수	다양	아니요	dc2.large 클러스터에 대한 최대 연결 수는 500개입니다. dc2.8xlarge 클러스터에 대한 최대 연결 수는 2,000개입니다.
클러스터 내 Amazon Redshift 역할 수	1,000	예	클러스터별로 생성할 수 있는 최대 Amazon Redshift 역할 수입니다. 역할 기반 액세스 제어(RBAC) 역할에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 역할 기반 액세스 제어(RBAC) 를 참조하세요.

Amazon Redshift Serverless 객체 할당량

Amazon Redshift에는 Amazon Redshift Serverless 네임스페이스에서의 여러 객체 유형 사용을 제한하는 할당량이 있습니다. 각각에 대한 기본값이 있습니다.

할당량 이름	AWS 기본 값	조정 가능	설명
데이터베이스 수	100	아니요	Amazon Redshift Serverless 네임스페이스에서 허용되는 최대 데이터베이스 수입니다. 여기에는 데이터 공유에서 생성된 데이터베이스는 제외됩니다.
스키마 수	9,900	아니요	Amazon Redshift Serverless 네임스페이스에서 허용되는 최대 스키마 수입니다.

할당량 이름	AWS 기본 값	조정 가능	설명
테이블 수	200,000	아니요	Amazon Redshift Serverless 네임스페이스에서 허용되는 최대 테이블 수입니다.
4개의 Redshift 처리 장치 (RPU)를 사용하는 데이터 웨어하우스용 Redshift 관리형 스토리지 단위	최대 32TB	아니요	4RPU가 있는 데이터 웨어하우스에 지원되는 최대 RMS 스토리지입니다. 32TB 한도 내에서 Amazon Redshift는 내부적으로 수백 MB의 스토리지를 사용합니다.
8개, 16개 또는 24개의 Redshift 처리 장치 (RPU)를 사용하는 데이터 웨어하우스용 Redshift 관리형 스토리지 단위	최대 128TB	아니요	8RPU, 16RPU 또는 24RPU가 있는 데이터 웨어하우스에 지원되는 최대 RMS 스토리지입니다. 128TB 한도 내에서 Amazon Redshift는 내부적으로 수백 MB의 스토리지를 사용합니다.
유희 또는 비활성 세션에 대한 시간 제한	1시간	아니요	사용자에 대한 유희 세션 시간 제한 값을 설정하는 방법에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 ALTER USER 섹션을 참조하세요. 사용자 설정이 우선 적용됩니다.
실행 중인 쿼리의 제한 시간	86,399초(24시간)	아니요	Amazon Redshift가 쿼리를 종료하기 전에 실행 중인 쿼리의 최대 시간입니다.

할당량 이름	AWS 기본 값	조정 가능	설명
유휴 트랜잭션에 대한 시간 제한	6시간	아니요	Amazon Redshift Serverless가 트랜잭션과 연결된 세션을 종료하기 전에 진행 중인 트랜잭션의 최대 비활성 기간입니다. 이 설정은 사용자가 정의한 유휴 시간 제한 설정보다 우선 적용됩니다.
최대 연결 수	2000	아니요	작업 그룹에 연결할 수 있는 최대 연결 수입니다.
작업 그룹 수	25	예	지원되는 작업 그룹 수입니다.
네임스페이스 수	25	예	지원되는 네임스페이스 수입니다.
작업 그룹 내 Amazon Redshift 역할 수	1,000	예	작업 그룹별로 생성할 수 있는 최대 Amazon Redshift 역할 수입니다. 역할 기반 액세스 제어(RBAC) 역할에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 역할 기반 액세스 제어(RBAC) 를 참조하세요.
AWS 계정당 기본 Redshift 처리 장치(RPU) 수	지난 6개월 동안의 최대 집계 기본 RPU의 1.5배 또는 3,200RPU 중 더 큰 값입니다.	예	Redshift Serverless 기본 RPU의 AWS 계정 할당량은 지난 6개월 동안의 최대 집계 기본 RPU의 1.5배 또는 3,200RPU 중 더 큰 값입니다.

시간 초과 구성이 Amazon Redshift Serverless 청구에 어떤 영향을 미치는지에 대한 자세한 내용은 [Amazon Redshift Serverless에 대한 청구](#) 섹션을 참조하세요.

Amazon Redshift 데이터 API의 할당량

Amazon Redshift에는 Redshift 데이터 API 사용을 제한하는 할당량이 있습니다. 각각에 대한 기본값이 있습니다. Amazon Redshift 데이터 API에 대한 자세한 내용은 [Amazon Redshift Data API 사용](#) 섹션을 참조하세요.

할당량 이름	AWS 기본 값	조정 가능	설명
BatchExecuteStatement API의 초당 트랜잭션 (TPS)	20	아니요	조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.
CancelStatement API의 초당 트랜잭션 (TPS)	3	아니요	조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.
DescribeStatement API의 초당 트랜잭션 (TPS)	100	아니요	조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.
DescribeTable API의 초당 트랜잭션 (TPS)	3	아니요	조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.
ExecuteStatement API의 초당	30	아니요	조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.

할당량 이름	AWS 기본 값	조정 가능	설명
트랜잭션 (TPS)			
GetStatementResult API의 초당 트랜잭션(TPS)	20	아니요	조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.
ListDatabases API의 초당 트랜잭션 (TPS)	3	아니요	조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.
ListSchemas API의 초당 트랜잭션(TPS)	3	아니요	조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.
ListStatements API의 초당 트랜잭션 (TPS)	3	아니요	조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.
ListTables API의 초당 트랜잭션(TPS)	3	아니요	조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.

쿼리 편집기 v2 객체에 대한 할당량

Amazon Redshift에는 Amazon Redshift 쿼리 편집기 v2에서의 여러 객체 유형 사용을 제한하는 할당량이 있습니다. 각각에 대한 기본값이 있습니다.

할당량 이름	AWS 기본 값	조정 가능	설명
연결	500	예	현재 리전의 이 계정에서 쿼리 편집기 v2를 사용하여 생성할 수 있는 최대 연결 수입니다.
계정당 활성 보안 주체	50	예	현재 리전의 이 계정에서 쿼리 에디터 v2를 사용할 수 있는 최대 동시 보안 주체 수입니다.
저장된 쿼리	2,500	예	현재 리전의 이 계정에서 쿼리 편집기 v2를 사용하여 생성할 수 있는 최대 저장된 쿼리 수입니다.
쿼리 버전	20	예	현재 리전의 이 계정에서 쿼리 편집기 v2를 사용하여 생성할 수 있는 쿼리당 최대 버전 수입니다.
저장된 차트	500	예	현재 리전의 이 계정에서 쿼리 편집기 v2를 사용하여 생성할 수 있는 최대 저장된 차트 수입니다.
쿼리당 가져온 데이터 크기	100	아니요	현재 리전에서 이 계정의 쿼리 편집기 v2가 가져온 쿼리당 최대 데이터 크기(메가바이트 단위)입니다.
행 100개당 가져온 데이터 크기	5	아니요	이 계정 및 현재 리전에서 쿼리 편집기 v2가 가져온 쿼리 결과 페이지 하나(행 100개)당 최대 데이터 크기(메가바이트 단위)입니다.
최대 동시 연결	3	아니요	사용자당 최대 데이터베이스 연결 수(격리된 세션 포함)입니다. 이 값은 쿼리 편집기 v2 관리자가 Account settings(계정 설정)에서 1~10으로 설정할 수 있습니다. 관리자가 설정한 한도에 도달하면, SQL을 실행할 때 격리된 세션 대신 공유 세션을 사용하는 것이 좋습니다. 연결에 대한 자세한 내용은 쿼리 편집기 v2 열기 섹션을

할당량 이름	AWS 기본 값	조정 가능	설명
			참조하세요. 제한 설정에 대한 자세한 내용을 알아보려면 계정 설정 섹션을 참조하세요.

Amazon Redshift Spectrum 객체 할당량 및 제한

Amazon Redshift Spectrum에는 다음과 같은 할당량과 제한이 있습니다.

- AWS Glue Data Catalog를 사용할 때 AWS 계정당 최대 데이터베이스 수입입니다. 이 값은 Amazon Web Services 일반 참조의 [AWS Glue 서비스 할당량](#)을 참조하세요.
- AWS Glue Data Catalog를 사용할 때 데이터베이스당 최대 테이블 수입입니다. 이 값은 Amazon Web Services 일반 참조의 [AWS Glue 서비스 할당량](#)을 참조하세요.
- AWS Glue Data Catalog를 사용할 때 테이블당 최대 파티션 수입입니다. 이 값은 Amazon Web Services 일반 참조의 [AWS Glue 서비스 할당량](#)을 참조하세요.
- AWS Glue Data Catalog를 사용할 때 AWS 계정당 최대 파티션 수입입니다. 이 값은 Amazon Web Services 일반 참조의 [AWS Glue 서비스 할당량](#)을 참조하세요.
- AWS Glue Data Catalog를 사용할 때 외부 테이블의 최대 열 수, 가상 열이 활성화된 경우 1,597, 가상 열이 활성화되지 않은 경우 1,600입니다.
- AWS Glue Data Catalog를 사용할 때 ION 또는 JSON 파일에서 문자열 값의 최대 크기는 16KB입니다. 이 한도에 도달하면 문자열이 잘릴 수 있습니다.
- ALTER TABLE 문을 사용하여 최대 100개의 파티션을 추가할 수 있습니다.
- 모든 S3 데이터는 Amazon Redshift 클러스터와 동일한 AWS 리전에 있어야 합니다.
- ION 및 JSON 타임스탬프는 [ISO8601](#) 형식을 사용해야 합니다.
- ORC 파일의 외부 압축은 지원되지 않습니다.
- 텍스트, OpenCSV 및 Regex SERDEs는 '177'보다 큰 8진수 구분 기호를 지원하지 않습니다.
- 파티션 열에 조건자를 지정하여 모든 파티션에서 읽지 않도록 해야 합니다.

예를 들어 다음 조건자는 ship_dtm 열을 필터링하지만 이 필터를 ship_yyyymm 파티션 열에는 적용하지 않습니다.

```
WHERE ship_dtm > '2018-04-01'.
```

필요 없는 파티션을 건너뛰려면 WHERE ship_yyyymm = '201804' 조건자를 추가해야 합니다. 이 조건자는 \ship_yyyymm=201804\ 파티션에 대한 읽기 작업을 제한합니다.

Apache Hive 메타스토어에는 이러한 제한이 적용되지 않습니다.

명명 제약 조건

다음 표는 Amazon Redshift의 명명 제약 조건을 설명한 것입니다.

클러스터 식별자	- 클러스터 식별자는 소문자로만 구성되어야 합니다. 1-63자의 영숫자 또는 하이픈으로 구성되어야 합니다. - 첫 번째 문자는 글자이어야 합니다. - 하이픈으로 끝나거나 하이픈이 2개 연속으로 이어져서는 안 됩니다. - AWS 계정에 속한 모든 클러스터와 비교하여 고유해야 합니다.
데이터베이스 이름	- 데이터베이스 이름은 1-64자의 영숫자로 구성되어야 합니다. - 소문자로만 구성되어야 합니다. - 예약어가 될 수는 없습니다. 예약어 목록은 Amazon Redshift 데이터베이스 개발자 안내서의 예약어 섹션을 참조하세요.
Redshift 관리형 VPC 엔드포인트의 엔드포인트 이름	- 엔드포인트 이름은 1~30자를 포함해야 합니다. - 유효한 문자는 A~Z, a~z, 0~9 및 하이픈(-)입니다. - 첫 번째 자리는 문자여야 합니다.

	이름은 2개 연속 하이픈을 포함하거나 하이픈으로 끝날 수 없습니다.
관리자 이름	- 관리자 이름은 소문자로만 구성해야 합니다. 1-128자의 영숫자로 구성되어야 합니다. - 첫 번째 문자는 글자이어야 합니다. - 예약어가 될 수는 없습니다. 예약어 목록은 Amazon Redshift 데이터베이스 개발자 안내서의 예약어 섹션을 참조하세요.
관리자 암호	- 관리자 암호는 8-64자로 구성해야 합니다. - 마스터 암호 - 마스터 암호 - 숫자를 1개 포함해야 합니다. '(작은따옴표), "(큰따옴표), \, / 또는 @을 제외하고 모든 ASCII 문자(ASCII 코드 33~126)를 사용할 수 있습니다.
파라미터 그룹 이름	- 파라미터 그룹 이름은 1-255자의 영숫자 또는 하이픈으로 구성되어야 합니다. - 소문자로만 구성되어야 합니다. - 첫 번째 문자는 글자이어야 합니다. - 하이픈으로 끝나거나 하이픈이 2개 연속으로 이어져서는 안 됩니다.

클러스터 보안 그룹 이름	• 클러스터 보안 그룹 이름은 255자 미만의 영숫자 문자 또는 하이픈으로 구성되어야 합니다. • 소문자로만 구성되어야 합니다. • Default는 사용하면 안 됩니다. • AWS 계정에서 생성되는 모든 보안 그룹과 비교하여 고유해야 합니다.
서브넷 그룹 이름	• 서브넷 그룹 이름은 255자 미만의 영숫자 문자 또는 하이픈으로 구성되어야 합니다. • 소문자로만 구성되어야 합니다. • Default는 사용하면 안 됩니다. • AWS 계정에서 생성되는 모든 서브넷 그룹과 비교하여 고유해야 합니다.
클러스터 스냅샷 식별자	• 클러스터 스냅샷 식별자는 255자 미만의 영숫자 문자 또는 하이픈으로 구성되어야 합니다. • 소문자로만 구성되어야 합니다. • Default는 사용하면 안 됩니다. • AWS 계정에서 생성되는 모든 스냅샷 식별자와 비교하여 고유해야 합니다.

Amazon Redshift에서 리소스 태그

AWS에서 태그는 키-값 페어로 구성된 사용자 정의 레이블입니다. Amazon Redshift는 태그 지정을 지원하여 리소스에 대한 메타데이터를 한 눈에 제공하고 비용 할당을 기반으로 결제 보고서를 분류합니다. 태그를 비용 할당에 사용하려면 먼저 AWS 결제 및 비용 관리 서비스에서 보고서 태그를 활성화해야 합니다. 결제 목적으로 태그를 설정 및 사용하는 방법에 대한 자세한 내용은 [사용자 지정 결제 보고서에 대한 비용 할당 태그 사용](#) 및 [월별 비용 할당 보고서 설정](#)을 참조하십시오.

Amazon Redshift 리소스에서는 태그가 필요하지 않지만 컨텍스트를 이해하는 데 도움이 될 수 있습니다. 비용 센터나 프로젝트 이름, 또는 기타 리소스 관련 정보에 대한 메타데이터를 사용해 리소스에 태그를 지정하기도 합니다. 예를 들어 테스트 환경과 프로덕션 환경에 속하는 리소스를 추적한다고 가정하겠습니다. 그렇다면 environment라는 이름의 키를 생성한 후 test 또는 production 값을 입력하여 각 환경에서 사용하는 리소스를 식별할 수 있습니다. 다른 AWS 서비스에서 태그를 지정하거나, 혹은 기본적인 비즈니스 범주가 있는 경우에는 Amazon Redshift 리소스에서도 일관성 유지를 위해 동일한 키-값 페어를 생성하는 것이 좋습니다.

태그는 클러스터 크기를 조정하거나, 혹은 동일한 리전에 속한 클러스터의 스냅샷을 복원한 후에도 리소스에 그대로 남습니다. 하지만 스냅샷을 다른 리전으로 복사하는 경우에는 태그가 사라지기 때문에 새로운 리전에서 태그를 다시 생성해야 합니다. 리소스를 삭제하는 경우에는 연결된 태그까지 모두 삭제됩니다.

각 리소스마다 태그 집합이 하나씩 있습니다. 여기에서 태그 집합이란 리소스에 할당되는 태그들이 하나 이상 모인 집합을 말합니다. 각 리소스는 태그 집합당 최대 50개의 태그를 가질 수 있습니다. 리소스를 생성할 때, 그리고 리소스가 생성된 후에도 태그 추가는 가능합니다. Amazon Redshift에서 태그를 추가할 수 있는 리소스 유형은 다음과 같습니다.

- CIDR/IP
- 클러스터
- 클러스터 보안 그룹
- 클러스터 보안 그룹 수신 규칙
- Amazon EC2 보안 그룹
- 하드웨어 보안 모듈(HSM) 연결
- HSM 클라이언트 인증서
- Parameter Group
- 스냅샷

- 서버넷 그룹
- 통합(제로 ETL 통합)

Amazon Redshift 콘솔에서 태그 지정을 사용하기 위해 사용자는 AWS 관리형 정책 AmazonRedshiftFullAccess를 연결할 수 있습니다. Amazon Redshift 콘솔 사용자에게 연결할 수 있는 태그 지정 권한이 제한된 예제 IAM 정책은 [예제 7: 사용자가 Amazon Redshift 콘솔로 리소스에 태그를 지정할 수 있도록 허용](#) 섹션을 참조하세요. 태그 지정에 대한 자세한 내용은 [AWS Resource Groups란 무엇입니까?](#)를 참조하세요.

태그 지정 요구 사항

태그를 지정할 때 요건은 다음과 같습니다.

- 키는 접두사로 `aws:`를 사용할 수 없습니다.
- 키는 태그 집합에 대해 고유해야 합니다.
- 키는 1~128자 사이의 허용된 문자이어야 합니다.
- 값은 0~256자 사이의 허용된 문자이어야 합니다.
- 값은 태그 집합마다 고유할 필요는 없습니다.
- 키와 값의 문자로는 Unicode 문자, 숫자, 공백 그리고 다음 기호가 허용됩니다. `_ . : / = + - @`.
- 키와 값은 대/소문자를 구분합니다.

리소스 태그 관리

다음 절차는 리소스 태그 사용 방법을 안내합니다.

Amazon Redshift 리소스에 대한 태그를 관리하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/redshiftv2/>에서 Amazon Redshift 콘솔을 엽니다.
2. 탐색 메뉴에서 구성(Configurations)을 선택한 다음 태그 관리(Manage tags)를 선택합니다.
3. 리소스에 대한 선택 사항을 입력하고 추가, 수정 또는 삭제할 태그를 선택합니다. 그런 다음 Manage tags of the resources that you chose(선택된 리소스의 태그 관리)를 선택합니다.

태그를 지정할 수 있는 리소스에는 클러스터, 파라미터 그룹, 서버넷 그룹, HSM 클라이언트 인증서, HSM 연결 및 스냅샷이 포함됩니다.

4. [태그 관리(Manage tags)] 탐색 페이지에서 [태그 변경 사항 검토 및 적용(Review and apply tag changes)]를 선택하고 [적용(Apply)]을 선택하여 변경 사항을 저장합니다.

Amazon Redshift와 AWS Backup의 통합

AWS Backup은 AWS 서비스, 클라우드 및 온프레미스에서 데이터 보호를 중앙 집중화하고 자동화하는 데 도움이 되는 완전관리형 서비스입니다.

Amazon Redshift용 AWS Backup을 사용하면 데이터 보호 정책을 구성하고 다양한 Amazon Redshift 리소스에 대한 활동을 한 곳에서 모니터링할 수 있습니다. Amazon Redshift 프로비저닝된 클러스터 및 서버리스 네임스페이스에서 스냅샷을 생성하고 저장할 수도 있습니다. 이를 통해 수동 프로세스 없이 이전에 별도로 수행해야 했던 백업 작업을 자동화하고 통합할 수 있습니다.

Note

RA3 프로비저닝된 클러스터 및 Amazon Redshift Serverless 작업 그룹에는 백업 없음 테이블이 지원되지 않습니다. RA3 클러스터 또는 서버리스 작업 그룹에서 백업 없음으로 표시된 테이블은 스냅샷을 생성하는 동안 항상 백업되고 스냅샷에서 복원할 때 항상 복원되는 영구 테이블로 처리됩니다. 백업이 없는 테이블의 스냅샷 비용을 방지하려면 스냅샷을 생성하기 전에 자릅니다.

백업 또는 복구 시점은 지정된 시간의 Amazon Redshift 클러스터와 같은 리소스의 콘텐츠를 나타냅니다. AWS Backup은 백업 볼트에 백업을 저장하며, 이는 비즈니스 요구 사항에 따라 구성할 수 있습니다. 복구 시점과 백업은 서로 통용되는 용어입니다. AWS Backup에 대한 자세한 내용은 AWS Backup 개발자 안내서의 [백업의 생성, 유지 관리 및 복원](#)을 참조하세요.

Amazon Redshift는 기본적으로 AWS Backup과 통합됩니다. 이를 통해 백업 계획을 정의하고 백업 계획에 Amazon Redshift 리소스를 할당할 수 있습니다. AWS Backup은 Amazon Redshift 수동 스냅샷 생성을 자동화하고 백업 계획에서 지정한 백업 볼트에 이러한 스냅샷을 안전하게 저장합니다. 볼트에 대한 자세한 내용은 AWS Backup 개발자 안내서의 [백업 볼트](#)를 참조하세요. 백업 계획에서 백업 빈도, 백업 기간, 수명 주기 또는 백업 볼트를 정의할 수 있습니다. 백업 계획에 대한 자세한 내용은 AWS Backup 백업 개발자 안내서의 [백업 계획](#)을 참조하세요.

AWS Backup을 사용하지 않고 Amazon Redshift Serverless 스냅샷을 생성하고 복원하는 방법에 대한 자세한 내용은 [스냅샷 및 복구 시점](#) 섹션을 참조하세요. AWS Backup을 사용하지 않고 Amazon Redshift 프로비저닝된 클러스터 스냅샷을 생성하고 복원하는 방법에 대한 자세한 내용은 [Amazon Redshift 스냅샷 및 백업](#) 섹션을 참조하세요.

주제

- [Amazon Redshift와 함께 AWS Backup 사용 시 고려 사항](#)
- [Amazon Redshift와 함께 AWS Backup 사용 시 제한 사항](#)
- [Amazon Redshift를 사용한 AWS Backup 관리](#)

Amazon Redshift와 함께 AWS Backup 사용 시 고려 사항

Amazon Redshift와 함께 AWS Backup 사용 시 다음 사항을 고려하세요.

- Amazon Redshift용 AWS Backup은 동일한 AWS 리전에서 AWS Backup과 Amazon Redshift를 모두 사용할 수 있는 경우에 사용할 수 있습니다. AWS Backup을 사용할 수 있는 곳에 대한 자세한 내용은 AWS 일반 참조의 [Amazon Redshift 엔드포인트 및 할당량](#)을 참조하세요.
- AWS Backup 사용을 시작하려면 모든 전제 조건을 충족했는지 확인하세요. 자세한 내용은 AWS Backup 개발자 안내서의 [사전 조건](#)을 참조하세요.
- 적극적으로 AWS Backup 서비스를 옵트인하세요. 옵트인 선택은 특정 계정 및 AWS 리전 리전에 적용됩니다. 하나의 계정으로 여러 리전에서 백업을 사용하려면 해당 계정으로 개별 리전을 옵트인해야 합니다. 자세한 내용은 AWS Backup 개발자 안내서의 [AWS Backup으로 서비스 관리 옵트인](#)을 참조하세요.
- Amazon Redshift의 AWS Backup 통합은 프로비저닝된 클러스터 및 서버리스 네임스페이스에 대한 수동 스냅샷만 지원합니다.
- AWS Backup을 사용하여 스냅샷 설정을 관리하면 Amazon Redshift를 사용하여 수동 스냅샷 설정을 계속 관리할 수 없습니다. 대신 AWS Backup 계획을 사용하여 설정을 계속 관리할 수 있습니다. 자세한 내용은 AWS Backup 개발자 안내서의 [백업 계획](#)을 참조하세요.
- 전체 데이터 웨어하우스 스냅샷을 서버리스 네임스페이스로 복원하는 것은 파괴적인 변경입니다. 데이터 웨어하우스 스냅샷을 해당 네임스페이스로 복원하면 대상 네임스페이스의 기존 데이터가 모두 손실됩니다. 이는 데이터 웨어하우스 스냅샷 복원에만 적용됩니다. 단일 테이블 스냅샷을 네임스페이스로 복원해도 기존 데이터는 삭제되지 않습니다.
- 스냅샷을 프로비저닝된 클러스터로 복원하려면 `RestoreFromClusterSnapshot` 권한이 있는 IAM 정책이 있어야 합니다. 스냅샷을 서버리스 네임스페이스로 복원하려면 `RestoreFromSnapshot` 권한이 있는 IAM 정책이 있어야 합니다. 이러한 권한은 소스 스냅샷 유형이 아닌 대상 데이터 웨어하우스 유형에 적용됩니다. 예를 들어 클러스터 스냅샷을 네임스페이스로 복원하려면 `RestoreFromClusterSnapshot`이 아닌 `RestoreFromSnapshot` 권한이 필요합니다. IAM 정책 관리에 대한 자세한 내용은 [Amazon Redshift의 Identity and Access Management](#) 섹션을 참조하세요.

Amazon Redshift와 함께 AWS Backup 사용 시 제한 사항

다음은 Amazon Redshift와 함께 AWS Backup을 사용할 때의 제한 사항입니다.

- AWS Backup을 사용하여 Amazon Redshift 자동 스냅샷을 관리할 수 없습니다. 자동 스냅샷을 관리하려면 태그를 사용하세요. 리소스 태그 지정에 대한 자세한 내용은 [Amazon Redshift에서 리소스 태그 지정](#) 단원을 참조하세요.
- 스냅샷에서 단일 테이블을 복원할 때는 프로비저닝된 클러스터 스냅샷에서 서버리스 네임스페이스로 또는 그 반대로 복원할 수 없습니다. 모든 구성에서 전체 스냅샷을 복원할 수 있습니다. 예를 들어 프로비저닝된 클러스터 스냅샷의 모든 데이터베이스를 서버리스 네임스페이스로 복원할 수 있지만 동일한 스냅샷에서 동일한 네임스페이스로 단일 테이블을 복원할 수는 없습니다.

Amazon Redshift를 사용한 AWS Backup 관리

Amazon Redshift 데이터 웨어하우스의 리소스를 보호하기 위해 AWS Backup 콘솔을 사용하거나 프로그래밍 방식으로 AWS Backup API 또는 AWS Command Line Interface(AWS CLI)를 사용할 수 있습니다. 리소스를 복구해야 할 경우 AWS Backup 콘솔이나 AWS CLI를 사용하여 필요한 리소스를 찾아서 복구할 수 있습니다. [자세한 내용은 AWS Command Line Interface 단원을 참조하세요.](#)

Amazon Redshift에 AWS Backup을 사용할 때 다음 작업을 수행할 수 있습니다.

- Amazon Redshift 스냅샷을 자동으로 시작하는 주기적인 백업을 생성합니다. 정기 백업은 장기 데이터 보존 요구 사항을 충족하는 데 유용합니다. 자세한 내용은 AWS Backup 개발자 안내서의 [Amazon Redshift 백업](#)을 참조하세요.
- 백업 계획을 중앙에서 구성하여 백업 예약 및 보존을 자동화합니다.
- 프로비저닝된 클러스터 또는 서버리스 네임스페이스를 선택한 저장된 백업으로 복원합니다. 스냅샷의 모든 데이터를 복원하거나 단일 테이블을 복원하도록 선택할 수 있습니다. 리소스를 백업하는 빈도를 설정합니다. 프로비저닝된 클러스터 스냅샷 복원에 대한 자세한 내용은 AWS Backup 개발자 안내서의 [Amazon Redshift 클러스터 복원](#)을 참조하세요. 서버리스 네임스페이스 스냅샷 복원에 대한 자세한 내용은 AWS Backup 개발자 안내서의 [Amazon Redshift Serverless 복원](#)을 참조하세요.

Amazon Redshift에 대한 클러스터 버전

Amazon Redshift는 정기적으로 클러스터 버전을 출시합니다. Amazon Redshift 클러스터는 시스템 유지 관리 기간 동안 패치됩니다. 패치의 타이밍은 AWS 리전 및 유지 관리 기간 설정에 따라 달라집니다. Amazon Redshift 콘솔에서 유지 관리 기간 설정을 보거나 변경할 수 있습니다. 유지 관리에 대한 자세한 내용은 [클러스터 유지 관리](#) 섹션을 참조하세요.

클러스터 세부 정보의 유지 관리(Maintenance) 탭에 있는 Amazon Redshift 콘솔에서 클러스터의 클러스터 버전을 볼 수 있습니다. 또는 다음 SQL 명령의 출력에서 클러스터 버전을 볼 수 있습니다.

```
SELECT version();
```

Note

Amazon Redshift가 발전함에 따라 Amazon Redshift 동작에 영향을 미치는 중요 업데이트가 도입됩니다. 이러한 변경 사항을 따라잡기 위해 조치를 취하고 워크로드에 잠재적인 중단을 방지하려면 [Amazon Redshift의 동작 변경 사항](#) 섹션을 참조하세요.

주제

- [Amazon Redshift 패치 199](#)
- [Amazon Redshift 패치 198](#)
- [Amazon Redshift 패치 197](#)
- [Amazon Redshift 패치 196](#)
- [Amazon Redshift 패치 195](#)
- [Amazon Redshift 패치 194](#)
- [Amazon Redshift 패치 193](#)
- [Amazon Redshift 패치 192](#)
- [Amazon Redshift 패치 191](#)
- [Amazon Redshift 패치 190](#)
- [Amazon Redshift 패치 189](#)
- [Amazon Redshift 패치 188](#)
- [Amazon Redshift 패치 187](#)
- [Amazon Redshift 패치 186](#)

- [Amazon Redshift 패치 185](#)
- [Amazon Redshift 패치 184](#)
- [Amazon Redshift 패치 183](#)
- [Amazon Redshift 패치 182](#)
- [Amazon Redshift 패치 181](#)
- [Amazon Redshift 패치 180](#)
- [Amazon Redshift 패치 179](#)
- [Amazon Redshift 패치 178](#)
- [Amazon Redshift 패치 177](#)
- [Amazon Redshift 패치 176](#)
- [Amazon Redshift 패치 175](#)
- [Amazon Redshift 패치 174](#)
- [Amazon Redshift 패치 173](#)
- [Amazon Redshift 패치 172](#)
- [Amazon Redshift 패치 171](#)
- [Amazon Redshift 패치 170](#)
- [Amazon Redshift 패치 169](#)
- [Amazon Redshift 패치 168](#)

Amazon Redshift 패치 199

이 패치의 클러스터 버전:

- 1.0.232773 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2026년 3월 4일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- Redshift CURRENT 유지 관리 트랙에서 새 쿼리(쿼리의 첫 번째 실행)의 성능을 높이는 [새 쿼리 컵 파일 개선 사항](#)을 릴리스했습니다. TRAILING 트랙은 향후 패치 릴리스에서 제공될 예정입니다.
- 생산자의 새 피부여자에 대해 Redshift 페더레이션 권한을 사용하여 소비자의 열 수준 권한 부여에 대한 지원을 활성화했습니다.

- 메타데이터 보안(MDS) 기능에서 범위가 지정된 권한 검사에 대한 지원이 활성화되었습니다.
- Amazon Redshift 페더레이션 권한은 Late Binding Views(LBV)를 통해 외부 테이블 쿼리를 지원합니다.
- 이제 ALTER EXTERNAL SCHEMA를 KINESIS 외부 스키마 유형과 함께 사용하여 기존 외부 스키마의 IAM_ROLE 및 REGION을 변경할 수 있습니다.
- Redshift RA3 및 Serverless 데이터 웨어하우스의 NO-BACKUP 테이블에 대한 복원 지원이 추가되었습니다. 이러한 인스턴스 유형은 NO-BACKUP 기능을 지원하지 않으므로 NO-BACKUP으로 표시된 테이블은 항상 스냅샷에 포함되며 이제 영구 테이블로 적절하게 복원됩니다.
- Amazon Redshift는 이제 IAM 역할 또는 IAM 사용자를 사용하여 연결할 때 외부 테이블에 쓰기를 지원합니다. 이전에는 외부 테이블의 읽기 작업만 IAM 역할 기반 인증으로 지원되었습니다. 이 업데이트를 통해 Redshift는 외부 테이블용 Amazon S3에 데이터를 쓸 때 IAM 세션 자격 증명을 올바르게 사용하여 전체 읽기 및 쓰기 액세스를 활성화합니다.
- Kafka 스트림 MV를 자동으로 새로 고치려면 이제 보다 효율적인 휴리스틱을 사용하여 수집할 새 레코드가 있는지 확인합니다. 이렇게 하면 이 검사를 위해 Kafka 브로커에서 전송되는 데이터의 양이 줄어들고 후속 MV 새로 고침 시작 시 지연 시간이 줄어듭니다.
- FIPS 호환 SHA384 암호화를 사용하도록 백업 및 복원 작업을 업그레이드하여 레거시 MD5 해싱 대체
- Lambda 사용자 정의 함수와 UNNEST 문을 결합한 쿼리의 성능 향상
- 시스템 테이블 및 뷰에 대해 Lambda 사용자 정의 함수를 사용하는 쿼리의 성능 향상
- 불필요한 데이터 스캔을 제거하여 ORDER BY 및 LIMIT 절을 사용할 때 쿼리 실행 성능 향상
- Amazon Redshift 구체화된 뷰(MV)에 대한 자동 REFRESH 쿼리는 이제 백그라운드 자율 프로세스 대신 사용자 쿼리로 처리됩니다. 자동 REFRESH 쿼리는 이제 다른 사용자 쿼리와 동일한 우선 순위로 실행되며 더 이상 자율성 백그라운드 프로세스에 의해 연기되지 않습니다. MV 자동 REFRESH 동작 변경 기능은 패치 릴리스 P198 이상의 현재 트랙에 있는 Amazon Redshift 프로비저닝된 클러스터에만 활성화됩니다. 현재 서버리스에서는 비활성화되어 있습니다.
- 스트리밍 수집, 자동 복사(COPY JOB) 및 구체화된 뷰 자동 새로 고침과 같은 사용자 중요 작업의 우선순위를 더 잘 지정하기 위해 자동 작업의 예약이 개선되었습니다.
- pg_last_query_id()가 다중 AZ 클러스터의 사용자 쿼리 ID 대신 다시 작성된 쿼리 ID를 잘못 반환하는 문제를 수정했습니다.
- SHOW GRANTS가 원파트 객체 참조를 처리하지 못하는 문제를 해결합니다.
- DROP TABLE 중에 삭제를 건너뛰고 타임스탬프 작업자가 다음 실행 시 정리할 수 있도록 하여 pg_class_extended의 동시 업데이트로 인해 카탈로그 재구축 작업자의 타임스탬프 자동 작업자와 DROP TABLE이 충돌하는 문제를 수정했습니다.

- 포함된 데이터베이스, 스키마 또는 테이블 이름에 키워드가 포함/제외될 때 발생하는 제로 ETL 채널 필터 구문 분석 문제를 수정했습니다.
- Vacuum 작업이 제로 ETL 테이블 작업과 동시에 실행될 때 발생할 수 있는 경합 조건을 수정했습니다.
- KMS 암호화 컨텍스트가 변경될 때 제로 ETL 통합이 오류 상태로 전환하는 대신 재동기화 작업을 잘못 시도하는 문제를 수정했습니다.
- 데이터 복제가 활성 상태일 때 클러스터를 다시 시작하는 동안 발생할 수 있는 제로 ETL 통합의 경합 조건을 수정했습니다.
- SHOW 테이블 수정

Amazon Redshift 패치 198

이 패치의 클러스터 버전:

- 1.0.239402 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2026년 3월 12일 릴리스됨
- 1.0.227967 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2026년 2월 24일 릴리스됨
- 1.0.224252 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2026년 2월 19일 릴리스됨
- 1.0.219674 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2026년 2월 11일 릴리스됨
- 1.0.212832 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2026년 2월 9일 릴리스됨
- 1.0.204436 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2026년 2월 3일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- Redshift CURRENT 유지 관리 트랙에서 새 쿼리(쿼리의 첫 번째 실행)의 성능을 높이는 [새 쿼리 컴파일 개선 사항](#)을 릴리스했습니다. TRAILING 트랙은 향후 패치 릴리스에서 제공될 예정입니다.
- 열을 상수 값과 비교할 때 적절한 데이터 정렬 규칙을 적용하여 INTERSECT 및 EXCEPT 쿼리를 개선합니다.

- 수동으로 생성된 구체화된 뷰의 자동 새로 고침을 활성화하여 동시성 조정 클러스터에서 실행할 수 있습니다.
- 기본 클러스터와 동시성 조정 클러스터 간의 직접 연결을 활성화하여 연결 성능을 개선하고 지연 시간을 줄이며 연결 처리량 제한을 제거합니다.
- 제로 ETL 통합의 오류 분류를 개선하여 근본 원인을 쉽게 식별하고 문제를 디버깅할 수 있습니다.
- 성공적으로 실행된 쿼리에 대해 SYS_QUERY_HISTORY에서 오류 메시지를 제거하여 더 깨끗한 쿼리 기록 레코드를 제공합니다.
- SYS_QUERY_HISTORY에서 버스트된 쿼리의 컴파일 시간 로깅 정확도를 개선하여 더 나은 성능 인사이트를 제공합니다.
- ARRAY_CONTAINS, ARRAY_POSITION, ARRAYS_OVERLAP, ARRAY_INTERSECTION, ARRAY_SORT, ARRAY_UNION, ARRAY_DISTINCT, ARRAY_EXCEPT, ARRAY_POSITIONS 배열 함수에 대한 지원이 추가되었습니다.
- SUPER 객체 내의 속성 개수를 반환하는 GET_NUMBER_ATTRIBUTES 함수에 대한 지원을 추가합니다.
- 실패한 트랜잭션에 대해 유틸리티 트랜잭션 제한 시간을 활성화하여 리소스 관리를 개선합니다.
- 재사용 가능한 형식 지정 파라미터를 정의할 수 있는 TEMPLATE 객체 유형에 대한 지원을 추가하므로 각 작업에 수동으로 지정할 필요가 없습니다.
- 공통 테이블 표현식(CTE)에서 Lambda 사용자 정의 함수(LUDF)를 간접적으로 호출하는 쿼리의 성능을 개선합니다.
- CASE 표현식의 THEN 절에서 Lambda 사용자 정의 함수(LUDF)를 간접적으로 호출하는 쿼리의 성능을 개선합니다.
- 새 Python 사용자 정의 함수의 생성은 더 이상 사용되지 않습니다. 기존 Python UDFs 2026년 6월 30일까지 지원됩니다. 지속적인 지원을 위해 Lambda 사용자 정의 함수(LUDF)로 마이그레이션합니다.
- 새 시스템 테이블인 SYS_LUDF_DETAIL을 사용하여 Lambda 사용자 정의 함수(LUDF)의 관찰성을 개선합니다. 이 표에는 쿼리에 사용되는 LUDF에 대한 정보와 지표가 기록됩니다.
- Amazon Redshift 구체화된 뷰(MV)에 대한 자동 REFRESH 쿼리는 이제 백그라운드 자율 프로세스 대신 사용자 쿼리로 처리됩니다. 자동 REFRESH 쿼리는 이제 다른 사용자 쿼리와 동일한 우선 순위로 실행되며 더 이상 자율성 백그라운드 프로세스에 의해 연기되지 않습니다. MV 자동 REFRESH 동작 변경 기능은 패치 릴리스 P198 이상의 현재 트랙에 있는 Amazon Redshift 프로비저닝된 클러스터에만 활성화됩니다. 현재 서버리스에서는 비활성화되어 있습니다.

Amazon Redshift 패치 197

이 패치의 클러스터 버전:

- 1.0.237947 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2026년 3월 10일 릴리스됨
- 1.0.198462 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2026년 1월 21일 릴리스됨
- 1.0.194394 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2026년 1월 13일 릴리스됨
- 1.0.187928 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2026년 1월 8일 릴리스됨
- 1.0.179517 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 12월 15일 릴리스됨
- 1.0.176066 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 12월 8일 릴리스됨
- 1.0.166219 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 11월 21일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 메시의 클러스터에 대한 글로벌 자율 제어인 Autonomics는 이제 전체 메시의 워크로드를 고려하여 Vacuum, Analyze 및 ATO를 최적화합니다.
- 여러 Amazon Redshift 데이터 웨어하우스에서 구체화된 뷰를 생성하고 새로 고칩니다.
- SUPER 데이터 형식 내에 최대 16,000,000바이트의 문자열 리터럴을 저장하는 지원이 추가되었습니다.
- SUPER 객체에 포함된 총 속성 수를 반환하는 GET_NUMBER_ATTRIBUTES(super_object) 함수가 추가되었습니다.
- 릴리스 패치 호환성 제한은 테이블 복원 작업에 적용됩니다. 테이블을 복원할 때 백업은 백업이 생성되었을 때와 동일한 패치 버전을 실행 중인 클러스터, 백업보다 한 단계 낮은 패치 버전을 실행 중인 클러스터, 백업보다 한 단계 높은 패치 버전을 실행 중인 클러스터에서만 사용할 수 있습니다.
- CHAR 데이터 유형을 사용할 때 패턴의 후행 공백을 적절하게 처리하도록 LIKE 연산자를 개선했습니다.

- 셀 계층 구조에서 작동하는 4개의 새로운 공간 H3 함수인 H3_Resolution, H3_ToParent, H3_ToChildren 및 H3_IsValid를 도입했습니다.
- 데이터 권한을 한 번 정의하고 AWS 계정의 모든 웨어하우스에 자동으로 적용할 수 있도록 하여 여러 Redshift 데이터 웨어하우스에서 권한 관리를 간소화하는 Amazon Redshift 페더레이션 권한을 활성화합니다.
- Amazon Redshift 페더레이션 권한 카탈로그 객체에 대한 객체 수준 권한 부여, 세분화된 열 수준 액세스 제어, 범위가 지정된 권한 부여 및 데이터베이스 수준 권한을 활성화합니다.
- Amazon Redshift 페더레이션 권한 카탈로그에서 CREATE, ALTER, ATTACH, DETACH 및 DROP RLS 정책 작업을 활성화합니다.
- Amazon Redshift 페더레이션 권한 카탈로그에서 관계에 대한 행 수준 보안을 켜거나 끌 수 있는 기능을 활성화합니다.
- Amazon Redshift 페더레이션 권한 카탈로그 객체에서 CREATE, ALTER, ATTACH, DETACH 및 DROP 마스킹 정책 작업을 활성화합니다.
- SHOW POLICIES를 활성화하여 연결된 데이터베이스의 RLS 및 마스킹 정책과 Amazon Redshift 페더레이션 권한 카탈로그 객체의 관계에 대한 연결을 표시할 수 있습니다.
- 데이터베이스 사용자가 IAM 역할을 수임하여 Amazon Redshift 페더레이션 권한 카탈로그 객체를 쿼리할 수 있습니다.
- 동일한 Redshift 클러스터 내에서 데이터베이스 간 권한 부여 가시성을 지원하도록 SHOW GRANTS 명령이 개선되었습니다.
- 열 수준 액세스 제어 검색을 간소화하기 위해 SHOW COLUMN GRANTS 명령에 대한 지원이 추가되었습니다.
- 저장된 프로시저 메타데이터 검색의 탐색을 간소화하기 위해 SHOW PROCEDURES 명령에 대한 지원이 추가되었습니다.
- 사용자 정의 함수 메타데이터 검색을 간소화하기 위해 SHOW FUNCTIONS 명령에 대한 지원이 추가되었습니다.
- 함수 및 저장 프로시저에 대한 파라미터 메타데이터를 표시하는 SHOW PARAMETERS 명령을 도입했습니다.
- 함수 및 저장 프로시저에 대한 제약 조건 메타데이터를 표시하는 SHOW CONSTRAINTS 명령을 도입했습니다.
- Amazon Redshift 페더레이션 권한 카탈로그 및 동일한 Redshift 클러스터 내의 데이터베이스 간에 CREATE/CREATE OR REPLACE/DROP VIEW 작업에 대한 지원이 추가되었습니다.
- 일부 INTERSECT 또는 EXCEPT 쿼리에서 상수 문자열 데이터 정렬 처리를 개선했습니다.

- AWS Glue Data Catalog 액세스에 필요한 FAS 자격 증명이 누락된 경우 오류를 발생시키는 대신 경고 메시지를 표시하도록 SHOW TABLES 명령이 개선되었습니다.
- 정렬 키, 배포 스타일, 인코딩 및 데이터 정렬 정보에 대한 추가 메타데이터 열로 SHOW COLUMNS 명령을 개선했습니다.
- SHOW GRANTS 출력에 grantor_name 열을 추가하고 모든 SHOW GRANTS 결과 세트에 표준화된 database_name 포함을 추가했습니다.
- 추가 열인 table_owner, last_altered_time, last_modified_time, dist_style, table_sub_type을 사용하여 SHOW TABLES 명령을 개선했습니다.
- Identity Center 인증 지원: Amazon Redshift는 이제 새 GetIdentityCenterAuthToken API 작업을 통해 IAM Identity Center 인증을 지원합니다. 이 기능을 사용하면 기존 Identity Center 자격 증명을 사용하여 Amazon Redshift 클러스터 및 서버리스 작업 그룹에 대한 페더레이션 액세스를 사용할 수 있습니다. 주요 이점으로는 Identity Center 자격 증명을 사용한 Single Sign-On 액세스, 암호화된 인증 토큰으로 보안 강화, 규정 준수를 위한 포괄적인 감사 로깅, 기존 Identity Center 설정과의 원활한 통합 등이 있습니다. API 작업은 특정 클러스터 식별자 또는 작업 그룹 이름으로 범위가 지정된 인증 토큰을 생성합니다. 요구 사항에는 Identity Center ID 정보가 포함된 ID 강화 자격 증명과 API 액세스를 위한 적절한 IAM 권한이 포함됩니다.

Amazon Redshift 패치 196

이 패치의 클러스터 버전:

- 1.0.179872 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 12월 16일 릴리스됨
- 1.0.171165 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 12월 8일 릴리스됨
- 1.0.163480 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 11월 19일 릴리스됨
- 1.0.160807 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 11월 13일 릴리스됨
- 1.0.155905 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 11월 10일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 상관관계가 있는 IN 및 EXISTS 하위 쿼리에 대한 계획을 세미 조인으로 분리하여 최적화했습니다.
- 모든 열에 대한 데이터 정렬 정보를 포함하도록 SHOW TABLE 명령을 확장했습니다.

Amazon Redshift 패치 195

이 패치의 클러스터 버전:

- 1.0.172541 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 12월 8일 릴리스됨
- 1.0.162991 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 11월 17일 릴리스됨
- 1.0.160706 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 11월 13일 릴리스됨
- 1.0.151179 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 11월 4일 릴리스됨
- 1.0.148180 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 11월 3일 릴리스됨
- 1.0.142459 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 10월 18일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- CREATE TABLE, CREATE TABLE AS SELECT, INSERT INTO, SQL 명령을 사용하여 Iceberg 테이블에 쓰기에 대한 지원을 추가합니다.
- 대/소문자를 구분하지 않는 데이터 정렬이 있는 데이터베이스의 SUPER 데이터 형식에 대한 지원을 추가합니다.
- IANA 시간대 데이터베이스의 데이터베이스 버전을 2025b로 업데이트합니다.
- TZDB_VERSION 함수에 대한 지원을 추가합니다. 이 함수는 사용 중인 현재 IANA 시간대 데이터베이스 버전을 표시합니다.
- Amazon Redshift는 이제 패치 호환성 제한을 적용합니다. 현재 클러스터보다 둘 이상의 패치 버전인 패치 버전에서 생성된 백업은 더 이상 테이블 복원 작업을 수행할 수 없습니다.

- JOIN 표현식을 지원하도록 UNNEST 기능을 확장합니다. 이전에는 UNNEST가 단일 테이블 참조로 제한되었습니다.
- JIT(Just In Time) ANALYZE 기능에 대한 지원을 추가합니다. JIT ANALYZE는 Amazon Redshift의 쿼리 옵티마이저가 효율적인 쿼리 실행 계획을 생성하는 데 필요한 최소한의 통계 세트를 자동으로, 지능적으로, 효율적으로 수집하여 고객이 Apache Iceberg OOB(out-of-box) 쿼리 및 워크로드에 대해 최상의 가격 대비 성능을 달성할 수 있도록 합니다.

Amazon Redshift 패치 194

이 패치의 클러스터 버전:

- 1.0.148169 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 10월 29일 릴리스됨
- 1.0.138443 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 10월 17일 릴리스됨
- 1.0.129791 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 10월 1일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- Amazon Redshift는 AuthorizedColumns 필드에 백슬래시가 포함된 AWS Lake Formation 관리형 관계에 액세스하는 쿼리에 더 이상 실패하지 않습니다.
- SUPER 열에 대한 COPY 명령의 NULL AS 옵션에 대한 지원이 추가되어 NULL 값을 나타내는 사용자 지정 문자열을 지정할 수 있습니다.
- 날짜 및 타임스탬프와 함께 사용할 때 EXTRACT 및 DATE_PART 함수의 ISOWEEK 및 ISOYEAR 날짜 부분에 대한 지원이 추가되었습니다.
- GROUP BY 쿼리에서 중첩된 필드 액세스를 올바르게 구문 분석하고 평가하도록 그룹화된 변수에 대한 PartiQL 탐색을 수정했습니다.
- SHOW CONSTRAINTS 명령을 사용하여 프라이머리 키 및 외래 키 제약 조건을 볼 수 있는 지원이 추가되었습니다.
- 데이터를 공유하는 구체화된 뷰에서 생성된 로컬 구체화된 뷰에 대한 지원이 추가되었습니다.
- 다른 테이블에서 데이터를 검색하고 조인 조건에서 Lambda 사용자 정의 함수를 간접적으로 호출하는 쿼리의 성능을 개선합니다.
- 데이터 공유 MV에서 MV에 대한 지원 추가

- DROP DATABASE 명령은 이제 백그라운드 작업을 사용하여 커밋 후 리더 노드 카탈로그 디렉터리를 제거합니다. 작업을 사용하면 카탈로그 디렉터리 정리와 관련하여 드물게 발생하는 버그를 피할 수 있습니다.

Amazon Redshift 패치 193

이 패치의 클러스터 버전:

- 1.0.136890 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 10월 13일 릴리스됨
- 1.0.127211 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 9월 29일 릴리스됨
- 1.0.125329 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 9월 24일 릴리스됨
- 1.0.121010 - Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 8월 26일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 원격 데이터베이스(동일 클러스터 내 및 다른 클러스터 간)에서 상위 테이블 하나를 복제하기 위해 CREATE TABLE LIKE를 사용한 테이블 생성에 대한 지원이 추가되었습니다.
- GeoJSON 형식의 SUPER 데이터형에서 지오메트리를 생성하는 ST_GeomFromGeoJSON 오버로드가 추가되었으며, 최대 1MB 크기의 지오메트리를 지원합니다.
- 자동 워크로드 관리(WLM)의 잘못된 구성 시 기본 폴백 옵션이 변경되었습니다. 이제 기본 대체 옵션은 수동 WLM 대신 자동 WLM입니다.
- 최신 IANA 시간대 데이터베이스 패치를 채택하여 시간대 계산을 업데이트했습니다. 이 변경 사항은 특정 시간대 및 기간에 대해 날짜 및 시간 변환이 작동하는 방식을 변경합니다. 자세한 내용은 [Amazon Redshift의 최신 IANA 시간대 데이터베이스 사용](#) 섹션을 참조하세요.
- 다차원 데이터 레이아웃(MDDL) 정렬에 대한 지원이 추가되었습니다. MDDL 정렬 키는 실제 쿼리 필터를 기반으로 데이터를 동적으로 정렬하여 쿼리 성능을 가속화합니다. 자세한 내용은 [Amazon Redshift 데이터베이스 개발자 가이드](#)의 다차원 데이터 레이아웃 저장 섹션을 참조하세요.

Amazon Redshift 패치 192

이 패치의 클러스터 버전:

- 1.0.124422 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 9월 22일 릴리스됨
- 1.0.122914 - 후행 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 9월 16일 릴리스됨
- 1.0.121035 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 8월 22일 릴리스됨
- 1.0.119650 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 8월 7일 릴리스됨
- 1.0.118447 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 7월 24일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 대소문자를 구분하지 않는 정렬된 열과 상수 문자열 열에서 JOIN을 실행할 때 일치하지 않는 해시 함수가 선택되는 문제를 해결했습니다.

Amazon Redshift 패치 191

이 패치의 클러스터 버전:

- 1.0.117891 - 후행 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 7월 29일 릴리스됨
- 1.0.117891 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 7월 22일 릴리스됨
- 1.0.117458 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 7월 15일 릴리스됨
- 1.0.116415 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 6월 26일 릴리스됨
- 1.0.116263 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 6월 25일 릴리스됨

- 1.0.115865 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 6월 18일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- SELECT 문의 FROM 절에 UNNEST 작업 지원이 추가되어 고객이 배열 요소를 개별 행으로 변환할 수 있습니다.
- 공통 테이블 표현식(CTE) 및 재귀적 CTE 내에서 SUPER 데이터 형식을 사용할 때 PartiQL 중첩 해제 구문의 오류 처리가 개선되었습니다. 특히 동일한 이름을 가진 테이블 별칭을 확인할 때 더욱 그렇습니다.
- 대용량 테이블에 대한 증분형 자동 분석 지원을 추가하여 자동 분석의 성능이 개선되었습니다.
- Amazon Redshift는 이제 중첩된 구체화된 뷰(MV)의 캐스케이드 새로 고침을 지원합니다.
- Amazon Redshift는 이제 Apache Iceberg 테이블에 정의된 구체화된 뷰(MV)의 자동 새로 고침을 지원합니다.
- Amazon Redshift와 Amazon RDS for Oracle의 제로 ETL 통합 지원이 추가되었습니다.
- Amazon Redshift와 Amazon RDS for PostgreSQL의 제로 ETL 통합 지원이 추가되었습니다.

Amazon Redshift 패치 190

이 패치의 클러스터 버전:

- 1.0.115140 - 후행 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 6월 16일 릴리스됨
- 1.0.115564 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 6월 14일 릴리스됨
- 1.0.113436 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 5월 15일 릴리스됨
- 1.0.112895 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 5월 7일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- Amazon Redshift Serverless AI 기반 규모 조정 및 최적화를 위한 개선된 모델이 추가되었습니다.

- 향상된 VPC 라우팅이 활성화된 Amazon Redshift를 사용할 때 자동 복사에 대한 지원이 추가되었습니다.
- 여러 테이블에서 동시에 여러 자동 테이블 최적화 작업을 실행할 수 있도록 지원이 추가되었습니다.
- 스트리밍 수집에서 대기 중인 자동 새로 고침 쿼리가 UNDO 쿼리를 트리거하고 해당 쿼리도 대기되어 잠금 경합을 유발하는 드문 상황이 해결되었습니다.
- Amazon Redshift와 Oracle Database@AWS의 제로 ETL 통합 지원이 추가되었습니다.

Amazon Redshift 패치 189

이 패치의 클러스터 버전:

- 1.0.112790 - 후행 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 5월 14일 릴리스됨
- 1.0.112790 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 5월 6일 릴리스됨
- 1.0.112086 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 4월 29일 릴리스됨
- 1.0.111040 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 4월 16일 릴리스됨
- 1.0.109768 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 3월 22일 릴리스됨
- 1.0.109284 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 3월 13일 릴리스됨
- 1.0.108971 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 3월 7일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- SUPER, GEOMETRY 및 GEOGRAPHY 데이터 유형을 사용한 쓰기 쿼리의 동시성 규모 조정에 대한 지원이 추가되었습니다.
- CREATE TABLE AS를 사용하여 테이블을 만드는 동안 SUPER, GEOMETRY 또는 GEOGRAPHY 열을 DISTKEY 또는 SORTKEY 열로 설정할 수 있었던 문제가 해결되었습니다.
- TRY_CAST 함수에 대한 지원이 추가되었습니다.
- 여러 테이블에서 여러 vacuum 명령을 동시에 실행할 수 있도록 지원합니다.

Amazon Redshift 패치 188

이 패치의 클러스터 버전:

- 1.0.109616 - 후행 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 3월 27일 릴리스됨
- 1.0.109616 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 3월 24일 릴리스됨
- 1.0.108470 - 후행 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 3월 13일 릴리스됨
- 1.0.108470 - 현재 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 3월 11일 릴리스됨
- 1.0.108950 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 3월 10일 릴리스됨
- 1.0.108790 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 3월 9일 릴리스됨
- 1.0.108470 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 3월 4일 릴리스됨
- 1.0.107910 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 2월 20일 릴리스됨
- 1.0.107360 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 2월 13일 릴리스됨
- 1.0.106767 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 2월 5일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- Amazon S3 Tables 카탈로그의 자동 탑재에 대한 지원이 추가되어 Amazon S3 Tables에서 관리되는 Apache Iceberg 테이블에서 쿼리를 더 쉽게 실행할 수 있습니다. 필요한 권한은 AWS Lake Formation 개발자 안내서의 [Prerequisites for managing Amazon Redshift namespaces in the AWS Glue Data Catalog](#)를 참조하세요.
- 이제 데이터 마스킹 검사를 우회하고 동적 데이터 마스킹(DDM) 연결이 있는 테이블을 데이터 공유에 추가할 수 있습니다.
- "건너뛰기 수준의 상관관계 참조"라고도 하는, 쿼리 블록을 건너뛰는 상관관계 참조의 사용을 허용하지 않습니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [지원되지 않는 상관관계를 가진 하위 쿼리 패턴](#)을 참조하세요.

- 데이터 공유 및 3부 표기법을 사용하는 상관관계가 있는 SUPER 데이터 유형 쿼리에 대한 지원이 추가되었습니다.
- EXCLUDE SQL 키워드에 대한 지원이 추가되었습니다.
- 새 GROUP BY ALL SQL 키워드에 대한 지원이 추가되었습니다.
- 역할로 구성된 워크로드 관리(WLM) 대기열의 성능을 개선하여 잠금 경합이 줄었습니다.
- 기록 모드가 켜져 있는 제로 ETL 테이블에서 자동으로 새로 고쳐진 구체화된 뷰에 대한 지원이 추가되었습니다.
- 제로 ETL 복제 중에 문자열에 포함된 null 문자에 대한 지원이 추가되었습니다.
- AI 기반 규모 조정 및 최적화를 위한 모델이 개선되었습니다.
- sys_query_history 뷰에 username 열이 추가되었습니다.

Amazon Redshift 패치 187

이 패치의 클러스터 버전:

- 1.0.107351 - Amazon Redshift 프로비저닝 클러스터 후행 트랙 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 2월 13일 릴리스됨
- 1.0.106452 - Amazon Redshift 프로비저닝 클러스터 후행 트랙 버전 - 2025년 2월 5일 릴리스됨
- 1.0.106980 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 2월 3일 릴리스됨
- 1.0.106452 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 1월 24일 릴리스됨
- 1.0.106073 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 1월 21일 릴리스됨
- 1.0.105722 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 1월 10일 릴리스됨
- 1.0.105373 - 현재 Amazon Redshift 프로비저닝 클러스터 버전 및 Amazon Redshift Serverless 작업 그룹 버전 - 2025년 1월 8일 릴리스됨
- 1.0.104930 - Amazon Redshift Serverless 작업 그룹 버전 - 2024년 12월 20일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 두 가지 새로운 공간 H3 함수(H3_center 및 H3_boundary)를 도입합니다.

- 동시성 크기 조정 클러스터에서 동적 디스크 캐시를 활성화합니다.
- 정렬된 테이블을 수집하는 동안 성능에 영향을 미칠 수 있는 메모리 사용량의 비효율성을 수정합니다.
- SELECT 절 내에 하위 쿼리가 있는 특정 쿼리(예: `SELECT ARRAY_FLATTEN((SELECT FROM ...))`)가 단일 SUPER 값과 관련된 특정 시나리오에서 `size >= min_partiql_size` XCHECK 오류를 잘못 트리거하는 문제를 해결합니다.
- null을 허용하지 않는 SUPER 표현식(`json_serialize()`, `json_size()`, `json_typeof()`, `is_object()` 및 기타 여러 표현식)이 `CASE ... END` 또는 `COALESCE()` 같은 인수 표현식과 결합될 때 때때로 잘못된 결과를 생성하는 문제를 해결합니다.
- 뷰에서 행 수준 보안을 활성화한 후 지연 바인딩 뷰에서 SELECT가 오류를 유발하는 문제를 수정합니다.
- 제로 ETL 통합의 CDC로 인해 리더 노드 CPU 사용량이 높아지는 문제를 해결합니다.
- 업데이트 중을 포함하여 모든 상태의 테이블을 쿼리하는 기능을 제로 ETL 통합에 추가합니다.
- Amazon Redshift의 최대 varchar 크기로 매핑하기 위해 큰 텍스트/문자열을 잘라내는 기능을 제로 ETL 통합에 추가합니다.
- 잘못된 UTF-8 문자를 선택한 지정된 문자로 바꾸는 기능을 제로 ETL 통합에 추가합니다.
- Amazon RDS for MySQL과의 제로 ETL 통합에서 '새로 고침 간격' 문제를 수정합니다.
- Aurora PostgreSQL과의 제로 ETL 통합에서 JSON/JSONB 데이터 유형에 대한 지원을 추가합니다.
- `IS [NOT] {TRUE|FALSE|UNKNOWN}`가 있는 표현식이 포함된 쿼리에 어설션 오류가 발생하는 문제를 해결합니다.
- 존재하지 않는 배열 요소 또는 속성, JSON 'null' 값 또는 빈 JSON 문자열에 액세스하는 경우 `json_extract_path_text()` 및 `json_extract_array_element_text()` 함수가 NULL 대신 빈 문자열 ''을 생성하는 문제를 해결합니다.
- 테이블에 데이터를 써야 할 때까지 잠금을 공유하고 진행하도록 허용하여 동일한 테이블에 쓰는 동시 트랜잭션의 INSERT/COPY 문 성능이 개선되었습니다.

Amazon Redshift 패치 186

이 패치의 클러스터 버전:

- 1.0.82096 - 후행 트랙 버전 - 2025년 1월 10일 릴리스됨
- 1.0.82000 - Amazon Redshift Serverless 버전 - 2025년 1월 10일 릴리스됨

- 1.0.81981 - 현재 트랙 버전 - 2025년 1월 10일 릴리스됨
- 1.0.81475 - 후행 트랙 버전 - 2025년 1월 6일 릴리스됨
- 1.0.81473 - Amazon Redshift Serverless 버전 - 2025년 1월 6일 릴리스됨
- 1.0.81462 - 현재 트랙 버전 - 2025년 1월 6일 릴리스됨
- 1.0.80643 - 후행 트랙 버전 - 2024년 12월 17일 릴리스됨
- 1.0.80583 - Amazon Redshift Serverless 버전 - 2024년 12월 17일 릴리스됨
- 1.0.80560 - 현재 트랙 버전 - 2024년 12월 17일 릴리스됨
- 1.0.80498 - Amazon Redshift Serverless 버전 - 2024년 12월 13일 릴리스됨
- 1.0.80491 - 현재 트랙 버전 - 2024년 12월 13일 릴리스됨
- 1.0.80036 - Amazon Redshift Serverless 버전 - 2024년 12월 6일 릴리스됨
- 1.0.80009 - 현재 트랙 버전 - 2024년 12월 6일 릴리스됨
- 1.0.79372 - Amazon Redshift Serverless 버전 - 2024년 11월 26일 릴리스됨
- 1.0.79237 - Amazon Redshift Serverless 버전 - 2024년 11월 24일 릴리스됨
- 1.0.79229 - 현재 트랙 버전 - 2024년 11월 24일 릴리스됨
- 1.0.79003 - Amazon Redshift Serverless 버전 - 2024년 11월 19일 릴리스됨
- 1.0.78987 - 현재 트랙 버전 - 2024년 11월 19일 릴리스됨
- 1.0.78890 - Amazon Redshift Serverless 버전 - 2024년 11월 18일 릴리스됨
- 1.0.78881 - 현재 트랙 버전 - 2024년 11월 18일 릴리스됨
- 1.0.78646 - Amazon Redshift Serverless 버전 - 2024년 11월 14일 릴리스됨
- 1.0.78641 - 현재 트랙 버전 - 2024년 11월 14일 릴리스됨
- 1.0.78178 - Amazon Redshift Serverless 버전 - 2024년 11월 12일 릴리스됨
- 1.0.78160 - 현재 트랙 버전 - 2024년 11월 12일 릴리스됨
- 1.0.77809 - Amazon Redshift Serverless 버전 - 2024년 10월 31일 릴리스됨
- 1.0.77777 - 현재 트랙 버전 - 2024년 10월 31일 릴리스됨
- 1.0.77292 - Amazon Redshift Serverless 버전 - 2024년 10월 24일 릴리스됨
- 1.0.77272 - 현재 트랙 버전 - 2024년 10월 24일 릴리스됨
- 1.0.77040 - Amazon Redshift Serverless 버전 - 2024년 10월 22일 릴리스됨
- 1.0.77028 - 현재 트랙 버전 - 2024년 10월 22일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- Amazon Aurora MySQL, Amazon Aurora PostgreSQL, Amazon RDS for MySQL 및 Amazon DynamoDB와의 제로 ETL 통합에서 구체화된 테이블 뷰의 자동 및 증분 새로 고침에 대한 지원을 추가합니다.
- Amazon Redshift가 `SELECT ... WHERE key = (SELECT ... correlated subquery)`와 같은 상관관계가 있는 단일 행 하위 쿼리를 사용하여 쿼리를 다시 작성하는 방법을 개선합니다. 이러한 쿼리는 하위 쿼리가 단일 행을 생성하는 경우에만 유효합니다. 재작성 개선으로 인해 이제 이 조건을 위반하는 특정 쿼리가 단일 행 하위 쿼리가 둘 이상의 행을 반환하는 오류로 실패할 수 있습니다. 이전에는 허용되었던 경우입니다. 이를 방지하려면 예를 들어 `MIN()` 또는 `MAX()` 집계를 추가하여 단일 행을 반환하도록 이러한 하위 쿼리를 수정해야 할 수 있습니다.
- Amazon Redshift에 SQL 식별자 'KAFKA'를 추가하여 직접 스트리밍 수집을 통해 외부 Kafka 소스에서 Amazon Redshift로의 스트리밍을 지원합니다. 이러한 소스에는 Confluent Managed Cloud 및 Apache Kafka와 같은 외부 Kafka 소스가 포함됩니다.
- 데이터 공유를 사용하여 다중 웨어하우스 쓰기에 대한 지원을 추가하므로 워크로드 요구 사항에 따라 다양한 유형 및 크기의 각기 다른 웨어하우스를 사용하여 쓰기 워크로드의 크기를 조정하고 추출, 전환, 적재(ETL) 워크로드에 대한 성능을 개선할 수 있습니다.
- 이제 Amazon Redshift를 사용하여 Amazon S3 Tables에서 관리되는 Apache Iceberg 테이블에서 `SELECT` 쿼리를 실행할 수 있습니다. 자세한 정보는 Amazon Simple Storage Service 사용 설명서의 [Accessing Amazon S3 tables with Amazon Redshift](#)를 참조하세요.
- Amazon Redshift 클러스터의 여러 데이터베이스에 쓸 수 있도록 Amazon Redshift에서 데이터베이스 간 쓰기 쿼리에 대한 지원을 추가합니다. 자세한 내용은 [데이터베이스 간 쿼리](#)를 참조하세요.

Amazon Redshift 패치 185

이 패치의 클러스터 버전:

- 1.0.79878 - 후행 트랙 버전 - 2024년 12월 3일 릴리스됨
- 1.0.79868 - Amazon Redshift Serverless 버전 - 2024년 12월 3일 릴리스됨
- 1.0.79845 - 현재 트랙 버전 - 2024년 12월 3일 릴리스됨
- 1.0.78946 - 후행 트랙 버전 - 2024년 11월 19일 릴리스됨
- 1.0.78354 - 후행 트랙 버전 - 2024년 11월 12일 릴리스됨
- 1.0.78130 - Amazon Redshift Serverless 버전 - 2024년 11월 12일 릴리스됨
- 1.0.78125 - 현재 트랙 버전 - 2024년 11월 12일 릴리스됨

- 1.0.78016 - Amazon Redshift Serverless 버전 - 2024년 11월 5일 릴리스됨
- 1.0.78014 - 현재 트랙 버전 - 2024년 11월 5일 릴리스됨
- 1.0.77707 - Amazon Redshift Serverless 버전 - 2024년 11월 4일 릴리스됨
- 1.0.77687 - 현재 트랙 버전 - 2024년 11월 4일 릴리스됨
- 1.0.77467 - Amazon Redshift Serverless 버전 - 2024년 11월 4일 릴리스됨
- 1.0.77433 - 현재 트랙 버전 - 2024년 11월 4일 릴리스됨
- 1.0.77467 - Amazon Redshift Serverless 버전 - 2024년 10월 29일 릴리스됨
- 1.0.77433 - 현재 트랙 버전 - 2024년 10월 29일 릴리스됨
- 1.0.76991 - Amazon Redshift Serverless 버전 - 2024년 10월 21일 릴리스됨
- 1.0.76913 - 현재 트랙 버전 - 2024년 10월 21일 릴리스됨
- 1.0.76645 - Amazon Redshift Serverless 버전 - 2024년 10월 14일 릴리스됨
- 1.0.76642 - 현재 트랙 버전 - 2024년 10월 14일 릴리스됨
- 1.0.76242 - Amazon Redshift Serverless 버전 - 2024년 10월 10일 릴리스됨
- 1.0.76230 - 현재 트랙 버전 - 2024년 10월 10일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 데이터 레이크 테이블에 만들어진 구체화된 뷰(MV)의 증분 새로고침에 대한 지원이 추가됩니다.
- 구체화된 뷰의 배포 키 및 정렬 키를 변경할 수 있는 지원이 도입됩니다.
- Amazon Redshift 기계 학습(ML)과 Amazon Bedrock의 통합에 대한 지원이 추가되며, Amazon Redshift의 데이터와 함께 간단한 SQL 명령의 대규모 언어 모델(LLM)을 활용합니다.
- 오류를 발생시키지 않고 Amazon Redshift 공간 데이터에서 빈 세이프 파일을 처리할 수 있는 버그가 수정됩니다.
- 제로 ETL 통합에서 varbit 및 varbinary 데이터 형식이 빈 문자열 값을 NULL이 아닌 ""로 수집하도록 허용하는 버그가 수정됩니다.
- 결과 캐시를 사용하여 레이스 조건을 수정하여 데이터베이스 간 쿼리가 오래된 결과를 반환하여 제로 ETL 통합이 이뤄지도록 합니다.
- 제로 ETL 통합 재동기화 프로세스를 최적화하여 재동기화 시간이 단축됩니다.
- 복구 및 시스템 유지 관리 작업 후 제로 ETL 부트스트랩 시간이 개선됩니다. 이제 시스템에서 사용할 가능한 최신 CDC 시점까지 복구할 수 없는 대신 마지막 쿼리 작업 후 CDC를 복구할 수 있습니다.

- 새 시스템 테이블 `sys_integration_table_activity`와의 제로 ETL 통합을 위한 관찰성이 개선됩니다. 이 테이블은 제로 ETL 통합 테이블 삽입, 업데이트 및 삭제를 추적합니다.
- 페더레이션 역할 환경의 자동 생성이 개선됩니다. 이제 Amazon Redshift 관리자가 페더레이션 사용자 로그인 중에 페더레이션 역할의 자동 생성을 더 잘 제어할 수 있습니다. 이제 각 ID 제공업체에 대한 자동 생성 설정을 사용, 사용 해제, 필터 적용 및 구성할 수 있습니다.
- IAM Identity Center 사용자는 S3 액세스 권한 부여를 사용하여 IAM Identity Center 자격 증명으로 COPY, UNLOAD 및 CREATE LIBRARY를 실행할 수 있습니다.
- Amazon S3에서 파일을 지속적으로 수집할 수 있는 자동 복사(COPY JOB)를 이제 일반적으로 사용할 수 있습니다.

Amazon Redshift 패치 184

이 패치의 클러스터 버전:

- 1.0.77706 - 후행 트랙 버전 - 2024년 11월 12일 릴리스됨
- 1.0.76832 - 후행 트랙 버전 - 2024년 10월 17일 릴리스됨
- 1.0.76169 - Amazon Redshift Serverless 버전 - 2024년 10월 10일 릴리스됨
- 1.0.76142 - 현재 트랙 버전 - 2024년 10월 10일 릴리스됨
- 1.0.75677 - Amazon Redshift Serverless 버전 - 2024년 9월 27일 릴리스됨
- 1.0.75672 - 현재 트랙 버전 - 2024년 9월 27일 릴리스됨
- 1.0.75504 - Amazon Redshift Serverless 버전 - 2024년 9월 23일 릴리스됨
- 1.0.75449 - 현재 트랙 버전 - 2024년 9월 24일 릴리스됨
- 1.0.74765 - Amazon Redshift Serverless 버전 - 2024년 9월 12일 릴리스됨
- 1.0.74754 - 현재 트랙 버전 - 2024년 9월 12일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 스트리밍 데이터 수집을 위한 Amazon Redshift 스트리밍 구체화된 뷰(MV)는 VARBYTE 열 크기를 1,024,000바이트에서 증가시켰습니다. 이제 Amazon Redshift는 Amazon Kinesis Data Streams에서 최대 1,048,576바이트(1MiB)의 레코드를 수집할 수 있습니다. Redshift는 Amazon Managed Streaming for Apache Kafka에서 최대 16,777,216바이트(16MiB)까지 레코드를 수집할 수 있습니다. `ATA SQL ALTER TABLE <target_tbl> APPEND FROM <streaming_mv>` 명령을 사용하는 경우 <target_tbl>을 삭제하고 다시 생성하여 해당하는 더 큰 VARBYTE 열 크기를 보유합니다.

- 이제 Amazon Redshift 데이터 공유에는 Lake Formation에서 거버넌스되는 테이블을 포함하여 AWS Glue Data Catalog을 참조하는 Amazon S3 데이터 레이크 테이블과 뷰가 포함됩니다.
- DynamoDB와의 제로 ETL 통합에서 테이블에 구체화된 뷰의 자동 및 증분 새로고침에 대한 지원이 추가됩니다.
- Amazon Redshift에서 복제 새로고침 빈도를 지정하도록 제로 ETL 통합 테이블에서 '새로고침 간격'을 구성하는 지원이 추가됩니다. 새 통합을 위한 데이터베이스 생성 시 설정하거나 기존 통합의 데이터베이스를 변경할 수 있습니다.
- Amazon Managed Streaming for Apache Kafka에 대한 Amazon Redshift 스트리밍 수집에서 상호 전송 계층 보안(mTLS) 인증에 대한 지원이 추가됩니다.
- 쿼리의 텍스트 표현과 해당 파라미터의 값을 기반으로 SQL 쿼리의 고유 식별자인 쿼리 해시에 대한 지원이 추가됩니다. 이는 유사한 쿼리를 식별, 그룹화 및 분석하는 데 사용할 수 있습니다. 이제 쿼리 해시를 SYS_QUERY_HISTORY 뷰에서 확인할 수 있으며, 2개의 새 열이 추가됩니다.
 - user_query_hash - 쿼리 리터럴을 포함하여 사용자가 제출한 해시입니다.
 - generic_query_hash - 쿼리 리터럴 없이 사용자가 제출한 해시입니다.
- 제로 ETL이 CDC 복제 및 스캔을 실행하고 결과 캐시를 사용하여 테이블 쿼리를 수행하는 경우에 드물게 발생하는 시스템 교착 문제가 수정됩니다.
- Python UDF 쿼리에 대한 WLM 리소스를 사용할 수 없을 경우 Python 사용자 정의 함수(UDF) 쿼리가 다른 쿼리를 선점하는 워크로드 관리(WLM) 문제가 해결됩니다.
- 워크로드 관리(WLM)가 새로 생성된 사용자 역할에 매핑된 대기열로 쿼리를 라우팅하지 못하는 문제가 해결됩니다.
- 대규모 생산자 테이블을 쿼리하는 소규모 데이터 공유 소비자의 디스크 사용률이 개선됩니다.
- 크기가 더 큰 탄력적으로 조정된 프로비저닝 클러스터에 대한 INSERT 스테이트먼트의 성능이 개선됩니다.

Amazon Redshift 패치 183

이 패치의 클러스터 버전:

- 1.0.75655 - 후행 트랙 버전 – 2024년 9월 30일 릴리스됨
- 1.0.75388 - Amazon Redshift Serverless 버전 – 2024년 9월 25일 릴리스됨
- 1.0.75379 - 현재 트랙 버전 – 2024년 9월 25일 릴리스됨
- 1.0.74967 - Amazon Redshift Serverless 버전 – 2024년 9월 17일 릴리스됨
- 1.0.74927 - 현재 트랙 버전 – 2024년 9월 17일 릴리스됨

- 1.0.74518 - Amazon Redshift Serverless 버전 – 2024년 9월 11일 릴리스됨
- 1.0.74503 - 현재 트랙 버전 – 2024년 9월 11일 릴리스됨
- 1.0.74223 - Amazon Redshift Serverless 버전 – 2024년 9월 5일 릴리스됨
- 1.0.74159 - 현재 트랙 버전 – 2024년 9월 5일 릴리스됨
- 1.0.74126 – Amazon Redshift Serverless 버전 – 2024년 8월 30일 릴리스됨
- 1.0.74097 - 현재 트랙 버전 - 2024년 8월 30일 릴리스됨
- 1.0.73016 - Amazon Redshift Serverless 버전 - 2024년 8월 8일 릴리스됨
- 1.0.72982 - 현재 트랙 버전 - 2024년 8월 8일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- SVV_DATABASE_PRIVILEGES 및 SVV_SCHEMA_PRIVILEGES를 통한 범위 지정 권한 검색 지원을 추가합니다. 또한 SVV_DATABASE_PRIVILEGES 및 SVV_SCHEMA_PRIVILEGES에 `privilege_scope` 열을 도입합니다.
- 그룹화 열에 고유 값 수(NDV)가 적은 경우 고유 집계 연산을 실행하는 쿼리의 성능을 개선합니다.
- 크기가 2배 이상 탄력적으로 조정된 프로비저닝된 데이터 웨어하우스에 대한 INSERT/COPY 문의 성능을 개선합니다.
- 동적 데이터 마스킹 정책 내에서 세션 컨텍스트 변수를 지원합니다.
- MERGE 문의 소스로서 하위 쿼리 및 뷰에 대한 지원을 추가합니다.
- 프로비저닝된 동시성 확장 및 서버리스 자동 스케일링 컴퓨팅에 대한 MERGE 문을 포함하는 저장 프로시저를 지원합니다.
- COPY 명령과 크기가 조정되는 웨어하우스에 대한 워크로드 관리에서 더 나은 리소스 예측을 통해 쿼리 성능을 개선합니다.
- 사용 가능한 메모리가 제한된 클러스터에서 메모리 부족 오류에 대한 복원력 향상
- COPY 명령에 필드 구분 기호로 ASCII가 아닌 문자에 대한 지원을 추가합니다.
- COPY 명령을 사용하여 ISO-8859-1 문자 집합으로 인코딩된 데이터를 수집하는 기능을 추가합니다.
- URI를 지정하는 경우 MSK 외부 스키마 정의에서 CLUSTER_ARN을 지정해야 하는 요구 사항을 제거합니다.
- 제로 ETL 통합 테이블에서 스캔하는 동안 범위 스캔 필터를 적용할 수 있습니다.
- 제로 ETL 통합 테이블에 정렬 키 추가를 지원합니다.

- 제로 ETL 통합 데이터베이스를 만들 때 `serializable`, `collation` 등의 데이터베이스 옵션을 `CREATE DATABASE` 문으로 지정할 수 있도록 지원합니다.
- 제로 ETL 통합에서 데이터 필터가 2KB를 초과할 때 클러스터가 다시 시작되는 문제를 수정했습니다.

Amazon Redshift 패치 182

이 패치의 클러스터 버전:

- 1.0.73589 - 후행 트랙 버전 - 2024년 8월 22일 릴리스됨
- 1.0.73359 - Amazon Redshift Serverless 버전 - 2024년 8월 15일 릴리스됨
- 1.0.73348 - 현재 트랙 버전 - 2024년 8월 15일 릴리스됨
- 1.0.72917 - Amazon Redshift Serverless 버전 - 2024년 8월 12일 릴리스됨
- 1.0.72899 - 현재 트랙 버전 - 2024년 8월 12일 릴리스됨
- 1.0.72528 - Amazon Redshift Serverless 버전 - 2024년 8월 7일 릴리스됨
- 1.0.72503 - 현재 트랙 버전 - 2024년 8월 8일 릴리스됨
- 1.0.72239 - Amazon Redshift Serverless 버전 - 2024년 8월 1일 릴리스됨
- 1.0.71714 - Amazon Redshift Serverless 버전 - 2024년 7월 24일 릴리스됨
- 1.0.71629 - 현재 트랙 버전 - 2024년 7월 24일 릴리스됨
- 1.0.70953 - Amazon Redshift Serverless 버전 - 2024년 7월 11일 릴리스됨
- 1.0.70890 - 현재 트랙 버전 - 2024년 7월 11일 릴리스됨
- 1.0.70716 - Amazon Redshift Serverless 버전 - 2024년 7월 8일 릴리스됨
- 1.0.70695 - 현재 트랙 버전 - 2024년 7월 8일 릴리스됨
- 1.0.69945 - Amazon Redshift Serverless 버전 - 2024년 6월 27일 릴리스됨
- 1.0.69938 - 현재 트랙 버전 - 2024년 6월 27일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 사용자 정의 테이블이 더 이상 필요하지 않도록 `LISTAGG`, `MEDIAN`, `PERCENTILE_CONT` 및 `PERCENTILE_DISC`를 업데이트했습니다. 카탈로그 테이블을 참조하는 쿼리 또는 테이블을 참조하지 않는 쿼리에서도 이러한 함수를 사용할 수 있습니다.
- 동시성이 높은 워크로드를 위해 단일 세션 내에서 여러 쿼리의 임시 테이블을 통합하여 데이터 공유 읽기 쿼리의 쿼리 계획 시간을 단축했습니다.

- 자체 대규모 언어 모델을 사용할 수 있도록 Amazon SageMaker AI Jumpstart와 Redshift 기계 학습 (ML)과의 통합 기능이 정식 출시되었습니다.
- Redshift ML의 SUPER 입력 및 출력 데이터 유형이 새롭게 지원됩니다.
- 대상 테이블이 동적 데이터 마스킹 정책으로 보호되고 JOIN 절에서 참조되는 경우 JOIN 절을 사용하여 UPDATE 문을 지원할 수 있습니다.
- 통합이 소스에서 삭제된 후에도 Redshift에서 제로 ETL 데이터베이스를 쿼리할 수 있습니다.
- 제로 ETL 통합의 실패를 일으킬 수 있는 복제 오류를 해결했습니다. 이를 통해 통합의 복원력이 향상됩니다.
- 제로 ETL 통합을 생성한 사용자 이외의 사용자가 권한 부여 후 데이터를 쿼리할 수 있습니다.
- 제로 ETL 통합이 활성화된 Amazon Redshift 프로비저닝 클러스터에서 클러스터 재시작을 일으킬 수 있는 메모리 부족 문제를 해결했습니다.
- 소스 RDS 다중 AZ DB 클러스터에서 Redshift와의 RDS for MySQL 제로 ETL 통합을 생성할 수 있습니다. 다중 AZ DB 클러스터는 읽을 수 있는 복제 DB 인스턴스가 2개 있는 Amazon RDS의 반동기식 고가용성 배포 모드입니다.
- Amazon Redshift 스트리밍 구체화된 뷰를 Amazon MSK 주제와 연결하는 데 필요한 외부 스키마 정의에 Amazon MSK 클러스터의 브로커 URI를 지정하여 사용자가 Amazon Redshift 스트리밍 소비자 클라이언트에서 Amazon MSK 클러스터에 연결할 수 있습니다. 이 기능을 사용하면 인터넷 게이트웨이를 통해 Amazon MSK 클러스터의 GetBootstrapBroker API를 직접적으로 호출하여 Amazon MSK 부트스트랩 브로커 노드 이름을 얻을 필요가 없습니다.
- 기존 데이터베이스 사용자가 IAM Identity Center 인증 방법을 사용해 Amazon Redshift 쿼리 에디터 v2에서 데이터베이스에 연결할 때 서버리스 인스턴스를 재개할 수 있도록 Amazon Redshift Serverless 인스턴스를 재개할 때 발생하는 문제를 해결했습니다.
- 테이블 기반 샤딩으로 전환하여 Redshift 컴퓨팅에서 CDC 복제를 최적화하고 리소스 사용률을 줄였습니다.
- 워크로드 관리(WLM)에서 향상된 리소스 예측을 통해 쿼리 성능을 개선했습니다.
- ran out of WLM queues for restart 메시지와 함께 동시 규모 조정 클러스터에서 쿼리가 실패하는 문제를 해결했습니다.
- 고객이 잘못된 WLM 구성을 적용하려고 할 때 Amazon Redshift가 수동 WLM으로 대체되는 워크로드 관리(WLM) 문제를 해결했습니다.
- 계획된 유지 관리 또는 예상치 못한 운영 중단으로 인해 생산자가 가동을 중단한 경우에도 데이터 공유 소비자가 읽기 쿼리를 실행할 수 있습니다.
- 데이터를 집계하는 쿼리에서 ANY_VALUE 함수를 사용할 때 발생하는 드문 클러스터 재시작 문제 (예: COUNT(DISTINCT) 집계 함수)를 수정합니다.

Amazon Redshift 패치 181

이 패치의 클러스터 버전:

- 1.0.72031 - 현재 트랙 버전 - 2023년 8월 1일 릴리스됨
- 1.0.71912 - 후행 트랙 버전 - 2024년 8월 1일 릴리스됨
- 1.0.70665 - Amazon Redshift Serverless 버전 - 2024년 7월 8일 릴리스됨
- 1.0.70634 - 현재 트랙 버전 - 2024년 7월 8일 릴리스됨
- 1.0.69954 - Amazon Redshift Serverless 버전 - 2024년 6월 26일 릴리스됨
- 1.0.69952 - 현재 트랙 버전 - 2024년 6월 26일 릴리스됨
- 1.0.69497 - Amazon Redshift Serverless 버전 - 2024년 6월 18일 릴리스됨
- 1.0.69451 - 현재 트랙 버전 - 2024년 6월 18일 릴리스됨
- 1.0.69076 - Amazon Redshift Serverless 버전 - 2024년 6월 14일 릴리스됨
- 1.0.69065 - 현재 트랙 버전 - 2024년 6월 14일 릴리스됨
- 1.0.68555 - Amazon Redshift Serverless 버전 - 2024년 5월 31일 릴리스됨
- 1.0.68540 - 현재 트랙 버전 - 2024년 5월 31일 릴리스됨
- 1.0.68328 - Amazon Redshift Serverless 버전 - 2024년 5월 23일 릴리스됨
- 1.0.68205 - 현재 트랙 버전 - 2024년 5월 23일 릴리스됨
- 1.0.67796 - Amazon Redshift Serverless 버전 - 2024년 5월 15일 릴리스됨
- 1.0.67788 - 현재 트랙 버전 - 2024년 5월 15일 릴리스됨
- 1.0.67308 - Amazon Redshift Serverless 버전 - 2024년 5월 1일 릴리스됨
- 1.0.67305 - 현재 트랙 버전 - 2024년 5월 1일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- SUPER 객체 값에 대한 속성 이름의 대소문자를 수정하는 'lower_attribute_names()' 및 'upper_attribute_names()' 함수를 지원합니다.
- ID 열을 사용할 때 CREATE TABLE LIKE 관련 문제가 해결되었습니다. 이전에는 새 테이블이 소스 테이블의 식별자를 상속했습니다. 이로 인해 나중에 소스 테이블이 삭제되면 식별자가 새 테이블에서 유효하지 않게 되므로 문제가 발생했습니다.
- 일부 외부 테이블이 SVV_ALL_TABLES에 표시되지 않는 문제가 해결되었습니다.

- 클러스터 부트스트랩 시간이 개선되고 동시 워크로드가 많은 경우 쿼리 초기화 속도가 높아졌습니다.
- `split_part()` 함수를 RDS 및 Aurora MySQL의 페더레이션 소스에 전달할 때 오류를 일으키는 페더레이션 쿼리 관련 문제가 해결되었습니다.
- 프로비저닝된 동시성 규모 조정 클러스터와 서버리스 자동 규모 조정 컴퓨팅에서 `ALTER TABLE...ALTER DISTSTYLE KEY DISTKEY` 명령을 통해 사용자가 시작한 배포 키 변경을 지원합니다.
- 프로비저닝된 동시성 규모 조정 및 서버리스 자동 규모 조정 컴퓨팅에 대한 집계를 포함하는 수동으로 새로 고쳐진 구체화된 뷰를 지원합니다.
- 이제 제로 ETL을 통해 최대 16MB의 레코드를 처리하고 최대 16MB의 SUPER 값을 지원합니다.
- 스키마 및 테이블 이름과 같은 추가 세부 정보를 제공하여 Aurora MySQL의 제로 ETL에서 초기 동기화 중 나타나는 오류 메시지가 개선되었습니다.
- Amazon Redshift ML `CREATE MODEL`을 사용한 태그 지정을 지원합니다. 이 개선 사항을 통해 이제 Amazon Redshift ML에서 사용하는 Amazon SageMaker AI 리소스에 태그를 지정할 수 있습니다. 태그를 지정하면 리소스를 손쉽게 관리, 식별, 정리, 검색 및 필터링할 수 있습니다.
- AWS Lambda를 사용해 데이터 처리를 최적화하여 Lambda 사용자 정의 함수(UDF) 관련 쿼리의 성능이 개선되었습니다.
- 탄력적으로 크기가 조정된 서버리스 클러스터의 정렬된 테이블에서 데이터 모으기 중 메모리 사용률이 감소했습니다.
- `SYS_QUERY_HISTORY` 뷰의 `query_text` 열과 `SYS_QUERY_TEXT` 뷰의 `text` 열에서 줄바꿈 (`\n`)을 지원합니다.

Amazon Redshift 패치 180

이 패치의 클러스터 버전:

- 1.0.68520 - 후행 트랙 버전 - 2024년 5월 28일 릴리스됨
- 1.0.67699 - 후행 트랙 버전 - 2024년 5월 15일 릴리스됨
- 1.0.66960 - 후행 트랙 버전 - 2024년 4월 21일 릴리스됨
- 1.0.66954 - 현재 트랙 버전 - 2024년 4월 21일 릴리스됨
- 1.0.66276 - 현재 트랙 버전 - 2024년 4월 12일 릴리스됨
- 1.0.66290 - Amazon Redshift Serverless 버전 - 2024년 4월 10일 릴리스됨
- 1.0.63590 - 현재 트랙 버전 - 2024년 2월 19일 릴리스됨

- 1.0.63567 - Amazon Redshift Serverless 버전 - 2024년 2월 16일 릴리스됨
- 1.0.63282 - Amazon Redshift Serverless 버전 - 2024년 2월 13일 릴리스됨
- 1.0.63269 - 현재 트랙 버전 - 2024년 2월 13일 릴리스됨
- 1.0.63215 - Amazon Redshift Serverless 버전 - 2024년 2월 12일 릴리스됨
- 1.0.63205 - 현재 트랙 버전 - 2024년 2월 12일 릴리스됨
- 1.0.63030 - Amazon Redshift Serverless 버전 - 2024년 2월 7일 릴리스됨
- 1.0.62913 - 현재 트랙 버전 - 2024년 2월 7일 릴리스됨
- 1.0.62922 - Amazon Redshift Serverless 버전 - 2024년 2월 5일 릴리스됨
- 1.0.62878 - 현재 트랙 버전 - 2024년 2월 5일 릴리스됨
- 1.0.62698 - Amazon Redshift Serverless 버전 - 2024년 1월 31일 릴리스됨
- 1.0.62614 - 현재 트랙 버전 - 2024년 1월 31일 릴리스됨
- 1.0.61687 - Amazon Redshift Serverless 버전 - 2024년 1월 5일 릴리스됨
- 1.0.61678 - 현재 트랙 버전 - 2024년 1월 5일 릴리스됨
- 1.0.61567 - Amazon Redshift Serverless 버전 - 2023년 12월 31일 릴리스됨
- 1.0.61559 - 현재 트랙 버전 - 2023년 12월 31일 릴리스됨
- 1.0.61430 - Amazon Redshift Serverless 버전 - 2023년 12월 29일 릴리스됨
- 1.0.61395 - 현재 트랙 버전 - 2023년 12월 29일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 반환된 사용자 이름이 더 이상 64자로 잘리지 않도록 CURRENT_USER를 변경합니다.
- 표준 뷰와 지연 바인딩 뷰에 데이터 마스킹 정책을 적용하는 기능을 추가합니다.
- SUPER 데이터 유형 열의 스칼라 속성에 동적 데이터 마스킹(DDM)을 적용하는 기능을 추가합니다.
- OBJECT_TRANSFORM SQL 함수를 추가합니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [OBJECT_TRANSFORM 함수](#)를 참조하세요.
- 중첩된 데이터에 AWS Lake Formation 세밀한 액세스 제어를 적용하고 Amazon Redshift 데이터 레이크 분석을 사용하여 쿼리하는 기능을 추가합니다.
- INTERVAL 데이터 유형을 추가합니다.
- 저장 프로시저의 흐름을 제어하는 예외 핸들러의 일종인 CONTINUE_HANDLER를 추가합니다. 이 기능을 사용하면 기존 문 블록을 종료하지 않고도 예외를 파악하고 처리할 수 있습니다.
- 개별 객체 외에도 범위(스키마 또는 데이터베이스)에 대한 권한을 정의하는 기능을 추가합니다. 이렇게 하면 범위 내의 모든 현재 및 미래 객체에 대한 권한을 사용자와 역할에 부여할 수 있습니다.

- 소비자 측 관리자가 공유 데이터베이스 객체에 대한 개별 권한을 소비자 측 사용자 및 역할에 부여할 수 있는 권한을 가진 데이터 공유에서 데이터베이스를 생성하는 기능을 추가합니다.
- 원격 BYOM 모델의 SUPER 반환 데이터 유형에 대한 지원을 추가합니다. 이렇게 하면 허용되는 SageMaker AI 모델의 범위가 확장되어 더 복잡한 반환 형식을 가진 모델을 포함하게 됩니다.
- 이제 소수 부분이 있거나 없는 숫자를 열의 숫자 데이터 유형에 대해 암시적으로 캐스팅하도록 외부 함수를 변경합니다. int2, int4, int8 열의 경우 숫자가 범위를 벗어나지 않는 한 소수 자릿수를 잘라서 숫자를 입력할 수 있습니다. float4 및 float8 열의 경우 소수 자릿수 없이 숫자를 입력할 수 있습니다.
- H3 계층형 지리공간 인덱싱 그리드 시스템에서 작동하는 세 개의 공간 함수, 즉 H3_FromLongLat, H3_FromPoint, H3_Polyfill을 추가합니다.

Amazon Redshift 패치 179

이 패치의 클러스터 버전:

- 1.0.62317 - Amazon Redshift Serverless 버전 - 2024년 1월 29일 릴리스됨
- 1.0.62312 - 후행 트랙 버전 - 2024년 1월 29일 릴리스됨
- 1.0.61631 - Amazon Redshift Serverless 버전 - 2024년 1월 5일 릴리스됨
- 1.0.61626 - 현재 트랙 버전 - 2024년 1월 5일 릴리스됨
- 1.0.61191 - 현재 트랙 버전 - 2023년 12월 16일 릴리스됨
- 1.0.61150 - Amazon Redshift Serverless 버전 - 2023년 12월 16일 릴리스됨
- 1.0.60982 - Amazon Redshift Serverless 버전 - 2023년 12월 13일 릴리스됨
- 1.0.60854 - 현재 트랙 버전 - 2023년 12월 10일 릴리스됨
- 1.0.60354 - Amazon Redshift Serverless 버전 - 2023년 11월 22일 릴리스됨
- 1.0.60353 - 현재 트랙 버전 - 2023년 11월 21일 릴리스됨
- 1.0.60293 - Amazon Redshift Serverless 버전 - 2023년 11월 21일 릴리스됨
- 1.0.60292 - 현재 트랙 버전 - 2023년 11월 22일 릴리스됨
- 1.0.60161 - Amazon Redshift Serverless 버전 - 2023년 11월 18일 릴리스됨
- 1.0.60140 - 현재 트랙 버전 - 2023년 11월 18일 릴리스됨
- 1.0.60139 - Amazon Redshift Serverless 버전 - 2023년 11월 18일 릴리스됨
- 1.0.59947 - Amazon Redshift Serverless 버전 - 2023년 11월 16일 릴리스됨
- 1.0.59945 - 현재 트랙 버전 - 2023년 11월 16일 릴리스됨
- 1.0.59118 - Amazon Redshift Serverless 버전 - 2023년 11월 9일 릴리스됨

- 1.0.59117 - 현재 트랙 버전 – 2023년 11월 9일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 적절한 권한이 있는 페더레이션 사용자가 행 수준 보안 및 동적 데이터 마스킹 시스템 보기를 볼 수 있도록 다음과 같은 지원을 추가합니다.
 - SVV_ATTACHED_MASKING_POLICY
 - SVV_MASKING_POLICY
 - SVV_RLS_ATTACHED_POLICY
 - SVV_RLS_POLICY
 - SVV_RLS_RELATION
- 이제 FROM 절에 스칼라 함수만 포함된 쿼리에서 오류가 발생하는 기능을 추가합니다.
- 영구 대상 테이블 기능이 있는 CREATE TABLE AS(CTAS) 문을 동시성 조정 클러스터에 추가합니다. 동시성 조정 클러스터가 이제 더 많은 쿼리를 지원합니다.
- RA3 클러스터에서 클래식 크기 조정을 실행한 후 테이블 재배포 상태를 추적하기 위해 다음 시스템 테이블을 추가합니다.
 - SYS_RESTORE_STATE 시스템 테이블은 테이블 수준 재배포 진행 상황을 보여줍니다.
 - SYS_RESTORE_LOG 시스템 테이블은 과거의 데이터 재배포 처리량을 보여줍니다.
- RA3 노드 유형에서 클래식 크기 조정을 실행한 후 EVEN 테이블에서 조각 스큐 최소화를 개선합니다. 이는 클래식 크기 조정을 실행한 패치 178 클러스터에도 적용됩니다.
- 동시성 조정 클러스터에 EXTENSION이 있는 UNLOAD에 대한 지원을 추가합니다.
- HashJoins 및 NestLoop 조인에 $\wedge$ UDF가 포함된 쿼리의 성능을 개선합니다.
- RA3 노드 유형에 대한 탄력적 크기 조성의 성능을 개선합니다.
- 데이터 공유 쿼리 성능을 개선합니다.
- 탄력적으로 크기가 조정된 프로비저닝된 클러스터와 서버리스 작업 그룹에서 수동으로 시작하는 분석 쿼리의 성능을 개선합니다.
- 워크로드 관리에서 더 나은 리소스 예측을 통해 자동 WLM 쿼리 성능을 개선합니다.
- 클러스터를 전용 테넌시 VPC에서 시작하도록 하는 기능을 제거합니다. 이 변경 사항은 VPC에 있는 EC2 인스턴스의 테넌시에는 영향을 주지 않습니다. `modify-vpc-tenancy` AWS CLI 명령을 사용하여 VPC의 테넌시를 기본값으로 수정할 수 있습니다.
- 이제 프로비저닝된 동시성 조정 클러스터와 서버리스 자동 크기 조정 컴퓨팅에서 구체화된 뷰 수동 새로 고침이 지원됩니다.

- INTERVAL 리터럴에 대한 지원을 EXTRACT 함수에 추가합니다. 예를 들어, EXTRACT('hours' from Interval '50 hours')는 2를 반환합니다. 50시간이 2일 2시간으로 해석되고 시간 구성 요소인 2가 추출되기 때문입니다.

Amazon Redshift 패치 178

이 패치의 클러스터 버전:

- 1.0.63327 - 현재 트랙 버전 - 2024년 2월 9일 릴리스됨
- 1.0.63313 - 후행 트랙 버전 - 2024년 2월 9일 릴리스됨
- 1.0.60977 - 후행 트랙 버전 - 2023년 12월 15일 릴리스됨
- 1.0.59596 - 현재 트랙 버전 - 2023년 11월 9일 릴리스됨
- 1.0.58593 - Amazon Redshift Serverless 버전 - 2023년 10월 23일 릴리스됨
- 1.0.58558 - 현재 트랙 버전 - 2023년 10월 23일 릴리스됨
- 1.0.57864 - 현재 트랙 버전 - 2023년 10월 12일 릴리스됨
- 1.0.57850 - Amazon Redshift Serverless 버전 - 2023년 10월 12일 릴리스됨
- 1.0.56952 - 현재 트랙 버전 - 2023년 9월 25일 릴리스됨
- 1.0.56970 - Amazon Redshift Serverless 버전 - 2023년 9월 25일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- Amazon Redshift는 생산자 인스턴스에서 동시 데이터 변경이 발생하는 동안 소비자 인스턴스의 메타데이터 새로 고침 속도를 높여 데이터 공유 쿼리 성능을 개선했습니다.
- 구체화된 뷰의 기본 테이블이 공유 데이터를 참조할 때 Amazon Redshift 데이터 공유 소비자 인스턴스에서 구체화된 뷰의 자동 및 증분 새로 고침에 대한 지원을 추가합니다.
- SUPER 데이터 유형으로 최대 16MB 크기의 대형 객체를 저장하기 위한 지원을 추가합니다. JSON, PARQUET, TEXT 및 CSV 소스 파일에서 수집하는 경우 반정형 데이터 또는 문서를 SUPER 데이터 유형(최대 16MB)의 값으로 로드할 수 있습니다.
- 단일 노드 Amazon Redshift RA3 클러스터 안팎으로 확장하기 위한 탄력적 크기 조정 지원을 추가합니다.
- 단일 노드 Amazon Redshift RA3 클러스터는 이제 암호화 기능이 향상되어 암호화 중에 전체 암호화 시간이 단축되고 데이터 웨어하우스의 가용성이 향상됩니다.

- SUPER 데이터 유형에 저장된 데이터를 중첩하거나 피벗을 해제할 때 쿼리에 대한 지원을 개선합니다.
- SUPER 데이터 유형을 사용하여 구체화된 뷰를 새로 고치는 성능이 향상됩니다.
- ANY_VALUE 함수를 사용하여 INTERVAL 리터럴을 집계하기 위한 지원을 추가합니다.
- 스트리밍 수집이 이제 스트리밍 데이터를 제거하기 위해 새로운 SQL 명령인 DELETE FROM streaming_materialized_views WHERE <where filter clause>를 지원합니다.
- DECODE 함수는 등식 조건의 결과에 따라 특정 값을 다른 특정 값이나 기본값으로 변경합니다. 이제 DECODE에는 다음 세 가지 파라미터가 필요합니다.
 - expression
 - 검색
 - 결과
- 저장 프로시저에 데이터 오버플로 데이터 유형 변환 오류를 포착하고 예외 처리 블록 내에서 처리할 수 있는 기능을 추가합니다.
- 이제 enable_case_sensitive_identifier를 세션 기본 설정과 다르게 변경하면 행 수준 보안 또는 동적 데이터 마스킹으로 보호된 관계를 쿼리할 때 오류가 발생합니다. 또한 프로비저닝된 클러스터 또는 서버리스 네임스페이스에 행 수준 보안 또는 동적 데이터 마스킹 정책이 적용되는 경우 다음 구성이 차단됩니다.

ALTER USER <current_user> SET case-sensitive identifier.

- 이제 MERGE 명령이 대상 및 소스 테이블만 필요한 단순 구문을 지원합니다. 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서에서 [MERGE](#)를 참조하세요.
- 여러 사용자 또는 역할에 같은 우선 순위로 또는 우선 순위를 지정하지 않고 동일한 동적 데이터 마스킹 정책을 연결할 수 있는 지원을 추가합니다.
- 이제 ALTER TABLE ADD COLUMN을 통해 새 열을 추가할 때 COLLATION을 지정할 수 있습니다.
- 동시 크기 조정 클러스터와 Amazon Redshift Serverless에서 QMR 규칙 적용이 지연되는 문제를 수정합니다.
- Amazon Redshift 페더레이션 쿼리는 Amazon RDS for PostgreSQL과 Amazon Aurora PostgreSQL에서 타임스탬프를 사용하여 시간대에 대한 푸시다운 지원을 확대했습니다.
- 이제 페더레이션 쿼리에 숫자로 시작하는 Amazon RDS for MySQL과 Aurora MySQL 데이터베이스 이름을 사용할 수 있습니다.
- ANALYZE 작업에 대한 레코드 세부 정보를 포함하는 SYS_ANALYZE_HISTORY 뷰를 추가합니다.
- COPY 또는 ANALYZE COMPRESSION 명령 중 압축 분석 작업에 대한 레코드 세부 정보를 포함하는 SYS_ANALYZE_COMPRESSION_HISTORY 뷰를 추가합니다.

- 활성, 과거 및 다시 시작된 세션과 관련된 레코드 세부 정보를 포함하는 SYS_SESSION_HISTORY 뷰를 추가합니다.
- 커밋에 소요된 시간, 데이터, 커밋된 블록의 데이터 공유 수 및 격리 수준을 제공하는 트랜잭션 수준 분석과 관련된 레코드 세부 정보를 포함하는 SYS_TRANSACTION_HISTORY 뷰를 추가합니다.
- 데이터베이스의 각 스키마에 대한 할당량 및 현재 디스크 사용량과 관련된 레코드를 포함하는 SVV_REDSHIFT_SCHEMA_QUOTA 뷰를 추가합니다.
- 시작 시간, 종료 시간, 저장 프로시저 호출의 상태, 중첩된 저장 프로시저 호출의 호출 계층 구조 등 저장 프로시저 호출과 관련된 레코드를 포함하는 SYS_PROCEDURE_CALL 뷰를 추가합니다.
- 교차 리전 데이터 공유 사용량 추적과 관련된 레코드를 포함하는 SYS_CROSS_REGION_DATASHARING_USAGE 뷰를 추가합니다.
- 로깅된 저장 프로시저 메시지에 대한 추적 정보와 관련된 레코드를 포함하는 SYS_PROCEDURE_MESSAGES 뷰를 추가합니다.
- 해당하는 경우 사용자 정의 함수 호출, 오류, 경고 또는 추적의 시스템 로그 메시지 추적과 관련된 레코드를 포함하는 SYS_UDF_LOG 뷰를 추가합니다.
- 새로운 열인 IS_RECURSIVE, IS_NESTED, S3LIST_TIME, GET_PARTITION_TIME to SYS_EXTERNAL_QUERY_DETAIL을 추가합니다.
- Redshift Serverless를 위한 새로운 컴퓨팅 비용 제어 설정인 MaxRPU를 추가합니다. MaxRPU를 사용하면 Redshift Serverless가 작업 그룹별로 크기를 조정할 수 있는 최대 컴퓨팅 수준을 선택하여 다양한 시점의 데이터 웨어하우스 비용을 제어하는 상한 컴퓨팅 임계값을 선택적으로 지정할 수 있습니다.
- INTERVAL 리터럴의 출력을 숫자 간격 문자열로 수정합니다. 예를 들어, 이제 INTERVAL '1' YEAR를 지정하는 간격은 "00:00:00이 아닌 1 YEAR를 반환합니다. 또한 INTERVAL 리터럴의 출력은 지정된 가장 작은 INTERVAL 구성 요소까지 잘립니다. 예를 들어, INTERVAL '1 day 1 hour 1 minute 1.123 seconds' HOUR TO MINUTE은 1 day 01:01:00으로 잘립니다.

Amazon Redshift 패치 177

이 패치의 클러스터 버전:

- 1.0.57922 - 후행 트랙 버전 - 2023년 10월 12일 릴리스됨
- 1.0.57799 - Amazon Redshift Serverless 버전 - 2023년 10월 10일 릴리스됨
- 1.0.57798 - 현재 트랙 버전 - 2023년 10월 10일 릴리스됨
- 1.0.57085 - 후행 트랙 버전 - 2023년 9월 26일 릴리스됨
- 1.0.56899 - Amazon Redshift Serverless 버전 - 2023년 9월 21일 릴리스됨

- 1.0.56754 - 현재 트랙 버전 - 2023년 9월 21일 릴리스됨
- 1.0.56242 - 현재 트랙 버전 - 2023년 9월 11일 릴리스됨
- 1.0.55539 - Amazon Redshift Serverless 버전 - 2023년 8월 28일 릴리스됨
- 1.0.55524 - 현재 트랙 버전 - 2023년 8월 28일 릴리스됨
- 1.0.54899 - 현재 트랙 버전 - 2023년 8월 15일 릴리스됨
- 1.0.54899 - 현재 트랙 버전 - 2023년 8월 14일 릴리스됨
- 1.0.54899 - 현재 트랙 버전 - 2023년 8월 15일 릴리스됨
- 1.0.54239 - 현재 트랙 버전 - 2023년 8월 3일 릴리스됨
- 1.0.54321 - Amazon Redshift Serverless 버전 - 2023년 8월 3일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 구체화된 뷰의 모든 상태 전환에 대한 행을 포함하는 SYS_MV_STATE 뷰가 추가되었습니다. SYS_MV_STATE는 Amazon Redshift Serverless 및 Amazon Redshift 프로비저닝된 인스턴스에 대한 MV 새로 고침 모니터링에 사용할 수 있습니다.
- 사용자 생성, 사용자 삭제, 사용자 변경(이름 변경), 사용자 변경(속성 변경)에 대한 데이터베이스 사용자 변경에 대한 세부 정보를 기록하는 SYS_USERLOG 뷰가 추가되었습니다.
- 잘못된 UTF-8 문자가 ACCEPTINVCHARS 옵션을 사용하여 COPY 명령으로 대체된 경우를 기록하는 SYS_REPLACENCES 뷰가 추가되었습니다.
- COPY 명령을 사용하여 단순화된 공간 지오메트리 객체에 대한 정보가 포함된 SYS_SPATIAL_SIMPLIFY 뷰가 추가되었습니다.
- VACUUM 작업의 세부 정보 및 결과를 확인하는 데 사용할 수 있는 SYS_VACUUM_HISTORY 뷰가 추가되었습니다.
- 스키마 할당량을 초과한 경우 발생, 타임스탬프, XID 및 기타 유용한 정보를 기록하는 SYS_SCHEMA_QUOTA_VIOLATIONS 뷰가 추가되었습니다.
- 비동기식 클래식 크기 조정 중에 클러스터 내 각 테이블의 재배포 진행 상황을 모니터링하는 데 사용할 수 있는 SYS_RESTORE_STATE 뷰가 추가되었습니다.
- 적색편이 스펙트럼 스캔 오류에 대한 정보를 반환하는 SYS_EXTERNAL_QUERY_ERROR 뷰가 추가되었습니다.
- CREATE MODEL 명령에 태그 파라미터를 추가하여 이제 Autopilot 훈련 작업으로 훈련 비용을 추적할 수 있습니다.
- Amazon Redshift 클러스터에 대한 사용자 지정 도메인 이름(CNAME)이 추가되었습니다.

- Apache Iceberg에 대한 미리 보기 지원을 추가하여 고객이 Amazon Redshift 내에서 Apache Iceberg 테이블에 대한 분석 쿼리를 실행할 수 있습니다.
- 워크로드 관리(WLM)에서 파라미터 그룹과 함께 사용자 역할 사용에 대한 지원이 추가되었습니다.
- 고객이 데이터 레이크에서 쿼리를 더 쉽게 실행할 수 있도록 AWS Glue Data Catalog의 자동 마운팅에 대한 지원이 추가되었습니다.
- GROUP BY 절 없이 그룹화 함수를 사용하거나 WHERE 절에서 그룹화 연산을 사용하면 오류가 발생하는 기능이 추가되었습니다.
- 저장 프로시저에 0으로 나눗셈 오류를 포착하고 예외 처리 블록 내에서 처리할 수 있는 기능이 추가되었습니다.
- 소스 테이블이 데이터 공유 테이블일 때 쿼리가 동시성 스케일링을 사용하여 테이블에 데이터를 쓰지 못하게 하는 버그가 수정되었습니다.
- enable_case_sensitive_identifier에 문서화된 대/소문자 구분 식별자가 이제 MERGE 문에서 작동하도록 수정되었습니다.
- pg_get_late_binding_view_cols() 함수에 대한 쿼리가 간혹 무시될 수 있는 버그가 수정되었습니다. 이제 언제든지 해당 쿼리를 취소할 수 있습니다.
- 생산자에서 진공 작업을 실행할 때 소비자에서 실행되는 데이터 공유 쿼리의 성능이 개선되었습니다.
- 특히 생산자에서 동시 데이터 변경이 있거나 소비자에게 연결된 동시성 확장 인스턴스로 오프로드할 때 데이터 공유 및 동시성 확장 쿼리의 성능이 개선되었습니다.

Amazon Redshift 패치 176

이 패치의 클러스터 버전:

- 1.0.56738 - 후행 트랙 버전 – 2023년 9월 21일 릴리스됨
- 1.0.55837 - 후행 트랙 버전 – 2023년 9월 11일 릴리스됨
- 1.0.54776 - 현재 트랙 버전 - 2023년 8월 15일 릴리스됨
- 1.0.54052 - 현재 트랙 버전 – 2023년 7월 26일 릴리스됨
- 1.0.53642 - Amazon Redshift Serverless 버전 – 2023년 7월 20일 릴리스됨
- 1.0.53301 - 현재 트랙 버전 – 2023년 7월 20일 릴리스됨
- 1.0.52943 - Amazon Redshift Serverless 버전 – 2023년 7월 7일 릴리스됨
- 1.0.52931 - 현재 트랙 버전 – 2023년 7월 7일 릴리스됨

- 1.0.52194 - Amazon Redshift Serverless 버전 - 2023년 6월 21일 릴리스됨
- 1.0.51986 - 현재 트랙 버전 - 2023년 6월 16일 릴리스됨
- 1.0.51594 - 현재 트랙 버전 - 2023년 6월 9일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 빈 그룹화 집합에 대해 GROUP BY ()를 작성할 때 발생하는 오류 처리가 개선되었습니다. 이전에는 무시되었지만 이제는 파서 오류가 반환됩니다.
- SUPER 열을 사용하여 구체화된 뷰를 점진적으로 새로 고치는 성능이 향상되었습니다.
- ALTER TABLE <target_tbl> APPEND FROM <streaming_mv> - (ATA) SQL 명령은 이제 테이블을 소스로 사용하는 것 외에 스트리밍 구체화된 뷰(MV)를 소스로 삼아 소스의 모든 레코드를 대상 테이블로 이동할 수 있도록 지원합니다. 스트리밍 MV에서 ATA를 지원하므로 사용자는 데이터 증가를 관리하기 위해 스트리밍 MV의 모든 레코드를 다른 테이블로 이동하여 신속하게 제거할 수 있습니다.
- TRUNCATE <streaming_mv> - 이제 SQL 명령에서 테이블 외에도 스트리밍 구체화된 뷰(MV)의 모든 레코드를 잘라낼 수 있습니다. TRUNCATE는 스트리밍 MV 구조를 그대로 유지하면서 스트리밍 MV의 모든 레코드를 삭제합니다. 스트리밍 MV에서 TRUNCATE를 실행하면 고객이 스트리밍 MV의 모든 레코드를 신속하게 제거하여 데이터 증가를 관리할 수 있습니다.
- SELECT 명령에 QUALIFY 절에 대한 기능이 추가되었습니다.
- Amazon Forecast와 통합하여 시계열 예측을 위한 Redshift 기계 학습을 지원합니다.
- AWS Glue Data Catalog 자동 마운트가 지원되므로 외부 스키마 참조를 생성하기 위한 추가 단계 없이 데이터 레이크 쿼리를 단순화할 수 있습니다.
- 이제 RLS 정책 변경이 지원됩니다. 자세한 내용은 설명서에서 [ALTER RLS POLICY](#)를 참조하세요.
- Lambda UDF는 이제 CREATE FUNCTION 명령문에서 STABLE 함수 변동성 파라미터를 지원합니다. CREATE FUNCTION 명령문에서 STABLE 파라미터를 사용하고 동일한 인수를 사용하여 Lambda UDF를 여러 번 호출하면 예상되는 Lambda UDF 함수 호출 수가 감소합니다. STABLE 함수 변동성 카테고리는 [CREATE FUNCTION 파라미터](#)에 자세히 설명되어 있습니다.
- 여러 Lambda UDF 성능이 개선되었습니다. 특히 행 수준 보안(RLS) 정책으로 보호되는 테이블을 쿼리할 때 레코드 배치 지원이 개선되었습니다.
- Amazon Redshift RA3 클러스터의 전체 암호화 시간이 단축되고 암호화 중에 데이터 웨어하우스의 가용성이 향상됩니다. 자세한 내용은 [Amazon Redshift 데이터베이스 암호화](#)를 참조하세요.
- 새로운 시스템 뷰 SYS_MV_REFRESH_HISTORY가 Redshift에 추가되었습니다.
SYS_MV_REFRESH_HISTORY 뷰는 구체화된 보기의 새로 고침 작업에 대한 행을 포함합니다.

SYS_MV_REFRESH_HISTORY를 사용하여 구체화된 뷰의 새로 고침 기록을 확인할 수 있습니다. SYS_MV_REFRESH_HISTORY는 모든 사용자에게 표시됩니다. 슈퍼유저는 모든 행을 볼 수 있지만 일반 사용자는 자체 데이터만 볼 수 있습니다.

새 열 SPILLED_BLOCK_LOCAL_DISK가 시스템 뷰 SYS_QUERY_DETAIL에 추가되었습니다. SPILLED_BLOCK_LOCAL_DISK라는 새 열을 통해 고객은 로컬 디스크에 유출된 블록을 확인할 수 있습니다. SYS_QUERY_DETAIL을 사용하여 단계 수준에서 쿼리에 대한 세부 정보를 볼 수 있습니다. SYS_QUERY_DETAIL은 모든 사용자에게 표시됩니다. 슈퍼 사용자는 모든 행을 볼 수 있지만 일반 사용자는 액세스 권한을 가지고 있는 메타데이터에 한해 볼 수 있습니다.

- 새로운 시스템 뷰인 SYS_QUERY_TEXT가 Amazon Redshift Serverless 및 Amazon Redshift 프로비저닝에 추가되었습니다. SYS_QUERY_TEXT 뷰는 프로비저닝된 클러스터의 경우 [SVL_STATEMENTTEXT](#)와 유사합니다. SYS_QUERY_TEXT 뷰의 sequence 열을 사용하여 전체 SQL 명령문 텍스트를 가져올 수 있습니다.

Amazon Redshift 패치 175

이 패치의 클러스터 버전:

- 1.0.53064 - 현재 트랙 버전 - 2023년 7월 7일 릴리스됨
- 1.0.51973 - 현재 트랙 버전 - 2023년 6월 16일 릴리스됨
- 1.0.51781 - 현재 트랙 버전 - 2023년 6월 10일 릴리스됨
- 1.0.51314 - Amazon Redshift Serverless 버전 - 2023년 6월 3일 릴리스됨
- 1.0.51304 - 현재 트랙 버전 - 2023년 6월 2일 릴리스됨
- 1.0.50708 - 현재 트랙 버전 - 2023년 5월 19일 릴리스됨
- 1.0.50300 - 현재 트랙 버전 - 2023년 5월 8일 릴리스됨
- 1.0.49710 - Amazon Redshift Serverless 버전 - 2023년 4월 28일 릴리스됨
- 1.0.49676 - 현재 트랙 버전 - 2023년 4월 28일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 사소한 버그를 수정했습니다.
- Amazon Redshift 스트리밍 수집은 이제 크로스 리전 스트리밍 수집을 지원하여 소스 Amazon Kinesis Data Streams(KDS) 또는 Amazon Managed Streaming for Apache Kafka(MSK) 주제가 Amazon Redshift 데이터 웨어하우스가 있는 AWS 리전과 다른 AWS 리전에 위치할 수 있습니다.

[Amazon Kinesis Data Streams에서 스트리밍 수집 시작](#) 설명서가 개정되었으며 REGION 키워드의 사용 방법을 설명합니다.

- 이집트 일광 절약 시간이 조정되었습니다.
- RA3 클러스터의 전체 암호화 시간이 개선되었습니다.

Amazon Redshift 패치 174

1.0.51296– 2023년 6월 2일 릴리스됨

후행 트랙에 릴리스되었습니다. 릴리스 정보가 없습니다.

1.0.50468 – 2023년 5월 12일 릴리스됨

유지 관리 릴리스 릴리스 정보가 없습니다.

1.0.49780, 1.0.49868 및 1.0.49997 - 2023년 4월 28일 릴리스됨

이 버전에 대한 릴리스 정보:

- Lambda UDF에 대한 배치 처리 지원이 개선되었습니다.
- Lambda UDF에 대한 증분 배치 처리 기능이 도입되었습니다.
- 소스 데이터 변경 사항을 Amazon Redshift 테이블에 적용하는 새로운 MERGE SQL 명령이 도입되었습니다.
- Amazon Redshift 데이터 웨어하우스의 민감한 데이터를 보호하는 프로세스를 간소화하는 새로운 동적 데이터 마스킹 기능이 도입되었습니다.
- Lake Formation을 사용한 데이터 공유를 위한 새로운 중앙 집중식 액세스 제어를 통해 Lake Formation API 및 AWS Console을 사용하여 Amazon Redshift 데이터 공유의 테이블 및 뷰에 대한 권한 부여 관리, 액세스 제어 보기 및 권한 감사를 수행할 수 있습니다.
- 이집트 일광 절약 시간이 조정되었습니다.

1.0.49087 – 2023년 4월 12일 릴리스됨

유지 관리 릴리스 릴리스 정보가 없습니다.

1.0.48805 – 2023년 4월 5일 릴리스됨

이 버전에 대한 릴리스 정보:

- Amazon Redshift는 LZO 또는 ZSTD와 같은 대체 압축 인코딩에 비해 문자열 기반 데이터 처리 속도를 5~63배 높이는 Amazon Redshift의 새로운 압축 인코딩인 BYTEDICT를 사용하여 문자열이 많은 쿼리의 성능을 추가로 개선했습니다. 이 기능에 관한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [압축 인코딩](#)을 참조하시기 바랍니다.

1.0.48004 – 2023년 3월 17일 릴리스됨

유지 관리 릴리스 릴리스 정보가 없습니다.

1.0.47470 – 2023년 3월 11일 릴리스됨

이 버전에 대한 릴리스 정보:

- `pg_catalog.svv_table_info` 쿼리 성능이 개선되었습니다. 또한 새 열 `create_time`이 추가되었습니다. 테이블을 생성할 때 이 열에는 날짜/시간 스탬프가 UTC로 저장됩니다.
- 페더레이션 쿼리에 세션 수준 제한 시간을 지정하는 기능이 추가되었습니다.

Amazon Redshift 패치 173

1.0.49788 – 2023년 4월 28일 릴리스됨

이 버전에 대한 릴리스 정보:

- 이집트 일광 절약 시간이 조정되었습니다.

1.0.49074 – 2023년 4월 12일 릴리스됨

이 버전에 대한 릴리스 정보:

- 시간대 구성이 IANA 라이브러리 릴리스 2022g로 업데이트되었습니다.

1.0.48766 – 2023년 4월 5일 릴리스됨

유지 관리 릴리스 릴리스 정보가 없습니다.

1.0.48714 – 2023년 4월 5일 릴리스됨

유지 관리 릴리스 릴리스 정보가 없습니다.

1.0.48022 – 2023년 3월 17일 릴리스됨

유지 관리 릴리스 릴리스 정보가 없습니다.

1.0.47357 – 2023년 3월 7일 릴리스됨

유지 관리 릴리스 릴리스 정보가 없습니다.

1.0.46987 - 2023년 2월 24일에 릴리스됨

유지 관리 릴리스 릴리스 정보가 없습니다.

1.0.46806 - 2023년 2월 18일에 릴리스됨

유지 관리 릴리스 릴리스 정보가 없습니다.

1.0.46607 - 2023년 2월 13일에 릴리스됨

이 버전에 대한 릴리스 정보:

- 이제 배포 스타일이 DISTSTYLE KEY로 설정된 경우 인터리브된 정렬 키를 수동으로 설정한 테이블을 복합 정렬 키로 자동 변환하여 이러한 테이블의 성능을 개선합니다. 이 작업은 스냅샷을 Amazon Redshift Serverless로 복원할 때 수행됩니다.

1.0.45698 — 2023년 1월 20일 릴리스됨

이 버전에 대한 릴리스 정보:

- UNLOAD 명령에 파일 확장명 매개 변수를 추가하여, 파일 확장자가 파일 이름에 자동으로 추가됩니다.
- RLS 보호 객체를 데이터 공유에 추가하는 경우, 또는 RLS 보호 객체가 이미 데이터 공유에 속한 경우 RLS 보호 객체 보호가 기본적으로 지원됩니다. 이제 관리자는 데이터 공유에 대한 RLS를 해제하여 소비자가 보호 객체에 액세스하도록 허용할 수 있습니다.
- 모니터링을 위한 새 시스템 테이블인 SVV_ML_MODEL_INFO, SVV_MV_DEPENDENCY 및 SYS_LOAD_DETAIL을 추가합니다. 또한 시스템 테이블 SYS_QUERY_DETAIL에 data_skewness 및 time_skewness 열을 추가합니다.

Amazon Redshift 패치 172

이 패치의 클러스터 버전:

- 1.0.46534 - 2023년 2월 18일에 릴리스됨
- 1.0.46523 - 2023년 2월 13일에 릴리스됨
- 1.0.46206 — 2023년 2월 1일에 릴리스됨
- 1.0.45603 — 2023년 1월 20일 릴리스됨
- 1.0.44924 – 2022년 12월 19일 릴리스됨
- 1.0.44903 – 2022년 12월 18일 릴리스됨
- 1.0.44540 – 2022년 12월 13일 릴리스됨
- 1.0.44126 – 2022년 11월 23일 릴리스됨
- 1.0.43980 – 2022년 11월 17일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- CTAS에서 만든 테이블은 기본적으로 AUTO입니다.
- 구체화된 뷰에서 행 수준 보안(RLS)에 대한 지원을 추가합니다.
- 리전 간 데이터 공유를 개선하기 위해 S3 제한 시간을 늘립니다.
- 새로운 공간 함수 ST_GeomFromGeohash를 추가합니다.
- 기본 성능을 개선하기 위해 복합 기본 키에서 배포 키의 자동 선택을 개선합니다.
- 복합 기본 키가 있는 테이블의 배포 키에 자동 기본 키를 추가하여 기본 성능을 개선합니다.
- 데이터가 변경되더라도 더 많은 쿼리를 확장할 수 있도록 동시성 확장을 개선합니다.
- 데이터 공유 쿼리 성능을 향상시킵니다.
- 분류 모델에 대한 기계 학습 비율 지표를 추가합니다.
- 모니터링을 위한 새 시스템 테이블 추가: SVV_USER_INFO, SVV_MV_INFO, SYS_CONNECTION_LOG, SYS_DATASHARE_USAGE_PRODUCER, SYS_DATASHARE_USAGE_CONSUMER, SYS_DATASHARE_CHANGE_LOG
- Parquet 및 ORC 파일 유형을 대상으로 외부 테이블의 VARBYTE 열을 쿼리하는 기능 지원을 추가합니다.

Amazon Redshift 패치 171

이 패치의 클러스터 버전:

- 1.0.43931 – 2022년 11월 16일 릴리스됨
- 1.0.43551 – 2022년 11월 5일 릴리스됨
- 1.0.43331 – 2022년 9월 29일 릴리스됨
- 1.0.43029 - 2022년 9월 26일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- CONNECT BY 지원: CONNECT BY SQL 구문에 대한 지원을 추가하여 해당 데이터 세트 내의 상위-하위 관계를 기반으로 데이터 웨어하우스의 계층적 데이터를 재귀적으로 쿼리할 수 있습니다.

Amazon Redshift 패치 170

이 패치의 클러스터 버전:

- 1.0.43922 – 2022년 9월 21일 릴리스됨
- 1.0.43573 – 2022년 11월 7일 릴리스됨
- 1.0.41881 – 2022년 9월 20일 릴리스됨
- 1.0.41465 - 2022년 9월 7일 릴리스됨
- 1.0.40325 – 2022년 7월 27일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- ST_geomfromGeoJSON: GeoJSON 표현의 VARCHAR에서 Amazon Redshift 공간 지오메트리 객체를 생성합니다.

Amazon Redshift 패치 169

이 패치의 클러스터 버전:

- 1.0.41050 – 2022년 9월 7일 릴리스됨

- 1.0.40083 – 2022년 7월 16일 릴리스됨
- 1.0.39734 – 2022년 7월 7일 릴리스됨
- 1.0.39380 – 2022년 6월 23일 릴리스됨
- 1.0.39251 – 2022년 6월 15일 릴리스됨
- 1.0.39009 – 2022년 6월 8일 릴리스됨

이 패치의 새로운 기능 및 개선 사항

- 역할 기반 액세스 제어를 지원하기 위해 Alter Default Privileges 명령에 대한 파라미터로서 역할을 추가합니다.
- Parquet 및 ORC 파일에서 복사할 때 잘못된 UTF-8 문자를 바꿀 수 있도록 ACCEPTINVCHARS 파라미터를 추가합니다.
- OBJECT(k,v) 함수를 추가하여 키-값 페어에서 SUPER 객체를 구성합니다.

Amazon Redshift 패치 168

이 패치의 클러스터 버전:

- 1.0.38698 – 2022년 5월 25일 릴리스됨
- 1.0.38551 – 2022년 5월 20일 릴리스됨
- 1.0.38463 – 2022년 5월 18일 릴리스됨
- 1.0.38361 – 2022년 5월 13일 릴리스됨
- 1.0.38199 – 2022년 5월 9일 릴리스됨
- 1.0.38112 – 2022년 5월 6일 릴리스됨
- 1.0.37684 – 2022년 4월 20일 출시됨

이 패치의 새로운 기능 및 개선 사항

- Amazon Redshift ML에서 Linear Learner 모델 유형에 대한 지원이 추가되었습니다.
- SQL 트랜잭션 격리 수준에 SNAPSHOT 옵션이 추가되었습니다.
- VARBYTE 및 VARCHAR 데이터에 대한 새로운 해싱 알고리즘으로 farmhashFingerprint64가 추가되었습니다.
- 구체화된 보기의 증분 새로 고침에서 AVG 함수를 지원합니다.

- Redshift Spectrum 외부 테이블에 대한 상관 하위 쿼리를 지원합니다.
- 기본 쿼리 성능을 개선하기 위해 Amazon Redshift는 자동으로 특정 테이블에 대한 단일 열 기본 키를 배포 키로 선택합니다.

Amazon Redshift의 동작 변경 사항

Amazon Redshift가 계속 발전하고 개선됨에 따라 성능, 보안 및 사용자 경험을 향상시키기 위해 특정 동작의 변경 사항이 도입됩니다. 이 페이지는 이러한 중요한 업데이트에 대한 최신 정보를 제공하고, 조치를 취하고, 워크로드에 대한 잠재적 중단을 방지할 수 있는 포괄적인 리소스 역할을 합니다.

예정된 동작 변경 사항

다음은 향후 동작 변경 사항에 대한 설명입니다.

주제

- [2026년 6월 30일 이후 Scalar Python UDF의 지원 종료](#)
- [2026년 2월 27일 이후 구체화된 뷰\(MV\) 자동 새로 고침 동작 변경](#)
- [Amazon Redshift는 2026년 2월 16일 이후 데이터 공유를 통해 소비자 정보에 액세스하는 함수를 지원하지 않습니다.](#)
- [2026년 1월 31일부터 적용되는 Transport Layer Security\(TLS\) 버전 변경 사항](#)
- [Amazon Redshift는 2025년 10월 30일 이후 새 스칼라 Python UDF 생성을 지원하지 않음](#)

2026년 6월 30일 이후 Scalar Python UDF의 지원 종료

Amazon Redshift는 2026년 6월 30일 이후에 Python UDF에 대한 지원을 종료합니다. 대안으로 Lambda UDF를 사용하는 것이 좋습니다.

Lambda UDF는 Python UDF에 비해 다음과 같은 장점이 있습니다.

- Lambda UDF는 UDF 로직 내에서 외부 서비스 및 API에 연결할 수 있습니다.
- Lambda UDF는 Lambda 컴퓨팅 리소스를 사용합니다. 컴퓨팅 또는 메모리 집약적인 Lambda UDF는 Amazon Redshift의 쿼리 성능 또는 리소스 동시성에 영향을 주지 않습니다.
- Lambda UDF는 Python 코드 실행을 지원합니다. Lambda UDF는 특정 사용 사례에 따라 여러 Python 런타임을 지원합니다. 자세한 내용은 AWS Lambda 개발자 안내서의 [Python으로 빌드](#)를 참조하세요.
- 별도의 서비스 경계에서 사용자 지정 코드 실행을 격리할 수 있습니다. 이렇게 하면 유지 관리, 모니터링, 예산 책정 및 권한 관리가 간소화됩니다.

Lambda UDF 생성 및 사용에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Scalar Lambda UDF](#)를 참조하세요. 기존 Python UDF를 Lambda UDF로 변환하는 방법에 대한 자세한 내용은 [블로그 게시물](#)을 참조하세요.

2026년 2월 27일 이후 구체화된 뷰(MV) 자동 새로 고침 동작 변경

2026년 2월 27일부터 Amazon Redshift 구체화된 뷰에 대한 자동 REFRESH 쿼리는 백그라운드 자율 프로세스가 아닌 사용자 쿼리로 실행됩니다. 따라서 자동 REFRESH 쿼리는 이제 다른 사용자 쿼리와 동일한 우선 순위로 실행됩니다.

이 변경 사항은 자동 REFRESH가 활성화된 구체화된 뷰의 최신성을 개선하여 이전 동작과 비교하여 기본 테이블의 최신 변경 사항을 최신 상태로 유지하는 데 도움이 됩니다.

참고: MV 자동 REFRESH 동작 변경 기능은 패치 릴리스 P198 이상의 현재 트랙에 있는 Amazon Redshift 프로비저닝된 클러스터에만 활성화됩니다. 현재 서버리스에서는 비활성화되어 있습니다.

Amazon Redshift는 2026년 2월 16일 이후 데이터 공유를 통해 소비자 정보에 액세스하는 함수를 지원하지 않습니다.

2026년 2월 16일부터 Amazon Redshift는 데이터 공유를 통해 소비자 사용자, 역할 또는 그룹 정보에 액세스하는 `user_is_member_of` 및 관련 함수의 사용을 더 이상 지원하지 않습니다.

2026년 1월 31일부터 적용되는 Transport Layer Security(TLS) 버전 변경 사항

2026년 1월 31일부터 Amazon Redshift에서 Transport Layer Security(TLS) 버전 1.2가 적용됩니다. TLS 버전 1.0 또는 1.1을 사용하는 수신 연결은 거부됩니다. 이는 Amazon Redshift 프로비저닝된 클러스터 및 서버리스 작업 그룹에 모두 적용됩니다. TLS를 사용하지 않는 Amazon Redshift 데이터 웨어하우스는 이 변경의 영향을 받지 않습니다.

이 업데이트는 TLS 버전 1.0 또는 1.1을 사용하여 Amazon Redshift에 연결하는 경우 영향을 미칠 수 있습니다.

현재 사용 중인 TLS 버전을 확인하려면 다음을 수행합니다.

Amazon Redshift 프로비저닝된 경우: `STL_CONNECTION_LOG` 시스템 테이블 [1]에서 `sslversion` 열을 확인합니다.

Amazon Redshift Serverless 작업 그룹의 경우: `SYS_CONNECTION_LOG` 시스템 테이블 [2]에서 `ssl_version` 열을 확인합니다.

이 변경 후 Amazon Redshift 데이터 웨어하우스에 대한 중단 없는 액세스를 유지하려면 아래 나열된 단계를 따르세요.

1. TLS 1.2 이상을 지원하도록 클라이언트 업데이트
2. TLS 1.2 이상을 지원하는 최신 드라이버 버전 설치

가능하면 최신 버전의 Amazon Redshift 드라이버 [3]를 사용하는 것이 좋습니다.

[1] https://docs.aws.amazon.com/redshift/latest/dg/r_STL_CONNECTION_LOG.html

[2] https://docs.aws.amazon.com/redshift/latest/dg/SYS_CONNECTION_LOG.html

[3] <https://docs.aws.amazon.com/redshift/latest/mgmt/configuring-connections.html>

Amazon Redshift는 2025년 10월 30일 이후 새 스칼라 Python UDF 생성을 지원하지 않음

Amazon Redshift는 2025년 10월 30일 이후에 새 Python UDF 생성을 더 이상 지원하지 않습니다. 기존 Python UDF는 정상적으로 계속 작동합니다. 이 날짜 이전에 기존 Python UDF를 Lambda UDF로 마이그레이션하는 것이 좋습니다.

Lambda UDF는 Python UDF에 비해 다음과 같은 장점이 있습니다.

- Lambda UDF는 UDF 로직 내에서 외부 서비스 및 API에 연결할 수 있습니다.
- Lambda UDF는 Lambda 컴퓨팅 리소스를 사용합니다. 컴퓨팅 또는 메모리 집약적인 Lambda UDF는 Amazon Redshift의 쿼리 성능 또는 리소스 동시성에 영향을 주지 않습니다.
- Lambda UDF는 Python 코드 실행을 지원합니다. Lambda UDF는 특정 사용 사례에 따라 여러 Python 런타임을 지원합니다. 자세한 내용은 AWS Lambda 개발자 안내서의 [Python으로 빌드](#)를 참조하세요.
- 별도의 서비스 경계에서 사용자 지정 코드 실행을 격리할 수 있습니다. 이렇게 하면 유지 관리, 모니터링, 예산 책정 및 권한 관리가 간소화됩니다.

Lambda UDF 생성 및 사용에 대한 자세한 내용은 Amazon Redshift 데이터베이스 개발자 안내서의 [Scalar Lambda UDF](#)를 참조하세요. 기존 Python UDF를 Lambda UDF로 변환하는 방법에 대한 자세한 내용은 [블로그 게시물](#)을 참조하세요.

최근 동작 변경 사항

주제

- [Amazon Redshift는 2025년 8월 26일 이후 최신 IANA 시간대 데이터베이스를 사용합니다.](#)
- [2025년 8월 15일 이후 적용되는 Amazon Redshift Serverless RPU 변경 사항](#)
- [2025년 8월 10일 이후 적용되는 데이터베이스 감사 로깅 변경 사항](#)
- [2025년 6월 27일 이후 적용되는 서버리스 작업 그룹에 대한 가상 프라이빗 클라우드 엔드포인트 변경 사항](#)
- [2025년 5월 2일 이후에 적용되는 쿼리 모니터링 변경 사항](#)
- [2025년 1월 10일 후에 적용되는 보안 변경 사항](#)

Amazon Redshift는 2025년 8월 26일 이후 최신 IANA 시간대 데이터베이스를 사용합니다.

2025년 8월 26일부터 Amazon Redshift는 최신 IANA 시간대 데이터베이스 패치를 채택하여 시간대를 계산합니다. 이 변경 사항은 특정 시간대 및 기간에 대해 날짜 및 시간 변환이 작동하는 방식을 변경합니다. 이 업데이트는 [CONVERT_TIMEZONE](#) 함수 또는 [TIMEZONE](#) 및 [AT TIME ZONE](#) 명령으로 수행되는 것과 같은 명시적 시간대 변환을 비롯해 유형 캐스팅 작업 중, 특히 [TIMESTAMP](#)와 [TIMESTAMPTZ](#) 형식 간에 발생하는 암시적 변환에 영향을 줍니다.

다음은 시간대 및 기간 조합에 대한 업데이트 목록입니다.

- 시간대는 이제 2038년 이후 일광 절약 시간(DST)을 올바르게 준수합니다. 이전에는 2038년 이후 DST를 관찰한 시간대가 없었습니다.
- America/Toronto 시간대와 이 시간대를 연결하는 시간대는 자정이 아닌 현지 시간으로 오전 2시에 1947~1950에서 DST를 전환합니다.
- Amazon Redshift는 이제 모든 시간대에 대한 표준화 전 기간의 현지 평균 시간(LMT)을 올바르게 반영합니다. 이 기간은 시간대에 따라 다르며, 대부분의 시간대는 19세기 중반 이전의 표준 상태로 전환됩니다.
- EET, CET WET 및 MET는 이제 약어 대신 일반 시간대로 취급됩니다.
- Amazon Redshift에는 다음 시간대 이름이 더 이상 존재하지 않습니다.
 - Asia/Riyadh87
 - Asia/Riyadh88

- Asia/Riyadh89
- Mideast/Riyadh87
- Mideast/Riyadh88
- Mideast/Riyadh89
- US/Pacific-New

IANA 시간대 데이터베이스에 대한 자세한 내용은 IANA 시간대 데이터베이스 웹 사이트의 [시간대 데이터베이스](#)를 참조하세요.

2025년 8월 15일 이후 적용되는 Amazon Redshift Serverless RPU 변경 사항

2025년 8월 15일부터 Amazon Redshift Serverless 기본 Redshift 처리 장치(RPU)의 AWS 계정 할당량은 지난 6개월 동안의 최대 집계 기본 RPU의 1.5배 또는 3,200RPU 중 더 큰 값입니다.

2025년 8월 10일 이후 적용되는 데이터베이스 감사 로깅 변경 사항

2025년 8월 10일부터 Amazon Redshift에서 데이터베이스 감사 로깅이 변경되므로 조치가 필요합니다. Amazon Redshift는 데이터베이스의 연결 및 사용자 활동에 대한 정보를 Amazon S3 버킷과 CloudWatch에 로깅합니다. 2025년 8월 10일 이후 Amazon Redshift가 Redshift IAM USER를 지정하는 버킷 정책이 있는 Amazon S3 버킷에 대한 데이터베이스 감사 로깅을 중단합니다. 감사 로깅을 위해 S3 버킷 정책 내에서 Redshift SERVICE-PRINCIPAL을 대신 사용하도록 정책을 업데이트하는 것이 좋습니다. 감사 로깅에 대한 자세한 내용은 [Amazon Redshift 감사 로깅을 위한 버킷 권한](#) 섹션을 참조하세요.

로깅 중단을 방지하려면 2025년 8월 10일 이전에 관련 리전의 Redshift 서비스 위탁자에 액세스 권한을 부여하도록 S3 버킷 정책을 검토하고 업데이트하세요. 데이터베이스 감사 로깅에 대한 자세한 내용은 [Amazon S3의 로그 파일](#) 섹션을 참조하세요.

질문이나 우려 사항이 있는 경우 [AWS Support](#) 링크를 통해 AWS Support에 문의하세요.

2025년 6월 27일 이후 적용되는 서버리스 작업 그룹에 대한 가상 프라이빗 클라우드 엔드포인트 변경 사항

2025년 6월 27일부터 Amazon Redshift는 서버리스 작업 그룹에 대한 가상 프라이빗 클라우드 엔드포인트(VPCE) 지원을 변경합니다. 이 날짜 이전에는 Amazon Redshift가 작업 그룹 생성 시 엔드포인트

를 단일 가용 영역(AZ)에 배포하고 점차 최대 3개의 AZ까지 VPCE 지원을 확장했습니다. 이 날짜 이후 Amazon Redshift는 작업 그룹 생성 시 지정된 최대 3개의 가용 영역에 VPCE를 배포합니다.

자세한 내용은 [Amazon Redshift Serverless 사용 시 고려 사항](#) 섹션을 참조하세요.

질문이나 우려 사항이 있는 경우 [AWS Support](#) 링크를 통해 AWS Support에 문의하세요.

주제

- [2025년 5월 2일 이후에 적용되는 쿼리 모니터링 변경 사항](#)
- [2025년 1월 10일 후에 적용되는 보안 변경 사항](#)

2025년 5월 2일 이후에 적용되는 쿼리 모니터링 변경 사항

2025년 5월 2일부터 기존 및 새로 생성된 Redshift Serverless 작업 그룹 모두에 대해 쿼리 한도 탭에서 쿼리 CPU 시간(max_query_cpu_time) 및 쿼리 CPU 사용량(max_query_cpu_percentage) 지표를 더 이상 제공하지 않습니다. 이 날짜 이후에는 모든 Redshift Serverless 작업 그룹에서 이러한 지표를 기반으로 모든 쿼리 한도를 자동으로 제거합니다.

쿼리 한도는 런어웨이 쿼리를 포착하도록 설계되었습니다. 그러나 쿼리 CPU 시간(max_query_cpu_time)과 쿼리 CPU 사용량(max_query_cpu_percentage)은 쿼리 수명 동안 다를 수 있으므로 런어웨이 쿼리를 포착하는 데 일관되게 효과적인 방법이 아닙니다. 런어웨이 쿼리를 포착하려면 일관되고 실행 가능한 정보를 제공하는 쿼리 모니터링 지표를 활용하는 것이 좋습니다. 다음은 몇 가지 예입니다.

- 쿼리 실행 시간(max_query_execution_time): 예상 기간 내에 쿼리가 완료되도록 합니다.
- 행 개수 반환(max_scan_row_count): 처리 중인 데이터의 규모를 모니터링합니다.
- 쿼리 대기열 시간(max_query_queue_time): 대기열에서 시간이 소요되는 쿼리를 식별합니다.

지원되는 지표의 전체 목록은 [Amazon Redshift Serverless에 대한 쿼리 모니터링 지표](#)를 참조하세요.

2025년 1월 10일 후에 적용되는 보안 변경 사항

보안은 Amazon Web Services(AWS)에서 가장 우선시하는 사안입니다. 이를 위해 추가 설정 없이 데이터 보안의 모범 사례를 준수하고 잠재적인 구성 오류 위험을 줄이는 데 도움이 되는 향상된 보안 기본값을 도입하여 Amazon Redshift 환경의 보안 태세를 더욱 강화할 예정입니다. 잠재적인 종단을 방지하려면 프로비저닝된 클러스터 및 서버리스 작업 그룹 만들기 구성, 스크립트 및 도구를 검토하여 유효일 이전에 새 기본 설정에 맞게 필요한 변경을 수행하세요.

퍼블릭 액세스가 기본적으로 비활성화됨

2025년 1월 10일 후에는 새로 만들어지는 모든 프로비저닝된 클러스터와 스냅샷에서 복원된 클러스터에 대해 [퍼블릭 액세스](#)가 기본적으로 비활성화됩니다. 이 릴리스로 인해 기본적으로 클러스터에 대한 연결이 동일한 가상 프라이빗 클라우드(VPC) 내의 클라이언트 애플리케이션에서만 허용됩니다. 다른 VPC의 애플리케이션에서 데이터 웨어하우스에 액세스하려면 [교차 VPC](#) 액세스를 구성하세요. 이 변경 사항은 CreateCluster 및 RestoreFromClusterSnapshot API 작업과 해당 SDK 및 AWS CLI 명령에 반영됩니다. Amazon Redshift 콘솔에서 프로비저닝된 클러스터를 만드는 경우 클러스터에는 기본적으로 퍼블릭 액세스가 비활성화되어 있습니다.

여전히 퍼블릭 액세스가 필요한 경우 CreateCluster 또는 RestoreFromClusterSnapshot API 작업을 실행할 때 PubliclyAccessible 파라미터를 true로 설정하여 기본값을 재정의해야 합니다. 퍼블릭 액세스가 가능한 클러스터의 경우 보안 그룹 또는 네트워크 액세스 제어 목록(ACL)을 사용하여 액세스를 제한하는 것이 좋습니다. 자세한 내용은 [VPC 보안 그룹](#) 및 [Amazon Redshift 클러스터에 대한 보안 그룹 통신 설정 구성 또는 Amazon Redshift Serverless 작업 그룹](#) 섹션을 참조하세요.

암호화 기본 제공

2025년 1월 10일 후에 Amazon Redshift는 새로 만들어지는 모든 Amazon Redshift 프로비저닝된 클러스터의 기본 설정으로 암호화를 활성화하여 데이터 및 클러스터 보안을 더욱 강화할 것입니다. 스냅샷에서 복원된 클러스터에는 적용되지 않습니다.

이 변경 사항으로 인해 KMS 키를 지정하지 않고 AWS Management Console, AWS CLI 또는 API를 사용하여 프로비저닝된 클러스터를 만들 때 클러스터를 복호화할 수 없게 됩니다. 클러스터는 AWS 소유 키를 사용하여 자동으로 암호화됩니다.

이 업데이트는 자동 스크립트를 사용하여 암호화되지 않은 클러스터를 만들거나 암호화되지 않은 클러스터와의 데이터 공유를 활용하는 경우 영향을 미칠 수 있습니다. 원활한 전환을 위해 암호화되지 않은 클러스터를 만드는 스크립트를 업데이트하세요. 또한 암호화되지 않은 새 소비자 클러스터를 정기적으로 만들고 데이터 공유에 사용하는 경우 생산자와 소비자 클러스터가 모두 암호화되어 있는지 구성을 검토하여 데이터 공유 활동이 중단되지 않도록 하세요. 자세한 내용은 [Amazon Redshift 데이터베이스 암호화](#) 섹션을 참조하세요.

SSL 연결 적용

2025년 1월 10일 후에 Amazon Redshift는 새로 만들어지는 프로비저닝된 클러스터 및 복원된 클러스터에 연결하는 클라이언트에 대해 기본적으로 SSL 연결을 적용합니다. 이 기본값 변경 사항은 서버리스 작업 그룹에도 적용됩니다.

이 변경 사항으로 인해 새로 만들어지거나 복원되는 모든 클러스터에 `default.redshift-2.0`이라는 새 기본 파라미터 그룹이 사용되며 `require_ssl` 파라미터는 기본적으로 `true`로 설정됩니다. 지정된 파라미터 그룹 없이 만들어지는 모든 새 클러스터는 `default.redshift-2.0` 파라미터 그룹을 자동으로 활용합니다. Amazon Redshift 콘솔을 통해 클러스터를 만들 때 새 `default.redshift-2.0` 파라미터 그룹이 자동으로 선택됩니다. 이 변경 사항은 `CreateCluster` 및 `RestoreFromClusterSnapshot` API 작업과 해당 SDK 및 AWS CLI 명령에도 반영됩니다. 기존 또는 사용자 지정 파라미터 그룹을 사용하는 경우 Amazon Redshift는 파라미터 그룹에 지정된 `require_ssl` 값을 계속 준수합니다. 필요에 따라 사용자 지정 파라미터 그룹의 `require_ssl` 값을 변경할 수 있는 옵션은 여전히 사용할 수 있습니다.

Amazon Redshift Serverless 사용자의 경우 `config-parameters`에서 `require_ssl`의 기본값이 `true`로 변경됩니다. `require_ssl`이 `false`로 설정된 새 작업 그룹을 만들라는 요청은 거부됩니다. 작업 그룹의 만들어진 후에는 `require_ssl` 값을 `false`로 변경할 수 없습니다. 자세한 내용은 [연결을 위한 보안 옵션 구성](#) 섹션을 참조하세요.

특정 사용 사례에 필요한 경우 여전히 클러스터 또는 작업 그룹 설정을 수정하여 기본 동작을 변경할 수 있습니다.

AWS SDK를 사용한 Amazon Redshift용 코드 예제

다음 코드 예시에서는 Amazon Redshift를 AWS 소프트웨어 개발 키트(SDK)와 함께 사용하는 방법을 보여줍니다.

기본 사항은 서비스 내에서 필수 작업을 수행하는 방법을 보여주는 코드 예제입니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적 호출하여 특정 작업을 수행하는 방법을 보여주는 코드 예제입니다.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [AWS SDK와 함께 이 서비스 사용](#)을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

코드 예시

- [AWS SDK를 사용한 Amazon Redshift용 기본 예제](#)
 - [Hello Amazon Redshift](#)
 - [AWS SDK와 함께 사용되는 Amazon Redshift의 기본 사항 알아보기](#)
 - [AWS SDK를 사용한 Amazon Redshift에 대한 작업](#)
 - [AWS SDK 또는 CLI와 함께 CreateCluster 사용](#)
 - [AWS SDK 또는 CLI와 함께 DeleteCluster 사용](#)
 - [AWS SDK 또는 CLI와 함께 DescribeClusters 사용](#)
 - [AWS SDK와 함께 DescribeStatement 사용](#)
 - [AWS SDK와 함께 ExecuteStatement 사용](#)
 - [AWS SDK와 함께 GetStatementResult 사용](#)
 - [AWS SDK와 함께 ListDatabases 사용](#)
 - [AWS SDK 또는 CLI와 함께 ModifyCluster 사용](#)
- [AWS SDK를 사용한 Amazon S3에 대한 시나리오](#)
 - [Amazon Redshift 항목 추적기 만들기](#)

AWS SDK를 사용한 Amazon Redshift용 기본 예제

다음 코드 예제에서는 AWS SDK와 함께 Amazon Redshift의 기본 사항을 사용하는 방법을 보여줍니다.

예제

- [Hello Amazon Redshift](#)
- [AWS SDK와 함께 사용되는 Amazon Redshift의 기본 사항 알아보기](#)
- [AWS SDK를 사용한 Amazon Redshift에 대한 작업](#)
 - [AWS SDK 또는 CLI와 함께 CreateCluster 사용](#)
 - [AWS SDK 또는 CLI와 함께 DeleteCluster 사용](#)
 - [AWS SDK 또는 CLI와 함께 DescribeClusters 사용](#)
 - [AWS SDK와 함께 DescribeStatement 사용](#)
 - [AWS SDK와 함께 ExecuteStatement 사용](#)
 - [AWS SDK와 함께 GetStatementResult 사용](#)
 - [AWS SDK와 함께 ListDatabases 사용](#)
 - [AWS SDK 또는 CLI와 함께 ModifyCluster 사용](#)

Hello Amazon Redshift

다음 코드 예에서는 Amazon Redshift 사용을 시작하는 방법을 보여줍니다.

.NET

SDK for .NET(v4)

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
/// <summary>
/// Main method to run the Hello Amazon Redshift example.
/// </summary>
```



```
/// <param name="args">Command line arguments (not used).</param>
public static async Task Main(string[] args)
{
    var redshiftClient = new AmazonRedshiftClient();

    Console.WriteLine("Hello, Amazon Redshift! Let's list available
clusters:");

    var clusters = new List<Cluster>();

    try
    {
        // Use pagination to retrieve all clusters.
        var clustersPaginator =
redshiftClient.Paginators.DescribeClusters(new DescribeClustersRequest());

        await foreach (var response in clustersPaginator.Responses)
        {
            if (response.Clusters != null)
                clusters.AddRange(response.Clusters);
        }

        Console.WriteLine($"{clusters.Count} cluster(s) retrieved.");

        foreach (var cluster in clusters)
        {
            Console.WriteLine($"{cluster.ClusterIdentifier} (Status:
{cluster.ClusterStatus})");
        }
    }
    catch (AmazonRedshiftException ex)
    {
        Console.WriteLine($"Couldn't list clusters. Here's why:
{ex.Message}");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"An error occurred: {ex.Message}");
    }
}
```

- API에 대한 세부 정보는 AWS SDK for .NET API 참조의 [DescribeClusters](#)를 참조하세요.

Go

SDK for Go V2

 Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package main

import (
    "context"
    "fmt"

    "github.com/aws/aws-sdk-go-v2/aws"
    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/redshift"
)

// main uses the AWS SDK for Go V2 to create a Redshift client
// and list up to 10 clusters in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
    ctx := context.Background()
    sdkConfig, err := config.LoadDefaultConfig(ctx)
    if err != nil {
        fmt.Println("Couldn't load default configuration. Have you set up your AWS account?")
        fmt.Println(err)
        return
    }
    redshiftClient := redshift.NewFromConfig(sdkConfig)
    count := 20
    fmt.Printf("Let's list up to %v clusters for your account.\n", count)
    result, err := redshiftClient.DescribeClusters(ctx,
        &redshift.DescribeClustersInput{
            MaxRecords: aws.Int32(int32(count)),
        })
}
```

```

if err != nil {
    fmt.Printf("Couldn't list clusters for your account. Here's why: %v\n", err)
    return
}
if len(result.Clusters) == 0 {
    fmt.Println("You don't have any clusters!")
    return
}
for _, cluster := range result.Clusters {
    fmt.Printf("\t%v : %v\n", *cluster.ClusterIdentifier, *cluster.ClusterStatus)
}
}

```

- API에 대한 세부 정보는 AWS SDK for Go API 참조의 [DescribeClusters](#)를 참조하세요.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.redshift.RedshiftClient;
import
    software.amazon.awssdk.services.redshift.paginators.DescribeClustersIterable;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class HelloRedshift {

```

```

public static void main(String[] args) {
    Region region = Region.US_EAST_1;
    RedshiftClient redshiftClient = RedshiftClient.builder()
        .region(region)
        .build();

    listClustersPaginator(redshiftClient);
}

public static void listClustersPaginator(RedshiftClient redshiftClient) {
    DescribeClustersIterable clustersIterable =
redshiftClient.describeClustersPaginator();
    clustersIterable.stream()
        .flatMap(r -> r.clusters().stream())
        .forEach(cluster -> System.out
            .println(" Cluster identifier: " + cluster.clusterIdentifier() +
" status = " + cluster.clusterStatus()));
}
}

```

- API에 대한 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeClusters](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

import boto3

def hello_redshift(redshift_client):
    """
    Use the AWS SDK for Python (Boto3) to create an Amazon Redshift client and
    list
    the clusters in your account. This list might be empty if you haven't created
    any clusters.
    """

```

This example uses the default settings specified in your shared credentials and config files.

```
:param redshift_client: A Boto3 Redshift Client object.
"""

print("Hello, Redshift! Let's list your clusters:")
paginator = redshift_client.get_paginator("describe_clusters")
clusters = []
for page in paginator.paginate():
    clusters.extend(page["Clusters"])

print(f"{len(clusters)} cluster(s) were found.")

for cluster in clusters:
    print(f"  {cluster['ClusterIdentifier']}")

if __name__ == "__main__":
    hello_redshift(boto3.client("redshift"))
```

- API 세부 정보는 AWS SDK for Python(Boto3) API 참조의 [DescribeClusters](#)를 참조하세요.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [AWS SDK와 함께 이 서비스 사용](#)을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 함께 사용되는 Amazon Redshift의 기본 사항 알아보기

다음 코드 예제는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- Redshift 클러스터를 생성합니다.
- 클러스터의 데이터베이스를 나열합니다.
- Movies라는 테이블을 생성합니다.
- Movies 테이블을 채웁니다.
- Movies 테이블을 연도별로 쿼리합니다.
- Redshift 클러스터를 수정합니다.
- Amazon Redshift 클러스터를 삭제합니다.

.NET

SDK for .NET(v4)

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

Redshift 래퍼 클래스를 생성하여 작업을 관리합니다.

```
/// <summary>
/// Wrapper class for Amazon Redshift operations.
/// </summary>
public class RedshiftWrapper
{
    private readonly IAmazonRedshift _redshiftClient;
    private readonly IAmazonRedshiftDataAPIService _redshiftDataClient;

    /// <summary>
    /// Constructor for RedshiftWrapper.
    /// </summary>
    /// <param name="redshiftClient">Amazon Redshift client.</param>
    /// <param name="redshiftDataClient">Amazon Redshift Data API client.</param>
    public RedshiftWrapper(IAmazonRedshift redshiftClient,
        IAmazonRedshiftDataAPIService redshiftDataClient)
    {
        _redshiftClient = redshiftClient;
        _redshiftDataClient = redshiftDataClient;
    }

    /// <summary>
    /// Create a new Amazon Redshift cluster.
    /// </summary>
    /// <param name="clusterIdentifier">The identifier for the cluster.</param>
    /// <param name="databaseName">The name of the database.</param>
    /// <param name="masterUsername">The master username.</param>
    /// <param name="masterUserPassword">The master user password.</param>
    /// <param name="nodeType">The node type for the cluster.</param>
    /// <returns>The cluster that was created.</returns>
    public async Task<Cluster> CreateClusterAsync(string clusterIdentifier,
        string databaseName,
```

```

        string masterUsername, string masterUserPassword, string nodeType =
"ra3.large")
    {
        try
        {
            var request = new CreateClusterRequest
            {
                ClusterIdentifier = clusterIdentifier,
                DBName = databaseName,
                MasterUsername = masterUsername,
                MasterUserPassword = masterUserPassword,
                NodeType = nodeType,
                NumberOfNodes = 1,
                ClusterType = "single-node"
            };

            var response = await _redshiftClient.CreateClusterAsync(request);
            Console.WriteLine($"Created cluster {clusterIdentifier}");
            return response.Cluster;
        }
        catch (ClusterAlreadyExistsException ex)
        {
            Console.WriteLine($"Cluster already exists: {ex.Message}");
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't create cluster. Here's why:
{ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Describe Amazon Redshift clusters.
    /// </summary>
    /// <param name="clusterIdentifier">Optional cluster identifier to describe a
specific cluster.</param>
    /// <returns>A list of clusters.</returns>
    public async Task<List<Cluster>> DescribeClustersAsync(string?
clusterIdentifier = null)
    {
        try
        {

```

```

        var clusters = new List<Cluster>();
        var request = new DescribeClustersRequest();
        if (!string.IsNullOrEmpty(clusterIdentifier))
        {
            request.ClusterIdentifier = clusterIdentifier;
        }

        var clustersPaginator =
_redshiftClient.Paginators.DescribeClusters(request);
        await foreach (var response in clustersPaginator.Responses)
        {
            if (response.Clusters != null)
                clusters.AddRange(response.Clusters);
        }

        Console.WriteLine($"{clusters.Count} cluster(s) retrieved.");
        foreach (var cluster in clusters)
        {
            Console.WriteLine($"{cluster.ClusterIdentifier} (Status:
{cluster.ClusterStatus})");
        }

        return clusters;
    }
    catch (ClusterNotFoundException ex)
    {
        Console.WriteLine($"Cluster {clusterIdentifier} not found:
{ex.Message}");
        throw;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Couldn't describe clusters. Here's why:
{ex.Message}");
        throw;
    }
}

/// <summary>
/// Modify an Amazon Redshift cluster.
/// </summary>
/// <param name="clusterIdentifier">The identifier for the cluster.</param>
/// <param name="preferredMaintenanceWindow">The preferred maintenance
window.</param>

```



```

    /// <returns>True if successful.</returns>
    public async Task<bool> ModifyClusterAsync(string clusterIdentifier, string
preferredMaintenanceWindow)
    {
        try
        {
            var request = new ModifyClusterRequest
            {
                ClusterIdentifier = clusterIdentifier,
                PreferredMaintenanceWindow = preferredMaintenanceWindow
            };

            var response = await _redshiftClient.ModifyClusterAsync(request);
            Console.WriteLine($"The modified cluster was successfully modified
and has {response.Cluster.PreferredMaintenanceWindow} as the maintenance
window");
            return true;
        }
        catch (ClusterNotFoundException ex)
        {
            Console.WriteLine($"Cluster {clusterIdentifier} not found:
{ex.Message}");
            return false;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't modify cluster. Here's why:
{ex.Message}");
            return false;
        }
    }

    /// <summary>
    /// Delete an Amazon Redshift cluster without a final snapshot.
    /// </summary>
    /// <param name="clusterIdentifier">The identifier for the cluster.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteClusterWithoutSnapshotAsync(string
clusterIdentifier)
    {
        try
        {
            var request = new DeleteClusterRequest
            {

```

```

        ClusterIdentifier = clusterIdentifier,
        SkipFinalClusterSnapshot = true
    };

    var response = await _redshiftClient.DeleteClusterAsync(request);
    Console.WriteLine($"The {clusterIdentifier} was deleted");
    return true;
}
catch (ClusterNotFoundException ex)
{
    Console.WriteLine($"Cluster not found: {ex.Message}");
    return false;
}
catch (Exception ex)
{
    Console.WriteLine($"Couldn't delete cluster. Here's why:
{ex.Message}");
    return false;
}
}

/// <summary>
/// List databases in a Redshift cluster.
/// </summary>
/// <param name="clusterIdentifier">The cluster identifier.</param>
/// <param name="dbUser">The database user.</param>
/// <param name="dbUser">The database name for authentication.</param>
/// <returns>A list of database names.</returns>
public async Task<List<string>> ListDatabasesAsync(string clusterIdentifier,
string dbUser, string databaseName)
{
    try
    {
        var request = new ListDatabasesRequest
        {
            ClusterIdentifier = clusterIdentifier,
            DbUser = dbUser,
            Database = databaseName
        };

        var response = await _redshiftDataClient.ListDatabasesAsync(request);
        var databases = new List<string>();

        foreach (var database in response.Databases)

```

```

        {
            Console.WriteLine($"The database name is : {database}");
            databases.Add(database);
        }

        return databases;
    }
    catch (Amazon.RedshiftDataAPIService.Model.ValidationException ex)
    {
        Console.WriteLine($"Validation error: {ex.Message}");
        throw;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Couldn't list databases. Here's why:
{ex.Message}");
        throw;
    }
}

/// <summary>
/// Create a table in the Redshift database.
/// </summary>
/// <param name="clusterIdentifier">The cluster identifier.</param>
/// <param name="database">The database name.</param>
/// <param name="dbUser">The database user.</param>
/// <returns>The statement ID.</returns>
public async Task<string> CreateTableAsync(string clusterIdentifier, string
database, string dbUser)
{
    try
    {
        var sqlStatement = @"
            CREATE TABLE Movies (
                id INTEGER PRIMARY KEY,
                title VARCHAR(250) NOT NULL,
                year INTEGER NOT NULL
            )";

        var request = new ExecuteStatementRequest
        {
            ClusterIdentifier = clusterIdentifier,
            Database = database,
            DbUser = dbUser,

```

```

        Sql = sqlStatement
    };

    var response = await
_redshiftDataClient.ExecuteStatementAsync(request);
    await WaitForStatementToCompleteAsync(response.Id);
    Console.WriteLine("Table created: Movies");
    return response.Id;
}
catch (Amazon.RedshiftDataAPIService.Model.ValidationException ex)
{
    Console.WriteLine($"Validation error: {ex.Message}");
    throw;
}
catch (Exception ex)
{
    Console.WriteLine($"Couldn't create table. Here's why:
{ex.Message}");
    throw;
}
}

/// <summary>
/// Insert a record into the Movies table using parameterized query.
/// </summary>
/// <param name="clusterIdentifier">The cluster identifier.</param>
/// <param name="database">The database name.</param>
/// <param name="dbUser">The database user.</param>
/// <param name="id">The movie ID.</param>
/// <param name="title">The movie title.</param>
/// <param name="year">The movie year.</param>
/// <returns>The statement ID.</returns>
public async Task<string> InsertMovieAsync(string clusterIdentifier, string
database, string dbUser,
    int id, string title, int year)
{
    try
    {
        var sqlStatement = "INSERT INTO Movies (id, title, year) VALUES
(:id, :title, :year)";

        var request = new ExecuteStatementRequest
        {
            ClusterIdentifier = clusterIdentifier,

```

```

        Database = database,
        DbUser = dbUser,
        Sql = sqlStatement,
        Parameters = new List<SqlParameter>
        {
            new SqlParameter { Name = "id", Value = id.ToString() },
            new SqlParameter { Name = "title", Value = title },
            new SqlParameter { Name = "year", Value = year.ToString() }
        }
    };

    var response = await
_redshiftDataClient.ExecuteStatementAsync(request);
    await WaitForStatementToCompleteAsync(response.Id);
    Console.WriteLine($"Inserted: {title} ({year})");
    return response.Id;
}
catch (Amazon.RedshiftDataAPIService.Model.ValidationException ex)
{
    Console.WriteLine($"Validation error: {ex.Message}");
    throw;
}
catch (Exception ex)
{
    Console.WriteLine($"Couldn't insert movie. Here's why:
{ex.Message}");
    throw;
}
}

/// <summary>
/// Query movies by year using parameterized query.
/// </summary>
/// <param name="clusterIdentifier">The cluster identifier.</param>
/// <param name="database">The database name.</param>
/// <param name="dbUser">The database user.</param>
/// <param name="year">The year to query.</param>
/// <returns>A list of movie titles.</returns>
public async Task<List<string>> QueryMoviesByYearAsync(string
clusterIdentifier, string database,
    string dbUser, int year)
{
    try
    {

```

```
var sqlStatement = "SELECT title FROM Movies WHERE year = :year";

var request = new ExecuteStatementRequest
{
    ClusterIdentifier = clusterIdentifier,
    Database = database,
    DbUser = dbUser,
    Sql = sqlStatement,
    Parameters = new List<SqlParameter>
    {
        new SqlParameter { Name = "year", Value = year.ToString() }
    }
};

var response = await
_redshiftDataClient.ExecuteStatementAsync(request);
Console.WriteLine($"The identifier of the statement is
{response.Id}");

await WaitForStatementToCompleteAsync(response.Id);

var results = await GetStatementResultAsync(response.Id);
var movieTitles = new List<string>();

foreach (var row in results)
{
    if (row.Count > 0)
    {
        var title = row[0].StringValue;
        Console.WriteLine($"The Movie title field is {title}");
        movieTitles.Add(title);
    }
}

return movieTitles;
}
catch (Amazon.RedshiftDataAPIService.Model.ValidationException ex)
{
    Console.WriteLine($"Validation error: {ex.Message}");
    throw;
}
catch (Exception ex)
{

```

```

        Console.WriteLine($"Couldn't query movies. Here's why:
{ex.Message}");
        throw;
    }
}

/// <summary>
/// Describe a statement execution.
/// </summary>
/// <param name="statementId">The statement ID.</param>
/// <returns>The statement description.</returns>
public async Task<DescribeStatementResponse> DescribeStatementAsync(string
statementId)
{
    try
    {
        var request = new DescribeStatementRequest
        {
            Id = statementId
        };

        var response = await
_redshiftDataClient.DescribeStatementAsync(request);
        return response;
    }
    catch (Amazon.RedshiftDataAPIService.Model.ResourceNotFoundException ex)
    {
        Console.WriteLine($"Statement not found: {ex.Message}");
        throw;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Couldn't describe statement. Here's why:
{ex.Message}");
        throw;
    }
}

/// <summary>
/// Get the results of a statement execution.
/// </summary>
/// <param name="statementId">The statement ID.</param>
/// <returns>A list of result rows.</returns>

```

```

    public async Task<List<List<Field>>> GetStatementResultAsync(string
statementId)
    {
        try
        {
            var request = new GetStatementResultRequest
            {
                Id = statementId
            };

            var response = await
_redshiftDataClient.GetStatementResultAsync(request);
            return response.Records;
        }
        catch (Amazon.RedshiftDataAPIService.Model.ResourceNotFoundException ex)
        {
            Console.WriteLine($"Statement not found: {ex.Message}");
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't get statement result. Here's why:
{ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Wait for a statement to complete execution.
    /// </summary>
    /// <param name="statementId">The statement ID.</param>
    /// <returns>A task representing the asynchronous operation.</returns>
    private async Task WaitForStatementToCompleteAsync(string statementId)
    {
        var status = StatusString.SUBMITTED;
        DescribeStatementResponse? response = null;

        while (status == StatusString.SUBMITTED || status == StatusString.PICKED
|| status == StatusString.STARTED)
        {
            await Task.Delay(1000); // Wait 1 second
            response = await DescribeStatementAsync(statementId);
            status = response.Status;
            Console.WriteLine($"...{status}");
        }
    }

```



```
    }

    if (status == StatusString.FINISHED)
    {
        Console.WriteLine("The statement is finished!");
    }
    else
    {
        var errorMessage = response?.Error ?? "Unknown error";
        Console.WriteLine($"The statement failed with status: {status}");
        Console.WriteLine($"Error message: {errorMessage}");
    }
}

/// <summary>
/// Wait for a cluster to become available.
/// </summary>
/// <param name="clusterIdentifier">The cluster identifier.</param>
/// <returns>A task representing the asynchronous operation.</returns>
public async Task WaitForClusterAvailableAsync(string clusterIdentifier)
{
    Console.WriteLine($"Wait until {clusterIdentifier} is available. This may
take a few minutes.");

    var startTime = DateTime.Now;
    var clusters = await DescribeClustersAsync(clusterIdentifier);

    while (clusters[0].ClusterStatus != "available")
    {
        var elapsed = DateTime.Now - startTime;
        Console.WriteLine($"Elapsed Time: {elapsed:mm\\\:ss} - Waiting for
cluster...");

        await Task.Delay(5000); // Wait 5 seconds
        clusters = await DescribeClustersAsync(clusterIdentifier);
    }

    var totalElapsed = DateTime.Now - startTime;
    Console.WriteLine($"Cluster is available! Total Elapsed Time:
{totalElapsed:mm\\\:ss}");
}
}
```

Redshift 기본 사항을 시연하는 대화형 시나리오를 실행합니다.

```

/// <summary>
/// Amazon Redshift Getting Started Scenario.
/// </summary>
public class RedshiftBasics
{
    public static bool IsInteractive = true;
    public static RedshiftWrapper? Wrapper = null;
    public static ILogger logger = null!;
    private static readonly string _moviesFilePath = "../../../../../resources/sample_files/movies.json";

    /// <summary>
    /// Main method for the Amazon Redshift Getting Started scenario.
    /// </summary>
    /// <param name="args">Command line arguments.</param>
    public static async Task Main(string[] args)
    {
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonRedshift>()
                    .AddAWSService<IAmazonRedshiftDataAPIService>()
                    .AddTransient<RedshiftWrapper>()
            )
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<RedshiftBasics>();

        Wrapper = host.Services.GetRequiredService<RedshiftWrapper>();

        await RunScenarioAsync();
    }

    /// <summary>
    /// Run the complete Amazon Redshift scenario.
    /// </summary>
    public static async Task RunScenarioAsync()
    {
        // Set all variables to default values
        string userName = "awsuser";
        string userPassword = "AwsUser1000";
        string clusterIdentifier = "redshift-cluster-movies";
    }
}

```

```

        var databaseName = "dev";
        int recordCount = 50;
        int year = 2013;
        try
        {
            Console.WriteLine(

"=====");

            Console.WriteLine("Welcome to the Amazon Redshift SDK Getting Started
scenario.");
            Console.WriteLine(
                "This .NET program demonstrates how to interact with Amazon
Redshift by using the AWS SDK for .NET.");
            Console.WriteLine("Let's get started...");
            Console.WriteLine(

"=====");

            // Step 1: Get user credentials (if interactive)
            if (IsInteractive)
            {
                Console.WriteLine("Please enter a user name for the cluster
(default is awsuser):");
                var userInput = Console.ReadLine();
                if (!string.IsNullOrEmpty(userInput))
                    userName = userInput;

                Console.WriteLine("=====");
                Console.WriteLine("Please enter a user password for the cluster
(default is AwsUser1000):");
                var passwordInput = Console.ReadLine();
                if (!string.IsNullOrEmpty(passwordInput))
                    userPassword = passwordInput;

                Console.WriteLine("=====");

                // Step 2: Get cluster identifier
                Console.WriteLine("Enter a cluster id value (default is redshift-
cluster-movies):");
                var clusterInput = Console.ReadLine();
                if (!string.IsNullOrEmpty(clusterInput))
                    clusterIdentifier = clusterInput;
            }
        }
    }
}

```

```
    }
    else
    {
        Console.WriteLine($"Using default values: userName={userName},
clusterIdentifier={clusterIdentifier}");
    }

    // Step 3: Create Redshift cluster
    await Wrapper!.CreateClusterAsync(clusterIdentifier, databaseName,
userName, userPassword);

Console.WriteLine("=====

    // Step 4: Wait for cluster to become available

Console.WriteLine("=====
    await Wrapper.WaitForClusterAvailableAsync(clusterIdentifier);

Console.WriteLine("=====

    // Step 5: List databases

Console.WriteLine("=====
    Console.WriteLine($" When you created {clusterIdentifier}, the dev
database is created by default and used in this scenario.");
    Console.WriteLine(" To create a custom database, you need to have a
CREATEDB privilege.");
    Console.WriteLine(" For more information, see the documentation here:
https://docs.aws.amazon.com/redshift/latest/dg/r\_CREATE\_DATABASE.html.");
    if (IsInteractive)
    {
        Console.WriteLine("Press Enter to continue...");
        Console.ReadLine();
    }

Console.WriteLine("=====

Console.WriteLine("=====
    Console.WriteLine($"List databases in {clusterIdentifier}");
    if (IsInteractive)
    {
        Console.WriteLine("Press Enter to continue...");
        Console.ReadLine();
    }
}
```

```
    }
    await Wrapper.ListDatabasesAsync(clusterIdentifier, userName,
databaseName);

Console.WriteLine("=====

// Step 6: Create Movies table

Console.WriteLine("=====
Console.WriteLine("Now you will create a table named Movies.");
if (IsInteractive)
{
    Console.WriteLine("Press Enter to continue...");
    Console.ReadLine();
}
await Wrapper.CreateTableAsync(clusterIdentifier, databaseName,
userName);

Console.WriteLine("=====

// Step 7: Populate the Movies table

Console.WriteLine("=====
Console.WriteLine("Populate the Movies table using the Movies.json
file.");

if (IsInteractive)
{
    Console.WriteLine("Specify the number of records you would like
to add to the Movies Table.");
    Console.WriteLine("Please enter a value between 50 and 200.");
    Console.Write("Enter a value: ");

    var recordCountInput = Console.ReadLine();
    if (int.TryParse(recordCountInput, out var inputCount) &&
inputCount is >= 50 and <= 200)
    {
        recordCount = inputCount;
    }
    else
    {
        Console.WriteLine($"Invalid input. Using default value of
{recordCount}.");
    }
}
```

```

    }
    else
    {
        Console.WriteLine($"Using default record count: {recordCount}");
    }

    await PopulateMoviesTableAsync(clusterIdentifier, databaseName,
userName, recordCount);
    Console.WriteLine($"{recordCount} records were added to the Movies
table.");

Console.WriteLine("=====

// Step 8 & 9: Query movies by year

Console.WriteLine("=====
    Console.WriteLine("Query the Movies table by year. Enter a value
between 2012-2014.");

    if (IsInteractive)
    {
        Console.Write("Enter a year: ");
        var yearInput = Console.ReadLine();
        if (int.TryParse(yearInput, out var inputYear) && inputYear is >=
2012 and <= 2014)
        {
            year = inputYear;
        }
        else
        {
            Console.WriteLine($"Invalid input. Using default value of
{year}.");
        }
    }
    else
    {
        Console.WriteLine($"Using default year: {year}");
    }

    await Wrapper.QueryMoviesByYearAsync(clusterIdentifier, databaseName,
userName, year);

Console.WriteLine("=====

```

```

        // Step 10: Modify the cluster

Console.WriteLine("=====
    Console.WriteLine("Now you will modify the Redshift cluster.");
    if (IsInteractive)
    {
        Console.WriteLine("Press Enter to continue...");
        Console.ReadLine();
    }
    await Wrapper.ModifyClusterAsync(clusterIdentifier, "wed:07:30-
wed:08:00");

Console.WriteLine("=====

        // Step 11 & 12: Delete cluster confirmation

Console.WriteLine("=====
    if (IsInteractive)
    {
        Console.WriteLine("Would you like to delete the Amazon Redshift
cluster? (y/n)");
        var deleteResponse = Console.ReadLine();
        if (deleteResponse?.ToLower() == "y")
        {
            await
Wrapper.DeleteClusterWithoutSnapshotAsync(clusterIdentifier);
        }
    }
    else
    {
        Console.WriteLine("Deleting the Amazon Redshift cluster...");
        await
Wrapper.DeleteClusterWithoutSnapshotAsync(clusterIdentifier);
    }

Console.WriteLine("=====

Console.WriteLine("=====
    Console.WriteLine("This concludes the Amazon Redshift SDK Getting
Started scenario.");

Console.WriteLine("=====
    }

```

```

        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred during the scenario:
{ex.Message}");
            Console.WriteLine("Deleting the Amazon Redshift cluster...");
            await Wrapper!.DeleteClusterWithoutSnapshotAsync(clusterIdentifier);
            throw;
        }
    }

    /// <summary>
    /// Populate the Movies table with data from the JSON file.
    /// </summary>
    /// <param name="clusterIdentifier">The cluster identifier.</param>
    /// <param name="database">The database name.</param>
    /// <param name="dbUser">The database user.</param>
    /// <param name="recordCount">Number of records to insert.</param>
    private static async Task PopulateMoviesTableAsync(string clusterIdentifier,
string database, string dbUser, int recordCount)
    {
        if (!File.Exists(_moviesFilePath))
        {
            throw new FileNotFoundException($"Required movies data file not found
at: {_moviesFilePath}");
        }

        var jsonContent = await File.ReadAllTextAsync(_moviesFilePath);
        var options = new JsonSerializerOptions
        {
            PropertyNameCaseInsensitive = true
        };
        var movies = JsonSerializer.Deserialize<List<Movie>>(jsonContent,
options);

        if (movies == null || movies.Count == 0)
        {
            throw new InvalidOperationException("Failed to parse movies JSON file
or file is empty.");
        }

        var insertCount = Math.Min(recordCount, movies.Count);

        for (int i = 0; i < insertCount; i++)
        {

```



```

        var movie = movies[i];
        await Wrapper!.InsertMovieAsync(clusterIdentifier, database, dbUser,
i, movie.Title, movie.Year);
    }
}

/// <summary>
/// Movie data model.
/// </summary>
private class Movie
{
    public string Title { get; set; } = string.Empty;
    public int Year { get; set; }
}
}

```

- API 세부 정보는 AWS SDK for .NET API 참조의 다음 주제를 참조하세요.

- [CreateCluster](#) :
- [DescribeClusters](#)
- [DescribeStatement](#)
- [ExecuteStatement](#)
- [GetStatementResult](#)
- [ListDatabasesPaginator](#)
- [ModifyCluster](#)

Go

SDK for Go V2

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package scenarios
```

```
import (  
    "context"  
    "encoding/json"  
    "errors"  
    "fmt"  
    "log"  
    "math/rand"  
    "strings"  
    "time"  
  
    "github.com/aws/aws-sdk-go-v2/aws"  
    redshift_types "github.com/aws/aws-sdk-go-v2/service/redshift/types"  
    redshiftdata_types "github.com/aws/aws-sdk-go-v2/service/redshiftdata/types"  
    "github.com/aws/aws-sdk-go-v2/service/secretsmanager"  
    "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"  
    "github.com/awsdocs/aws-doc-sdk-examples/gov2/redshift/actions"  
  
    "github.com/aws/aws-sdk-go-v2/service/redshift"  
    "github.com/aws/aws-sdk-go-v2/service/redshiftdata"  
)  
  
// IScenarioHelper abstracts input and wait functions from a scenario so that  
// they  
// can be mocked for unit testing.  
type IScenarioHelper interface {  
    GetName() string  
}  
  
const rMax = 100000  
  
type ScenarioHelper struct {  
    Prefix string  
    Random *rand.Rand  
}  
  
// GetName returns a unique name formed of a prefix and a random number.  
func (helper ScenarioHelper) GetName() string {  
    return fmt.Sprintf("%v%v", helper.Prefix, helper.Random.Intn(rMax))  
}  
  
// RedshiftBasicsScenario separates the steps of this scenario into individual  
// functions so that  
// they are simpler to read and understand.
```

```

type RedshiftBasicsScenario struct {
    sdkConfig      aws.Config
    helper         IScenarioHelper
    questioner     demotools.IQuestioner
    pauser         demotools.IPausable
    filesystem     demotools.IFileSystem
    redshiftActor  *actions.RedshiftActions
    redshiftDataActor *actions.RedshiftDataActions
    secretsmanager *SecretsManager
}

// SecretsManager is used to retrieve username and password information from a
// secure service.
type SecretsManager struct {
    SecretsManagerClient *secretsmanager.Client
}

// RedshiftBasics constructs a new Redshift Basics runner.
func RedshiftBasics(sdkConfig aws.Config, questioner demotools.IQuestioner,
    pauser demotools.IPausable, filesystem demotools.IFileSystem, helper
    IScenarioHelper) RedshiftBasicsScenario {
    scenario := RedshiftBasicsScenario{
        sdkConfig:      sdkConfig,
        helper:         helper,
        questioner:     questioner,
        pauser:         pauser,
        filesystem:     filesystem,
        secretsmanager: &SecretsManager{SecretsManagerClient:
            secretsmanager.NewFromConfig(sdkConfig)},
        redshiftActor:  &actions.RedshiftActions{RedshiftClient:
            redshift.NewFromConfig(sdkConfig)},
        redshiftDataActor: &actions.RedshiftDataActions{RedshiftDataClient:
            redshiftdata.NewFromConfig(sdkConfig)},
    }
    return scenario
}

// Movie makes it easier to use Movie objects given in json format.
type Movie struct {
    ID    int    `json:"id"`
    Title string `json:"title"`
    Year  int    `json:"year"`
}

```

```
// User makes it easier to get the User data back from SecretsManager and use it
// later.
type User struct {
    Username string `json:"userName"`
    Password string `json:"userPassword"`
}

// Run runs the RedshiftBasics interactive example that shows you how to use
// Amazon
// Redshift and how to interact with its common endpoints.
//
// 0. Retrieve username and password information to access Redshift.
// 1. Create a cluster.
// 2. Wait for the cluster to become available.
// 3. List the available databases in the region.
// 4. Create a table named "Movies" in the "dev" database.
// 5. Populate the movies table from the "movies.json" file.
// 6. Query the movies table by year.
// 7. Modify the cluster's maintenance window.
// 8. Optionally clean up all resources created during this demo.
//
// This example creates an Amazon Redshift service client from the specified
// sdkConfig so that
// you can replace it with a mocked or stubbed config for unit testing.
//
// It uses a questioner from the `demotools` package to get input during the
// example.
// This package can be found in the ../../demotools folder of this repo.
func (runner *RedshiftBasicsScenario) Run(ctx context.Context) {

    user := User{}
    secretId := "s3express/basics/secrets"
    clusterId := "demo-cluster-1"
    maintenanceWindow := "wed:07:30-wed:08:00"
    databaseName := "dev"
    tableName := "Movies"
    fileName := "Movies.json"
    nodeType := "ra3.xlplus"
    clusterType := "single-node"

    defer func() {
        if r := recover(); r != nil {
```

```

    log.Println("Something went wrong with the demo.")
    _, isMock := runner.questioner.(*demotools.MockQuestioner)
    if isMock || runner.questioner.AskBool("Do you want to see the full error
message (y/n)?", "y") {
        log.Println(r)
    }
    runner.cleanUpResources(ctx, clusterId, databaseName, tableName,
user.Username, runner.questioner)
}
}()

// Retrieve the userName and userPassword from SecretsManager
output, err := runner.secretsmanager.SecretsManagerClient.GetSecretValue(ctx,
&secretsmanager.GetSecretValueInput{
    SecretId: aws.String(secretId),
})
if err != nil {
    log.Printf("There was a problem getting the secret value: %s", err)
    log.Printf("Please make sure to create a secret named 's3express/basics/
secrets' with keys of 'userName' and 'userPassword'.")
    panic(err)
}

err = json.Unmarshal([]byte(*output.SecretString), &user)
if err != nil {
    log.Printf("There was a problem parsing the secret value from JSON: %s", err)
    panic(err)
}

// Create the Redshift cluster
_, err = runner.redshiftActor.CreateCluster(ctx, clusterId, user.Username,
user.Password, nodeType, clusterType, true)
if err != nil {
    var clusterAlreadyExistsFault *redshift_types.ClusterAlreadyExistsFault
    if errors.As(err, &clusterAlreadyExistsFault) {
        log.Println("Cluster already exists. Continuing.")
    } else {
        log.Println("Error creating cluster.")
        panic(err)
    }
}

// Wait for the cluster to become available

```

```
waiter :=
redshift.NewClusterAvailableWaiter(runner.redshiftActor.RedshiftClient)
err = waiter.Wait(ctx, &redshift.DescribeClustersInput{
    ClusterIdentifier: aws.String(clusterId),
}, 5*time.Minute)
if err != nil {
    log.Println("An error occurred waiting for the cluster.")
    panic(err)
}

// Get some info about the cluster
describeOutput, err := runner.redshiftActor.DescribeClusters(ctx, clusterId)
if err != nil {
    log.Println("Something went wrong trying to get information about the
cluster.")
    panic(err)
}
log.Println("Here's some information about the cluster.")
log.Printf("The cluster's status is %s",
*describeOutput.Clusters[0].ClusterStatus)
log.Printf("The cluster was created at %s",
*describeOutput.Clusters[0].ClusterCreateTime)

// List databases
log.Println("List databases in", clusterId)
runner.questioner.Ask("Press Enter to continue...")
err = runner.redshiftDataActor.ListDatabases(ctx, clusterId, databaseName,
user.Username)
if err != nil {
    log.Printf("Failed to list databases: %v\n", err)
    panic(err)
}

// Create the "Movies" table
log.Println("Now you will create a table named " + tableName + ".")
runner.questioner.Ask("Press Enter to continue...")
err = nil
result, err := runner.redshiftDataActor.CreateTable(ctx, clusterId,
databaseName, tableName, user.Username, runner.pauser, []string{"title
VARCHAR(256)", "year INT"})
if err != nil {
    log.Printf("Failed to create table: %v\n", err)
    panic(err)
}
```

```

describeInput := redshiftdata.DescribeStatementInput{
    Id: result.Id,
}
query := actions.RedshiftQuery{
    Context: ctx,
    Input:   describeInput,
    Result:  result,
}
err = runner.redshiftDataActor.WaitForQueryStatus(query, runner.pauser, true)
if err != nil {
    log.Printf("Failed to execute query: %v\n", err)
    panic(err)
}
log.Printf("Successfully executed query\n")

// Populate the "Movies" table
runner.PopulateMoviesTable(ctx, clusterId, databaseName, tableName,
user.Username, fileName)

// Query the "Movies" table by year
log.Println("Query the Movies table by year.")
year := runner.questioner.AskInt(
    fmt.Sprintf("Enter a value between %v and %v:", 2012, 2014),
    demotools.InIntRange{Lower: 2012, Upper: 2014})
runner.QueryMoviesByYear(ctx, clusterId, databaseName, tableName, user.Username,
year)

// Modify the cluster's maintenance window
runner.redshiftActor.ModifyCluster(ctx, clusterId, maintenanceWindow)

// Delete the Redshift cluster if confirmed
runner.cleanUpResources(ctx, clusterId, databaseName, tableName, user.Username,
runner.questioner)

log.Println("Thanks for watching!")
}

// cleanUpResources asks the user if they would like to delete each resource
// created during the scenario, from most
// impactful to least impactful. If any choice to delete is made, further
// deletion attempts are skipped.

```

```

func (runner *RedshiftBasicsScenario) cleanUpResources(ctx context.Context,
clusterId string, databaseName string, tableName string, userName string,
questioner demotools.IQuestioner) {
    deleted := false
    var err error = nil
    if questioner.AskBool("Do you want to delete the entire cluster? This will clean
up all resources. (y/n)", "y") {
        deleted, err = runner.redshiftActor.DeleteCluster(ctx, clusterId)
        if err != nil {
            log.Printf("Error deleting cluster: %v", err)
        }
    }
    if !deleted && questioner.AskBool("Do you want to delete the dev table? This
will clean up all inserted records but keep your cluster intact. (y/n)", "y") {
        deleted, err = runner.redshiftDataActor.DeleteTable(ctx, clusterId,
databaseName, tableName, userName)
        if err != nil {
            log.Printf("Error deleting movies table: %v", err)
        }
    }
    if !deleted && questioner.AskBool("Do you want to delete all rows in the Movies
table? This will clean up all inserted records but keep your cluster and table
intact. (y/n)", "y") {
        deleted, err = runner.redshiftDataActor.DeleteDataRows(ctx, clusterId,
databaseName, tableName, userName, runner.pauser)
        if err != nil {
            log.Printf("Error deleting data rows: %v", err)
        }
    }
    if !deleted {
        log.Print("Please manually delete any unwanted resources.")
    }
}

// loadMoviesFromJSON takes the <fileName> file and populates a slice of Movie
objects.
func (runner *RedshiftBasicsScenario) loadMoviesFromJSON(fileName string,
filesystem demotools.IFileSystem) ([]Movie, error) {
    file, err := filesystem.OpenFile("../resources/sample_files/" + fileName)
    if err != nil {
        return nil, err
    }
    defer filesystem.CloseFile(file)

```



```

var movies []Movie
err = json.NewDecoder(file).Decode(&movies)
if err != nil {
    return nil, err
}

return movies, nil
}

// PopulateMoviesTable reads data from the <fileName> file and inserts records
into the "Movies" table.
func (runner *RedshiftBasicsScenario) PopulateMoviesTable(ctx context.Context,
clusterId string, databaseName string, tableName string, userName string,
fileName string) {
log.Println("Populate the " + tableName + " table using the " + fileName + "
file.")
numRecords := runner.questioner.AskInt(
    fmt.Sprintf("Enter a value between %v and %v:", 10, 100),
    demotools.InIntRange{Lower: 10, Upper: 100})

movies, err := runner.loadMoviesFromJSON(fileName, runner.filesystem)
if err != nil {
    log.Printf("Failed to load movies from JSON: %v\n", err)
    panic(err)
}

var sqlStatements []string

for i, movie := range movies {
    if i >= numRecords {
        break
    }

    sqlStatement := fmt.Sprintf(`INSERT INTO %s (title, year) VALUES ('%s', %d);`,
        tableName,
        strings.Replace(movie.Title, "'", "''", -1), // Double any single quotes to
        escape them
        movie.Year)

    sqlStatements = append(sqlStatements, sqlStatement)
}

```

```

input := &redshiftdata.BatchExecuteStatementInput{
    ClusterIdentifier: aws.String(clusterId),
    Database:          aws.String(databaseName),
    DbUser:            aws.String(userName),
    Sqls:              sqlStatements,
}

result, err := runner.redshiftDataActor.ExecuteBatchStatement(ctx, *input)
if err != nil {
    log.Printf("Failed to execute batch statement: %v\n", err)
    panic(err)
}

describeInput := redshiftdata.DescribeStatementInput{
    Id: result.Id,
}

query := actions.RedshiftQuery{
    Context: ctx,
    Result:  result,
    Input:   describeInput,
}

err = runner.redshiftDataActor.WaitForQueryStatus(query, runner.pauser, true)
if err != nil {
    log.Printf("Failed to execute batch insert query: %v\n", err)
    return
}
log.Printf("Successfully executed batch statement\n")

log.Printf("%d records were added to the Movies table.\n", numRecords)
}

// QueryMoviesByYear retrieves only movies from the "Movies" table which match
// the given year.
func (runner *RedshiftBasicsScenario) QueryMoviesByYear(ctx context.Context,
    clusterId string, databaseName string, tableName string, userName string, year
    int) {

    sqlStatement := fmt.Sprintf(`SELECT title FROM %s WHERE year = %d;`, tableName,
    year)

```

```

input := &redshiftdata.ExecuteStatementInput{
    ClusterIdentifier: aws.String(clusterId),
    Database:          aws.String(databaseName),
    DbUser:            aws.String(userName),
    Sql:               aws.String(sqlStatement),
}

result, err := runner.redshiftDataActor.ExecuteStatement(ctx, *input)
if err != nil {
    log.Printf("Failed to query movies: %v\n", err)
    panic(err)
}

log.Println("The identifier of the statement is ", *result.Id)

describeInput := redshiftdata.DescribeStatementInput{
    Id: result.Id,
}

query := actions.RedshiftQuery{
    Context: ctx,
    Input:   describeInput,
    Result:  result,
}

err = runner.redshiftDataActor.WaitForQueryStatus(query, runner.pauser, true)
if err != nil {
    log.Printf("Failed to execute query: %v\n", err)
    panic(err)
}
log.Printf("Successfully executed query\n")

getResultOutput, err := runner.redshiftDataActor.GetStatementResult(ctx,
*result.Id)
if err != nil {
    log.Printf("Failed to query movies: %v\n", err)
    panic(err)
}
for _, row := range getResultOutput.Records {
    for _, col := range row {
        title, ok := col.(*redshiftdata_types.FieldMemberStringValue)
        if !ok {
            log.Println("Failed to parse the field")
        } else {
            log.Printf("The Movie title field is %s\n", title.Value)
        }
    }
}

```

```
}  
}  
}  
}
```

- API 세부 정보는 AWS SDK for Go API 참조의 다음 주제를 참조하세요.
 - [CreateCluster](#) :
 - [DescribeClusters](#)
 - [DescribeStatement](#)
 - [ExecuteStatement](#)
 - [GetStatementResult](#)
 - [ListDatabasesPaginator](#)
 - [ModifyCluster](#)

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

Amazon Redshift 기능을 시연하는 대화형 시나리오를 실행합니다.

```
import com.example.redshift.User;  
import com.google.gson.Gson;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import software.amazon.awssdk.regions.Region;  
import  
    software.amazon.awssdk.services.redshift.model.ClusterAlreadyExistsException;  
import software.amazon.awssdk.services.redshift.model.CreateClusterResponse;  
import software.amazon.awssdk.services.redshift.model.DeleteClusterResponse;  
import software.amazon.awssdk.services.redshift.model.ModifyClusterResponse;
```

```
import software.amazon.awssdk.services.redshift.model.RedshiftException;
import
    software.amazon.awssdk.services.redshiftdata.model.ExecuteStatementResponse;
import software.amazon.awssdk.services.redshiftdata.model.RedshiftDataException;
import java.util.Scanner;
import java.util.concurrent.CompletableFuture;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
import
    software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import
    software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 *
 * This example requires an AWS Secrets Manager secret that contains the
 * database credentials. If you do not create a
 * secret that specifies user name and password, this example will not work. For
 * details, see:
 *
 * https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating\_how-services-use-secrets\_RS.html
 *
 * This Java example performs these tasks:
 *
 * 1. Prompts the user for a unique cluster ID or use the default value.
 * 2. Creates a Redshift cluster with the specified or default cluster Id value.
 * 3. Waits until the Redshift cluster is available for use.
 * 4. Lists all databases using a pagination API call.
 * 5. Creates a table named "Movies" with fields ID, title, and year.
 * 6. Inserts a specified number of records into the "Movies" table by reading
 * the Movies JSON file.
 * 7. Prompts the user for a movie release year.
 * 8. Runs a SQL query to retrieve movies released in the specified year.
 * 9. Modifies the Redshift cluster.
 * 10. Prompts the user for confirmation to delete the Redshift cluster.
 * 11. If confirmed, deletes the specified Redshift cluster.
```

```

*/

public class RedshiftScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
"-");
    private static final Logger logger =
LoggerFactory.getLogger(RedshiftScenario.class);

    static RedshiftActions redshiftActions = new RedshiftActions();
    public static void main(String[] args) throws Exception {
        final String usage = ""

        Usage:
            <jsonFilePath> <secretName>\s

        Where:
            jsonFilePath - The path to the Movies JSON file (you can locate
that file in ../../../../resources/sample_files/movies.json)
            secretName - The name of the secret that belongs to Secret
Manager that stores the user name and password used in this scenario.
            """;

        if (args.length != 2) {
            logger.info(usage);
            return;
        }

        String jsonFilePath = args[0];
        String secretName = args[1];
        Scanner scanner = new Scanner(System.in);
        logger.info(DASHES);
        logger.info("Welcome to the Amazon Redshift SDK Basics scenario.");
        logger.info("")

        This Java program demonstrates how to interact with Amazon Redshift
by using the AWS SDK for Java (v2).\s

        Amazon Redshift is a fully managed, petabyte-scale data warehouse
service hosted in the cloud.

        The program's primary functionalities include cluster creation,
verification of cluster readiness,\s
        list databases, table creation, data population within the table, and
execution of SQL statements.

        Furthermore, it demonstrates the process of querying data from the
Movie table.\s

```

```

        Upon completion of the program, all AWS resources are cleaned up.
        """);

    logger.info("Lets get started...");
    logger.info("")
        First, we will retrieve the user name and password from Secrets
Manager.

        Using Amazon Secrets Manager to store Redshift credentials provides
several security benefits.

        It allows you to securely store and manage sensitive information,
such as passwords, API keys, and
        database credentials, without embedding them directly in your
application code.

        More information can be found here:

        https://docs.aws.amazon.com/secretsmanager/latest/userguide/
integrating_how-services-use-secrets_RS.html
        """);
    Gson gson = new Gson();
    User user = gson.fromJson(String.valueOf(getSecretValues(secretName)),
User.class);
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    try {
        runScenario(user, scanner, jsonFilePath);
    } catch (RuntimeException e) {
        e.printStackTrace();
    } catch (Throwable e) {
        throw new RuntimeException(e);
    }
}

    private static void runScenario(User user, Scanner scanner, String
jsonFilePath) throws Throwable {
        String databaseName = "dev";
        System.out.println(DASHES);
        logger.info("Create a Redshift Cluster");
        logger.info("A Redshift cluster refers to the collection of computing
resources and storage that work together to process and analyze large volumes of
data.");

```

```
        logger.info("Enter a cluster id value or accept the default by hitting  
Enter (default is redshift-cluster-movies): ");  
        String userClusterId = scanner.nextLine();  
        String clusterId = userClusterId.isEmpty() ? "redshift-cluster-movies" :  
userClusterId;  
        try {  
            CompletableFuture<CreateClusterResponse> future =  
redshiftActions.createClusterAsync(clusterId, user.getUserName(),  
user.getUserPassword());  
            CreateClusterResponse response = future.join();  
            logger.info("Cluster successfully created. Cluster Identifier {} ",  
response.cluster().clusterIdentifier());  
  
        } catch (RuntimeException rt) {  
            Throwable cause = rt.getCause();  
            if (cause instanceof ClusterAlreadyExistsException) {  
                logger.info("The Cluster {} already exists. Moving on...",  
clusterId);  
            } else {  
                logger.info("An unexpected error occurred: " + rt.getMessage());  
            }  
        }  
        logger.info(DASHES);  
  
        logger.info(DASHES);  
        logger.info("Wait until {} is available.", clusterId);  
        waitForInputToContinue(scanner);  
        try {  
            CompletableFuture<Void> future =  
redshiftActions.waitForClusterReadyAsync(clusterId);  
            future.join();  
            logger.info("Cluster is ready!");  
  
        } catch (RuntimeException rt) {  
            Throwable cause = rt.getCause();  
            if (cause instanceof RedshiftException redshiftEx) {  
                logger.info("Redshift error occurred: Error message: {}, Error  
code {}", redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());  
            } else {  
                logger.info("An unexpected error occurred: " + rt.getMessage());  
            }  
            throw cause;  
        }  
        logger.info(DASHES);
```



```

        logger.info(DASHES);
        String databaseInfo = ""
            When you created $clusteridD, the dev database is created by default
            and used in this scenario.\s

            To create a custom database, you need to have a CREATEDB privilege.\s
            For more information, see the documentation here: https://docs.aws.amazon.com/redshift/latest/dg/r\_CREATE\_DATABASE.html.
            """.replace("$clusteridD", clusterId);

        logger.info(databaseInfo);
        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("List databases in {} ",clusterId);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Void> future =
redshiftActions.listAllDatabasesAsync(clusterId, user.getUserName(), "dev");
            future.join();
            logger.info("Databases listed successfully.");

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof RedshiftDataException redshiftEx) {
                logger.error("Redshift Data error occurred: {} Error code: {}",
redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
            } else {
                logger.error("An unexpected error occurred: {}",
rt.getMessage());
            }
            throw cause;
        }
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("Now you will create a table named Movies.");
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<ExecuteStatementResponse> future =
redshiftActions.createTableAsync(clusterId, databaseName, user.getUserName());
            future.join();

```

```

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof RedshiftDataException redshiftEx) {
            logger.info("Redshift Data error occurred: {} Error code: {}",
redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: {}", rt.getMessage());
        }
        throw cause;
    }
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("Populate the Movies table using the Movies.json file.");
    logger.info("Specify the number of records you would like to add to the
Movies Table.");
    logger.info("Please enter a value between 50 and 200.");
    int numRecords;
    do {
        logger.info("Enter a value: ");
        while (!scanner.hasNextInt()) {
            logger.info("Invalid input. Please enter a value between 50 and
200.");
            logger.info("Enter a year: ");
            scanner.next();
        }
        numRecords = scanner.nextInt();
    } while (numRecords < 50 || numRecords > 200);
    try {
        redshiftActions.popTableAsync(clusterId, databaseName,
user.getUserName(), jsonFilePath, numRecords).join(); // Wait for the operation
to complete
    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof RedshiftDataException redshiftEx) {
            logger.info("Redshift Data error occurred: {} Error code: {}",
redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: {}", rt.getMessage());
        }
        throw cause;
    }
    waitForInputToContinue(scanner);

```

```
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("Query the Movies table by year. Enter a value between
2012-2014.");
        int movieYear;
        do {
            logger.info("Enter a year: ");
            while (!scanner.hasNextInt()) {
                logger.info("Invalid input. Please enter a valid year between
2012 and 2014.");
                logger.info("Enter a year: ");
                scanner.next();
            }
            movieYear = scanner.nextInt();
            scanner.nextLine();
        } while (movieYear < 2012 || movieYear > 2014);

        String id;
        try {
            CompletableFuture<String> future =
redshiftActions.queryMoviesByYearAsync(databaseName, user.getUserName(),
movieYear, clusterId);
            id = future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof RedshiftDataException redshiftEx) {
                logger.info("Redshift Data error occurred: {} Error code: {}",
redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: {}", rt.getMessage());
            }
            throw cause;
        }

        logger.info("The identifier of the statement is " + id);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Void> future =
redshiftActions.checkStatementAsync(id);
            future.join();

        } catch (RuntimeException rt) {
```

```

        Throwable cause = rt.getCause();
        if (cause instanceof RedshiftDataException redshiftEx) {
            logger.info("Redshift Data error occurred: {} Error code: {}",
redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: {}", rt.getMessage());
        }
        throw cause;
    }
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<Void> future = redshiftActions.getResultsAsync(id);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof RedshiftDataException redshiftEx) {
            logger.info("Redshift Data error occurred: {} Error code: {}",
redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: {}", rt.getMessage());
        }
        throw cause;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("Now you will modify the Redshift cluster.");
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<ModifyClusterResponse> future =
redshiftActions.modifyClusterAsync(clusterId);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof RedshiftDataException redshiftEx) {
            logger.info("Redshift Data error occurred: {} Error code: {}",
redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: {}", rt.getMessage());
        }
        throw cause;
    }

```

```

    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("Would you like to delete the Amazon Redshift cluster? (y/
n)");
    String delAns = scanner.nextLine().trim();
    if (delAns.equalsIgnoreCase("y")) {
        logger.info("You selected to delete {} ", clusterId);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<DeleteClusterResponse> future =
redshiftActions.deleteRedshiftClusterAsync(clusterId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof RedshiftDataException redshiftEx) {
                logger.info("Redshift Data error occurred: {} Error code:
{}", redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: {}",
rt.getMessage());
            }
            throw cause;
        }
    } else {
        logger.info("The {} was not deleted", clusterId);
    }
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("This concludes the Amazon Redshift SDK Basics scenario.");
    logger.info(DASHES);
}

private static SecretsManagerClient getSecretClient() {
    Region region = Region.US_EAST_1;
    return SecretsManagerClient.builder()
        .region(region)
        .build();
}

```

```

private static void waitForInputToContinue(Scanner scanner) {
    while (true) {
        System.out.println("");
        System.out.println("Enter 'c' followed by <ENTER> to continue:");
        String input = scanner.nextLine();

        if (input.trim().equalsIgnoreCase("c")) {
            System.out.println("Continuing with the program...");
            System.out.println("");
            break;
        } else {
            // Handle invalid input.
            System.out.println("Invalid input. Please try again.");
        }
    }
}

// Get the Amazon Redshift credentials from AWS Secrets Manager.
private static String getSecretValues(String secretName) {
    SecretsManagerClient secretClient = getSecretClient();
    GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
        .secretId(secretName)
        .build();

    GetSecretValueResponse valueResponse =
secretClient.getSecretValue(valueRequest);
    return valueResponse.secretString();
}
}

```

Amazon Redshift SDK 메서드의 래퍼 클래스입니다.

```

public class RedshiftActions {

    private static final Logger logger =
LoggerFactory.getLogger(RedshiftActions.class);
    private static RedshiftDataAsyncClient redshiftDataAsyncClient;

    private static RedshiftAsyncClient redshiftAsyncClient;

    private static RedshiftAsyncClient getAsyncClient() {
        if (redshiftAsyncClient == null) {

```

```
        SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
            .maxConcurrency(100)
            .connectionTimeout(Duration.ofSeconds(60))
            .readTimeout(Duration.ofSeconds(60))
            .writeTimeout(Duration.ofSeconds(60))
            .build();

        ClientOverrideConfiguration overrideConfig =
ClientOverrideConfiguration.builder()
            .apiCallTimeout(Duration.ofMinutes(2))
            .apiCallAttemptTimeout(Duration.ofSeconds(90))
            .retryStrategy(RetryMode.STANDARD)
            .build();

        redshiftAsyncClient = RedshiftAsyncClient.builder()
            .httpClient(httpClient)
            .overrideConfiguration(overrideConfig)
            .build();
    }
    return redshiftAsyncClient;
}

private static RedshiftDataAsyncClient getAsyncDataClient() {
    if (redshiftDataAsyncClient == null) {
        SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
            .maxConcurrency(100)
            .connectionTimeout(Duration.ofSeconds(60))
            .readTimeout(Duration.ofSeconds(60))
            .writeTimeout(Duration.ofSeconds(60))
            .build();

        ClientOverrideConfiguration overrideConfig =
ClientOverrideConfiguration.builder()
            .apiCallTimeout(Duration.ofMinutes(2))
            .apiCallAttemptTimeout(Duration.ofSeconds(90))
            .retryStrategy(RetryMode.STANDARD)
            .build();

        redshiftDataAsyncClient = RedshiftDataAsyncClient.builder()
            .httpClient(httpClient)
            .overrideConfiguration(overrideConfig)
            .build();
    }
    return redshiftDataAsyncClient;
}
```

```

    }

    /**
     * Creates a new Amazon Redshift cluster asynchronously.
     * @param clusterId the unique identifier for the cluster
     * @param username the username for the administrative user
     * @param userPassword the password for the administrative user
     * @return a CompletableFuture that represents the asynchronous operation of
     creating the cluster
     * @throws RuntimeException if the cluster creation fails
     */
    public CompletableFuture<CreateClusterResponse> createClusterAsync(String
clusterId, String username, String userPassword) {
        CreateClusterRequest clusterRequest = CreateClusterRequest.builder()
            .clusterIdentifier(clusterId)
            .masterUsername(username)
            .masterUserPassword(userPassword)
            .nodeType("ra3.4xlarge")
            .publiclyAccessible(true)
            .numberOfNodes(2)
            .build();

        return getAsyncClient().createCluster(clusterRequest)
            .whenComplete((response, exception) -> {
                if (response != null) {
                    logger.info("Created cluster ");
                } else {
                    throw new RuntimeException("Failed to create cluster: " +
exception.getMessage(), exception);
                }
            });
    }

    /**
     * Waits asynchronously for the specified cluster to become available.
     * @param clusterId the identifier of the cluster to wait for
     * @return a {@link CompletableFuture} that completes when the cluster is
     ready
     */
    public CompletableFuture<Void> waitForClusterReadyAsync(String clusterId) {
        DescribeClustersRequest clustersRequest =
DescribeClustersRequest.builder()
            .clusterIdentifier(clusterId)
            .build();
    }

```



```

        logger.info("Waiting for cluster to become available. This may take a few
minutes.");
        long startTime = System.currentTimeMillis();

        // Recursive method to poll the cluster status.
        return checkClusterStatusAsync(clustersRequest, startTime);
    }

    private CompletableFuture<Void>
checkClusterStatusAsync(DescribeClustersRequest clustersRequest, long startTime)
{
    return getAsyncClient().describeClusters(clustersRequest)
        .thenCompose(clusterResponse -> {
            List<Cluster> clusterList = clusterResponse.clusters();
            boolean clusterReady = false;
            for (Cluster cluster : clusterList) {
                if ("available".equals(cluster.clusterStatus())) {
                    clusterReady = true;
                    break;
                }
            }

            if (clusterReady) {
                logger.info(String.format("Cluster is available!"));
                return CompletableFuture.completedFuture(null);
            } else {
                long elapsedTimeMillis = System.currentTimeMillis() -
startTime;

                long elapsedSeconds = elapsedTimeMillis / 1000;
                long minutes = elapsedSeconds / 60;
                long seconds = elapsedSeconds % 60;
                System.out.printf("\rElapsed Time: %02d:%02d - Waiting for
cluster...", minutes, seconds);
                System.out.flush();

                // Wait 1 second before the next status check
                return CompletableFuture.runAsync(() -> {
                    try {
                        TimeUnit.SECONDS.sleep(1);
                    } catch (InterruptedException e) {
                        throw new RuntimeException("Error during sleep: " +
e.getMessage(), e);
                    }
                });
            }
        });
}

```

```

        }).thenCompose(ignored ->
checkClusterStatusAsync(clustersRequest, startTime));
    }
    }).exceptionally(exception -> {
        throw new RuntimeException("Failed to get cluster status: " +
exception.getMessage(), exception);
    });
}

/**
 * Lists all databases asynchronously for the specified cluster, database
user, and database.
 * @param clusterId the identifier of the cluster to list databases for
 * @param dbUser the database user to use for the list databases request
 * @param database the database to list databases for
 * @return a {@link CompletableFuture} that completes when the database
listing is complete, or throws a {@link RuntimeException} if there was an error
 */
public CompletableFuture<Void> listAllDatabasesAsync(String clusterId, String
dbUser, String database) {
    ListDatabasesRequest databasesRequest = ListDatabasesRequest.builder()
        .clusterIdentifier(clusterId)
        .dbUser(dbUser)
        .database(database)
        .build();

    // Asynchronous paginator for listing databases.
    ListDatabasesPublisher databasesPaginator =
getAsyncDataClient().listDatabasesPaginator(databasesRequest);
    CompletableFuture<Void> future = databasesPaginator.subscribe(response ->
{
        response.databases().forEach(db -> {
            logger.info("The database name is {} ", db);
        });
    });

    // Return the future for asynchronous handling.
    return future.exceptionally(exception -> {
        throw new RuntimeException("Failed to list databases: " +
exception.getMessage(), exception);
    });
}

/**

```

```

    * Creates an asynchronous task to execute a SQL statement for creating a new
    table.
    *
    * @param clusterId    the identifier of the Amazon Redshift cluster
    * @param databaseName the name of the database to create the table in
    * @param userName     the username to use for the database connection
    * @return a {@link CompletableFuture} that completes with the result of the
    SQL statement execution
    * @throws RuntimeException if there is an error creating the table
    */
    public CompletableFuture<ExecuteStatementResponse> createTableAsync(String
clusterId, String databaseName, String userName) {
        ExecuteStatementRequest createTableRequest =
ExecuteStatementRequest.builder()
            .clusterIdentifier(clusterId)
            .dbUser(userName)
            .database(databaseName)
            .sql("CREATE TABLE Movies (" +
                "id INT PRIMARY KEY, " +
                "title VARCHAR(100), " +
                "year INT)")
            .build();

        return getAsyncDataClient().executeStatement(createTableRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    throw new RuntimeException("Error creating table: " +
exception.getMessage(), exception);
                } else {
                    logger.info("Table created: Movies");
                }
            });
    }

    /**
    * Asynchronously pops a table from a JSON file.
    *
    * @param clusterId    the ID of the cluster
    * @param databaseName the name of the database
    * @param userName     the username
    * @param fileName     the name of the JSON file
    * @param number       the number of records to process
    * @return a CompletableFuture that completes with the number of records
    added to the Movies table

```

```

    */
    public CompletableFuture<Integer> popTableAsync(String clusterId, String
databaseName, String userName, String fileName, int number) {
        return CompletableFuture.supplyAsync(() -> {
            try {
                JsonParser parser = new JsonFactory().createParser(new
File(fileName));

                JsonNode rootNode = new ObjectMapper().readTree(parser);
                Iterator<JsonNode> iter = rootNode.iterator();
                return iter;
            } catch (IOException e) {
                throw new RuntimeException("Failed to read or parse JSON
file: " + e.getMessage(), e);
            }
        }).thenCompose(iter -> processNodesAsync(clusterId, databaseName,
userName, iter, number))
        .whenComplete((result, exception) -> {
            if (exception != null) {
                logger.info("Error {} ", exception.getMessage());
            } else {
                logger.info("{} records were added to the Movies table." ,
result);
            }
        });
    }

    private CompletableFuture<Integer> processNodesAsync(String clusterId, String
databaseName, String userName, Iterator<JsonNode> iter, int number) {
        return CompletableFuture.supplyAsync(() -> {
            int t = 0;
            try {
                while (iter.hasNext()) {
                    if (t == number)
                        break;
                    JsonNode currentNode = iter.next();
                    int year = currentNode.get("year").asInt();
                    String title = currentNode.get("title").asText();

                    // Use SqlParameter to avoid SQL injection.
                    List<SqlParameter> parameterList = new ArrayList<>();
                    String sqlStatement = "INSERT INTO Movies
VALUES( :id , :title, :year);";
                    SqlParameter idParam = SqlParameter.builder()
                        .name("id")

```

```

        .value(String.valueOf(t))
        .build();

        SqlParameter titleParam = SqlParameter.builder()
            .name("title")
            .value(title)
            .build();

        SqlParameter yearParam = SqlParameter.builder()
            .name("year")
            .value(String.valueOf(year))
            .build();
        parameterList.add(idParam);
        parameterList.add(titleParam);
        parameterList.add(yearParam);

        ExecuteStatementRequest insertStatementRequest =
        ExecuteStatementRequest.builder()
            .clusterIdentifier(clusterId)
            .sql(sqlStatement)
            .database(databaseName)
            .dbUser(userName)
            .parameters(parameterList)
            .build();

        getAsyncDataClient().executeStatement(insertStatementRequest);
        logger.info("Inserted: " + title + " (" + year + ")");
        t++;
    }
    } catch (RedshiftDataException e) {
        throw new RuntimeException("Error inserting data: " +
        e.getMessage(), e);
    }
    return t;
});
}

/**
 * Checks the status of an SQL statement asynchronously and handles the
 * completion of the statement.
 *
 * @param sqlId the ID of the SQL statement to check

```

```

    * @return a {@link CompletableFuture} that completes when the SQL
    statement's status is either "FINISHED" or "FAILED"
    */
    public CompletableFuture<Void> checkStatementAsync(String sqlId) {
        DescribeStatementRequest statementRequest =
        DescribeStatementRequest.builder()
            .id(sqlId)
            .build();

        return getAsyncDataClient().describeStatement(statementRequest)
            .thenCompose(response -> {
                String status = response.statusAsString();
                logger.info("... Status: {} ", status);

                if ("FAILED".equals(status)) {
                    throw new RuntimeException("The Query Failed. Ending
program");
                } else if ("FINISHED".equals(status)) {
                    return CompletableFuture.completedFuture(null);
                } else {
                    // Sleep for 1 second and recheck status
                    return CompletableFuture.runAsync(() -> {
                        try {
                            TimeUnit.SECONDS.sleep(1);
                        } catch (InterruptedException e) {
                            throw new RuntimeException("Error during sleep: " +
e.getMessage(), e);
                        }
                    }).thenCompose(ignore -> checkStatementAsync(sqlId)); //
Recursively call until status is FINISHED or FAILED
                }
            }).whenComplete((result, exception) -> {
                if (exception != null) {
                    // Handle exceptions
                    logger.info("Error: {} ", exception.getMessage());
                } else {
                    logger.info("The statement is finished!");
                }
            });
    }

    /**
     * Asynchronously retrieves the results of a statement execution.
     */

```

```

    * @param statementId the ID of the statement for which to retrieve the
    results
    * @return a {@link CompletableFuture} that completes when the statement
    result has been processed
    */
    public CompletableFuture<Void> getResultsAsync(String statementId) {
        GetStatementResultRequest resultRequest =
        GetStatementResultRequest.builder()
            .id(statementId)
            .build();

        return getAsyncDataClient().getStatementResult(resultRequest)
            .handle((response, exception) -> {
                if (exception != null) {
                    logger.info("Error getting statement result {}",
exception.getMessage());
                    throw new RuntimeException("Error getting statement result: "
+ exception.getMessage(), exception);
                }

                // Extract and print the field values using streams if the
                response is valid.
                response.records().stream()
                    .flatMap(List::stream)
                    .map(Field::stringValue)
                    .filter(value -> value != null)
                    .forEach(value -> System.out.println("The Movie title field
is " + value));

                return response;
            }).thenAccept(response -> {
                // Optionally add more logic here if needed after handling the
                response
            });
    }

    /**
     * Asynchronously queries movies by a given year from a Redshift database.
     *
     * @param database    the name of the database to query
     * @param dbUser      the user to connect to the database with
     * @param year        the year to filter the movies by
     * @param clusterId   the identifier of the Redshift cluster to connect to

```

```

    * @return a {@link CompletableFuture} containing the response ID of the
    executed SQL statement
    */
    public CompletableFuture<String> queryMoviesByYearAsync(String database,
                                                            String dbUser,
                                                            int year,
                                                            String
clusterId) {

        String sqlStatement = "SELECT * FROM Movies WHERE year = :year";
        SqlParameter yearParam = SqlParameter.builder()
            .name("year")
            .value(String.valueOf(year))
            .build();

        ExecuteStatementRequest statementRequest =
ExecuteStatementRequest.builder()
            .clusterIdentifier(clusterId)
            .database(database)
            .dbUser(dbUser)
            .parameters(yearParam)
            .sql(sqlStatement)
            .build();

        return CompletableFuture.supplyAsync(() -> {
            try {
                ExecuteStatementResponse response =
getAsyncDataClient().executeStatement(statementRequest).join(); // Use join() to
wait for the result
                return response.id();
            } catch (RedshiftDataException e) {
                throw new RuntimeException("Error executing statement: " +
e.getMessage(), e);
            }
        }).exceptionally(exception -> {
            logger.info("Error: {}", exception.getMessage());
            return "";
        });
    }

    /**
     * Modifies an Amazon Redshift cluster asynchronously.
     *
     * @param clusterId the identifier of the cluster to be modified

```



```

    * @return a {@link CompletableFuture} that completes when the cluster
    modification is complete
    */
    public CompletableFuture<ModifyClusterResponse> modifyClusterAsync(String
clusterId) {
        ModifyClusterRequest modifyClusterRequest =
ModifyClusterRequest.builder()
            .clusterIdentifier(clusterId)
            .preferredMaintenanceWindow("wed:07:30-wed:08:00")
            .build();

        return getAsyncClient().modifyCluster(modifyClusterRequest)
            .whenComplete((clusterResponse, exception) -> {
                if (exception != null) {
                    if (exception.getCause() instanceof RedshiftException) {
                        logger.info("Error: {} ", exception.getMessage());
                    } else {
                        logger.info("Unexpected error: {} ",
exception.getMessage());
                    }
                } else {
                    logger.info("The modified cluster was successfully modified
and has "
                                + clusterResponse.cluster().preferredMaintenanceWindow()
+ " as the maintenance window");
                }
            });
    }

    /**
     * Deletes a Redshift cluster asynchronously.
     *
     * @param clusterId the identifier of the Redshift cluster to be deleted
     * @return a {@link CompletableFuture} that represents the asynchronous
     operation of deleting the Redshift cluster
     */
    public CompletableFuture<DeleteClusterResponse>
deleteRedshiftClusterAsync(String clusterId) {
        DeleteClusterRequest deleteClusterRequest =
DeleteClusterRequest.builder()
            .clusterIdentifier(clusterId)
            .skipFinalClusterSnapshot(true)
            .build();
    }

```

```

        return getAsyncClient().deleteCluster(deleteClusterRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    // Handle exceptions
                    if (exception.getCause() instanceof RedshiftException) {
                        logger.info("Error: {}", exception.getMessage());
                    } else {
                        logger.info("Unexpected error: {}",
exception.getMessage());
                    }
                } else {
                    // Handle successful response
                    logger.info("The status is {}",
response.cluster().clusterStatus());
                }
            });
    }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 다음 주제를 참조하세요.
 - [CreateCluster](#) :
 - [DescribeClusters](#)
 - [DescribeStatement](#)
 - [ExecuteStatement](#)
 - [GetStatementResult](#)
 - [ListDatabasesPaginator](#)
 - [ModifyCluster](#)

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
class RedshiftScenario:
    """Runs an interactive scenario that shows how to get started with
    Redshift."""

    def __init__(self, redshift_wrapper, redshift_data_wrapper):
        self.redshift_wrapper = redshift_wrapper
        self.redshift_data_wrapper = redshift_data_wrapper

    def redshift_scenario(self, json_file_path):
        database_name = "dev"

        print(DASHES)
        print("Welcome to the Amazon Redshift SDK Getting Started example.")
        print(
            """
This Python program demonstrates how to interact with Amazon Redshift
using the AWS SDK for Python (Boto3).

Amazon Redshift is a fully managed, petabyte-scale data warehouse
service hosted in the cloud.

The program's primary functionalities include cluster creation,
verification of cluster readiness, listing databases, table creation,
populating data within the table, and executing SQL statements.

It also demonstrates querying data from the Movies table.

Upon completion, all AWS resources are cleaned up.
"""
        )
        if not os.path.isfile(json_file_path):
            logging.error(f"The file {json_file_path} does not exist.")
            return

        print("Let's get started...")
        user_name = q.ask("Please enter your user name (default is awsuser):")
        user_name = user_name if user_name else "awsuser"

        print(DASHES)
        user_password = q.ask(
            "Please enter your user password (default is AwsUser1000):"
        )
        user_password = user_password if user_password else "AwsUser1000"
```

```

        print(DASHES)
        print(
            """A Redshift cluster refers to the collection of computing resources
and storage that work
            together to process and analyze large volumes of data."""
        )
        cluster_id = q.ask(
            "Enter a cluster identifier value (default is redshift-cluster-
movies): "
        )
        cluster_id = cluster_id if cluster_id else "redshift-cluster-movies"

        self.redshift_wrapper.create_cluster(
            cluster_id, "ra3.4xlarge", user_name, user_password, True, 2
        )

        print(DASHES)
        print(f"Wait until {cluster_id} is available. This may take a few
minutes...")
        q.ask("Press Enter to continue...")

        self.wait_cluster_available(cluster_id)

        print(DASHES)

        print(
            f"""
            When you created {cluster_id}, the dev database is created by default and
            used in this scenario.

            To create a custom database, you need to have a CREATEDB privilege.
            For more information, see the documentation here:
            https://docs.aws.amazon.com/redshift/latest/dg/r_CREATE_DATABASE.html.
            """
        )
        q.ask("Press Enter to continue...")
        print(DASHES)

        print(DASHES)
        print(f"List databases in {cluster_id}")
        q.ask("Press Enter to continue...")
        databases = self.redshift_data_wrapper.list_databases(
            cluster_id, database_name, user_name

```

```

    )
    print(f"The cluster contains {len(databases)} database(s).")
    for database in databases:
        print(f"    Database: {database}")
    print(DASHES)

    print(DASHES)
    print("Now you will create a table named Movies.")
    q.ask("Press Enter to continue...")

    self.create_table(cluster_id, database_name, user_name)

    print(DASHES)

    print("Populate the Movies table using the Movies.json file.")
    print(
        "Specify the number of records you would like to add to the Movies
Table."
    )
    print("Please enter a value between 50 and 200.")

    while True:
        try:
            num_records = int(q.ask("Enter a value: ", q.is_int))
            if 50 <= num_records <= 200:
                break
            else:
                print("Invalid input. Please enter a value between 50 and
200.")
        except ValueError:
            print("Invalid input. Please enter a value between 50 and 200.")

    self.populate_table(
        cluster_id, database_name, user_name, json_file_path, num_records
    )

    print(DASHES)
    print("Query the Movies table by year. Enter a value between 2012-2014.")

    while True:
        movie_year = int(q.ask("Enter a year: ", q.is_int))
        if 2012 <= movie_year <= 2014:
            break
        else:

```

```
print("Invalid input. Please enter a valid year between 2012 and
2014.")

# Function to query database
sql_id = self.query_movies_by_year(
    database_name, user_name, movie_year, cluster_id
)

print(f"The identifier of the statement is {sql_id}")

print("Checking statement status...")
self.wait_statement_finished(sql_id)
result = self.redshift_data_wrapper.get_statement_result(sql_id)

self.display_movies(result)

print(DASHES)

print(DASHES)
print("Now you will modify the Redshift cluster.")
q.ask("Press Enter to continue...")

preferred_maintenance_window = "wed:07:30-wed:08:00"
self.redshift_wrapper.modify_cluster(cluster_id,
preferred_maintenance_window)

print(DASHES)

print(DASHES)
delete = q.ask("Do you want to delete the cluster? (y/n) ", q.is_yesno)

if delete:
    print(f"You selected to delete {cluster_id}")
    q.ask("Press Enter to continue...")
    self.redshift_wrapper.delete_cluster(cluster_id)
else:
    print(f"Cluster {cluster_id} cluster_id was not deleted")

print(DASHES)
print("This concludes the Amazon Redshift SDK Getting Started scenario.")
print(DASHES)

def create_table(self, cluster_id, database, username):
    self.redshift_data_wrapper.execute_statement(
```

```

        cluster_identifier=cluster_id,
        database_name=database,
        user_name=username,
        sql="CREATE TABLE Movies (statement_id INT PRIMARY KEY, title
VARCHAR(100), year INT)",
    )

    print("Table created: Movies")

def populate_table(self, cluster_id, database, username, file_name, number):
    with open(file_name) as f:
        data = json.load(f)

    i = 0
    for record in data:
        if i == number:
            break

        statement_id = i
        title = record["title"]
        year = record["year"]
        i = i + 1
        parameters = [
            {"name": "statement_id", "value": str(statement_id)},
            {"name": "title", "value": title},
            {"name": "year", "value": str(year)},
        ]

        self.redshift_data_wrapper.execute_statement(
            cluster_identifier=cluster_id,
            database_name=database,
            user_name=username,
            sql="INSERT INTO Movies VALUES(:statement_id, :title, :year)",
            parameter_list=parameters,
        )

    print(f"{i} records inserted into Movies table")

def wait_cluster_available(self, cluster_id):
    """
    Waits for a cluster to be available.

    :param cluster_id: The cluster identifier.

```

```

Note: The cluster_available waiter can also be used.
It is not used in this case to allow an elapsed time message.
"""

cluster_ready = False
start_time = time.time()

while not cluster_ready:
    time.sleep(30)
    cluster = self.redshift_wrapper.describe_clusters(cluster_id)
    status = cluster[0]["ClusterStatus"]
    if status == "available":
        cluster_ready = True
    elif status != "creating":
        raise Exception(
            f"Cluster {cluster_id} creation failed with status {status}."
        )

    elapsed_seconds = int(round(time.time() - start_time))
    minutes = int(elapsed_seconds // 60)
    seconds = int(elapsed_seconds % 60)

    print(f"Elapsed Time: {minutes}:{seconds:02d} - status {status}...")

    if minutes > 30:
        raise Exception(
            f"Cluster {cluster_id} is not available after 30 minutes."
        )

def query_movies_by_year(self, database, username, year, cluster_id):
    sql = "SELECT * FROM Movies WHERE year = :year"

    params = [{"name": "year", "value": str(year)}]

    response = self.redshift_data_wrapper.execute_statement(
        cluster_identifier=cluster_id,
        database_name=database,
        user_name=username,
        sql=sql,
        parameter_list=params,
    )

    return response["Id"]

```



```

    @staticmethod
    def display_movies(response):
        metadata = response["ColumnMetadata"]
        records = response["Records"]

        title_column_index = None
        for i in range(len(metadata)):
            if metadata[i]["name"] == "title":
                title_column_index = i
                break

        if title_column_index is None:
            print("No title column found.")
            return

        print(f"Found {len(records)} movie(s).")
        for record in records:
            print(f"    {record[title_column_index]['stringValue']}")

    def wait_statement_finished(self, sql_id):
        while True:
            time.sleep(1)
            response = self.redshift_data_wrapper.describe_statement(sql_id)
            status = response["Status"]
            print(f"Statement status is {status}.")

            if status == "FAILED":
                print(f"The query failed because {response['Error']}. Ending
program")
                raise Exception("The Query Failed. Ending program")
            elif status == "FINISHED":
                break

```

시나리오 구현을 보여주는 메인 함수입니다.

```

def main():
    redshift_client = boto3.client("redshift")
    redshift_data_client = boto3.client("redshift-data")
    redshift_wrapper = RedshiftWrapper(redshift_client)
    redshift_data_wrapper = RedshiftDataWrapper(redshift_data_client)

```

```

redshift_scenario = RedshiftScenario(redshift_wrapper, redshift_data_wrapper)
redshift_scenario.redshift_scenario(
    f"{os.path.dirname(__file__)}/../../resources/sample_files/
movies.json"
)

```

시나리오에 사용된 래퍼 함수입니다.

```

def create_cluster(
    self,
    cluster_identifier,
    node_type,
    master_username,
    master_user_password,
    publicly_accessible,
    number_of_nodes,
):
    """
    Creates a cluster.

    :param cluster_identifier: The name of the cluster.
    :param node_type: The type of node in the cluster.
    :param master_username: The master username.
    :param master_user_password: The master user password.
    :param publicly_accessible: Whether the cluster is publicly accessible.
    :param number_of_nodes: The number of nodes in the cluster.
    :return: The cluster.
    """

    try:
        cluster = self.client.create_cluster(
            ClusterIdentifier=cluster_identifier,
            NodeType=node_type,
            MasterUsername=master_username,
            MasterUserPassword=master_user_password,
            PubliclyAccessible=publicly_accessible,
            NumberOfNodes=number_of_nodes,
        )
        return cluster
    except ClientError as err:

```

```

        logging.error(
            "Couldn't create a cluster. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

def describe_clusters(self, cluster_identifier):
    """
    Describes a cluster.

    :param cluster_identifier: The cluster identifier.
    :return: A list of clusters.
    """
    try:
        kwargs = {}
        if cluster_identifier:
            kwargs["ClusterIdentifier"] = cluster_identifier

        paginator = self.client.get_paginator("describe_clusters")
        clusters = []
        for page in paginator.paginate(**kwargs):
            clusters.extend(page["Clusters"])

        return clusters

    except ClientError as err:
        logging.error(
            "Couldn't describe a cluster. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

def execute_statement(
    self, cluster_identifier, database_name, user_name, sql,
    parameter_list=None
):
    """
    Executes a SQL statement.

    :param cluster_identifier: The cluster identifier.

```

```

:param database_name: The database name.
:param user_name: The user's name.
:param sql: The SQL statement.
:param parameter_list: The optional SQL statement parameters.
:return: The SQL statement result.
"""

try:
    kwargs = {
        "ClusterIdentifier": cluster_identifier,
        "Database": database_name,
        "DbUser": user_name,
        "Sql": sql,
    }
    if parameter_list:
        kwargs["Parameters"] = parameter_list
    response = self.client.execute_statement(**kwargs)
    return response
except ClientError as err:
    logging.error(
        "Couldn't execute statement. Here's why: %s: %s",
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise

def describe_statement(self, statement_id):
    """
    Describes a SQL statement.

    :param statement_id: The SQL statement identifier.
    :return: The SQL statement result.
    """
    try:
        response = self.client.describe_statement(Id=statement_id)
        return response
    except ClientError as err:
        logging.error(
            "Couldn't describe statement. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

```

```
def get_statement_result(self, statement_id):
    """
    Gets the result of a SQL statement.

    :param statement_id: The SQL statement identifier.
    :return: The SQL statement result.
    """
    try:
        result = {
            "Records": [],
        }
        paginator = self.client.get_paginator("get_statement_result")
        for page in paginator.paginate(Id=statement_id):
            if "ColumnMetadata" not in result:
                result["ColumnMetadata"] = page["ColumnMetadata"]
            result["Records"].extend(page["Records"])
        return result
    except ClientError as err:
        logging.error(
            "Couldn't get statement result. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

def modify_cluster(self, cluster_identifier, preferred_maintenance_window):
    """
    Modifies a cluster.

    :param cluster_identifier: The cluster identifier.
    :param preferred_maintenance_window: The preferred maintenance window.
    """
    try:
        self.client.modify_cluster(
            ClusterIdentifier=cluster_identifier,
            PreferredMaintenanceWindow=preferred_maintenance_window,
        )
    except ClientError as err:
        logging.error(
            "Couldn't modify a cluster. Here's why: %s: %s",
            err.response["Error"]["Code"],
```

```

        err.response["Error"]["Message"],
    )
    raise

def list_databases(self, cluster_identifier, database_name, database_user):
    """
    Lists databases in a cluster.

    :param cluster_identifier: The cluster identifier.
    :param database_name: The database name.
    :param database_user: The database user.
    :return: The list of databases.
    """
    try:
        paginator = self.client.get_paginator("list_databases")
        databases = []
        for page in paginator.paginate(
            ClusterIdentifier=cluster_identifier,
            Database=database_name,
            DbUser=database_user,
        ):
            databases.extend(page["Databases"])

        return databases
    except ClientError as err:
        logging.error(
            "Couldn't list databases. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

def delete_cluster(self, cluster_identifier):
    """
    Deletes a cluster.

    :param cluster_identifier: The cluster identifier.
    """
    try:
        self.client.delete_cluster(
            ClusterIdentifier=cluster_identifier,
            SkipFinalClusterSnapshot=True

```

```

    )
except ClientError as err:
    logging.error(
        "Couldn't delete a cluster. Here's why: %s: %s",
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise

```

- API 세부 정보는 AWSSDK for Python (Boto3) API 참조의 다음 주제를 참조하세요.
 - [CreateCluster](#) :
 - [DescribeClusters](#)
 - [DescribeStatement](#)
 - [ExecuteStatement](#)
 - [GetStatementResult](#)
 - [ListDatabasesPaginator](#)
 - [ModifyCluster](#)

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [AWS SDK와 함께 이 서비스 사용](#)을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK를 사용한 Amazon Redshift에 대한 작업

다음 코드 예시에서는 AWS SDK를 통해 개별 Amazon Redshift 작업을 수행하는 방법을 보여줍니다. 각 예제에는 GitHub에 대한 링크가 포함되어 있습니다. 여기에서 코드 설정 및 실행에 대한 지침을 찾을 수 있습니다.

이들 발췌문은 Amazon Redshift API를 직접적으로 호출하며, 컨텍스트에서 실행되어야 하는 더 큰 프로그램에서 발췌한 코드입니다. [AWS SDK를 사용한 Amazon S3에 대한 시나리오](#)에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

다음 예제에는 가장 일반적으로 사용되는 작업만 포함되어 있습니다. 전체 목록은 [Amazon Redshift API 참조](#)를 참조하세요.

예제

- [AWS SDK 또는 CLI와 함께 CreateCluster 사용](#)

- [AWS SDK 또는 CLI와 함께 DeleteCluster 사용](#)
- [AWS SDK 또는 CLI와 함께 DescribeClusters 사용](#)
- [AWS SDK와 함께 DescribeStatement 사용](#)
- [AWS SDK와 함께 ExecuteStatement 사용](#)
- [AWS SDK와 함께 GetStatementResult 사용](#)
- [AWS SDK와 함께 ListDatabases 사용](#)
- [AWS SDK 또는 CLI와 함께 ModifyCluster 사용](#)

AWS SDK 또는 CLI와 함께 **CreateCluster** 사용

다음 코드 예시는 CreateCluster의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [기본 사항 알아보기](#)

.NET

SDK for .NET(v4)

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

/// <summary>
/// Create a new Amazon Redshift cluster.
/// </summary>
/// <param name="clusterIdentifier">The identifier for the cluster.</param>
/// <param name="databaseName">The name of the database.</param>
/// <param name="masterUsername">The master username.</param>
/// <param name="masterUserPassword">The master user password.</param>
/// <param name="nodeType">The node type for the cluster.</param>
/// <returns>The cluster that was created.</returns>
public async Task<Cluster> CreateClusterAsync(string clusterIdentifier,
string databaseName,
```



```

        string masterUsername, string masterUserPassword, string nodeType =
"ra3.large")
    {
        try
        {
            var request = new CreateClusterRequest
            {
                ClusterIdentifier = clusterIdentifier,
                DBName = databaseName,
                MasterUsername = masterUsername,
                MasterUserPassword = masterUserPassword,
                NodeType = nodeType,
                NumberOfNodes = 1,
                ClusterType = "single-node"
            };

            var response = await _redshiftClient.CreateClusterAsync(request);
            Console.WriteLine($"Created cluster {clusterIdentifier}");
            return response.Cluster;
        }
        catch (ClusterAlreadyExistsException ex)
        {
            Console.WriteLine($"Cluster already exists: {ex.Message}");
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't create cluster. Here's why:
{ex.Message}");
            throw;
        }
    }
}

```

- API에 대한 세부 정보는 AWS SDK for .NET API 참조의 [CreateCluster](#)를 참조하세요.

CLI

AWS CLI

최소 파라미터를 사용하여 클러스터 생성 이 예시에서는 최소 파라미터 세트로 클러스터를 만듭니다. 기본 출력 형식은 JSON입니다. 명령:

```
aws redshift create-cluster --node-type dw.hs1.xlarge --number-of-nodes 2 --
master-username adminuser --master-user-password TopSecret1 --cluster-identifier
mycluster
```

결과:

```
{
  "Cluster": {
    "NodeType": "dw.hs1.xlarge",
    "ClusterVersion": "1.0",
    "PubliclyAccessible": "true",
    "MasterUsername": "adminuser",
    "ClusterParameterGroups": [
      {
        "ParameterApplyStatus": "in-sync",
        "ParameterGroupName": "default.redshift-1.0"
      }
    ],
    "ClusterSecurityGroups": [
      {
        "Status": "active",
        "ClusterSecurityGroupName": "default"
      }
    ],
    "AllowVersionUpgrade": true,
    "VpcSecurityGroups": [],
    "PreferredMaintenanceWindow": "sat:03:30-sat:04:00",
    "AutomatedSnapshotRetentionPeriod": 1,
    "ClusterStatus": "creating",
    "ClusterIdentifier": "mycluster",
    "DBName": "dev",
    "NumberOfNodes": 2,
    "PendingModifiedValues": {
      "MasterUserPassword": "\\*****"
    }
  },
  "ResponseMetadata": {
    "RequestId": "7cf4bcfc-64dd-11e2-bea9-49e0ce183f07"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateCluster](#)를 참조하세요.

Go

SDK for Go V2

 Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
import (  
    "context"  
    "errors"  
    "log"  
    "time"  
  
    "github.com/aws/aws-sdk-go-v2/aws"  
    "github.com/aws/aws-sdk-go-v2/service/redshift"  
    "github.com/aws/aws-sdk-go-v2/service/redshift/types"  
)  
  
// RedshiftActions wraps Redshift service actions.  
type RedshiftActions struct {  
    RedshiftClient *redshift.Client  
}  
  
// CreateCluster sends a request to create a cluster with the given clusterId  
// using the provided credentials.  
func (actor RedshiftActions) CreateCluster(ctx context.Context, clusterId string,  
    userName string, userPassword string, nodeType string, clusterType string,  
    publiclyAccessible bool) (*redshift.CreateClusterOutput, error) {  
    // Create a new Redshift cluster  
    input := &redshift.CreateClusterInput{  
        ClusterIdentifier:  aws.String(clusterId),  
        MasterUserPassword: aws.String(userPassword),  
        MasterUsername:      aws.String(userName),  
        NodeType:            aws.String(nodeType),  
    }
```

```

ClusterType:      aws.String(clusterType),
PubliclyAccessible: aws.Bool(publiclyAccessible),
}
var opErr *types.ClusterAlreadyExistsFault
output, err := actor.RedshiftClient.CreateCluster(ctx, input)
if err != nil && errors.As(err, &opErr) {
    log.Println("Cluster already exists")
    return nil, nil
} else if err != nil {
    log.Printf("Failed to create Redshift cluster: %v\n", err)
    return nil, err
}

log.Printf("Created cluster %s\n", *output.Cluster.ClusterIdentifier)
return output, nil
}

```

- API에 대한 세부 정보는 AWS SDK for Go API 참조의 [CreateCluster](#)를 참조하세요.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

클러스터를 생성합니다.

```

/**
 * Creates a new Amazon Redshift cluster asynchronously.
 * @param clusterId      the unique identifier for the cluster
 * @param username       the username for the administrative user
 * @param userPassword   the password for the administrative user
 * @return a CompletableFuture that represents the asynchronous operation of
 *         creating the cluster
 * @throws RuntimeException if the cluster creation fails
 */

```

```

public CompletableFuture<CreateClusterResponse> createClusterAsync(String
clusterId, String username, String userPassword) {
    CreateClusterRequest clusterRequest = CreateClusterRequest.builder()
        .clusterIdentifier(clusterId)
        .masterUsername(username)
        .masterUserPassword(userPassword)
        .nodeType("ra3.4xlarge")
        .publiclyAccessible(true)
        .numberOfNodes(2)
        .build();

    return getAsyncClient().createCluster(clusterRequest)
        .whenComplete((response, exception) -> {
            if (response != null) {
                logger.info("Created cluster ");
            } else {
                throw new RuntimeException("Failed to create cluster: " +
exception.getMessage(), exception);
            }
        });
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [CreateCluster](#)를 참조하십시오.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

클라이언트를 생성합니다.

```

import { RedshiftClient } from "@aws-sdk/client-redshift";
// Set the AWS Region.
const REGION = "REGION";
//Set the Redshift Service Object
const redshiftClient = new RedshiftClient({ region: REGION });

```

```
export { redshiftClient };
```

클러스터를 생성합니다.

```
// Import required AWS SDK clients and commands for Node.js
import { CreateClusterCommand } from "@aws-sdk/client-redshift";
import { redshiftClient } from "../libs/redshiftClient.js";

const params = {
  ClusterIdentifier: "CLUSTER_NAME", // Required
  NodeType: "NODE_TYPE", //Required
  MasterUsername: "MASTER_USER_NAME", // Required - must be lowercase
  MasterUserPassword: "MASTER_USER_PASSWORD", // Required - must contain at least
  one uppercase letter, and one number
  ClusterType: "CLUSTER_TYPE", // Required
  IAMRoleARN: "IAM_ROLE_ARN", // Optional - the ARN of an IAM role with
  permissions your cluster needs to access other AWS services on your behalf, such
  as Amazon S3.
  ClusterSubnetGroupName: "CLUSTER_SUBNET_GROUPNAME", //Optional - the name of a
  cluster subnet group to be associated with this cluster. Defaults to 'default'
  if not specified.
  DBName: "DATABASE_NAME", // Optional - defaults to 'dev' if not specified
  Port: "PORT_NUMBER", // Optional - defaults to '5439' if not specified
};

const run = async () => {
  try {
    const data = await redshiftClient.send(new CreateClusterCommand(params));
    console.log(
      `Cluster ${data.Cluster.ClusterIdentifier} successfully created`,
    );
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();
```

- API 세부 정보는 AWS SDK for JavaScript API 참조의 [CreateCluster](#)를 참조하십시오.

Kotlin

SDK for Kotlin

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

클러스터를 생성합니다.

```
suspend fun createCluster(  
    clusterId: String?,  
    masterUsernameVal: String?,  
    masterUserPasswordVal: String?,  
) {  
    val clusterRequest =  
        CreateClusterRequest {  
            clusterIdentifier = clusterId  
            availabilityZone = "us-east-1a"  
            masterUsername = masterUsernameVal  
            masterUserPassword = masterUserPasswordVal  
            nodeType = "ra3.4xlarge"  
            publiclyAccessible = true  
            numberOfNodes = 2  
        }  
  
    RedshiftClient.fromEnvironment { region = "us-east-1" }.use { redshiftClient  
->  
        val clusterResponse = redshiftClient.createCluster(clusterRequest)  
        println("Created cluster ${clusterResponse.cluster?.clusterIdentifier}")  
    }  
}
```

- API 세부 정보는 AWS SDK for Kotlin API 참조의 [CreateCluster](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
class RedshiftWrapper:
    """
    Encapsulates Amazon Redshift cluster operations.
    """

    def __init__(self, redshift_client):
        """
        :param redshift_client: A Boto3 Redshift client.
        """
        self.client = redshift_client

    def create_cluster(
        self,
        cluster_identifier,
        node_type,
        master_username,
        master_user_password,
        publicly_accessible,
        number_of_nodes,
    ):
        """
        Creates a cluster.

        :param cluster_identifier: The name of the cluster.
        :param node_type: The type of node in the cluster.
        :param master_username: The master username.
        :param master_user_password: The master user password.
        :param publicly_accessible: Whether the cluster is publicly accessible.
        :param number_of_nodes: The number of nodes in the cluster.
        :return: The cluster.
        """
```



```

try:
    cluster = self.client.create_cluster(
        ClusterIdentifier=cluster_identifier,
        NodeType=node_type,
        MasterUsername=master_username,
        MasterUserPassword=master_user_password,
        PubliclyAccessible=publicly_accessible,
        NumberOfNodes=number_of_nodes,
    )
    return cluster
except ClientError as err:
    logging.error(
        "Couldn't create a cluster. Here's why: %s: %s",
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise

```

다음 코드는 RedshiftWrapper 객체를 인스턴스화합니다.

```

client = boto3.client("redshift")
redshift_wrapper = RedshiftWrapper(client)

```

- API 세부 정보는 AWS SDK for Python(Boto3) API 참조의 [CreateCluster](#)를 참조하세요.

SAP ABAP

SDK for SAP ABAP

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

클러스터를 생성합니다.

```

TRY.
  " Example values: iv_cluster_identifier = 'my-redshift-cluster'
  " Example values: iv_node_type = 'ra3.4xlarge'
  " Example values: iv_master_username = 'awsuser'
  " Example values: iv_master_password = 'AwsUser1000'
  " Example values: iv_publicly_accessible = abap_true
  " Example values: iv_number_of_nodes = 2
  oo_result = lo_rsh->createcluster(
    iv_clusteridentifier = iv_cluster_identifier
    iv_nodetype = iv_node_type
    iv_masterusername = iv_master_username
    iv_masteruserpassword = iv_master_password
    iv_publiclyaccessible = iv_publicly_accessible
    iv_numberofnodes = iv_number_of_nodes
  ).
  MESSAGE 'Redshift cluster created successfully.' TYPE 'I'.
CATCH /aws1/cx_rshclustalrddyexfault.
  MESSAGE 'Cluster already exists.' TYPE 'I'.
CATCH /aws1/cx_rshclstquotaexcdfault.
  MESSAGE 'Cluster quota exceeded.' TYPE 'I'.
ENDTRY.

```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 [CreateCluster](#)를 참조하세요.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [AWS SDK와 함께 이 서비스 사용](#)을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 함께 **DeleteCluster** 사용

다음 코드 예시는 DeleteCluster의 사용 방법을 보여 줍니다.

.NET

SDK for .NET(v4)

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

/// <summary>
/// Delete an Amazon Redshift cluster without a final snapshot.
/// </summary>
/// <param name="clusterIdentifier">The identifier for the cluster.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteClusterWithoutSnapshotAsync(string
clusterIdentifier)
{
    try
    {
        var request = new DeleteClusterRequest
        {
            ClusterIdentifier = clusterIdentifier,
            SkipFinalClusterSnapshot = true
        };

        var response = await _redshiftClient.DeleteClusterAsync(request);
        Console.WriteLine($"The {clusterIdentifier} was deleted");
        return true;
    }
    catch (ClusterNotFoundException ex)
    {
        Console.WriteLine($"Cluster not found: {ex.Message}");
        return false;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Couldn't delete cluster. Here's why:
{ex.Message}");
        return false;
    }
}

```

- API 세부 정보는 AWS SDK for .NET API 참조의 [DeleteCluster](#)를 참조하세요.

CLI

AWS CLI

클러스터 삭제 및 최종 클러스터 스냅샷 생성 방지 이 예시에서는 클러스터를 삭제하고 최종 클러스터 스냅샷이 생성되지 않도록 데이터를 강제로 삭제합니다. 명령:

```
aws redshift delete-cluster --cluster-identifier mycluster --skip-final-cluster-snapshot
```

클러스터 삭제, 최종 클러스터 스냅샷 허용 이 예시에서는 클러스터를 삭제하지만 최종 클러스터 스냅샷을 지정합니다. 명령:

```
aws redshift delete-cluster --cluster-identifier mycluster --final-cluster-snapshot-identifier myfinalsnapshot
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteCluster](#)를 참조하세요.

Go

SDK for Go V2

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
import (
    "context"
    "errors"
    "log"
    "time"

    "github.com/aws/aws-sdk-go-v2/aws"
    "github.com/aws/aws-sdk-go-v2/service/redshift"
    "github.com/aws/aws-sdk-go-v2/service/redshift/types"
)

// RedshiftActions wraps Redshift service actions.
type RedshiftActions struct {
    RedshiftClient *redshift.Client
}
```

```
// DeleteCluster deletes the given cluster.
func (actor RedshiftActions) DeleteCluster(ctx context.Context, clusterId string)
(bool, error) {
    input := redshift.DeleteClusterInput{
        ClusterIdentifier:      aws.String(clusterId),
        SkipFinalClusterSnapshot: aws.Bool(true),
    }
    _, err := actor.RedshiftClient.DeleteCluster(ctx, &input)
    var opErr *types.ClusterNotFoundFault
    if err != nil && errors.As(err, &opErr) {
        log.Println("Cluster was not found. Where could it be?")
        return false, err
    } else if err != nil {
        log.Printf("Failed to delete Redshift cluster: %v\n", err)
        return false, err
    }
    waiter := redshift.NewClusterDeletedWaiter(actor.RedshiftClient)
    err = waiter.Wait(ctx, &redshift.DescribeClustersInput{
        ClusterIdentifier: aws.String(clusterId),
    }, 5*time.Minute)
    if err != nil {
        log.Printf("Wait time exceeded for deleting cluster, continuing: %v\n", err)
    }
    log.Printf("The cluster %s was deleted\n", clusterId)
    return true, nil
}
```

- API 세부 정보는 AWS SDK for Go API 참조의 [DeleteCluster](#)를 참조하세요.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

클러스터를 삭제합니다.

```
/**
 * Deletes a Redshift cluster asynchronously.
 *
 * @param clusterId the identifier of the Redshift cluster to be deleted
 * @return a {@link CompletableFuture} that represents the asynchronous
operation of deleting the Redshift cluster
 */
public CompletableFuture<DeleteClusterResponse>
deleteRedshiftClusterAsync(String clusterId) {
    DeleteClusterRequest deleteClusterRequest =
DeleteClusterRequest.builder()
        .clusterIdentifier(clusterId)
        .skipFinalClusterSnapshot(true)
        .build();

    return getAsyncClient().deleteCluster(deleteClusterRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                // Handle exceptions
                if (exception.getCause() instanceof RedshiftException) {
                    logger.info("Error: {}", exception.getMessage());
                } else {
                    logger.info("Unexpected error: {}",
exception.getMessage());
                }
            } else {
                // Handle successful response
                logger.info("The status is {}",
response.cluster().clusterStatus());
            }
        });
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DeleteCluster](#)를 참조하십시오.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

클라이언트를 생성합니다.

```
import { RedshiftClient } from "@aws-sdk/client-redshift";
// Set the AWS Region.
const REGION = "REGION";
//Set the Redshift Service Object
const redshiftClient = new RedshiftClient({ region: REGION });
export { redshiftClient };
```

클러스터를 생성합니다.

```
// Import required AWS SDK clients and commands for Node.js
import { DeleteClusterCommand } from "@aws-sdk/client-redshift";
import { redshiftClient } from "../libs/redshiftClient.js";

const params = {
  ClusterIdentifier: "CLUSTER_NAME",
  SkipFinalClusterSnapshot: false,
  FinalClusterSnapshotIdentifier: "CLUSTER_SNAPSHOT_ID",
};

const run = async () => {
  try {
    const data = await redshiftClient.send(new DeleteClusterCommand(params));
    console.log("Success, cluster deleted. ", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();
```

- API 세부 정보는 AWS SDK for JavaScript API 참조의 [DeleteCluster](#)를 참조하십시오.

Kotlin

SDK for Kotlin

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

클러스터를 삭제합니다.

```
suspend fun deleteRedshiftCluster(clusterId: String?) {
    val request =
        DeleteClusterRequest {
            clusterIdentifier = clusterId
            skipFinalClusterSnapshot = true
        }

    RedshiftClient.fromEnvironment { region = "us-west-2" }.use { redshiftClient
->
        val response = redshiftClient.deleteCluster(request)
        println("The status is ${response.cluster?.clusterStatus}")
    }
}
```

- API 세부 정보는 AWS SDK for Kotlin API 참조의 [DeleteCluster](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
class RedshiftWrapper:
    """
    Encapsulates Amazon Redshift cluster operations.
    """

    def __init__(self, redshift_client):
        """
        :param redshift_client: A Boto3 Redshift client.
        """
        self.client = redshift_client

    def delete_cluster(self, cluster_identifier):
        """
        Deletes a cluster.

        :param cluster_identifier: The cluster identifier.
        """
        try:
            self.client.delete_cluster(
                ClusterIdentifier=cluster_identifier,
                SkipFinalClusterSnapshot=True
            )
        except ClientError as err:
            logging.error(
                "Couldn't delete a cluster. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

다음 코드는 RedshiftWrapper 객체를 인스턴스화합니다.

```
client = boto3.client("redshift")
redshift_wrapper = RedshiftWrapper(client)
```

- API 세부 정보는 AWS SDK for Python(Boto3) API 참조의 [DeleteCluster](#)를 참조하세요.

SAP ABAP

SDK for SAP ABAP

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

클러스터를 삭제합니다.

```
TRY.
  " Example values: iv_cluster_identifier = 'my-redshift-cluster'
  lo_rsh->deletecluster(
    iv_clusteridentifier = iv_cluster_identifier
    iv_skipfinalclustersnapshot = abap_true
  ).
  MESSAGE 'Redshift cluster deleted successfully.' TYPE 'I'.
CATCH /aws1/cx_rshclustnotfoundfault.
  MESSAGE 'Cluster not found.' TYPE 'I'.
CATCH /aws1/cx_rshinvcluststatefault.
  MESSAGE 'Invalid cluster state for deletion.' TYPE 'I'.
ENDTRY.
```

- API 세부 정보는 [AWS SDK for SAP ABAP API 참조](#)의 DeleteCluster를 참조하세요.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [AWS SDK와 함께 이 서비스 사용](#)을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 함께 **DescribeClusters** 사용

다음 코드 예시는 DescribeClusters의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [기본 사항 알아보기](#)

.NET

SDK for .NET(v4)

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

/// <summary>
/// Describe Amazon Redshift clusters.
/// </summary>
/// <param name="clusterIdentifier">Optional cluster identifier to describe a
specific cluster.</param>
/// <returns>A list of clusters.</returns>
public async Task<List<Cluster>> DescribeClustersAsync(string?
clusterIdentifier = null)
{
    try
    {
        var clusters = new List<Cluster>();
        var request = new DescribeClustersRequest();
        if (!string.IsNullOrEmpty(clusterIdentifier))
        {
            request.ClusterIdentifier = clusterIdentifier;
        }

        var clustersPaginator =
_redshiftClient.Paginators.DescribeClusters(request);
        await foreach (var response in clustersPaginator.Responses)
        {
            if (response.Clusters != null)

```

```

        clusters.AddRange(response.Clusters);
    }

    Console.WriteLine($"{clusters.Count} cluster(s) retrieved.");
    foreach (var cluster in clusters)
    {
        Console.WriteLine($"\\t{cluster.ClusterIdentifier} (Status:
{cluster.ClusterStatus})");
    }

    return clusters;
}
catch (ClusterNotFoundException ex)
{
    Console.WriteLine($"Cluster {clusterIdentifier} not found:
{ex.Message}");
    throw;
}
catch (Exception ex)
{
    Console.WriteLine($"Couldn't describe clusters. Here's why:
{ex.Message}");
    throw;
}
}
}

```

- API에 대한 세부 정보는 AWS SDK for .NET API 참조의 [DescribeClusters](#)를 참조하세요.

CLI

AWS CLI

모든 클러스터의 설명 가져오기 이 예시에서는 계정의 모든 클러스터에 대한 설명을 반환합니다. 기본 출력 형식은 JSON입니다. 명령:

```
aws redshift describe-clusters
```

결과:

```
{
  "Clusters": [
```

```
{
  "NodeType": "dw.hs1.xlarge",
  "Endpoint": {
    "Port": 5439,
    "Address": "mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com"
  },
  "ClusterVersion": "1.0",
  "PubliclyAccessible": "true",
  "MasterUsername": "adminuser",
  "ClusterParameterGroups": [
    {
      "ParameterApplyStatus": "in-sync",
      "ParameterGroupName": "default.redshift-1.0"
    }
  ],
  "ClusterSecurityGroups": [
    {
      "Status": "active",
      "ClusterSecurityGroupName": "default"
    }
  ],
  "AllowVersionUpgrade": true,
  "VpcSecurityGroups": [],
  "AvailabilityZone": "us-east-1a",
  "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
  "PreferredMaintenanceWindow": "sat:03:30-sat:04:00",
  "AutomatedSnapshotRetentionPeriod": 1,
  "ClusterStatus": "available",
  "ClusterIdentifier": "mycluster",
  "DBName": "dev",
  "NumberOfNodes": 2,
  "PendingModifiedValues": {}
} ],
"ResponseMetadata": {
  "RequestId": "65b71cac-64df-11e2-8f5b-e90bd6c77476"
}
}
```

--output text 옵션을 사용하여 텍스트 형식으로 동일한 정보를 얻을 수도 있습니다. 명령:

--output text 옵션. 명령:

옵션. 명령:

```
aws redshift describe-clusters --output text
```

결과:

```
dw.hs1.xlarge      1.0      true      adminuser      True      us-east-1a
2013-01-22T21:59:29.559Z      sat:03:30-sat:04:00      1      available
mycluster      dev      2
ENDPOINT      5439      mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com
in-sync      default.redshift-1.0
active      default
PENDINGMODIFIEDVALUES
RESPONSEMETADATA      934281a8-64df-11e2-b07c-f7fbdd006c67
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeClusters](#)를 참조하세요.

Go

SDK for Go V2

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
import (
    "context"
    "errors"
    "log"
    "time"

    "github.com/aws/aws-sdk-go-v2/aws"
    "github.com/aws/aws-sdk-go-v2/service/redshift"
    "github.com/aws/aws-sdk-go-v2/service/redshift/types"
)

// RedshiftActions wraps Redshift service actions.
type RedshiftActions struct {
    RedshiftClient *redshift.Client
}
```

```
// DescribeClusters returns information about the given cluster.
func (actor RedshiftActions) DescribeClusters(ctx context.Context, clusterId
string) (*redshift.DescribeClustersOutput, error) {
    input, err := actor.RedshiftClient.DescribeClusters(ctx,
        &redshift.DescribeClustersInput{
            ClusterIdentifier: aws.String(clusterId),
        })
    var opErr *types.AccessToClusterDeniedFault
    if errors.As(err, &opErr) {
        println("Access to cluster denied.")
        panic(err)
    } else if err != nil {
        println("Failed to describe Redshift clusters.")
        return nil, err
    }
    return input, nil
}
```

- API에 대한 세부 정보는 AWS SDK for Go API 참조의 [DescribeClusters](#)를 참조하세요.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

클러스터를 설명하세요.

```
/**
 * Waits asynchronously for the specified cluster to become available.
 * @param clusterId the identifier of the cluster to wait for
 * @return a {@link CompletableFuture} that completes when the cluster is
 * ready
```

```

    */
    public CompletableFuture<Void> waitForClusterReadyAsync(String clusterId) {
        DescribeClustersRequest clustersRequest =
DescribeClustersRequest.builder()
            .clusterIdentifier(clusterId)
            .build();

        logger.info("Waiting for cluster to become available. This may take a few
minutes.");
        long startTime = System.currentTimeMillis();

        // Recursive method to poll the cluster status.
        return checkClusterStatusAsync(clustersRequest, startTime);
    }

    private CompletableFuture<Void>
checkClusterStatusAsync(DescribeClustersRequest clustersRequest, long startTime)
{
    return getAsyncClient().describeClusters(clustersRequest)
        .thenCompose(clusterResponse -> {
            List<Cluster> clusterList = clusterResponse.clusters();
            boolean clusterReady = false;
            for (Cluster cluster : clusterList) {
                if ("available".equals(cluster.clusterStatus())) {
                    clusterReady = true;
                    break;
                }
            }

            if (clusterReady) {
                logger.info(String.format("Cluster is available!"));
                return CompletableFuture.completedFuture(null);
            } else {
                long elapsedTimeMillis = System.currentTimeMillis() -
startTime;

                long elapsedSeconds = elapsedTimeMillis / 1000;
                long minutes = elapsedSeconds / 60;
                long seconds = elapsedSeconds % 60;
                System.out.printf("\rElapsed Time: %02d:%02d - Waiting for
cluster...", minutes, seconds);
                System.out.flush();

                // Wait 1 second before the next status check
                return CompletableFuture.runAsync(() -> {

```



```

        try {
            TimeUnit.SECONDS.sleep(1);
        } catch (InterruptedException e) {
            throw new RuntimeException("Error during sleep: " +
e.getMessage(), e);
        }
    }).thenCompose(ignored ->
checkClusterStatusAsync(clustersRequest, startTime));
    }
    }).exceptionally(exception -> {
        throw new RuntimeException("Failed to get cluster status: " +
exception.getMessage(), exception);
    });
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeClusters](#)를 참조하십시오.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

클라이언트를 생성합니다.

```

import { RedshiftClient } from "@aws-sdk/client-redshift";
// Set the AWS Region.
const REGION = "REGION";
//Set the Redshift Service Object
const redshiftClient = new RedshiftClient({ region: REGION });
export { redshiftClient };

```

클러스터에 대해 설명합니다.

```

// Import required AWS SDK clients and commands for Node.js

```

```
import { DescribeClustersCommand } from "@aws-sdk/client-redshift";
import { redshiftClient } from "../libs/redshiftClient.js";

const params = {
  ClusterIdentifier: "CLUSTER_NAME",
};

const run = async () => {
  try {
    const data = await redshiftClient.send(new DescribeClustersCommand(params));
    console.log("Success", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- API 세부 정보는 AWS SDK for JavaScript API 참조의 [DescribeClusters](#)를 참조하십시오.

Kotlin

SDK for Kotlin

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

클러스터를 설명하세요.

```
suspend fun describeRedshiftClusters() {
    RedshiftClient.fromEnvironment { region = "us-west-2" }.use { redshiftClient
    ->
        val clusterResponse =
        redshiftClient.describeClusters(DescribeClustersRequest {})
        val clusterList = clusterResponse.clusters

        if (clusterList != null) {
            for (cluster in clusterList) {
```

```

        println("Cluster database name is ${cluster.dbName}")
        println("Cluster status is ${cluster.clusterStatus}")
    }
}
}
}

```

- API 세부 정보는 AWS SDK for Kotlin API 참조의 [DescribeClusters](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

class RedshiftWrapper:
    """
    Encapsulates Amazon Redshift cluster operations.
    """

    def __init__(self, redshift_client):
        """
        :param redshift_client: A Boto3 Redshift client.
        """
        self.client = redshift_client

    def describe_clusters(self, cluster_identifier):
        """
        Describes a cluster.

        :param cluster_identifier: The cluster identifier.
        :return: A list of clusters.
        """
        try:
            kwargs = {}
            if cluster_identifier:

```

```

        kwargs["ClusterIdentifier"] = cluster_identifier

    paginator = self.client.get_paginator("describe_clusters")
    clusters = []
    for page in paginator.paginate(**kwargs):
        clusters.extend(page["Clusters"])

    return clusters

except ClientError as err:
    logging.error(
        "Couldn't describe a cluster. Here's why: %s: %s",
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise

```

다음 코드는 RedshiftWrapper 객체를 인스턴스화합니다.

```

client = boto3.client("redshift")
redshift_wrapper = RedshiftWrapper(client)

```

- API 세부 정보는 AWS SDK for Python(Boto3) API 참조의 [DescribeClusters](#)를 참조하세요.

SAP ABAP

SDK for SAP ABAP

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

클러스터를 설명하세요.

TRY.

```

" Example values: iv_cluster_identifier = 'my-redshift-
cluster' (optional)
oo_result = lo_rsh->describeclusters(
    iv_clusteridentifier = iv_cluster_identifier
).
lt_clusters = oo_result->get_clusters( ).
lv_cluster_count = lines( lt_clusters ).
MESSAGE |Retrieved { lv_cluster_count } cluster(s).| TYPE 'I'.
CATCH /aws1/cx_rshclustnotfoundfault.
MESSAGE 'Cluster not found.' TYPE 'I'.
ENDTRY.

```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 [DescribeClusters](#)를 참조하세요.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [AWS SDK와 함께 이 서비스 사용](#)을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 함께 **DescribeStatement** 사용

다음 코드 예시는 DescribeStatement의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [기본 사항 알아보기](#)

.NET

SDK for .NET(v4)

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

/// <summary>
/// Describe a statement execution.
/// </summary>
/// <param name="statementId">The statement ID.</param>

```

```

    /// <returns>The statement description.</returns>
    public async Task<DescribeStatementResponse> DescribeStatementAsync(string
statementId)
    {
        try
        {
            var request = new DescribeStatementRequest
            {
                Id = statementId
            };

            var response = await
_redshiftDataClient.DescribeStatementAsync(request);
            return response;
        }
        catch (Amazon.RedshiftDataAPIService.Model.ResourceNotFoundException ex)
        {
            Console.WriteLine($"Statement not found: {ex.Message}");
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't describe statement. Here's why:
{ex.Message}");
            throw;
        }
    }
}

```

- API 세부 정보는 AWS SDK for .NET API 참조의 [DescribeStatement](#)를 참조하세요.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
/**
```

```

    * Checks the status of an SQL statement asynchronously and handles the
    completion of the statement.
    *
    * @param sqlId the ID of the SQL statement to check
    * @return a {@link CompletableFuture} that completes when the SQL
    statement's status is either "FINISHED" or "FAILED"
    */
    public CompletableFuture<Void> checkStatementAsync(String sqlId) {
        DescribeStatementRequest statementRequest =
DescribeStatementRequest.builder()
            .id(sqlId)
            .build();

        return getAsyncDataClient().describeStatement(statementRequest)
            .thenCompose(response -> {
                String status = response.statusAsString();
                logger.info("... Status: {} ", status);

                if ("FAILED".equals(status)) {
                    throw new RuntimeException("The Query Failed. Ending
program");
                } else if ("FINISHED".equals(status)) {
                    return CompletableFuture.completedFuture(null);
                } else {
                    // Sleep for 1 second and recheck status
                    return CompletableFuture.runAsync(() -> {
                        try {
                            TimeUnit.SECONDS.sleep(1);
                        } catch (InterruptedException e) {
                            throw new RuntimeException("Error during sleep: " +
e.getMessage(), e);
                        }
                    }).thenCompose(ignore -> checkStatementAsync(sqlId)); //
Recursively call until status is FINISHED or FAILED
                }
            }).whenComplete((result, exception) -> {
                if (exception != null) {
                    // Handle exceptions
                    logger.info("Error: {} ", exception.getMessage());
                } else {
                    logger.info("The statement is finished!");
                }
            });
    }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeStatement](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
class RedshiftDataWrapper:
    """Encapsulates Amazon Redshift data."""

    def __init__(self, client):
        """
        :param client: A Boto3 RedshiftDataWrapper client.
        """
        self.client = client

    def describe_statement(self, statement_id):
        """
        Describes a SQL statement.

        :param statement_id: The SQL statement identifier.
        :return: The SQL statement result.
        """
        try:
            response = self.client.describe_statement(Id=statement_id)
            return response
        except ClientError as err:
            logging.error(
                "Couldn't describe statement. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```


다음 코드는 RedshiftDataWrapper 객체를 인스턴스화합니다.

```
client = boto3.client("redshift-data")
redshift_data_wrapper = RedshiftDataWrapper(client)
```

- API 세부 정보는 AWS SDK for Python(Boto3) API 참조의 [DescribeStatement](#)를 참조하세요.

SAP ABAP

SDK for SAP ABAP

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
TRY.
    " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxxxx'
    oo_result = lo_rsd->describestatement(
        iv_id = iv_statement_id
    ).
    lv_status = oo_result->get_status( ).
    MESSAGE |Statement status: { lv_status }| TYPE 'I'.
CATCH /aws1/cx_rsdresourcenotfoundex.
    MESSAGE 'Statement not found.' TYPE 'I'.
CATCH /aws1/cx_rsdinternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
ENDTRY.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 [DescribeStatement](#)를 참조하세요.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [AWS SDK와 함께 이 서비스 사용](#)을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 함께 **ExecuteStatement** 사용

다음 코드 예시는 ExecuteStatement의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서 컨텍스트 내 이 작업을 확인할 수 있습니다.

- [기본 사항 알아보기](#)

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

SQL 스테이트먼트를 실행하여 데이터베이스 테이블을 만듭니다.

```
/**
 * Creates an asynchronous task to execute a SQL statement for creating a new
 * table.
 *
 * @param clusterId    the identifier of the Amazon Redshift cluster
 * @param dbName       the name of the database to create the table in
 * @param userName     the username to use for the database connection
 * @return a {@link CompletableFuture} that completes with the result of the
 * SQL statement execution
 * @throws RuntimeException if there is an error creating the table
 */
public CompletableFuture<ExecuteStatementResponse> createTableAsync(String
clusterId, String dbName, String userName) {
    ExecuteStatementRequest createTableRequest =
ExecuteStatementRequest.builder()
        .clusterIdentifier(clusterId)
        .dbUser(userName)
        .database(dbName)
        .sql("CREATE TABLE Movies (" +
            "id INT PRIMARY KEY, " +
            "title VARCHAR(100), " +
```

```

        "year INT)")
        .build();

    return getAsyncDataClient().executeStatement(createTableRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                throw new RuntimeException("Error creating table: " +
exception.getMessage(), exception);
            } else {
                logger.info("Table created: Movies");
            }
        });
}

```

SQL 스테이트먼트를 실행하여 데이터베이스 테이블에 데이터를 삽입합니다.

```

/**
 * Asynchronously pops a table from a JSON file.
 *
 * @param clusterId    the ID of the cluster
 * @param databaseName the name of the database
 * @param userName     the username
 * @param fileName     the name of the JSON file
 * @param number       the number of records to process
 * @return a CompletableFuture that completes with the number of records
added to the Movies table
 */
public CompletableFuture<Integer> popTableAsync(String clusterId, String
databaseName, String userName, String fileName, int number) {
    return CompletableFuture.supplyAsync(() -> {
        try {
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));

            JsonNode rootNode = new ObjectMapper().readTree(parser);
            Iterator<JsonNode> iter = rootNode.iterator();
            return iter;
        } catch (IOException e) {
            throw new RuntimeException("Failed to read or parse JSON
file: " + e.getMessage(), e);
        }
    }).thenCompose(iter -> processNodesAsync(clusterId, databaseName,
userName, iter, number))
}

```

```

        .whenComplete((result, exception) -> {
            if (exception != null) {
                logger.info("Error {} ", exception.getMessage());
            } else {
                logger.info("{} records were added to the Movies table." ,
result);
            }
        });
    }

    private CompletableFuture<Integer> processNodesAsync(String clusterId, String
databaseName, String userName, Iterator<JsonNode> iter, int number) {
        return CompletableFuture.supplyAsync(() -> {
            int t = 0;
            try {
                while (iter.hasNext()) {
                    if (t == number)
                        break;
                    JsonNode currentNode = iter.next();
                    int year = currentNode.get("year").asInt();
                    String title = currentNode.get("title").asText();

                    // Use SqlParameter to avoid SQL injection.
                    List<SqlParameter> parameterList = new ArrayList<>();
                    String sqlStatement = "INSERT INTO Movies
VALUES( :id , :title, :year);";
                    SqlParameter idParam = SqlParameter.builder()
                        .name("id")
                        .value(String.valueOf(t))
                        .build();

                    SqlParameter titleParam = SqlParameter.builder()
                        .name("title")
                        .value(title)
                        .build();

                    SqlParameter yearParam = SqlParameter.builder()
                        .name("year")
                        .value(String.valueOf(year))
                        .build();
                    parameterList.add(idParam);
                    parameterList.add(titleParam);
                    parameterList.add(yearParam);

```

```

        ExecuteStatementRequest insertStatementRequest =
ExecuteStatementRequest.builder()
    .clusterIdentifier(clusterId)
    .sql(sqlStatement)
    .database(databaseName)
    .dbUser(userName)
    .parameters(parameterList)
    .build();

getAsyncDataClient().executeStatement(insertStatementRequest);
        logger.info("Inserted: " + title + " (" + year + ")");
        t++;
    }
    } catch (RedshiftDataException e) {
        throw new RuntimeException("Error inserting data: " +
e.getMessage(), e);
    }
    return t;
});
}

```

SQL 스테이트먼트를 실행하여 데이터베이스 테이블을 쿼리합니다.

```

/**
 * Asynchronously queries movies by a given year from a Redshift database.
 *
 * @param database    the name of the database to query
 * @param dbUser      the user to connect to the database with
 * @param year        the year to filter the movies by
 * @param clusterId   the identifier of the Redshift cluster to connect to
 * @return a {@link CompletableFuture} containing the response ID of the
executed SQL statement
 */
public CompletableFuture<String> queryMoviesByYearAsync(String database,
                                                         String dbUser,
                                                         int year,
                                                         String
clusterId) {

    String sqlStatement = "SELECT * FROM Movies WHERE year = :year";
    SqlParameter yearParam = SqlParameter.builder()

```

```

        .name("year")
        .value(String.valueOf(year))
        .build();

    ExecuteStatementRequest statementRequest =
    ExecuteStatementRequest.builder()
        .clusterIdentifier(clusterId)
        .database(database)
        .dbUser(dbUser)
        .parameters(yearParam)
        .sql(sqlStatement)
        .build();

    return CompletableFuture.supplyAsync(() -> {
        try {
            ExecuteStatementResponse response =
            getAsyncDataClient().executeStatement(statementRequest).join(); // Use join() to
            wait for the result
            return response.id();
        } catch (RedshiftDataException e) {
            throw new RuntimeException("Error executing statement: " +
            e.getMessage(), e);
        }
    }).exceptionally(exception -> {
        logger.info("Error: {}", exception.getMessage());
        return "";
    });
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [ExecuteStatement](#)를 참조하세요.

SAP ABAP

SDK for SAP ABAP

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

TRY.
  " Example values: iv_cluster_identifier = 'redshift-cluster-movies'
  " Example values: iv_database_name = 'dev'
  " Example values: iv_user_name = 'awsuser'
  " Example values: iv_sql = 'SELECT * FROM movies WHERE year = :year'
  " Example values: it_parameter_list - SQL parameters for parameterized
queries

  " Only pass parameters if the list is not empty
  IF it_parameter_list IS NOT INITIAL.
    oo_result = lo_rsd->executestatement(
      iv_clusteridentifier = iv_cluster_identifier
      iv_database = iv_database_name
      iv_dbuser = iv_user_name
      iv_sql = iv_sql
      it_parameters = it_parameter_list
    ).
  ELSE.
    oo_result = lo_rsd->executestatement(
      iv_clusteridentifier = iv_cluster_identifier
      iv_database = iv_database_name
      iv_dbuser = iv_user_name
      iv_sql = iv_sql
    ).
  ENDIF.

  lv_statement_id = oo_result->get_id( ).
  MESSAGE |Statement executed. ID: { lv_statement_id }| TYPE 'I'.
CATCH /aws1/cx_rsdexecutestatementex.
  MESSAGE 'Statement execution error.' TYPE 'I'.
CATCH /aws1/cx_rsdresourcenotfoundex.
  MESSAGE 'Resource not found.' TYPE 'I'.
ENDTRY.

```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 [ExecuteStatement](#)를 참조하세요.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [AWS SDK와 함께 이 서비스 사용](#)을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 함께 **GetStatementResult** 사용

다음 코드 예시는 GetStatementResult의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [기본 사항 알아보기](#)

.NET

SDK for .NET(v4)

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
/// <summary>
/// Get the results of a statement execution.
/// </summary>
/// <param name="statementId">The statement ID.</param>
/// <returns>A list of result rows.</returns>
public async Task<List<List<Field>>> GetStatementResultAsync(string
statementId)
{
    try
    {
        var request = new GetStatementResultRequest
        {
            Id = statementId
        };

        var response = await
_redshiftDataClient.GetStatementResultAsync(request);
        return response.Records;
    }
    catch (Amazon.RedshiftDataAPIService.Model.ResourceNotFoundException ex)
    {
        Console.WriteLine($"Statement not found: {ex.Message}");
    }
}
```



```

        throw;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Couldn't get statement result. Here's why:
{ex.Message}");
        throw;
    }
}

```

- API에 대한 세부 정보는 AWS SDK for .NET API 참조의 [GetStatementResult](#)를 참조하세요.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

문 결과를 확인합니다.

```

/**
 * Asynchronously retrieves the results of a statement execution.
 *
 * @param statementId the ID of the statement for which to retrieve the
results
 * @return a {@link CompletableFuture} that completes when the statement
result has been processed
 */
public CompletableFuture<Void> getResultsAsync(String statementId) {
    GetStatementResultRequest resultRequest =
GetStatementResultRequest.builder()
        .id(statementId)
        .build();

    return getAsyncDataClient().getStatementResult(resultRequest)
        .handle((response, exception) -> {
            if (exception != null) {

```

```

        logger.info("Error getting statement result {} ",
exception.getMessage());
        throw new RuntimeException("Error getting statement result: "
+ exception.getMessage(), exception);
    }

    // Extract and print the field values using streams if the
response is valid.
    response.records().stream()
        .flatMap(List::stream)
        .map(Field::stringValue)
        .filter(value -> value != null)
        .forEach(value -> System.out.println("The Movie title field
is " + value));

    return response;
}).thenAccept(response -> {
    // Optionally add more logic here if needed after handling the
response
});
}

```

- API에 대한 세부 정보는 AWS SDK for Java 2.x API 참조의 [GetStatementResult](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

class RedshiftDataWrapper:
    """Encapsulates Amazon Redshift data."""

    def __init__(self, client):
        """

```

```

        :param client: A Boto3 RedshiftDataWrapper client.
        """
        self.client = client

    def get_statement_result(self, statement_id):
        """
        Gets the result of a SQL statement.

        :param statement_id: The SQL statement identifier.
        :return: The SQL statement result.
        """
        try:
            result = {
                "Records": [],
            }
            paginator = self.client.get_paginator("get_statement_result")
            for page in paginator.paginate(Id=statement_id):
                if "ColumnMetadata" not in result:
                    result["ColumnMetadata"] = page["ColumnMetadata"]
                result["Records"].extend(page["Records"])
            return result
        except ClientError as err:
            logging.error(
                "Couldn't get statement result. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise

```

다음 코드는 RedshiftDataWrapper 객체를 인스턴스화합니다.

```

client = boto3.client("redshift-data")
redshift_data_wrapper = RedshiftDataWrapper(client)

```

- API 세부 정보는 AWS SDK for Python(Boto3) API 참조의 [GetStatementResult](#)를 참조하세요.

SAP ABAP

SDK for SAP ABAP

 Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

문 결과를 확인합니다.

```
TRY.
    " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-
    xxxxxxxxxxxxxx'
    " Handle pagination for large result sets

DO.
    lo_result_page = lo_rsd->getstatementresult(
        iv_id = iv_statement_id
        iv_nexttoken = lv_next_token
    ).

    " Collect records from this page
    lt_page_records = lo_result_page->get_records( ).
    APPEND LINES OF lt_page_records TO lt_all_records.

    " Check if there are more pages
    lv_next_token = lo_result_page->get_nexttoken( ).
    IF lv_next_token IS INITIAL.
        EXIT. " No more pages
    ENDIF.
ENDDO.

" For the last call, set oo_result for return value
oo_result = lo_result_page.
lv_record_count = lines( lt_all_records ).
MESSAGE |Retrieved { lv_record_count } record(s).| TYPE 'I'.
CATCH /aws1/cx_rsdresourcenotfoundex.
    MESSAGE 'Statement not found or results not available.' TYPE 'I'.
CATCH /aws1/cx_rsdinternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
ENDTRY.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 [GetStatementResult](#)를 참조하세요.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [AWS SDK와 함께 이 서비스 사용](#)을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 함께 **ListDatabases** 사용

다음 코드 예시는 ListDatabases의 사용 방법을 보여 줍니다.

.NET

SDK for .NET(v4)

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
/// <summary>
/// List databases in a Redshift cluster.
/// </summary>
/// <param name="clusterIdentifier">The cluster identifier.</param>
/// <param name="dbUser">The database user.</param>
/// <param name="dbName">The database name for authentication.</param>
/// <returns>A list of database names.</returns>
public async Task<List<string>> ListDatabasesAsync(string clusterIdentifier,
string dbUser, string dbName)
{
    try
    {
        var request = new ListDatabasesRequest
        {
            ClusterIdentifier = clusterIdentifier,
            DbUser = dbUser,
            Database = dbName
        };

        var response = await _redshiftDataClient.ListDatabasesAsync(request);
```

```

        var databases = new List<string>();

        foreach (var database in response.Databases)
        {
            Console.WriteLine($"The database name is : {database}");
            databases.Add(database);
        }

        return databases;
    }
    catch (Amazon.RedshiftDataAPIService.Model.ValidationException ex)
    {
        Console.WriteLine($"Validation error: {ex.Message}");
        throw;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Couldn't list databases. Here's why:
{ex.Message}");
        throw;
    }
}

```

- API 세부 정보는 AWS SDK for .NET API 참조의 [ListDatabases](#)를 참조하시기 바랍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

/**
 * Lists all databases asynchronously for the specified cluster, database
 * user, and database.
 * @param clusterId the identifier of the cluster to list databases for
 * @param dbUser the database user to use for the list databases request
 * @param database the database to list databases for

```

```

    * @return a {@link CompletableFuture} that completes when the database
    listing is complete, or throws a {@link RuntimeException} if there was an error
    */
    public CompletableFuture<Void> listAllDatabasesAsync(String clusterId, String
    dbUser, String database) {
        ListDatabasesRequest databasesRequest = ListDatabasesRequest.builder()
            .clusterIdentifier(clusterId)
            .dbUser(dbUser)
            .database(database)
            .build();

        // Asynchronous paginator for listing databases.
        ListDatabasesPublisher databasesPaginator =
        getAsyncDataClient().listDatabasesPaginator(databasesRequest);
        CompletableFuture<Void> future = databasesPaginator.subscribe(response ->
        {
            response.databases().forEach(db -> {
                logger.info("The database name is {} ", db);
            });
        });

        // Return the future for asynchronous handling.
        return future.exceptionally(exception -> {
            throw new RuntimeException("Failed to list databases: " +
            exception.getMessage(), exception);
        });
    }

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [ListDatabases](#)를 참조하시기 바랍니다.

SAP ABAP

SDK for SAP ABAP

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

TRY.

```

" Example values: iv_cluster_identifier = 'redshift-cluster-movies'
" Example values: iv_database_name = 'dev'
" Example values: iv_database_user = 'awsuser'
oo_result = lo_rsd->listdatabases(
    iv_clusteridentifier = iv_cluster_identifier
    iv_database = iv_database_name
    iv_dbuser = iv_database_user
).
lt_databases = oo_result->get_databases( ).
lv_db_count = lines( lt_databases ).
MESSAGE |Retrieved { lv_db_count } database(s).| TYPE 'I'.
CATCH /aws1/cx_rsddatabaseconnex.
MESSAGE 'Database connection error.' TYPE 'I'.
CATCH /aws1/cx_rsdresourcenotfoundex.
MESSAGE 'Cluster not found.' TYPE 'I'.
ENDTRY.

```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 [ListDatabases](#)를 참조하세요.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [AWS SDK와 함께 이 서비스 사용](#)을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 함께 **ModifyCluster** 사용

다음 코드 예시는 ModifyCluster의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [기본 사항 알아보기](#)

.NET

SDK for .NET(v4)

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.


```
/// <summary>
/// Modify an Amazon Redshift cluster.
/// </summary>
/// <param name="clusterIdentifier">The identifier for the cluster.</param>
/// <param name="preferredMaintenanceWindow">The preferred maintenance
window.</param>
/// <returns>True if successful.</returns>
public async Task<bool> ModifyClusterAsync(string clusterIdentifier, string
preferredMaintenanceWindow)
{
    try
    {
        var request = new ModifyClusterRequest
        {
            ClusterIdentifier = clusterIdentifier,
            PreferredMaintenanceWindow = preferredMaintenanceWindow
        };

        var response = await _redshiftClient.ModifyClusterAsync(request);
        Console.WriteLine($"The modified cluster was successfully modified
and has {response.Cluster.PreferredMaintenanceWindow} as the maintenance
window");
        return true;
    }
    catch (ClusterNotFoundException ex)
    {
        Console.WriteLine($"Cluster {clusterIdentifier} not found:
{ex.Message}");
        return false;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Couldn't modify cluster. Here's why:
{ex.Message}");
        return false;
    }
}
```

- API에 대한 세부 정보는 AWS SDK for .NET API 참조의 [ModifyCluster](#)를 참조하세요.

CLI

AWS CLI

클러스터에 보안 그룹 연결 이 예시에서는 클러스터 보안 그룹을 지정된 클러스터에 연결하는 방법을 보여줍니다. 명령:

```
aws redshift modify-cluster --cluster-identifier mycluster --cluster-security-groups mysecuritygroup
```

클러스터의 유지 관리 기간 수정 여기에서는 클러스터의 주간 기본 유지 관리 기간을 일요일 오후 11시 15분에 시작하여 월요일 오전 3시 15분에 끝나는 최소 4시간으로 변경하는 방법을 보여줍니다. 명령:

```
aws redshift modify-cluster --cluster-identifier mycluster --preferred-maintenance-window Sun:23:15-Mon:03:15
```

클러스터의 마스터 암호 변경 이 예시에서는 클러스터의 마스터 암호를 변경하는 방법을 보여줍니다. 명령:

```
aws redshift modify-cluster --cluster-identifier mycluster --master-user-password A1b2c3d4
```

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyCluster](#)를 참조하세요.

Go

SDK for Go V2

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
import (  
    "context"  
    "errors"
```

```
"log"
"time"

"github.com/aws/aws-sdk-go-v2/aws"
"github.com/aws/aws-sdk-go-v2/service/redshift"
"github.com/aws/aws-sdk-go-v2/service/redshift/types"
)

// RedshiftActions wraps Redshift service actions.
type RedshiftActions struct {
    RedshiftClient *redshift.Client
}

// ModifyCluster sets the preferred maintenance window for the given cluster.
func (actor RedshiftActions) ModifyCluster(ctx context.Context, clusterId string,
    maintenanceWindow string) *redshift.ModifyClusterOutput {
    // Modify the cluster's maintenance window
    input := &redshift.ModifyClusterInput{
        ClusterIdentifier:      aws.String(clusterId),
        PreferredMaintenanceWindow: aws.String(maintenanceWindow),
    }

    var opErr *types.InvalidClusterStateFault
    output, err := actor.RedshiftClient.ModifyCluster(ctx, input)
    if err != nil && errors.As(err, &opErr) {
        log.Println("Cluster is in an invalid state.")
        panic(err)
    } else if err != nil {
        log.Printf("Failed to modify Redshift cluster: %v\n", err)
        panic(err)
    }

    log.Printf("The cluster was successfully modified and now has %s as the
    maintenance window\n", *output.Cluster.PreferredMaintenanceWindow)
    return output
}
```

- API에 대한 세부 정보는 AWS SDK for Go API 참조의 [ModifyCluster](#)를 참조하세요.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

클러스터를 수정하세요.

```
/**
 * Modifies an Amazon Redshift cluster asynchronously.
 *
 * @param clusterId the identifier of the cluster to be modified
 * @return a {@link CompletableFuture} that completes when the cluster
 * modification is complete
 */
public CompletableFuture<ModifyClusterResponse> modifyClusterAsync(String
clusterId) {
    ModifyClusterRequest modifyClusterRequest =
ModifyClusterRequest.builder()
        .clusterIdentifier(clusterId)
        .preferredMaintenanceWindow("wed:07:30-wed:08:00")
        .build();

    return getAsyncClient().modifyCluster(modifyClusterRequest)
        .whenComplete((clusterResponse, exception) -> {
            if (exception != null) {
                if (exception.getCause() instanceof RedshiftException) {
                    logger.info("Error: {} ", exception.getMessage());
                } else {
                    logger.info("Unexpected error: {} ",
exception.getMessage());
                }
            } else {
                logger.info("The modified cluster was successfully modified
and has "
                    + clusterResponse.cluster().preferredMaintenanceWindow()
+ " as the maintenance window");
            }
        });
}
```

```
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [ModifyCluster](#)를 참조하십시오.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

클라이언트를 생성합니다.

```
import { RedshiftClient } from "@aws-sdk/client-redshift";
// Set the AWS Region.
const REGION = "REGION";
//Set the Redshift Service Object
const redshiftClient = new RedshiftClient({ region: REGION });
export { redshiftClient };
```

클러스터를 수정합니다.

```
// Import required AWS SDK clients and commands for Node.js
import { ModifyClusterCommand } from "@aws-sdk/client-redshift";
import { redshiftClient } from "../libs/redshiftClient.js";

// Set the parameters
const params = {
  ClusterIdentifier: "CLUSTER_NAME",
  MasterUserPassword: "NEW_MASTER_USER_PASSWORD",
};

const run = async () => {
  try {
    const data = await redshiftClient.send(new ModifyClusterCommand(params));
    console.log("Success was modified.", data);
  }
}
```

```

    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

- API 세부 정보는 AWS SDK for JavaScript API 참조의 [ModifyCluster](#)를 참조하십시오.

Kotlin

SDK for Kotlin

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

클러스터를 수정합니다.

```

suspend fun modifyCluster(clusterId: String?) {
    val modifyClusterRequest =
        ModifyClusterRequest {
            clusterIdentifier = clusterId
            preferredMaintenanceWindow = "wed:07:30-wed:08:00"
        }

    RedshiftClient { region = "us-west-2" }.use { redshiftClient ->
        val clusterResponse = redshiftClient.modifyCluster(modifyClusterRequest)
        println(
            "The modified cluster was successfully modified and has
            ${clusterResponse.cluster?.preferredMaintenanceWindow} as the maintenance
            window",
        )
    }
}

```

- API 세부 정보는 AWS SDK for Kotlin API 참조의 [ModifyCluster](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
class RedshiftWrapper:
    """
    Encapsulates Amazon Redshift cluster operations.
    """

    def __init__(self, redshift_client):
        """
        :param redshift_client: A Boto3 Redshift client.
        """
        self.client = redshift_client

    def modify_cluster(self, cluster_identifier, preferred_maintenance_window):
        """
        Modifies a cluster.

        :param cluster_identifier: The cluster identifier.
        :param preferred_maintenance_window: The preferred maintenance window.
        """
        try:
            self.client.modify_cluster(
                ClusterIdentifier=cluster_identifier,
                PreferredMaintenanceWindow=preferred_maintenance_window,
            )
        except ClientError as err:
            logging.error(
                "Couldn't modify a cluster. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

다음 코드는 RedshiftWrapper 객체를 인스턴스화합니다.

```
client = boto3.client("redshift")
redshift_wrapper = RedshiftWrapper(client)
```

- API 세부 정보는 AWS SDK for Python(Boto3) API 참조의 [ModifyCluster](#)를 참조하세요.

SAP ABAP

SDK for SAP ABAP

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

클러스터를 수정합니다.

```
TRY.
  " Example values: iv_cluster_identifier = 'my-redshift-cluster'
  " Example values: iv_pref_maintenance_wn = 'wed:07:30-wed:08:00'
  lo_rsh->modifycluster(
    iv_clusteridentifier = iv_cluster_identifier
    iv_preferredmaintenancewin00 = iv_pref_maintenance_wn
  ).
  MESSAGE 'Redshift cluster modified successfully.' TYPE 'I'.
CATCH /aws1/cx_rshclustnotfoundfault.
  MESSAGE 'Cluster not found.' TYPE 'I'.
CATCH /aws1/cx_rshinvcluststatefault.
  MESSAGE 'Invalid cluster state for modification.' TYPE 'I'.
ENDTRY.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 [ModifyCluster](#)을 참조하세요.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [AWS SDK와 함께 이 서비스 사용](#)을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK를 사용한 Amazon S3에 대한 시나리오

다음 코드 예시는 AWS SDK로 Amazon Redshift에서 일반적인 시나리오를 구현하는 방법을 보여줍니다. 이러한 시나리오에서는 Amazon Redshift 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 작업을 수행하는 방법을 보여줍니다. 각 시나리오에는 전체 소스 코드에 대한 링크가 포함되어 있습니다. 여기에서 코드를 설정 및 실행하는 방법에 대한 지침을 찾을 수 있습니다.

시나리오는 컨텍스트에 맞는 서비스 작업을 이해하는 데 도움이 되도록 중급 수준의 경험을 대상으로 합니다.

예시

- [Amazon Redshift 항목 추적기 만들기](#)

Amazon Redshift 항목 추적기 만들기

다음 코드 예제에서는 Amazon Redshift 데이터베이스를 사용하는 작업 항목을 추적하고 보고하는 웹 애플리케이션을 만드는 방법을 보여줍니다.

Java

SDK for Java 2.x

Amazon Redshift 데이터베이스에 저장된 작업 항목을 추적하고 보고하는 웹 애플리케이션을 만드는 방법을 보여줍니다.

Amazon Redshift 데이터를 쿼리하고 React 애플리케이션에서 사용하도록 Spring REST API를 설정하는 방법에 대한 지침과 전체 소스 코드는 [GitHub](#)에서 전체 예제를 참조하세요.

이 예제에서 사용되는 서비스

- Amazon Redshift
- Amazon SES

Kotlin

SDK for Kotlin

Amazon Redshift 데이터베이스에 저장된 작업 항목을 추적하고 보고하는 웹 애플리케이션을 만드는 방법을 보여줍니다.

Amazon Redshift 데이터를 쿼리하고 React 애플리케이션에서 사용하도록 Spring REST API를 설정하는 방법에 대한 지침과 전체 소스 코드는 [GitHub](#)에서 전체 예제를 참조하세요.

이 예제에서 사용되는 서비스

- Amazon Redshift
- Amazon SES

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [AWS SDK와 함께 이 서비스 사용](#)을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

문서 기록

Note

Amazon Redshift의 새로운 기능에 대한 설명은 [새로운 소식](#)을 참조하세요.

다음 테이블에서는 2018년 6월 이후 Amazon Redshift 관리 가이드에 대한 중요한 설명서 변경 사항에 대해 설명합니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

API 버전: 2012-12-01

Amazon Redshift 데이터베이스 개발자 안내서의 변경 사항 목록은 [Amazon Redshift 데이터베이스 개발자 안내서 문서 기록](#)을 참조하세요.

변경 사항	설명	날짜
Amazon Redshift 패치 199가 릴리스되었습니다.	새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 Amazon Redshift 패치 199 를 참조하세요.	2026년 3월 4일
Amazon Redshift 패치 198가 릴리스되었습니다.	새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 Amazon Redshift 패치 198 를 참조하세요.	2026년 2월 3일
Amazon Redshift 관리형 정책 업데이트	새로운 AmazonRedshiftFederatedAuth	2025년 11월 21일

orization 관리형 정책이
추가되었습니다.

[Amazon Redshift 패치 197가
릴리스되었습니다.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 197](#)를 참조하세요.

2025년 11월 20일

[Amazon Redshift 패치 196가
릴리스되었습니다.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 196](#)를 참조하세요.

2025년 11월 10일

[Amazon Redshift 패치 195가
릴리스되었습니다.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 195](#)를 참조하세요.

2025년 10월 24일

[Amazon Redshift 패치 194가
릴리스되었습니다.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 194](#)를 참조하세요.

2025년 10월 1일

[Amazon Redshift 패치 193 릴리스됨.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 193](#)을 참조하세요.

2025년 8월 26일

[Amazon Redshift 패치 192 릴리스됨.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 192](#)를 참조하세요.

2025년 7월 30일

[Amazon Redshift 패치 191 릴리스됨.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 191](#)을 참조하세요.

2025년 6월 24일

[Amazon Redshift 패치 190 릴리스됨.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 190](#)을 참조하세요.

2025년 5월 7일

[Amazon Redshift 관리형 정책 업데이트](#)

lakeformation:GetDataAccess 권한을 포함한 AmazonRedshiftServiceLinkedRolePolicy 관리형 정책의 업데이트입니다.

2025년 3월 13일

[Amazon Redshift 패치 189 릴리스됨.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 189](#)를 참조하세요.

2025년 3월 7일

[Amazon Redshift 패치 188 릴리스됨.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 188](#)을 참조하세요.

2025년 2월 5일

[Amazon Redshift 패치 187 릴리스됨.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift patch 187](#)을 참조하세요.

2024년 12월 20일

[Amazon Redshift 관리형 정책 업데이트](#)

glue:GetCatalog 및 glue:GetCatalogs 권한을 포함한 AmazonRedshiftServiceLinkedRolePolicy 관리형 정책의 업데이트입니다.

2024년 12월 3일

[Amazon Redshift 패치 186 릴리스됨.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 186](#)을 참조하세요.

2024년 10월 22일

[Amazon Redshift 패치 185 릴리스됨.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 185](#)를 참조하시기 바랍니다.

2024년 10월 9일

[Amazon Redshift 패치 184 릴리스됨.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 184](#)를 참조하시기 바랍니다.

2024년 9월 12일

[Amazon Redshift 패치 183 릴리스됨](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 183](#)을 참조하세요.

2024년 8월 8일

[Amazon Redshift 패치 182 릴리스됨](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 182](#)를 참조하세요.

2024년 6월 26일

[Amazon Redshift 패치 181 릴리스됨](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 181](#)을 참조하세요.

2024년 5월 1일

[Amazon Redshift 관리형 정책 업데이트](#)

AWS 할당량 또는 한도에 액세스하려면 `servicequotas:GetServiceQuota` 권한과 함께 `AmazonRedshiftServiceLinkedRolePolicy` 관리형 정책으로 업데이트합니다.

2024년 3월 8일

[쿼리 편집기 v2 관리형 정책 업데이트](#)

`redshift-serverless:ListNamespaces` 및 `redshift-serverless:ListWorkgroups` 권한을 사용하여 `AmazonRedshiftQueryEditorV2FullAccess`, `AmazonRedshiftQueryEditorV2NoSharing`, `AmazonRedshiftQueryEditorV2ReadSharing`, `AmazonRedshiftQueryEditorV2ReadWriteSharing` 관리형 정책을 업데이트합니다.

2024년 2월 21일

[Amazon Redshift 읽기 전용 액세스 관리형 정책 업데이트](#)

Amazon Redshift Advisor 권장 사항을 나열할 수 있는 `redshift:ListRecommendations` 권한이 있는 `AmazonRedshiftReadOnlyAccess` 관리형 정책 업데이트.

2024년 2월 7일

[Amazon Redshift 패치 180 릴리스됨](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 180](#)을 참조하세요.

2023년 12월 29일

[Amazon Redshift 패치 179 릴리스됨](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 179](#)를 참조하세요.

2023년 11월 9일

[Amazon Redshift 관리형 정책 업데이트](#)

ec2:AssignIpv6Addresses 및 ec2:UnassignIpv6Addresses 권한을 포함한 AmazonRedshiftServiceLinkedRolePolicy 관리형 정책의 업데이트입니다.

2023년 10월 31일

[Amazon Redshift 패치 178 릴리스됨](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 178](#)을 참조하세요.

2023년 9월 25일

[쿼리 편집기 v2 관리형 정책 업데이트](#)

sqlworkbench:GetAutocompletionMetadata 및 sqlworkbench:GetAutocompletionResource 권한을 사용하여 AmazonRedshiftQueryEditorV2NoSharing, AmazonRedshiftQueryEditorV2ReadSharing 및 AmazonRedshiftQueryEditorV2ReadWriteSharing 관리형 정책을 업데이트합니다.

2023년 8월 16일

[Amazon Redshift 관리형 정책 업데이트](#)

AWS Secrets Manager에 관리자 보안 인증 보안 암호를 만들고 관리할 수 있는 권한을 부여하기 위해 AmazonRedshiftServiceLinkedRolePolicy 관리형 정책을 업데이트합니다.

2023년 8월 14일

[Amazon Redshift 패치 177 릴리스됨](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 177](#)을 참조하세요.

2023년 8월 3일

[Amazon Redshift 패치 176 릴리스됨](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 176](#)을 참조하세요.

2023년 6월 8일

[Amazon Redshift 패치 175 릴리스됨](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 175](#)를 참조하세요.

2023년 4월 28일

[Amazon Redshift 관리형 정책 업데이트](#)

ec2 네트워크 관련 작업에 대한 권한을 제거하도록 AmazonRedshiftServiceLinkedRolePolicy 관리형 정책 업데이트. 이들은 특히 Purpose:RedshiftMigrateToVpc 리소스 태그와 연결되었습니다.

2023년 4월 27일

[Amazon Redshift 데이터 API 관리형 정책 업데이트](#)

redshift:GetClusterCredentialsWithIAM 권한을 포함한 AmazonRedshiftDataFullAccess 관리형 정책의 업데이트입니다.

2023년 4월 7일

[쿼리 편집기 v2 관리형 정책 업데이트](#)

sqlworkbench:GetSchemaInference 권한을 사용하여 AmazonRedshiftQueryEditorV2 NoSharing , AmazonRedshiftQueryEditorV2 ReadSharing 및 AmazonRedshiftQueryEditorV2ReadWriteSharing 관리형 정책을 업데이트합니다.

2023년 3월 21일

[Amazon Redshift 패치 174가 릴리스되었습니다.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 174](#)를 참조하세요.

2023년 3월 11일

[쿼리 편집기 v2 관리형 정책 업데이트](#)

sqlworkbench:AssociateNotebookWithTab 권한을 사용하여 AmazonRedshiftQueryEditorV2 NoSharing , AmazonRedshiftQueryEditorV2 ReadSharing 및 AmazonRedshiftQueryEditorV2ReadWriteSharing 관리형 정책을 업데이트합니다.

2023년 2월 2일

[Amazon Redshift 패치 1730](#)
[릴리스되었습니다.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 173](#)을 참조하세요.

2023년 1월 20일

[Amazon Redshift 패치 172가](#)
[릴리스되었습니다.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 172](#)를 참조하세요.

2022년 11월 17일

[Amazon Redshift 패치 1710](#)
[릴리스되었습니다.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 171](#)을 참조하세요.

2022년 11월 9일

[Amazon Redshift 패치 1700](#)
[릴리스되었습니다.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 170](#)을 참조하세요.

2022년 7월 20일

[Amazon Redshift 패치 169가 릴리스되었습니다.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 169](#)를 참조하세요.

2022년 6월 8일

[Amazon Redshift 패치 168이 릴리스되었습니다.](#)

새로운 Amazon Redshift 패치가 배포되고 있습니다. 모든 Amazon Redshift 지원 AWS 리전에서 새 버전을 사용할 수 있으려면 몇 주가 걸립니다. 이 버전에 대한 자세한 내용은 [Amazon Redshift 패치 168](#)을 참조하세요.

2022년 4월 19일

[Amazon Redshift 드라이버로 인증 프로파일 지원](#)

이제 인증 프로파일을 사용하여 Amazon Redshift에 연결할 수 있습니다.

2021년 8월 2일

[제공 Amazon Redshift용 크로스 VPC 엔드포인트 지원AWS PrivateLink](#)

이제 Amazon Redshift에서 Redshift 관리형 VPC 엔드포인트를 사용할 수 있습니다.

2021년 4월 1일

[Amazon Redshift 쿼리 편집기 개선 사항 지원](#)

이제 Enhanced VPC Routing, 더 긴 쿼리 실행 시간, 더 많은 클러스터 노드 유형을 제공하는 쿼리 편집기를 사용할 수 있습니다.

2021년 2월 17일

[파트너와의 콘솔 통합 지원](#)

Amazon Redshift 콘솔을 사용하여 파트너와 통합할 수 있습니다.

2020년 12월 9일

[가용 영역 간에 클러스터를 이동하는 기능 지원](#)

이제 가용 영역 간에 RA3 클러스터를 이동할 수 있습니다.

2020년 12월 9일

ra3.xlplus 노드 유형 지원	이제 ra3.xlplus 노드 유형을 생성할 수 있습니다.	2020년 12월 9일
JDBC 드라이버 버전 2.0 지원	이제 JDBC 드라이버 버전 2.0을 구성할 수 있습니다.	2020년 11월 5일
Lambda UDF 및 토큰화 지원	이제 Lambda UDF를 작성하여 데이터의 외부 토큰화를 사용할 수 있습니다.	2020년 10월 26일
SQL 문의 실행 예약 지원	이제 Amazon Redshift 콘솔에서 쿼리를 예약할 수 있습니다.	2020년 10월 22일
Amazon Redshift용 데이터 API 지원	이제 기본 제공 데이터 API를 사용하여 Amazon Redshift에 액세스할 수 있습니다. 설명서 업데이트에는 Amazon Redshift 데이터 API 참조가 포함됩니다.	2020년 9월 10일
Amazon Redshift 콘솔 쿼리 모니터링 지원	새로운 쿼리 모니터링 그래프를 설명하도록 안내서를 업데이트했습니다.	2020년 5월 7일
사용 한도 지원	사용 한도를 설명하도록 안내서를 업데이트했습니다.	2020년 4월 23일
멀티 팩터 인증	멀티 팩터 인증 지원에 대해 설명하도록 안내서를 업데이트했습니다.	2020년 4월 20일
탄력적 크기 조정에서 이제 노드 유형 변경 지원	탄력적 크기 조정 설명을 업데이트했습니다.	2020년 4월 6일
관리형 스토리지가 있는 ra3.4xlarge 노드 유형 지원	ra3.4xlarge 노드 유형을 포함하도록 안내서를 업데이트했습니다.	2020년 4월 2일

일시 중지 및 다시 시작 지원	일시 중지 및 다시 시작 클러스터 작업에 대해 설명하도록 안내서를 업데이트했습니다.	2020년 3월 11일
Microsoft Azure AD를 자격 증명 공급자로 지원	Microsoft Azure AD를 자격 증명 공급자로 사용하는 단계를 설명하도록 안내서를 업데이트했습니다.	2020년 2월 10일
RA3 노드 유형 지원	새 RA3 노드 유형을 설명하도록 안내서를 업데이트했습니다.	2019년 12월 3일
새로운 콘솔 지원	새 Amazon Redshift 콘솔을 설명하도록 안내서를 업데이트했습니다.	2019년 11월 11일
보안 정보 업데이트	보안 정보 설명서 업데이트.	2019년 6월 24일
스냅샷 기능 향상	Amazon Redshift에서는 이제 스냅샷을 관리하고 예약하는 여러 기능이 향상되었습니다.	2019년 4월 4일
동시성 확장	워크로드 관리(WLM)에서 동시성 확장 모드를 사용하도록 구성할 수 있습니다. 자세한 내용은 워크로드 관리 구성 을 참조하세요.	2019년 3월 21일
업데이트된 JDBC 및 ODBC 드라이버	이제 Amazon Redshift에서 새로운 버전의 JDBC 및 ODBC 드라이버를 지원합니다. 자세한 내용은 JDBC 연결 구성 을 참조하세요.	2019년 2월 4일

[지연된 유지 관리](#)

클러스터의 유지 관리 기간을 다시 예약해야 하는 경우 유지 관리를 최대 14일까지 지연할 수 있는 옵션이 있습니다. 하드웨어를 업데이트해야 하거나 지연 기간 중 다른 필수 업데이트를 수행해야 하는 경우 Amazon에서는 필요한 변경을 수행할 수 있도록 알려드립니다. 이러한 업데이트 기간 중에는 클러스터를 사용할 수 없습니다. 자세한 내용은 [유지 관리 지연](#)을 참조하세요.

2018년 11월 20일

[사전 알림](#)

Amazon Redshift에서는 일부 이벤트에 대해 사전에 알림을 제공합니다. 이러한 이벤트의 이벤트 범주는 pending입니다. 예를 들어, 클러스터의 노드 중 하나에 대해 하드웨어 업데이트가 필요한 경우 사전 알림을 보냅니다. 다른 Amazon Redshift 이벤트와 마찬가지로 보류 중인 이벤트를 구독할 수 있습니다. 자세한 내용은 [Amazon Redshift 이벤트 알림 구독](#) 섹션을 참조하세요.

2018년 11월 20일

탄력적 크기 조정

탄력적 크기 조정은 클러스터의 크기를 조정할 수 있는 가장 빠른 방법입니다. 탄력적 크기 조정은 기존 클러스터에서 노드를 추가 또는 제거한 다음 새 노드로 데이터를 자동으로 다시 분산시킵니다. 새 클러스터를 생성하지 않기 때문에 일반적으로 탄력적 크기 조정 작업은 불과 몇 분 내에 빠르게 완료됩니다. 자세한 내용은 [클러스터 크기 조정](#)을 참조하세요.

2018년 11월 15일

크기 조정 작업 취소

이제 크기 조정 작업은 진행 중 취소가 가능합니다. 자세한 내용은 [크기 조정 작업 개요](#)를 참조하세요.

2018년 11월 2일

암호화를 변경하기 위한 클러스터 수정

AWS 관리형 키 또는 고객 관리형 키를 사용하여 AWS Key Management Service(AWS KMS) 암호화를 사용하도록 암호화되지 않은 클러스터를 수정할 수 있습니다. KMS 암호화를 사용하도록 클러스터를 수정하면 Amazon Redshift에서는 암호화된 새 클러스터로 데이터를 자동으로 마이그레이션합니다. 또한 클러스터를 수정하여 암호화된 클러스터로 암호화되지 않은 클러스터를 마이그레이션할 수도 있습니다.

2018년 10월 16일

[Amazon Redshift Spectrum에서 향상된 VPC 라우팅 지원](#)

이제, 클러스터에 대해 Enhanced VPC Routing을 활성화한 상태로 Redshift Spectrum을 사용할 수 있습니다. 추가 구성 단계를 수행해야 할 수 있습니다. 자세한 내용은 [Amazon Redshift Spectrum에서 Enhanced VPC Routing 사용](#) 섹션을 참조하세요.

2018년 10월 10일

[쿼리 편집기](#)

이제 Amazon Redshift 관리 콘솔에서 SQL 쿼리를 실행할 수 있습니다.

2018년 10월 4일

[워크로드 실행 분석 차트](#)

이제, 콘솔에서 워크로드 실행 분석 차트를 확인하면 워크로드의 성능을 자세히 파악할 수 있습니다. 자세한 내용은 [워크로드 성능 분석](#)을 참조하세요.

2018년 7월 30일

[유지 관리 트랙](#)

이제 유지 관리 트랙을 선택하여 항상 클러스터를 Amazon Redshift의 최신 버전으로 업데이트할지 이전 버전으로 업데이트할지를 결정할 수 있습니다. 자세한 내용은 [클러스터 유지 관리 트랙 선택](#)을 참조하세요.

2018년 7월 26일

다음 표에서는 2018년 7월 이전 Amazon Redshift 관리 가이드의 중요 변경 사항에 대해 설명합니다.

변경	설명	릴리스 날짜
새로운 CloudWatch 지표	쿼리 성능을 모니터링하기 위한 새로운 CloudWatch 지표가 추가되었습니다. 자세한 내용은 Amazon Redshift의 성능 데이터 섹션을 참조하세요.	2018년 5월 17일

변경	설명	릴리스 날짜
HSM 암호화	Amazon Redshift는 하드웨어 보안 모듈(HSM) 키 관리를 위해 AWS CloudHSM만 지원합니다. 자세한 내용은 Amazon Redshift 데이터베이스 암호화 섹션을 참조하세요.	2018년 3월 6일
IAM 역할 함께 묶기	클러스터에 연결된 IAM 역할에 필요한 리소스에 대한 액세스 권한이 없는 경우 다른 계정에 속한 다른 역할을 함께 묶을 수 있습니다. 그러면 클러스터는 함께 묶은 역할을 일시적으로 수임하여 데이터에 액세스합니다. 역할을 함께 묶어 교차 계정 액세스를 부여할 수도 있습니다. 체인의 각 역할은 클러스터가 체인의 끝에 있는 역할을 수임할 때까지 체인의 다음 역할을 수임합니다. 최대 10개의 역할을 함께 묶을 수 있습니다. 자세한 내용은 Amazon Redshift에서 IAM 역할 연결 섹션을 참조하세요.	2018년 2월 23일
새로운 DC2 노드 유형	최신 고밀도 컴퓨팅(DC) 노드 유형은 DC1과 동일한 가격으로 훨씬 뛰어난 성능을 제공합니다. 성능 향상을 활용하기 위해 DC1 클러스터를 새로운 DC2 노드 유형으로 마이그레이션할 수 있습니다. 자세한 내용은 Amazon Redshift의 클러스터 및 노드 섹션을 참조하세요.	2017년 10월 17일
ACM 인증서	Amazon Redshift는 클러스터의 SSL 인증서를 AWS Certificate Manager(ACM)가 발급한 인증서로 바꿉니다. ACM은 대부분의 최신 시스템에서 신뢰하는 퍼블릭 인증 기관(CA)입니다. 계속해서 SSL을 사용하여 클러스터에 연결하려면 현재 신뢰하는 루트 CA 인증서를 업데이트해야 할 수 있습니다. 자세한 내용은 SSL 연결을 위해 ACM 인증서로 이전 섹션을 참조하세요.	2017년 9월 18일

변경	설명	릴리스 날짜
서비스 연결 역할	서비스 연결 역할은 Amazon Redshift에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 Amazon Redshift에서 사전 정의하며, 해당 서비스가 Amazon Redshift 클러스터를 대신해 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 가지고 있습니다. 자세한 내용은 Amazon Redshift에 대해 서비스 연결 역할 사용 섹션을 참조하세요.	2017년 9월 18일
IAM 데이터베이스 사용자 인증	사용자 스스로 자격 증명을 새로 만들고 IAM 자격 증명을 바탕으로 데이터베이스에 로그인하도록 시스템 권한을 구성할 수 있습니다. 또한 SAML 2.0을 준수하는 자격 증명 공급자를 통해 페더레이션 Single Sign-On을 이용해 로그인하도록 시스템을 구성할 수 있습니다. 자세한 내용은 IAM 인증을 이용한 데이터베이스 사용자 자격 증명 생성 섹션을 참조하세요.	2017년 8월 11일
Enhanced VPC Routing을 지원하는 테이블 수준 복원	테이블 수준 복원은 현재 Enhanced VPC Routing으로 네트워크 트래픽 제어 를 사용하는 클러스터에서 지원됩니다. 자세한 내용은 스냅샷에서 테이블 복원 섹션을 참조하세요.	2017년 7월 19일
쿼리 모니터링 규칙	WLM 쿼리 모니터링 규칙을 사용하여 WLM 대기열에 대한 지표 기반 성능 경계를 정의하고 쿼리가 이러한 경계(로그, 건너뛰기 또는 중단)를 벗어날 때 수행할 작업을 지정할 수 있습니다. 쿼리 모니터링 규칙은 워크로드 관리(WLM) 구성 시 정의합니다. 자세한 내용은 워크로드 관리 섹션을 참조하세요.	2017년 21월 4일
Enhanced VPC Routing	Amazon Redshift Enhanced VPC Routing을 사용하면 Amazon Redshift는 클러스터와 데이터 리포지토리 사이의 COPY 및 UNLOAD 트래픽이 모두 Amazon VPC를 통과하도록 강제합니다. 자세한 내용은 Redshift Enhanced VPC Routing으로 네트워크 트래픽 제어 섹션을 참조하세요.	2016년 9월 15일

변경	설명	릴리스 날짜
새로운 연결 로그 필드	연결 로그 감사 로그에 SSL 연결을 추적할 수 있는 필드 2개가 새롭게 추가되었습니다. 따라서 평상시대로 감사 로그를 Amazon Redshift 테이블에 로드할 때도 sslcompression 열과 sslexpansion 열을 대상 테이블에 추가해야 합니다.	2016년 5월 5일
COPY 및 UNLOAD를 위한 IAM 역할	이제 클러스터가 다른 AWS 서비스에 대한 액세스 인증에 사용할 수 있도록 AWS Identity and Access Management(IAM) 역할을 하나 이상 지정할 수 있습니다. IAM 역할은 COPY, UNLOAD 또는 CREATE LIBRARY 명령을 사용할 때 더욱 안전한 인증 수단으로서 손색이 없습니다. 자세한 내용은 Amazon Redshift가 사용자를 대신하여 AWS 서비스에 액세스할 수 있도록 권한 부여 및 IAM 역할을 사용하여 COPY, UNLOAD, CREATE EXTERNAL FUNCTION 및 CREATE EXTERNAL SCHEMA 작업에 대한 권한 부여 섹션을 참조하세요.	2016년 3월 29일
테이블에서 복원	클러스터 스냅샷에서 활성 클러스터의 새로운 테이블로 테이블을 복원할 수 있습니다. 자세한 내용은 스냅샷에서 테이블 복원 섹션을 참조하세요.	2016년 3월 10일
정책의 IAM 조건 사용	IAM 정책에서 조건 요소를 사용하여 리소스에 대한 액세스를 추가로 제한할 수 있습니다. 자세한 내용은 IAM 정책 조건을 사용하여 세분화된 액세스 제어 구현 섹션을 참조하세요.	2015년 12월 10일
공개 액세스 가능 여부 수정	VPC의 기존 클러스터를 수정하여 공개 액세스 여부를 변경할 수 있습니다. 자세한 내용은 클러스터 수정 섹션을 참조하세요.	2015년 11월 20일
설명서 수정 사항	다양한 설명서 수정 사항이 공개되었습니다.	2015년 8월 28일

변경	설명	릴리스 날짜
설명서 업데이트	네트워크 설정 구성에 대한 문제 해결 지침이 업데이트되면서 최대 전송 유닛(MTU) 크기가 다른 호스트도 연결에 필요한 패킷 크기를 결정할 수 있습니다. 자세한 내용은 쿼리가 중단되거나, 간혹 클러스터까지 전송되지 않습니다 섹션을 참조하세요.	2015년 8월 25일
설명서 업데이트	파라미터 그룹 섹션 전체가 구성 및 명료성을 개선하기 위해 개정되었습니다. 자세한 내용은 Amazon Redshift 파라미터 그룹 섹션을 참조하세요.	2015년 8월 17일
WLM 동적 속성	WLM 구성 파라미터가 이제 일부 속성을 동적으로 적용할 수 있도록 지원합니다. 그 밖의 속성은 정적 변경을 그대로 유지하며, 따라서 구성 변경 사항을 적용하려면 연결된 클러스터를 재부팅해야 합니다. 자세한 내용은 WLM 동적 및 정적 속성 및 Amazon Redshift 파라미터 그룹 섹션을 참조하세요.	2015년 8월 3일
KMS로 암호화된 클러스터를 다른 AWS 리전으로 복사	AWS KMS로 암호화된 클러스터를 다른 AWS 리전으로 복사할 수 있는 스냅샷 복사 권한의 구성에 대한 내용이 추가되었습니다. 자세한 내용은 AWS KMS로 암호화된 스냅샷을 다른 AWS 리전으로 복사 섹션을 참조하세요.	2015년 7월 28일
설명서 업데이트	데이터베이스 암호화 섹션이 업데이트되면서 Amazon Redshift가 키 관리를 위해 AWS KMS 또는 HSM을 사용하는 방법, 그리고 이 두 가지 옵션을 각각 사용하는 암호화 프로세스에 대한 설명이 더욱 명료해졌습니다. 자세한 내용은 Amazon Redshift 데이터베이스 암호화 섹션을 참조하세요.	2015년 7월 28일

변경	설명	릴리스 날짜
새로운 노드 유형	Amazon Redshift에서 이제 새로운 노드 유형인 DS2를 제공합니다. 이번 릴리스에서 처음 소개하는 새로운 이름을 사용할 수 있도록 기존 노드 유형에 대한 설명서 참조 자료도 업데이트되었습니다. 또한 노드 유형의 조합을 이해하기 쉽게 설명하고 기본 할당량 제한을 명확히 밝히기 위해 해당 섹션도 개정되었습니다. 자세한 내용은 Amazon Redshift의 클러스터 및 노드 섹션을 참조하세요.	2015년 6월 9일
예약 노드	새로운 예약 노드 제공에 대한 내용이 추가되었습니다. 또한 유효한 예약 노드를 이해하기 쉽게 설명하고 서로 비교할 수 있도록 해당 섹션이 개정되었으며, 온디맨드 방식의 노드 요금과 예약 노드 요금이 실제 결제에서 어떻게 달라지는지 예를 들어 설명하고 있습니다. 자세한 내용은 예약 노드 섹션을 참조하세요.	2015년 6월 9일
설명서 수정 사항	다양한 설명서 수정 사항이 공개되었습니다.	2015년 4월 30일
새로운 기능	이번 Amazon Redshift 릴리스에서는 Amazon Redshift에 최적화된 ODBC 및 JDBC 드라이버를 새로 소개합니다. 자세한 내용은 SQL 클라이언트 도구를 사용하여 Amazon Redshift 데이터 웨어하우스에 연결 섹션을 참조하세요.	2015년 2월 26일
새로운 기능	이번 Amazon Redshift 릴리스는 쿼리 실행에 따른 세부 정보를 확인하고 분석할 수 있는 클러스터 성능 지표를 제공합니다. 자세한 내용은 쿼리 및 로드 보기 섹션을 참조하세요.	2015년 2월 26일
설명서 업데이트	AWS의 공통 Amazon Redshift 서비스 작업 및 리소스에 대한 액세스 권한을 부여하는 정책 예가 새롭게 추가되었습니다. 자세한 내용은 고객 관리형 정책 예제 섹션을 참조하세요.	2015년 1월 16일

변경	설명	릴리스 날짜
설명서 업데이트	최대 전송 단위(MTU)를 설정하여 TCP/IP 정보 프레임 을 비활성화하는 지침이 업데이트되었습니다. 자세한 내용은 EC2를 사용하여 클러스터 생성 및 쿼리가 중단 되거나, 간혹 클러스터까지 전송되지 않습니다 섹션을 참조하세요.	2015년 1월 16일
설명서 업데이트	wlm_json_configuration 파라미터에 대한 내용 이 개정되었고, Linux, Mac OS X 및 Microsoft Windows 운영 체제 기반 AWS CLI를 사용하여 이 파라미터를 구성하는 예 구문이 제공됩니다. 자세한 내용은 워크로드 관리 섹션을 참조하세요.	2015년 1월 13일
설명서 업데이트	누락된 이벤트 알림 메시지 및 설명이 추가되었습니다. 자세한 내용은 Amazon Redshift 프로비저닝 클러스터 이벤트 알림 섹션을 참조하세요.	2015년 1월 8일
설명서 업데이트	Amazon Redshift 작업 및 리소스에 대한 IAM 정책 지침이 업데이트되었습니다. 구성 및 명료성을 개선하기 위해 해당 섹션이 개정되었습니다. 자세한 내용은 Amazon Redshift의 보안 섹션을 참조하세요.	2014년 11월 21일
새로운 기능	이번 Amazon Redshift 릴리스에는 AWS Key Management Service(AWS KMS) 암호화 키를 사용하여 클러스터를 암호화하는 기능이 도입됩니다. AWS KMS는 클라우드에 맞게 조정된 키 관리 시스템을 제공하기 위해 안전하고 가용성이 높은 하드웨어 및 소프트웨어를 결합합니다. AWS KMS 및 Amazon Redshift의 암호화 옵션에 대한 자세한 내용은 Amazon Redshift 데이터베이스 암호화 및 클러스터 작업 섹션을 참조하세요.	2014년 11월 12일

변경	설명	릴리스 날짜
새로운 기능	이번 Amazon Redshift 릴리스에는 클러스터나 스냅샷 같은 리소스의 태그 관리 기능이 도입됩니다. 태그는 사용자 정의 메타데이터를 제공하여 비용 할당을 기준으로 결제 보고서를 분류할 뿐만 아니라 리소스를 한 눈에 정확하게 파악하는 데도 효과적입니다. 자세한 내용은 Amazon Redshift에서 리소스 태그 섹션을 참조하세요.	2014년 11월 4일
새로운 기능	dw1.8xlarge 및 dw2.8xlarge 노드 크기일 때 최대 노드 수가 128개로 늘어났습니다. 자세한 내용은 Amazon Redshift의 클러스터 및 노드 섹션을 참조하세요.	2014년 10월 30일
새로운 기능	Amazon Redshift 콘솔에서 쿼리와 로드를 종료할 수 있는 기능이 추가되었습니다. 자세한 내용은 쿼리 및 로드 보기 및 로드 작업 중 클러스터 지표 보기 섹션을 참조하세요.	2014년 10월 28일
설명서 수정 사항	다양한 설명서 수정 사항이 공개되었습니다.	2014년 10월 17일
새로운 내용	클러스터 종료 및 삭제에 대한 내용이 추가되었습니다. 자세한 내용은 클러스터 종료 및 삭제 섹션을 참조하세요.	2014년 8월 14일
설명서 업데이트	클러스터에서 버전 업그레이드 허용 설정의 동작에 대한 설명이 이해하기 쉽게 바뀌었습니다. 자세한 내용은 Amazon Redshift 프로비저닝 클러스터 섹션을 참조하세요.	2014년 8월 14일
설명서 업데이트	Amazon Redshift 콘솔의 클러스터 작업에 대한 절차, 스크린샷 및 주제 구성이 개정되었습니다. 자세한 내용은 클러스터 작업 섹션을 참조하세요.	2014년 7월 11일
새로운 내용	클러스터의 읽기 전용 모드 시간을 최소화하여 클러스터의 크기를 조정하는 방법을 포함하여 Amazon Redshift 클러스터의 크기 조정에 대한 튜토리얼이 새롭게 추가되었습니다. 자세한 내용은 클러스터 크기 조정 섹션을 참조하세요.	2014년 6월 27일

변경	설명	릴리스 날짜
새로운 기능	클러스터의 이름을 변경하는 기능이 추가되었습니다. 자세한 내용은 클러스터 이름 변경 및 클러스터 수정 섹션을 참조하세요.	2014년 6월 2일
새로운 기능	스냅샷에서 클러스터를 복원할 때 다른 파라미터 그룹과 보안 그룹을 선택하는 옵션이 추가되었습니다. 자세한 내용은 스냅샷에서 클러스터 복원 섹션을 참조하세요.	2014년 5월 12일
새로운 기능	기본 Amazon CloudWatch 경보를 구성하여 Amazon Redshift 클러스터에서 사용된 디스크 공간 비율을 모니터링하는 방법에 대한 섹션이 새롭게 추가되었습니다. 이 경보는 클러스터 생성 프로세스에 새롭게 도입된 옵션입니다. 자세한 내용은 기본 디스크 공간 경보 섹션을 참조하세요.	2014년 4월 28일
설명서 업데이트	Amazon Redshift의 ECDHE(Elliptic Curve Diffie—Hellman Ephemeral) 지원에 대한 정보를 명확히 했습니다. 자세한 내용은 SSL 섹션을 참조하세요.	2014년 4월 22일
새로운 기능	Amazon Redshift의 ECDH(Elliptic Curve Diffie—Hellman Ephemeral) 지원에 대한 설명이 추가되었습니다. 자세한 내용은 SSL 섹션을 참조하세요.	2014년 4월 18일
설명서 업데이트	SQL 클라이언트 도구를 사용하여 Amazon Redshift 데이터 웨어하우스에 연결 섹션의 주제가 개정을 통해 재구성되었습니다. JDBC 및 ODBC 연결에 대한 추가 정보를 비롯해 연결 문제 해결을 위한 문제 해결 섹션이 새롭게 추가되었습니다.	2014년 4월 15일
설명서 업데이트	안내서 전체의 IAM 정책 예에 버전이 추가되었습니다.	2014년 4월 3일
설명서 업데이트	클러스터 크기 조정 시 요금 산정 방식에 대한 정보가 추가되었습니다. 자세한 내용은 예약 노드 섹션을 참조하세요.	2014년 4월 2일

변경	설명	릴리스 날짜
새로운 기능	개별 커서마다 저장 가능한 최대 결과 집합 크기(MB)를 설정하는 새로운 파라미터인 <code>max_cursor_result_set_size</code> 에 대한 섹션이 추가되었습니다. 이 파라미터 값은 클러스터에서 동시에 활성화되는 커서의 수에도 영향을 미칩니다. 자세한 내용은 Amazon Redshift 파라미터 그룹 섹션을 참조하세요.	2014년 3월 28일
새로운 기능	현재 클러스터 엔진 버전과 데이터베이스 개정 번호가 모두 기재되는 클러스터 버전 필드에 대한 설명이 추가되었습니다. 자세한 내용은 Amazon Redshift 프로비저닝 클러스터 섹션을 참조하세요.	2014년 3월 21일
새로운 기능	클러스터의 상태 탭에 크기 조정 진행 상황에 대한 정보를 표시할 수 있도록 크기 조정 절차가 업데이트되었습니다. 자세한 내용은 클러스터 크기 조정 섹션을 참조하세요.	2014년 3월 21일
설명서 업데이트	Amazon Redshift란 무엇입니까? 섹션이 재구성을 통해 업데이트되었고, Amazon Redshift 프로비저닝 클러스터 개요 섹션이 개정되었습니다. 다양한 설명서 수정 사항이 공개되었습니다.	2014년 21월 2일
새로운 기능	Amazon Redshift 클러스터의 새로운 노드 유형 및 크기가 추가되었고, 피드백을 기준으로 구성 및 명료성을 개선하기 위해 관련 클러스터 개요 주제를 다시 작성하였습니다. 자세한 내용은 Amazon Redshift 프로비저닝 클러스터 섹션을 참조하세요.	2014년 1월 23일
새로운 기능	Virtual Private Cloud에서 공개 액세스가 가능한 Amazon Redshift 클러스터에 탄력적 IP 주소(EIP)를 사용하는 것에 대한 정보가 추가되었습니다. Amazon Redshift의 EIP에 대한 자세한 내용은 VPC의 Redshift 리소스 및 VPC에서 Redshift 프로비저닝된 클러스터 또는 Amazon Redshift Serverless 작업 그룹 생성 섹션을 참조하세요.	2013년 20월 12일

변경	설명	릴리스 날짜
새로운 기능	Amazon Redshift의 AWS CloudTrail 로그에 대한 정보가 추가되었습니다. Amazon Redshift의 CloudTrail 지원에 대한 자세한 내용은 CloudTrail을 사용하여 로깅 섹션을 참조하세요.	2013년 12월 13일
새로운 기능	새로운 사용자 활동 로그에 대한 정보와 Amazon Redshift의 데이터베이스 감사 로깅 기능에 사용되는 enable_user_activity_logging 데이터베이스 파라미터에 대한 정보가 추가되었습니다. 데이터베이스 감사 로깅에 대한 자세한 내용은 데이터베이스 감사 로깅 섹션을 참조하세요. 데이터베이스 파라미터에 대한 자세한 내용은 Amazon Redshift 파라미터 그룹 섹션을 참조하세요.	2013년 12월 6일
새로운 기능	자동/수동 스냅샷을 보조 AWS 리전으로 자동 복사하도록 Amazon Redshift를 구성하는 방법에 대한 설명이 업데이트되었습니다. 리전 간 스냅샷 복사 구성에 대한 자세한 내용은 다른 AWS 리전에 스냅샷 복사 섹션을 참조하세요.	2013년 11월 14일
새로운 기능	연결 및 사용자 활동에 대한 Amazon Redshift 감사 로깅과 이 로그를 Amazon S3에 저장하는 방법을 설명하는 섹션이 추가되었습니다. 데이터베이스 감사 로깅에 대한 자세한 내용은 데이터베이스 감사 로깅 섹션을 참조하세요.	2013년 11월 11일
새로운 기능	하드웨어 보안 모듈(HSM)에서 암호화 키를 관리하거나, 암호화 키를 교체하는 등 새로운 기능이 추가된 Amazon Redshift 암호화에 대한 섹션이 추가되었습니다. 암호화, HSM 및 키 교체에 대한 자세한 내용은 Amazon Redshift 데이터베이스 암호화 , 하드웨어 보안 모듈을 사용한 암호화 및 키 교체 암호화 섹션을 참조하세요.	2013년 11월 11일

변경	설명	릴리스 날짜
새로운 기능	Amazon SNS를 사용하여 Amazon Redshift 이벤트 알림 메시지를 게시하는 방법에 대한 설명이 업데이트되었습니다. Amazon Redshift 이벤트 알림에 대한 자세한 내용은 Amazon Redshift 프로비저닝 클러스터 이벤트 알림 섹션을 참조하세요.	2013년 11월 11일
새로운 기능	IAM 리소스 수준 권한에 대한 설명이 업데이트되었습니다. Amazon Redshift IAM 권한에 대한 자세한 내용은 Amazon Redshift의 보안 섹션을 참조하세요.	2013년 8월 9일
새로운 기능	복원 진행 상황 지표에 대한 설명이 업데이트되었습니다. 자세한 내용은 스냅샷에서 클러스터 복원 섹션을 참조하세요.	2013년 8월 9일
새로운 기능	클러스터 스냅샷 공유를 비롯해 스냅샷 생성 진행 상황 지표에 대한 설명이 업데이트되었습니다. 자세한 내용은 스냅샷 공유 섹션을 참조하세요.	2013년 7월 17일
설명서 수정 사항	다양한 설명서 수정 사항이 공개되었습니다.	2013년 7월 8일
새로운 콘솔 화면	Amazon Redshift 콘솔의 변경 사항에 맞게 Amazon Redshift 관리 가이드가 업데이트되었습니다.	2013년 22월 4일
새 안내서	이 설명서는 Amazon Redshift 관리 안내서의 최초 릴리스입니다.	2013년 14월 2일